MC68HC05L2/D
Rev. 1

HC05

MC68HC05L2

TECHNICAL
DATA

MC68HC05L2
MC68HC705L2

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

Conventions

Where abbreviations are used in the text, an explanation can be found in the glossary, at the back of this manual. 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, a shaded cell in a register diagram indicates that the bit is either unused or reserved; 'u' is used to indicate an undefined state (on reset).

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC05L2/D)

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

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

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

Comments: _____

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

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

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

Completely ☐ ☐ ☐ ☐ Not at all

Comments: _____

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

Easy ☐ ☐ ☐ ☐ Difficult

Comments: _____

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

	Too little detail				Too much detail
SECTION 1 INTRODUCTION	☐	☐	☐	☐	☐
SECTION 2 MODES OF OPERATION AND PIN DESCRIPTIONS	☐	☐	☐	☐	☐
SECTION 3 MEMORY AND REGISTERS	☐	☐	☐	☐	☐
SECTION 4 INPUT/OUTPUT PORTS	☐	☐	☐	☐	☐
SECTION 5 CORE TIMER	☐	☐	☐	☐	☐
SECTION 6 PROGRAMMABLE TIMER	☐	☐	☐	☐	☐
SECTION 7 LIQUID CRYSTAL DISPLAY DRIVER	☐	☐	☐	☐	☐
SECTION 8 ANALOG TO DIGITAL CONVERTER	☐	☐	☐	☐	☐
SECTION 9 RESETS AND INTERRUPTS	☐	☐	☐	☐	☐
SECTION 10 CPU CORE AND INSTRUCTION SET	☐	☐	☐	☐	☐
SECTION 11 ELECTRICAL SPECIFICATIONS	☐	☐	☐	☐	☐
SECTION 12 MECHANICAL DATA	☐	☐	☐	☐	☐
SECTION 13 ORDERING INFORMATION	☐	☐	☐	☐	☐

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: _____

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

IBRS NUMBER PHQ-B/207/G
CCRI NUMERO PHQ-B/207/G

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

Motorola Ltd.,
Colvilles Road,
Kelvin Industrial Estate,
EAST KILBRIDE,
G75 8BR.
GREAT BRITAIN.

F.A.O. Technical Publications Manager
(re: MC68HC05L2/D)

MOTOROLA
Semiconductor Products Sector

– 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: _____

_____

Thank you for helping us improve our documentation,
Graham Livey, Technical Publications Manager, Motorola Ltd., Scotland.

For More Information On This Product,
Go to: www.freescale.com

– Finally, tuck this edge into opposite flap –

TABLE OF CONTENTS

Paragraph Number | TITLE | Page Number

6
PROGRAMMABLE TIMER

7
LIQUID CRYSTAL DISPLAY DRIVER

Paragraph Number | TITLE | Page Number

8
ANALOG TO DIGITAL CONVERTER

9
RESETS AND INTERRUPTS

10
CPU CORE AND INSTRUCTION SET

THIS PAGE LEFT BLANK INTENTIONALLY

LIST OF FIGURES

THIS PAGE LEFT BLANK INTENTIONALLY

LIST OF TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1
INTRODUCTION

The MC68HC05L2 is a member of the M68HC05 family of HCMOS microcomputers. It has 1.8 kbytes of user ROM and 96 bytes of RAM. In addition to its on-board memory, the MC68HC05L2 has a 16-bit programmable timer, a core timer and computer operating properly watchdog, and an A/D converter. With these features, and its LCD subsystem (capable of driving up to 45 segments), the MC68HC05L2 is particularly suited to consumer and automotive applications where a cost effective LCD drive capability is required. The device is available in a 42-pin SDIP package.

All references to the MC68HC05L2 apply equally to the MC68HC705L2, unless otherwise noted. *References specific to the MC68HC705L2 are italicised in the text.*

The MC68HC705L2 is an EPROM version of the MC68HC05L2 and is available in a windowed ceramic SDIP package.

1.1 Features

- Fully static design featuring the industry standard M68HC05 core
- On-chip oscillator and divide-by-two
- 1792 bytes of user ROM and vectors;
  1792 bytes of user EPROM and vectors (MC68HC705L2)
- 112 bytes of bootloader ROM
- 96 bytes of RAM
- LCD subsystem, with 3 x 15 segment drive capability and 8 bytes of dedicated display RAM
- 16-bit programmable timer with input capture
- 15-stage multipurpose core timer, with overflow, watchdog and real-time interrupt
- Computer Operating Properly (COP) watchdog timer (software selectable)
- Edge or edge-and-level sensitive interrupt trigger (software selectable)
- Single-channel, 8-bit analog-to-digital converter

- 13 bidirectional I/O lines
- Power saving STOP and WAIT modes
- Available in 42-pin plastic SDIP *and 42-pin ceramic SDIP* packages

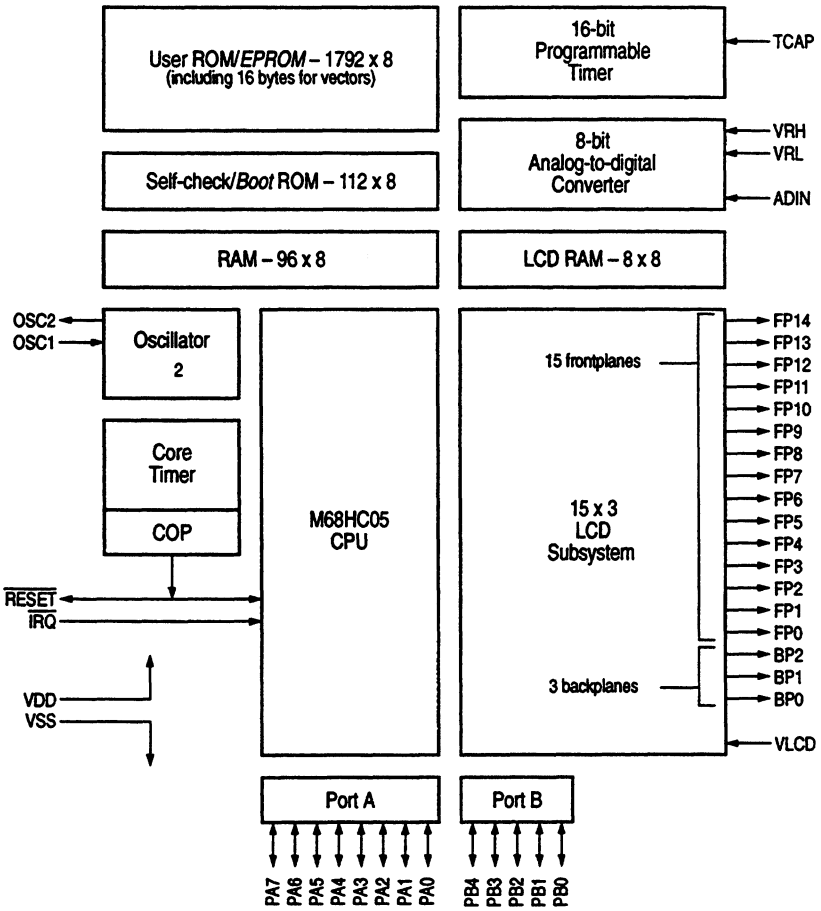


Figure 1-1 MC68HC05L2/*MC68HC705L2* block diagram

1.2 Mask option

There is only one mask option on the MC68HC05L2. This bit is programmed during manufacture and must be specified on the order form.

- STOP instruction – enable/disable.

The STOP enable/disable option does not exist on the MC68HC705L2. Here, the STOP instruction is enabled at all times.

1.3 Additional options

In addition to the mask option, there are two options that are controlled by write-once bits in the Core Timer Control and Status register (CTCSR).

- $\overline{\text{IRQ}}$ – Edge/edge-and-level sensitive
- COP watchdog timer – enable/disable

Full details of the above options are given in Section 5.3.1.

THIS PAGE LEFT BLANK INTENTIONALLY

2
MODES OF OPERATION AND PIN DESCRIPTIONS

2.1 Modes of operation

The MC68HC05L2 has two modes of operation, namely single chip and RAM bootloader mode. *The MC68HC705L2 also has two modes of operation; single chip and EPROM bootloader mode.* Table 2-1 and Table 2-1 show the conditions required to enter each mode on the rising edge of $\overline{\text{RESET}}$.

Table 2-1 MC68HC05L2 operating mode entry conditions

$\overline{\text{IRQ}}$/VPP	PB1	PB2	PB3	Mode	
V_{SS} to V_{DD}	x	x	x	Single chip	
$1.8V_{DD}$	1	x	0	RAM bootloader	Jump to RAM ($00A1)
$1.8V_{DD}$	1	x	1		Load RAM & execute ($00A1)

x = Don't care

Table 2-2 *MC68HC705L2 operating mode entry conditions*

$\overline{\text{IRQ}}$/VPP	*PB1*	*PB2*	*PB3*	*Mode*	
V_{SS} to V_{DD}	*x*	*x*	*x*	*Single chip*	
$1.8V_{DD}$	*1*	*1*	*1*	*EPROM bootloader*	*Program/verify*
$1.8V_{DD}$	*1*	*0*	*1*		*Verify only*

x = Don't care

2.1.1 Single chip mode

This is the normal operating mode of the MC68HC05L2. In this mode the device functions as a self-contained microcomputer (MCU) with all on-board peripherals, including the three 8-bit I/O ports and the 8-bit input-only port, available to the user. All address and data activity occurs within the MCU.

Single chip mode is entered on the rising edge of $\overline{\text{RESET}}$ if the voltage level on the $\overline{\text{IRQ}}$ pin is within the normal operating range.

Caution: *For the MC68HC705L2 all vectors are fetched from EPROM in single chip mode; therefore, the EPROM must be programmed (via the bootloader mode) before the device is powered up in single chip mode.*

2.1.2 RAM bootloader mode for the MC68HC05L2

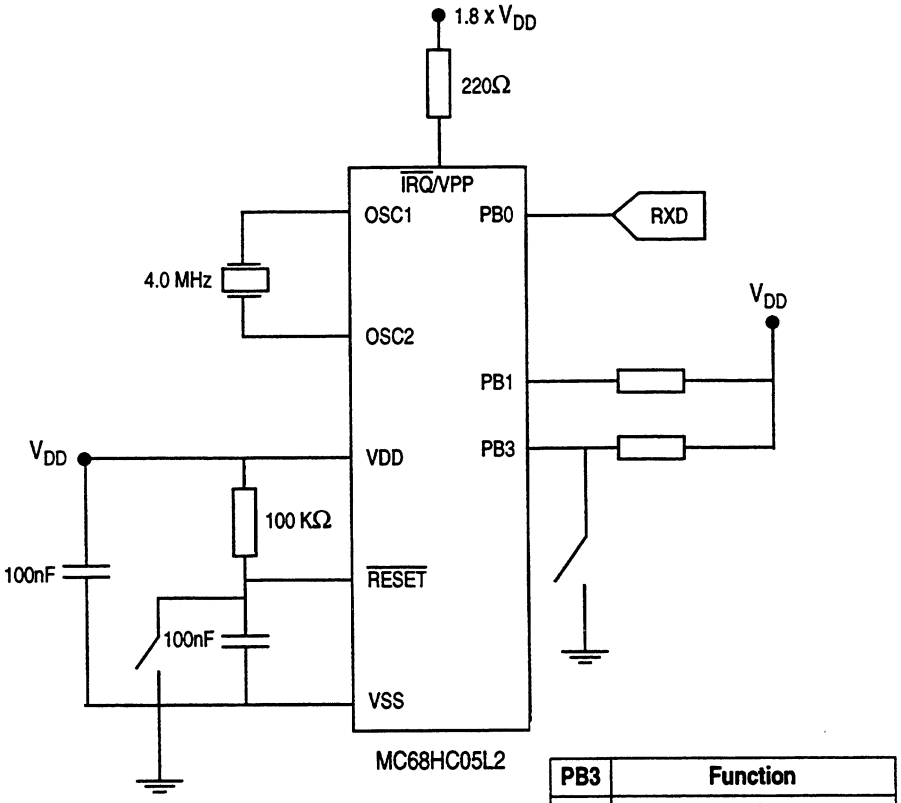
The RAM Bootloader mode for the MC68HC05L2 allows the user to perform simple load and execute instructions in RAM. To make use of this feature a circuit board should be constructed as shown in Figure 2-1. It is then possible, by correctly configuring port pins PB1 and PB3, to load a user program into RAM and then execute.

The RAM bootloader is selected when the device is put into bootloader mode with PB1 held high. If PB3 is low, the program counter is set to $00A1 and a previously loaded RAM program can be executed. If PB3 is high at reset a program is serially loaded from PB0 into the RAM and executed from $00A1.

The first byte to be loaded is the count byte which should hold the program length plus the count byte. Therefore, for a program length of $30, the count should equal $31. The maximum program size including the count byte is 94 bytes ($5E), since two bytes must be left for the stack during download.

The serial data format is 9600 baud, low start bit, 8 data bits, high stop bit. The data is in hexadecimal form, not ASCII.

In the RAM bootloader mode all interrupt vectors are mapped to pseudo-vectors in RAM (refer to Table 2-3). This allows programmers to use their own service-routine addresses. Each pseudo-vector is allowed three bytes of space, rather than the two bytes for normal vectors, because an explicit jump (JMP) opcode is needed to cause the desired jump to the user's service-routine address.



PB3	Function
0	Jump to RAM ($00A1)
1	Load RAM & execute ($00A1)

Figure 2-1 RAM bootloader circuit

Table 2-3 RAM bootloader mode jump vectors

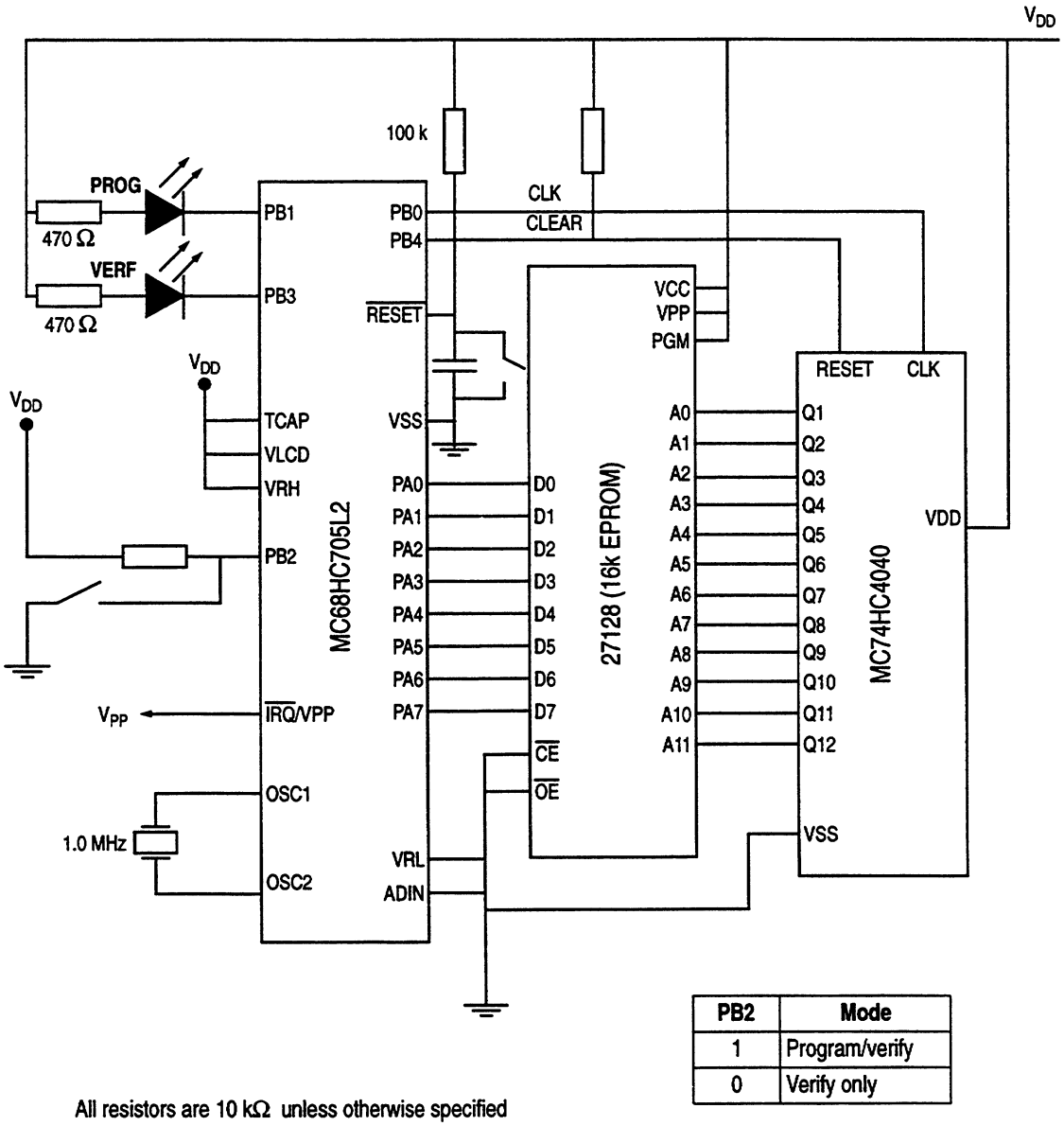
Address	Pseudo-vector
00AC	Programmable timer
00A9	Core Timer interrupt
00A6	IRQ interrupt
00A3	Software interrupt

2.1.3 *EPROM bootloader mode for the MC68HC705L2*

This mode is used for programming the on-board EPROM. In bootloader mode the operation of the device is the same as in single chip mode, except that the vectors are fetched from a reserved area of ROM at locations $0FE0 to $0FEF, instead of from the EPROM. The pin assignments are identical therefore to those of single chip mode (see Figure 2-3). Because the addresses in the

2

MC68HC705L2 and the EPROM containing the user code are incremented independently, it is essential that the data layout in the 27128 EPROM conforms exactly to the MC68HC705L2 memory map.The bootloader uses an external 14-bit counter to address the memory device containing the code to be copied. This counter requires a clock and a reset function. See Figure 2-2.



PB2	Mode
1	Program/verify
0	Verify only

All resistors are 10 kΩ unless otherwise specified

Figure 2-2 *MC68HC705L2 EPROM programming circuit*

2.1.3.1 *Bootloader functions*

The bootloader code deals with the copying of user code from an external EPROM into the on-chip EPROM. The bootloader function can only be used with an external EPROM. The bootloader performs a programming pass and then does a verify pass.

Pin PB2 is used to select various bootloader functions, including the programming mode (see Figure 2-2). Two other pins, PB1 and PB3, are used to drive the PROG and VERF LED outputs. While the EPROM is being programmed the PROG LED lights; when programming is complete the internal EPROM contents are compared to that of the external EPROM and, if they match exactly, the VERF LED lights up.

The EPROM must be erased before performing a program cycle.

2.2 Pin descriptions

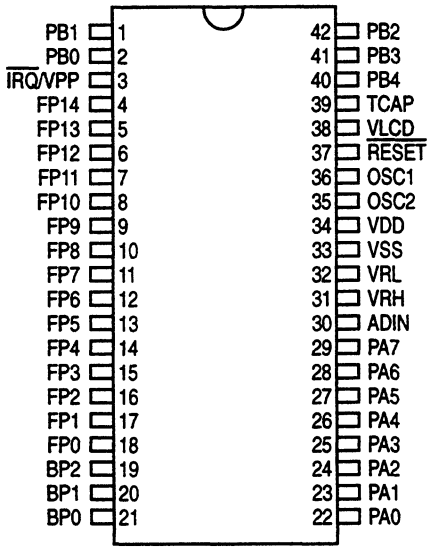


Figure 2-3 42-pin SDIP single chip and bootloader mode pin assignments

2.2.1 VDD and VSS

Power is supplied to the microcontroller using these two pins. VDD is the positive supply and VSS is ground.

It is in the nature of CMOS designs that very fast signal transitions occur on the MCU pins. These short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, special care must be taken to provide good power supply by-passing at the MCU. By-pass capacitors should have good high-frequency characteristics and be as close to the MCU as possible. By-passing requirements vary, depending on how heavily the MCU pins are loaded.

2.2.2 $\overline{\text{IRQ}}$

This is an input-only pin for external interrupt sources. Interrupt triggering can be either edge sensitive or edge-and-level sensitive (see Section 5.3.1). The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. *This pin also serves as the input pin for the EPROM programming voltage.*

2.2.3 OSC1, OSC2

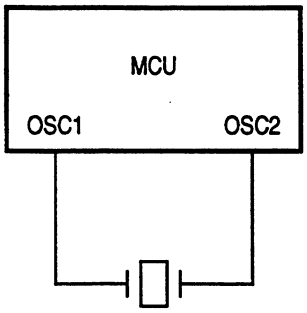
These pins provide control input for an on-chip oscillator circuit. A crystal, ceramic resonator or external clock signal connected to these pins supplies the oscillator clock. The oscillator frequency (f_{osc}) is divided by two to give the internal bus frequency (f_{op}). The on-chip oscillator circuit incorporates two 22pF capacitors, one from OSC1 to ground and the other from OSC2 to ground. A 10 MΩ resistor has also been included across OSC1 and OSC2.

2.2.3.1 Crystal

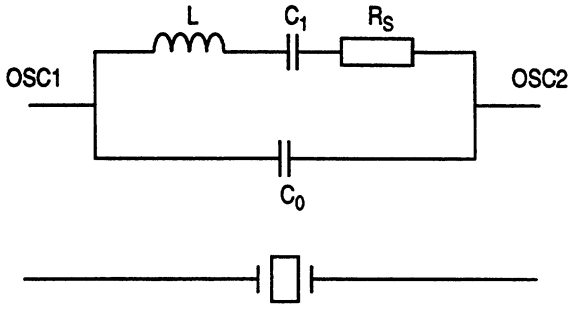
The circuit shown in Figure 2-4(a) is recommended when using either a crystal or a ceramic resonator. Figure 2-4(d) provides the recommended capacitance and feedback resistance values. The internal oscillator is designed to interface with an AT-cut parallel-resonant quartz crystal resonator in the frequency range specified for f_{osc} (see Section 11.4). Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and associated components should be mounted as close as possible to the input pins to minimise output distortion and start-up stabilisation time. The manufacturer of the particular crystal being considered should be consulted for specific information.

2.2.3.2 Ceramic Resonator

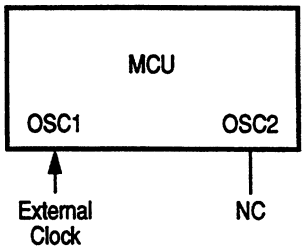
A ceramic resonator may be used instead of a crystal in cost sensitive applications. The circuit shown in Figure 2-4(a) is recommended when using either a crystal or a ceramic resonator. Figure 2-4(d) lists the recommended capacitance and feedback resistance values. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.



(a) Crystal/ceramic resonator oscillator connections



(b) Crystal equivalent circuit



(c) External clock source connections

Crystal

	2MHz	4MHz	Unit
R_S(max)	400	75	Ω
C_0	5	7	pF
C_1	8	12	nF
Q	30 000	40 000	—

Ceramic Resonator

	2 – 4MHz	Unit
R_S(typ)	10	Ω
C_0	40	pF
C_1	4.3	pF
Q	1250	—

(d) Crystal and ceramic resonator parameters

Figure 2-4 Oscillator connections

2.2.3.3 External Clock

An external clock should be applied to the OSC1 input, with the OSC2 pin left unconnected, as shown in Figure 2-4(c). The t_{OXOV} specification (see Section 11.4) does not apply when using an external clock input. The equivalent specification of the external clock source should be used in lieu of t_{OXOV}.

2.2.4 $\overline{\text{RESET}}$

This active low bidirectional pin is used to reset the MCU. Applying a logic zero to this pin forces the device to a known start-up state. An external RC-circuit can be connected to this pin to generate a power-on-reset (POR) if required. In this case, the time constant must be great enough to allow the oscillator circuit to stabilise. This input has an internal Schmitt trigger to improve noise immunity. When a reset condition occurs internally, i.e. from the COP watchdog, the $\overline{\text{RESET}}$ pin provides an active-low open drain output signal which may be used to reset external hardware.

2.2.5 PA0–PA7/PB0–PB4

These 13 I/O lines comprise ports A and B. The state of any pin is software programmable, and all the pins are configured as inputs during power-on or reset. In addition each port pin can be connected to an internal pull-up resistor (approximately 10 – 30kΩ) when configured as an input. This is done via the corresponding port configuration register (see Section 4.4.3 and Section 4.4.4 for further details).

2.2.6 TCAP

The TCAP pin controls the input capture feature of the on-chip programmable timer.

2.2.7 BP0–BP2

These are the three backplane drivers used to select the LCD multiplex ratio.

2.2.8 FP0–FP14

These are the 15 frontplane drivers of the device. These pins are dedicated to the LCD subsystem.

2.2.9 VLCD

This pin is the input for the LCD supply voltage.

2.2.10 ADIN

This pin provides the analog input to the A to D converter.

2.2.11 VRH, VRL

These two inputs provide the reference voltages for the analog to digital converter circuitry. VRL is used for the low reference voltage, typically 0V dc, and VRH the high reference.

2.3 Low power modes

2.3.1 STOP

The STOP instruction places the MCU in its lowest power consumption mode. In STOP mode, the internal oscillator is turned off, halting all internal processing, including timer (and COP watchdog timer) operation.

During the STOP mode, the core timer interrupt flags (CTOF and RTIF) and interrupt enable bits (CTOFE and RTIE) in the CTCSR, the timer flags for the 16-bit timer in the TSR register and the interrupt enable bits in the TCR register are cleared by internal hardware. This removes any pending timer interrupt requests and disables any further timer interrupts. The timer prescaler is cleared. The I-bit in the CCR is cleared to enable external interrupts. All other registers, the remaining bits in the CTCSR, and memory contents remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or a reset (see Figure 2-5).

2.3.2 WAIT

The WAIT instruction places the MCU in a low power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the 16-bit timer and the core timer remain active. An interrupt from either of the timers, if enabled, will cause the MCU to exit the WAIT mode.

During the WAIT mode, the I-bit in the CCR is cleared to enable interrupts. All other registers, memory and input/output lines remain in their previous state. The 16-bit timer or the core timer may be enabled to allow a periodic exit from the WAIT mode. See Figure 2-5.

2.3.3 Data retention

The contents of the RAM are retained at supply voltages as low as 2.0V dc. This is called the data retention mode, in which data is maintained but the device is not guaranteed to operate.

For lowest power consumption in data retention mode the device should be put into STOP mode before reducing the supply voltage, to ensure that all the clocks are stopped. If the device is not

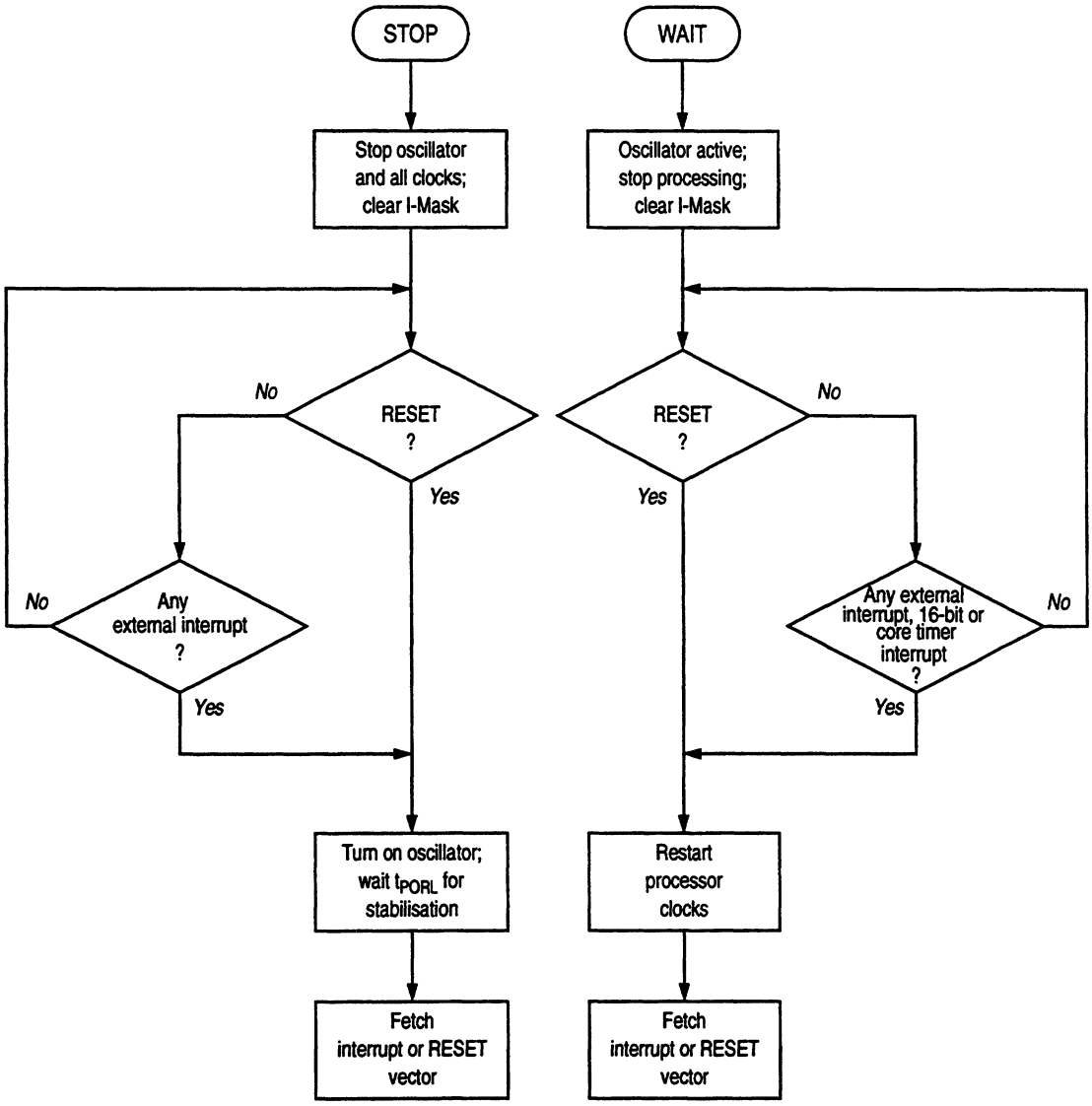


Figure 2-5 STOP and WAIT flowcharts

in STOP mode then it is recommended that $\overline{\text{RESET}}$ be held low, whilst the power supply is outwith the normal operating range, to ensure that processing is suspended in an orderly manner.

Recovery from data retention mode, after the power supply has been restored, is by an external interrupt, or by pulling the $\overline{\text{RESET}}$ line high.

3
MEMORY AND REGISTERS

The MC68HC05L2 has a 4 kbyte memory map consisting of registers (for I/O, control and status), LCD RAM, user RAM, user ROM *or EPROM*, *Bootloader ROM* and reset and interrupt vectors as shown in Figure 3-1.

3.1 Registers

All the I/O, control and status registers of the MC68HC05L2 are contained within the first 32-byte block of the memory map, as detailed in Table 3-1.

3.2 LCD RAM

The 8 bytes of LCD RAM are located at addresses $0030 to $0037. This RAM is used to store the 15 nibbles of data needed for the LCD. See Section 7.1 for further details of the organisation of these bits.

3.3 RAM

The user RAM consists of 96 bytes of memory, from $00A0 to $0100. This is shared with a 64-byte stack area. The stack begins at $00FF and may extend 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.

3.4 Non-volatile memory (NVM)

The NVM consists of 1776 bytes of ROM (MC68HC05L2) *or EPROM (MC68HC705L2)* from $0800 to $0F00, 112 bytes of bootloader ROM and 16 bytes of user vectors ($0FF0 to $0FFF).

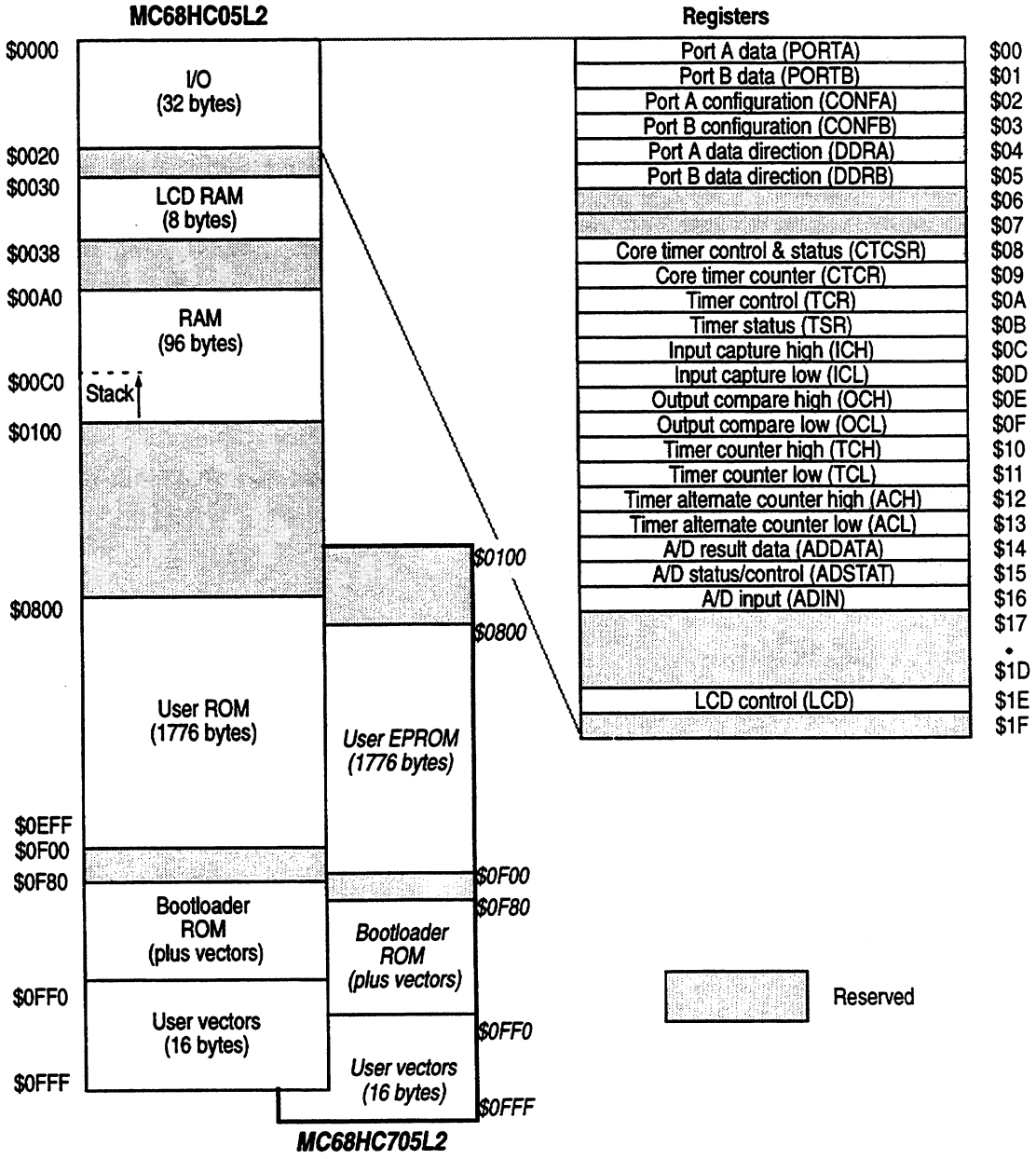


Figure 3-1 Memory map of the MC68HC05L2 and *MC68HC705L2*

3.4.1 *EPROM program/control register (PCR)*

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0070						0	ELAT	PGM	---- -000

3.4.1.1 *ELAT — EPROM latch control*

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

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

Causes address and data buses to be latched when a write to EPROM is carried out. EPROM cannot be read if ELAT = 1. This bit should not be set when no programming voltage is applied to the VPP pin.

3.4.1.2 *PGM — EPROM program control*

1 (set) – *Programming power connected to the EPROM array. PGM can only be set if ELAT is set. PGM is automatically cleared when ELAT = 0.*

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

Note: *ELAT and PGM cannot be set on the same write operation.*

Take the following steps to program a byte of EPROM:

1) Apply the programming voltage V_{PP} to the VPP pin.
2) Set the ELAT bit.
3) Write to the EPROM address.
4) Set the PGM bit for a time t_{PROG} to apply the programming voltage.
5) Clear the ELAT bit.

Note: The erased state of the EPROM is $FF.

bit 3	bit 2	bit 1	bit 0	State on Reset
				unaffected
				unaffected
				0000 0000
				0000 0000
				0000 0000
				0000 0000
COPE	IRQS	RT1	RT0	0000 0011
				0000 0000
0	0	IEDG	0	0000 00U0
0	0	0	0	uuu0 0000
			(bit 8)	unaffected
				unaffected
			(bit 8)	unaffected
				unaffected
			(bit 8)	1111 1111
				1111 1100
			(bit 8)	1111 1111
				1111 1100
				0000 0000
CH3	CH2	CH1	CH0	0000 0000
0	0	0	ADIN	0000 000U
FDISP	MUX4	MUX3	DISON	0000 0000
	0	ELAT	PGM	---- -000

4
INPUT/OUTPUT PORTS

In single-chip mode the MC68HC05L2 has a total of 13 I/O lines, arranged as one 8 bit I/O port (A) and one 5 bit I/O port (B). Each I/O line is individually programmable as either input or output, under software control of the data direction registers. In addition, each port pin can be connected to an internal pull-up resistor of approximately 10 – 30kΩ when configured as an input. This is controlled using the bits of the port configuration register.

To avoid glitches on the output pins, data should be written to the I/O port data register before writing ones to the corresponding data direction register bits to set the pins to output mode.

4.1 Input/output programming

The bidirectional port lines may be programmed as inputs or outputs under software control. The direction of each pin 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, thus configuring all port pins as inputs. The data direction registers can be 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. If the I/O pin is an input and a read-modify-write instruction is executed, the I/O pin will be read into the MC68HC05L2 CPU and the computed result will then be written to the data latch.

In addition to the normal I/O function, each port pin can be connected to an internal pull-up resistor (approximately 10 – 30kΩ) when configured as an input. This is controlled via the port configuration register (see Table 4-2).

The operation of the standard port hardware is shown schematically in Figure 4-1.

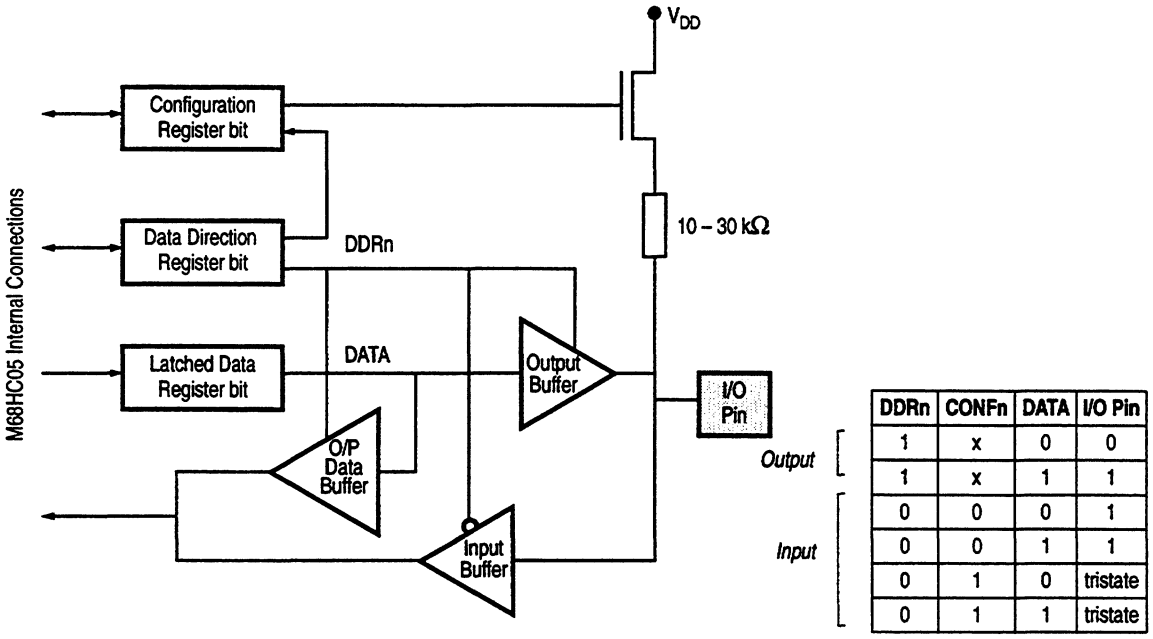


Figure 4-1 Standard I/O port structure

Table 4-1 shows the effect of reading from or writing to an I/O pin in various circumstances. Note that the read/write signal shown is internal and is not available to the user.

Table 4-1 I/O pin states

R/W	DDRn	CONFn	Action of MCU write to/read of data bit
0	0	0	The I/O pin is in input mode and is connected to an internal pull-up resistor. Data is written into the output data latch.
0	0	1	The I/O pin is in input mode. Data is written into the output data latch.
0	1	x	Data is written into the output data latch, and output to the I/O pin.
1	0	0	The state of the I/O pin is read. An internal pull-up resistor is connected to the input pin.
1	0	1	The state of the I/O pin is read.
1	1	x	The I/O pin is in output mode. The output data latch is read.

4.2 Port A

Port A is an 8-bit bidirectional port, with optional internal pull-up resistors, which comprises a data register (PORTA), a configuration register (CONFA) and a data direction register (DDRA). Reset does not affect the data register, but clears the data direction register and the configuration register, thereby returning the ports to inputs with pull-ups (see Table 4-2).

4.3 Port B

Port B is a 5-bit bidirectional port with optional internal pull-up resistors, which comprises a data register (PORTB), a configuration register (CONFB) and a data direction register (DDRB). Reset does not affect the data register, but clears the data direction register and the configuration register, thereby returning the ports to inputs with pull-ups (see Table 4-2)

4.4 Port registers

The following sections explain in detail the individual bits in the data and control registers associated with each I/O port.

4.4.1 Port A data register (PORTA)

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

Each bit can be configured as input or output via the corresponding data direction bit in the port data direction register (DDRA).

4.4.2 Port B data register (PORTB)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0001	0	0	0						unaffected

Each bit can be configured as input or output via the corresponding data direction bit in the port data direction register (DDRA). Bits 5, 6 and 7 are not available and always read 0.

4.4.3 Port A configuration register (CONFA)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0002									0000 0000

Providing the DDRA bit is clear, i.e. the port pin is an input, writing a '0' to any bit connects an internal pull-up resistor (approximately 10 – 30 kΩ) to the corresponding port B pin. Writing a '1' disconnects the pull-up resistor from the port pin (see Table 4-2).

Reset clears this register, thus connecting pull-up resistors to all port A pins.

Table 4-2 I/O port configurations

DDRn	CONFn	Function
0	0	Input with pull-up
0	1	Input without pull-up
1	0	Output
1	1	Output

4.4.4 Port B configuration register (CONFB)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0003	0	0	0						0000 0000

Providing the DDRB bit is clear, i.e. the port pin is an input, writing a '0' to any bit connects an internal pull-up resistor (approximately 10 – 30 kΩ) to the corresponding port B pin. Writing a '1' disconnects the pull-up resistor from the port pin (see Table 4-2). Writing to bit 5, 6 or 7 has no meaning or effect as these bits always read zero.

Reset clears this register, thus connecting pull-up resistors to all port B pins.

4.4.5 Port A data direction register (DDRA)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0004									0000 0000

Writing a '1' to any bit configures the corresponding port A pin as an output; conversely, writing any bit to '0' configures the corresponding port pin as an input.

Reset clears this register, thus configuring all pins as inputs.

4.4.6 Port B data direction register (DDRB)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0005	0	0	0						0000 0000

Writing a '1' to any bit configures the corresponding port B pin as an output; conversely, writing any bit to '0' configures the corresponding port pin as an input. Writing to bit 5, 6 or 7 has no meaning or effect as these bits always read zero.

Reset clears this register, thus configuring all pins as inputs.

4.5 Other port considerations

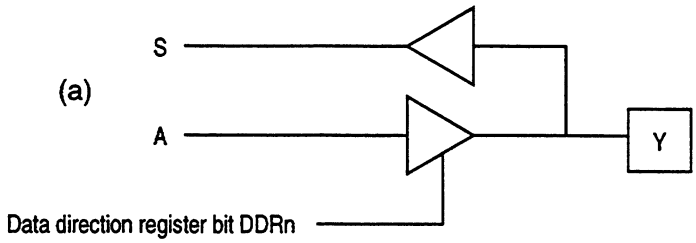
All ports are latched with the bus timing signal P02 (especially for inputs).

All output ports can emulate open drain outputs when the corresponding configuration register bits are set to '1'. This is achieved by writing a zero to the relevant output port latch. By toggling the corresponding data direction bit, the port pin will either be an output zero or tri-state (an input). This is shown diagrammatically in Figure 6-5.

When using a port pin as an open-drain output, certain precautions must be taken in the user software. If a read-modify-write instruction is used on a port where the open-drain is assigned and the pin at this time is programmed as an input, it will read it as a 'one'. The read-modify-write instruction will then write this 'one' into the output data latch on the next cycle. This would cause the open-drain pin not to output a 'zero' when desired

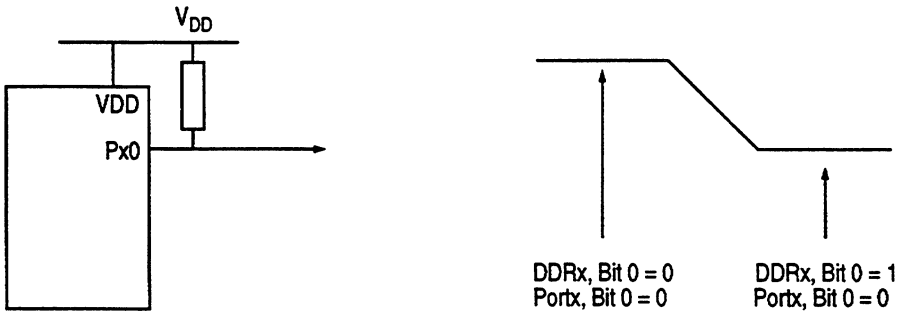
Note: Open-drain' outputs should not be pulled above V_{DD}.

(a)



(b)

DDRn	A	Y	
1	0	0	Normal operation – tristate
1	1	1	
0	0	tri state	
0	1	tri state	
1	0	low	Open drain
1	1	---	
0	0	high	
0	1	high	



(c)

Figure 4-2 Port logic levels

5
CORE TIMER

The MC68HC05L2 has a 15-stage ripple counter called the core timer (CTIMER). Features of this timer are: timer overflow, power-on reset (POR), real time interrupt (RTI), with four selectable interrupt rates, and a computer operating properly (COP) watchdog timer.

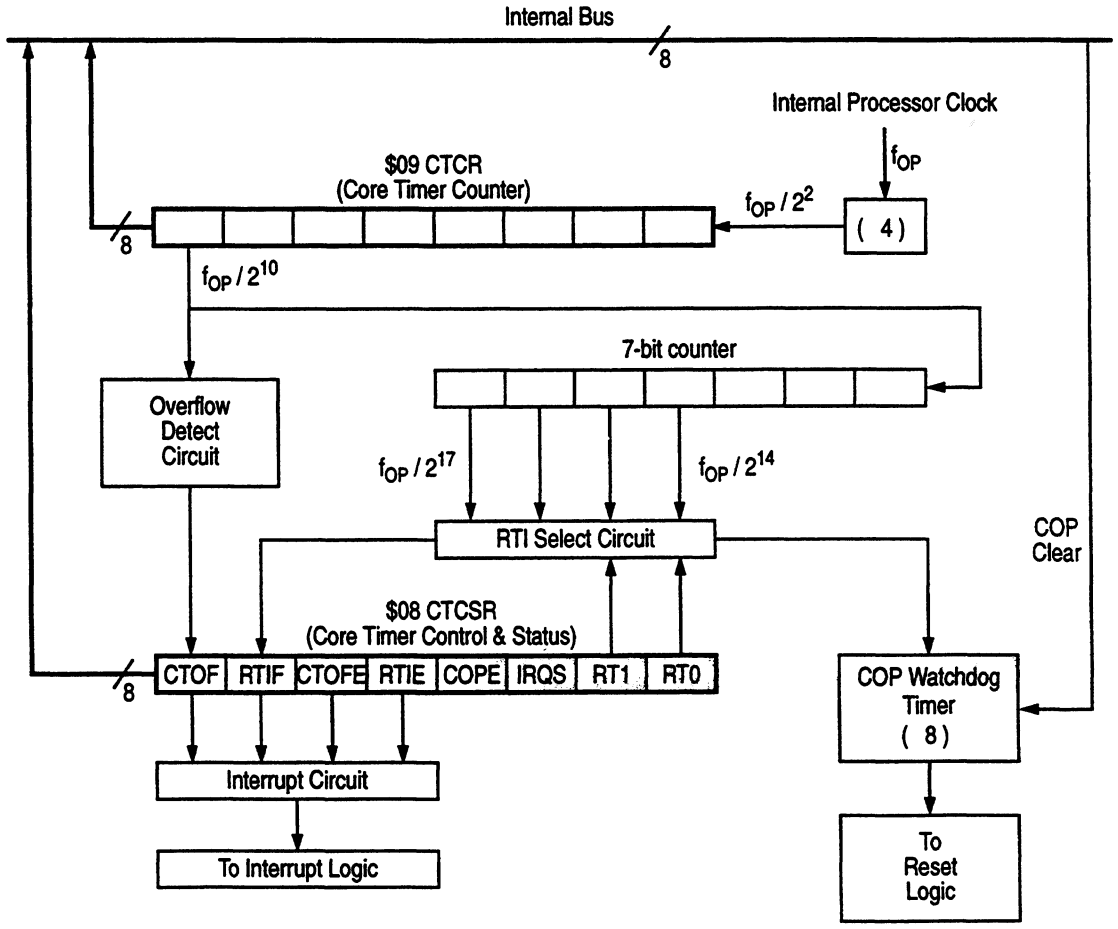


Figure 5-1 Core timer block diagram

As seen in Figure 5-1, 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 $f_{OP}/1024$. (The POR signal (t_{PORL}) is also derived from this register, at $f_{OP}/4064$.) The Counter Register circuit is followed by four more stages, with the resulting clock ($f_{OP}/16384$) 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 8 to drive the COP Watchdog Timer circuit. The RTI rate selector bits, and the RTI and CTIMER overflow enable bits and flags, are located in the CTIMER Control and Status Register (CTCSR) at location $08.

CTOF (core tmer overflow flag) is a clearable, read-only status bit and is set when the 8-bit ripple counter rolls over from $FF to $00. A CPU interrupt request will be generated if CTOFE is set. Clearing the CTOF is done by writing a '0' to it. Writing a '1' to CTOF has no effect on the bit's value. Reset clears CTOF.

When CTOFE (core timer overflow enable) is set, a CPU interrupt request is generated when the CTOF bit is set. Reset clears CTOFE.

The core timer counter register (CTCR) is a read-only register that contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked at $f_{OP}/4$ and can be used for various functions including a software input capture. Extended time periods can be attained using the CTIMER overflow function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter.

The power-on cycle clears the entire counter chain and begins clocking the counter. After t_{PORL} cycles, the power-on reset circuit is released, which again clears the counter chain and allows the device to come out of 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. When $\overline{\text{RESET}}$ is asserted at any time during operation (other than POR), the counter chain will be cleared.

5.1 Real time interrupts (RTI)

The Real Time Interrupt circuit consists of a three stage divider and a 1-of-4 selector. The clock frequency that drives the RTI circuit is $f_{OP}/2^{14}$ (or $f_{OP}/16384$), with three additional divider stages, giving a maximum interrupt period of 4 seconds at a bus frequency (f_{OP}) of 32kHz. Register details are given in Section 5.3.

5.2 Computer operating properly (COP) watchdog timer

The COP watchdog timer function is implemented by taking the output of the RTI circuit and further dividing it by eight, as shown in Figure 5-1. Note that the minimum COP timeout period is

seven times the RTI period. This is because the COP will be cleared asynchronously with respect to the value in the Core Timer counter register/RTI divider, hence the actual COP timeout period will vary between 7x and 8x the RTI period.

The COP function is selectable via the COPE bit in the core timer control and status register.

If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. Preventing a COP timeout is done by writing a '0' to bit 0 of address $0FF0. When the COP is cleared, only the final divide-by-eight stage is cleared (see Figure 5-1).

5.3 Core timer registers

5.3.1 Core timer control and status register (CTCSR)

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

5.3.1.1 CTOF — Core timer overflow

1 (set) – Core timer overflow has occurred.

0 (clear) – No core timer overflow interrupt has been generated.

This bit is set when the core timer counter register rolls over from $FF to $00; an interrupt request will be generated if CTOFE is set. When set, the bit may be cleared by writing a '0' to it.

5.3.1.2 RTIF — Real time interrupt flag

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

0 (clear) – No real time interrupt has been generated.

This bit is set when the output of the chosen stage becomes active; an interrupt request will be generated if RTIE is set. When set, the bit may be cleared by writing a '0' to it.

5.3.1.3 CTOFE — Core timer overflow enable

1 (set) – Core timer overflow interrupt is enabled.

0 (clear) – Core timer overflow interrupt is disabled.

Setting this bit enables the core timer overflow interrupt. A CPU interrupt request will then be generated whenever the CTOF bit becomes set. Clearing this bit disables the core timer overflow interrupt capability.

5.3.1.4 RTIE — Real time interrupt enable

1 (set) – Real time interrupt is enabled.

0 (clear) – Real time interrupt is disabled.

Setting this bit enables the real time interrupt. A CPU interrupt request will then be generated whenever the RTIF bit becomes set. Clearing this bit disables the real time interrupt capability.

5.3.1.5 COPE — Computer operating properly enable

1 (set) – COP watchdog enabled.

0 (clear) – COP watchdog disabled.

When set, COPE turns on the COP watchdog. This bit can be written to only once after reset to lock in the desired function. Reset clears COPE.

5.3.1.6 IRQS — $\overline{\text{IRQ}}$ sensitivity selection

1 (set) – $\overline{\text{IRQ}}$ is negative edge sensitive only.

0 (clear) – $\overline{\text{IRQ}}$ is negative edge-and-level sensitive.

This bit selects the type of input recognised by the $\overline{\text{IRQ}}$ pin. IRQS can be written to only once after reset to lock in the desired function. A read of this bit always returns a zero regardless of its state. Reset clears IRQS.

5.3.1.7 RT1, RT0 — Real time interrupt rate select

These two bits select one of four taps from the Real Time Interrupt circuitry. Reset sets both RT0 and RT1 to one, selecting the lowest periodic rate and therefore the maximum time in which to alter them if necessary. The COP reset times are also determined by these two bits. Care should be taken when altering RT0 and RT1 if a timeout is imminent, or the timeout period is uncertain. If the selected tap is modified during a cycle in which the counter is switching, an RTIF could be missed or an additional one could be generated. To avoid problems, the COP should be cleared before changing the RTI taps. See Table 5-1 for some example RTI periods.

Table 5-1 Example RTI periods

			Bus frequency f_{OP} = 2 MHz	
RT1	**RT0**	**Division Ratio**	**RTI Period**	**Minimum COP Period**
0	0	2^{14}	8.2ms	57.3ms
0	1	2^{15}	16.4ms	114.7ms
1	0	2^{16}	32.8ms	229.4ms
1	1	2^{17}	65.5ms	458.8ms

5.3.2 Core timer counter register (CTCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0009									0000 0000

The core timer counter register is a read-only register, which contains the current value of the 8-bit ripple counter at the beginning of the timer chain.

Reset clears this register.

5.4 Core timer during WAIT

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

5.5 Core timer during STOP

The timer is cleared when going into STOP mode. When STOP is exited by an external interrupt or an external reset, the internal oscillator will restart, followed by an internal processor stabilisation delay (t_{PORL}). The timer is then cleared and operation resumes.

THIS PAGE LEFT BLANK INTENTIONALLY

6
PROGRAMMABLE TIMER

Besides the core timer the MC68HC05L2 has a 16-bit programmable timer. This timer consists of a 16-bit read-only free-running counter, with a fixed divide-by-four prescaler, plus the input capture/output compare circuitry. Selected input edges cause the current counter value to be latched into a 16-bit input capture register so that software can later read this value to determine when the edge occurred. When the free running counter value matches the value in the output compare registers, an output compare interrupt will occur provided the OCIE bit is set. This results in the output compare flag (OCF) being set, signalling that a successful compare has taken place. Refer to Figure 6-1 for a block diagram of the timer.

Note: In order to maintain consistency with the naming conventions used in other M68HC05 family documents, the compare function on the MC68HC05L2 timer is referred to as 'output compare' although there is no output compare pin. The absence of a dedicated output compare pin means that any desired action resulting from a successful compare must be controlled in software using the output compare interrupt facility. This software could also nominate a port pin on which to output the result.

The timer has a 16-bit architecture, hence each specific functional segment is represented by two 8-bit registers. These registers contain the high and low byte of that functional segment. Accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

Note: The I-bit in the CCR 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.

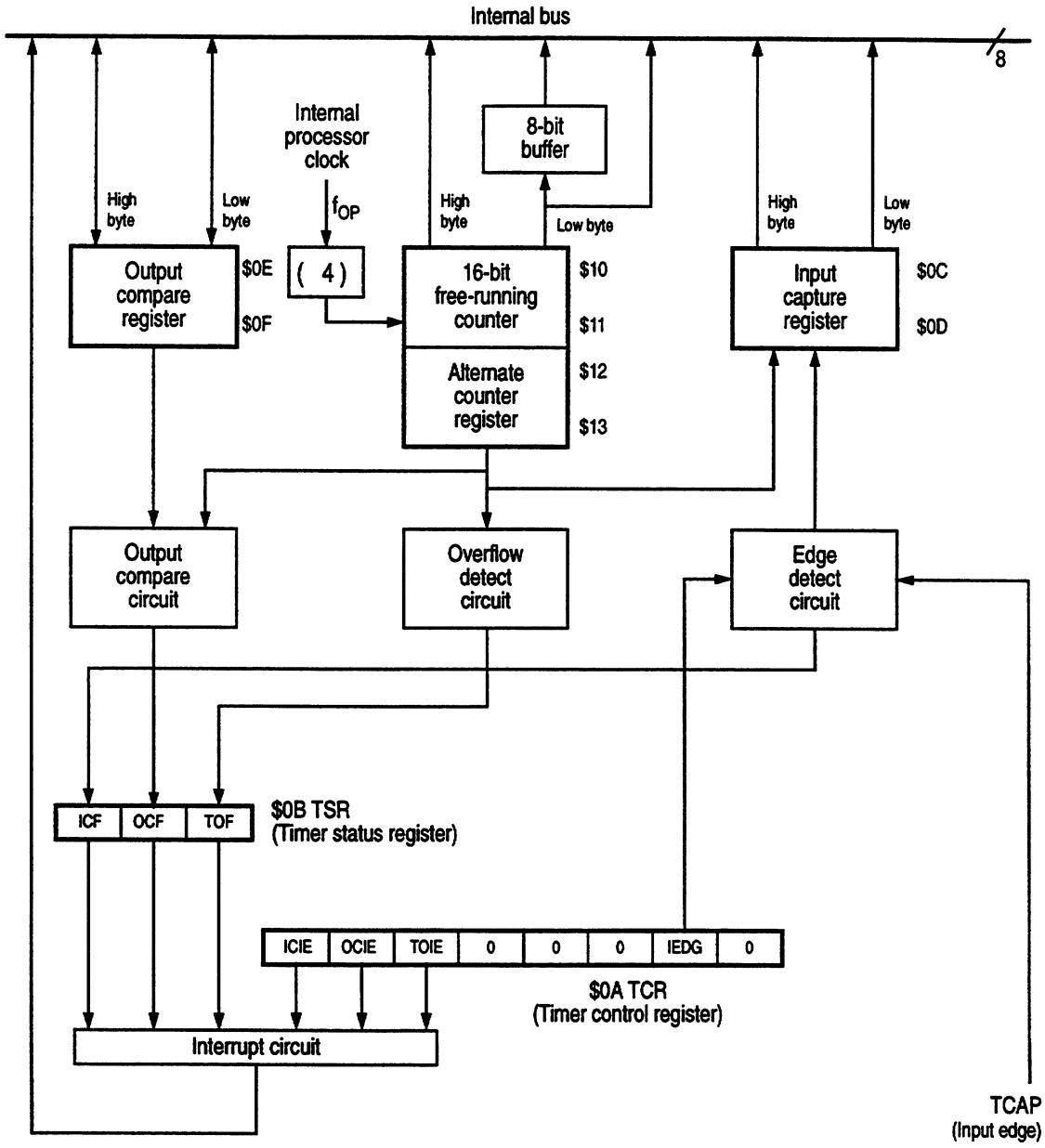


Figure 6-1 16-bit programmable timer block diagram

6.1 Counter

The key element in the programmable timer is a 16-bit, free-running counter, or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2 s if the internal bus clock is 2 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

6.1.1 Counter high register ($0010)
Counter low register ($0011)
Alternate counter high register ($0012)
Alternate counter low register ($0013)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0010	(bit 15)							(bit 8)	1111 1111
$0011									1111 1100
$0012	(bit 15)							(bit 8)	1111 1111
$0013									1111 1100

The double-byte, free-running counter can be read from either of two locations, $10 – $11 (counter register) or $12 – $13 (alternate counter register). A read from only the less significant byte (LSB) of the free-running counter ($19 or $1B) receives the count value at the time of the read. If a read of the free-running counter or alternate counter register first addresses the more significant byte (MSB) ($10 or $12), the LSB is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or alternate counter register LSB and thus completes a read sequence of the total counter value. In reading either the free-running counter or alternate counter register, if the MSB is read, the LSB must also be read to complete the sequence. If the timer overflow flag (TOF) is set when the counter register LSB is read, then a read of the TSR will clear the flag.

The alternate counter register differs from the counter register only in that a read of the LSB does not clear TOF. Therefore, to avoid the possibility of missing timer overflow interrupts due to clearing of TOF, the alternate counter register should be used where this is a critical issue.

The free-running counter is set 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. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, 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 is set.

6.1.1.1 Bits 8 – 15 — MSB of counter/alternate counter register

A read of only the more significant byte (MSB) transfers the LSB to a buffer, which remains fixed after the first MSB read, until the LSB is also read.

6.1.1.2 Bits 0 – 7 — LSB of counter/alternate counter register

A read of only the less significant byte (LSB) receives the count value at the time of reading.

6.2 Timer functions

The 16-bit programmable timer is monitored and controlled by a group of ten registers, full details of which are contained in the following paragraphs. An explanation of the timer functions is also given.

6.2.1 Timer control register (TCR)

The timer control register ($0A) is used to enable the input capture (ICIE), output compare (OCIE), and timer overflow (TOIE) interrupt enable functions as well as selecting input edge sensitivity (IEDG).

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$000A	ICIE	OCIE	TOIE	0	0	0	IEDG	0	0000 00U0

6.2.1.1 ICIE — Input capture interrupt enable

1 (set) – Input capture interrupt enabled.

0 (clear) – Input capture interrupt disabled.

6.2.1.2 OCIE — Output compare interrupt enable

1 (set) – Output compare interrupt enabled.

0 (clear) – Output compare interrupt disabled.

6.2.1.3 TOIE — Timer overflow interrupt enable

1 (set) – Timer overflow interrupt enabled.

0 (clear) – Timer overflow interrupt disabled.

6.2.1.4 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 register. When clear, a negative-going edge triggers the transfer.

6.2.2 Timer status register (TSR)

The timer status register ($0B) contains the status bits for the above three interrupt conditions — ICF, OCF, TOF.

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

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$000B	ICE	OCF	TOF	0	0	0	0	0	uuu0 0000

6.2.2.1 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 ($0D).

6.2.2.2 OCF — Output compare flag

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

0 (clear) – No output compare has occurred.

This bit is set when the output compare register contents match those 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 ($0F).

6.2.2.3 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 is set. TOF is cleared by reading the TSR and the counter low register ($11).

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 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.

6.2.3 Input capture function

'Input capture' is a technique whereby an external signal (connected to the 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.

6.2.4 Input capture high register ($000C)
Input capture low register ($000D)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$000C	(bit 15)							(bit 8)	unaffected
$000D									unaffected

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 input capture edge detector senses a valid transition. The level transition that triggers the counter transfer is defined by the input edge bit

(IEDG). The most significant 8 bits are stored in the input capture high register at $0C, the least significant in the input capture low register at $0D.

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 synchronisation. 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 ($0C), the counter transfer is inhibited until the LSB ($0D) 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 ($0D) does not inhibit the free-running counter transfer since the two actions occur on opposite edges of the internal bus clock.

Reset does not affect the contents of the input capture register, except when exiting STOP mode.

6.2.5 Output compare function

'Output compare' is a technique that may be used, for example, to generate an output waveform, or to signal when a specific time period has elapsed, by presetting the output compare register to the appropriate value.

The absence of an output compare pin on the MC68HC05L2 means that any desired action resulting from a successful compare must be controlled in software using the output compare interrupt facility. This software can also nominate a port pin on which to output the result.

6.2.6 Output compare high register ($000E)
Output compare low register ($000F)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$000E	(bit 15)							(bit 8)	unaffected
$000F									unaffected

The 16-bit output compare register is made up of two 8-bit registers at locations $0E (MSB) and $0F (LSB). 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 values should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding 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 ($0E), the output compare function is inhibited until the LSB ($0F) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($OF) will not inhibit the compare function. The processor can write to either byte of the 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 the output compare register are not defined at power on, and not affected by reset, care must be taken when initialising output compare functions with software. The following procedure is recommended:

1) write to output compare high to inhibit further compares;
2) read the timer status register to clear OCF (if set);
3) write to output compare low to enable the output compare function.

All bits of the output compare register are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

6.3 Timer during WAIT mode

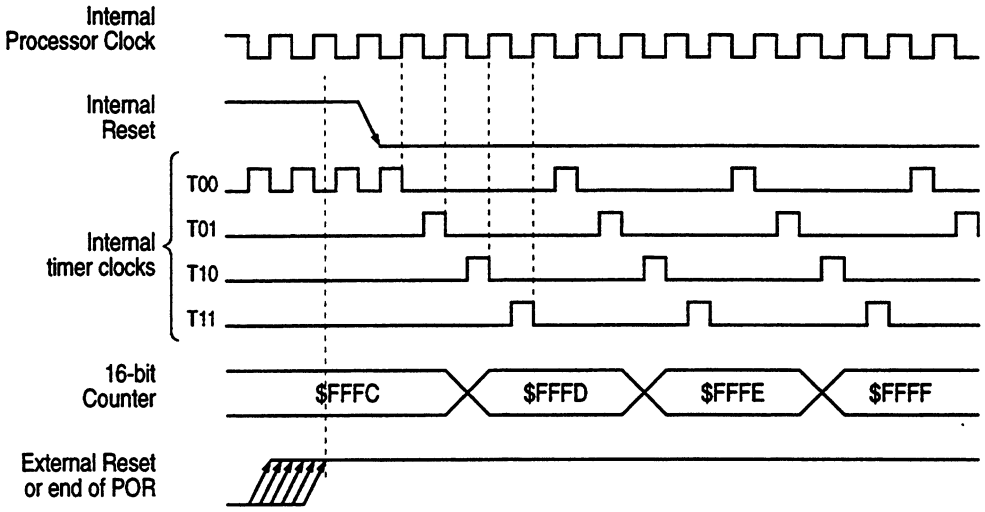
All CPU action is suspended, but the timers (core and 16-bit) remain active. An interrupt from either of the timers, if enabled, will cause the MCU to exit WAIT mode.

6.4 Timer during STOP mode

In the STOP mode all MCU clocks are stopped, hence the timer stops counting. If STOP is exited by an interrupt the counter retains the last count value. If the device is reset, then the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU. When the MCU does wake up, however, there is an active input capture flag and data from the first valid edge that occurred during the STOP period. If the device is reset to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

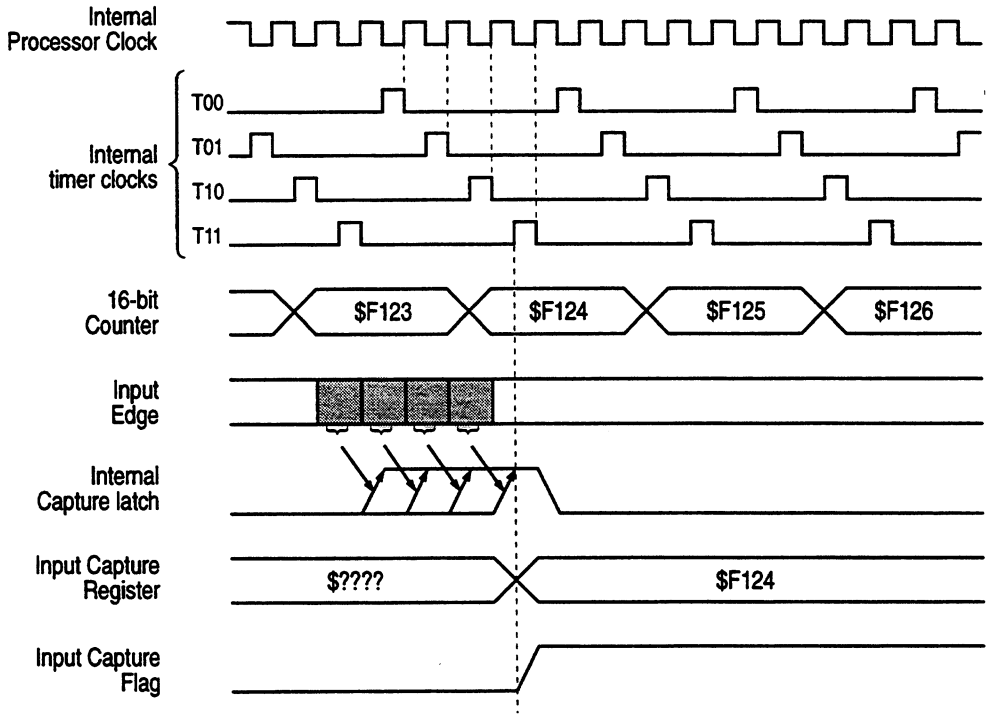
6.5 Timer state diagrams

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



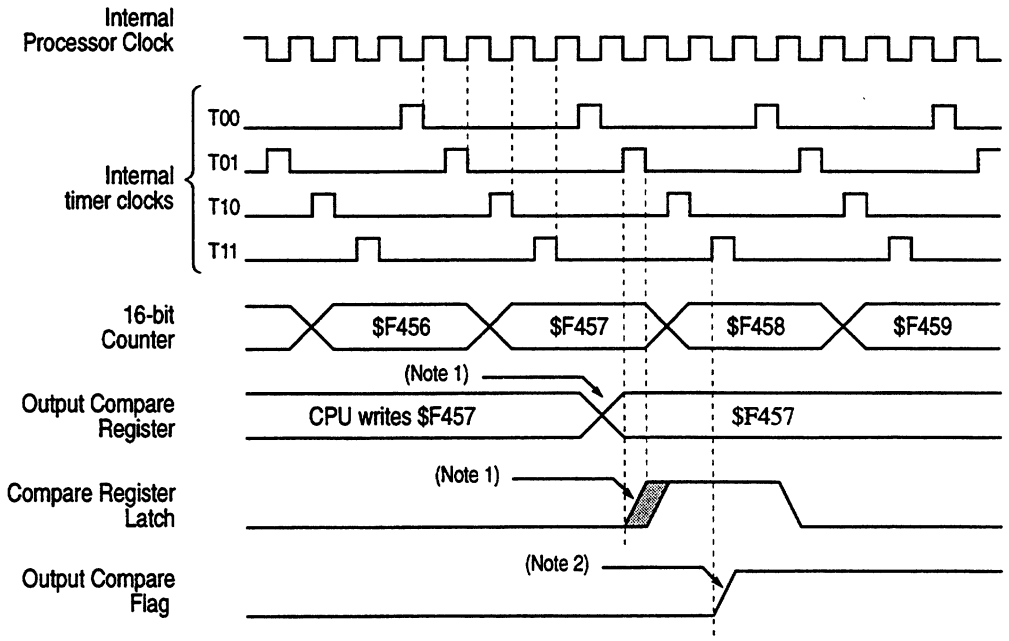
Note: The counter and timer control registers are the only ones affected by power-on or external reset.

Figure 6-2 Timer state timing diagram for reset



Note: If the input edge occurs in the shaded area from one timer state T10 to the next timer state T10, then the input capture flag will be set during the next T11 state.

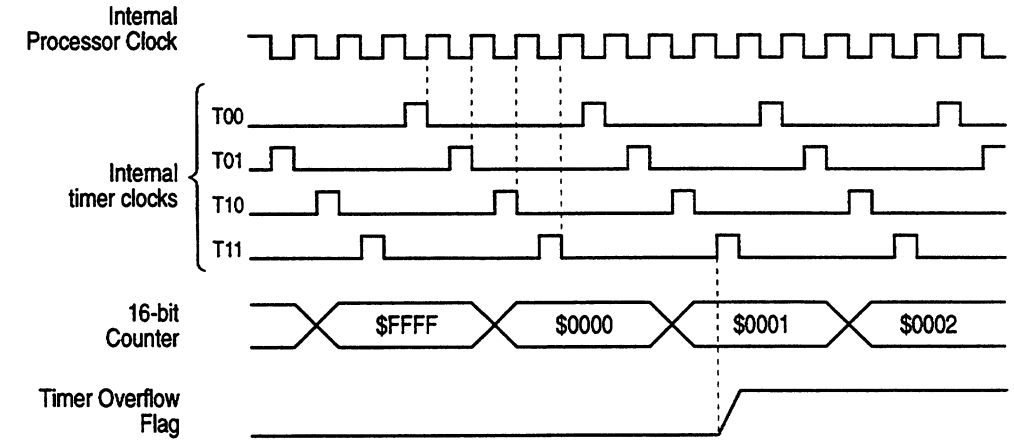
Figure 6-3 Timer state timing diagram for input capture



Note: (1)The CPU write to the compare registers may take place at any time, but a compare only occurs at timer state T01. Thus a four 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 ($F457 in this example).

Figure 6-4 Timer state timing diagram for output compare



Note: The timer overflow flag 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 timing diagram for timer overflow

7
LIQUID CRYSTAL DISPLAY DRIVER

The LCD driver module on the MC68HC05L2 can be configured with up to 15 frontplane drivers and up to 3 backplane drivers. This allows a maximum of 45 LCD segments to be driven. Each segment is controlled by a corresponding bit in the LCD RAM. At reset or on power-up, the drivers are configured in the default duplex mode, 1/2 bias with 2 backplanes and 13 frontplanes. At this stage all Port D pins that are shared with the LCD subsystem are configured as input only pins. Also at power-up or reset the ON/OFF control for the display (the DISON bit in the LCD Control (LCD) register) is cleared thus disabling the LCD drivers. Figure 7-1 shows a block diagram of the LCD system.

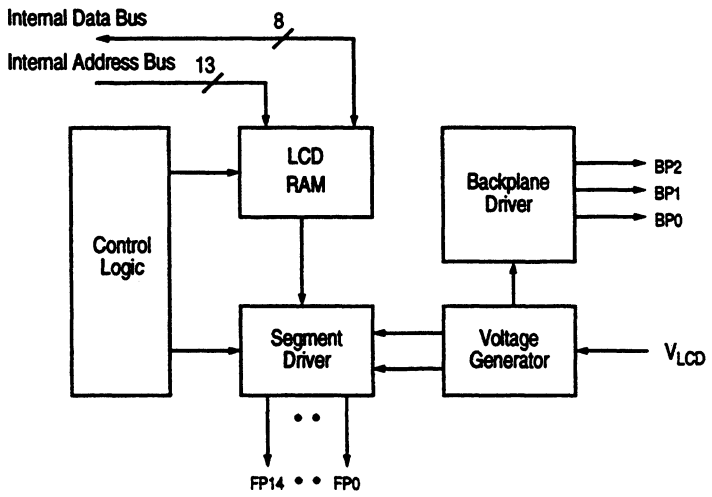


Figure 7-1 LCD block diagram

7.1 LCD RAM

Data to be displayed on the LCD must be written into the LCD RAM. The LCD RAM consists of 8 bytes of RAM (in the MC68HC05L2's memory map) at $0030-$0037. The 45 bits in the LCD RAM correspond to the 45 segments that can be driven by the frontplane/backplane drivers. Table 7-1 shows how the LCD RAM is organised. Writing a "1" to a given location will result in the corresponding display segment being activated when the DISON bit is set. The LCD RAM is a dual port RAM that interfaces with the internal address and data buses of the MCU. It is possible to read from LCD RAM locations for scrolling purposes. When DISON = 0, the LCD RAM can be used as main on-chip RAM.

Table 7-1 LCD RAM organisation

LCD RAM Address	Data							
	7	6	5	4	3	2	1	0
$0030	–	FP1–BP2	FP1–BP1	FP1–BP0	–	FP0–BP2	FP0–BP1	FP0–BP0
$0031	–	FP3–BP2	FP3–BP1	FP3–BP0	–	FP2–BP2	FP2–BP1	FP2–BP0
$0032	–	FP5–BP2	FP5–BP1	FP5–BP0	–	FP4–BP2	FP4–BP1	FP4–BP0
$0033	–	FP7–BP2	FP7–BP1	FP7–BP0	–	FP6–BP2	FP6–BP1	FP6–BP0
$0034	–	FP9–BP2	FP9–BP1	FP9–BP0	–	FP8–BP2	FP8–BP1	FP8–BP0
$0035	–	FP11–BP2	FP11–BP1	FP11–BP0	–	FP10–BP2	FP10–BP1	FP10–BP0
$0036	–	FP13–BP2	FP13–BP1	FP13–BP0	–	FP12–BP2	FP12–BP1	FP12–BP0
$0037	–	–	–	–	–	FP14–BP2	FP14–BP1	FP14–BP0

7.2 LCD operation

The LCD driver module can operate in three modes providing different multiplex ratios and number of backplanes as follows:

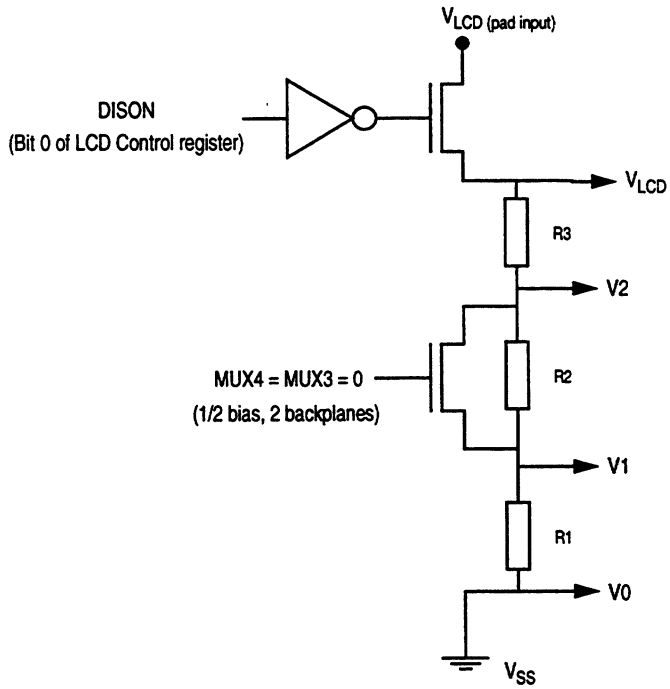
- 1/2 bias, 2 backplanes
- 1/3 bias, 2 backplanes
- 1/3 bias, 3 backplanes

The operating mode is selected using the multiplex ratio bits (MUX3 and MUX4) in the LCD control register as shown in Table 7-2.

It is recommended that the DISON bit in the LCD register is cleared (to inhibit the LCD drivers) until the multiplex rate is selected. The voltage levels required for the different multiplex rates are generated internally by a resistive divider chain between V_{LCD} and V_{SS}. When DISON is set, the display is switched on and the resistive divider chain activated with a maximum voltage, V_{LCD}, and a minimum, V_{SS}. V_{LCD} allows specific voltage thresholds to be applied to the LCD and can be any voltage up to V_{DD}. The 2-way multiplex with 1/3 bias and the 3-way multiplex options require four

voltage levels, whereas the 2-way multiplex with 1/2 bias needs only three levels. Figure 7-2 shows the resistive divider chain network that is used to produce the various LCD waveforms outlined in Section 7.3.

Caution: V_{LCD} may not exceed the positive power supply voltage V_{DD}.



Note: In the 1/2 bias, 2 backplane mode, R2 is shorted out and V2 = V1.

Figure 7-2 Voltage level selection

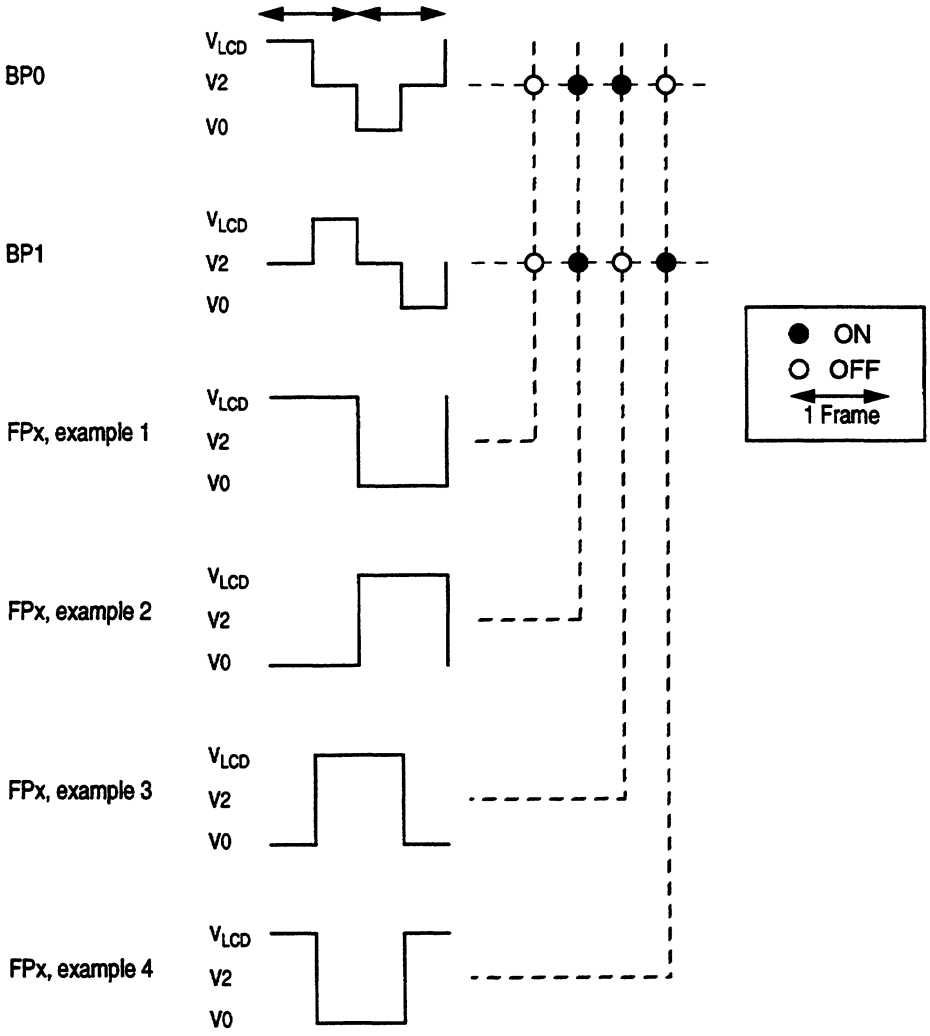
7.3 Timing signals and LCD voltage waveforms

The LCD timing signals are all derived from the main system clock; with a bus frequency of 2 MHz (f_{osc} = 4 MHz) the frame rate will be 61 Hz for 2-way multiplexing and 91 Hz for 3-way multiplexing (see Table 7-2). An extra divide by two stage can be included in the LCD clock generator by setting FDISP in the LCD register. This will result in the frame rate being halved. For example, when 3-way multiplexing is being used, a frame rate of 45.5 Hz instead of 91 Hz can be obtained. Alternatively, the same frame rates can be achieved using a higher bus frequency. See Section 7.4.1.

Figure 7-3 to Figure 7-5 show the backplane waveforms and some examples of frontplane waveforms for each of the operating modes.

The backplane waveforms are continuous and repetitive (every 2 frames); they are fixed within each operating mode and are not affected by the data in the LCD RAM.

The frontplane waveforms are dependent on the LCD segments to be driven as defined in the LCD RAM. Each 'on' segment must have a differential driving voltage (BP-FP) applied to it once in each frame; the LCD driver module hardware uses the data in the LCD RAM to construct the frontplane waveform to meet this criterion.



Note: In this mode V1=V2

Figure 7-3 LCD waveform with 2 backplanes, 1/2 Bias

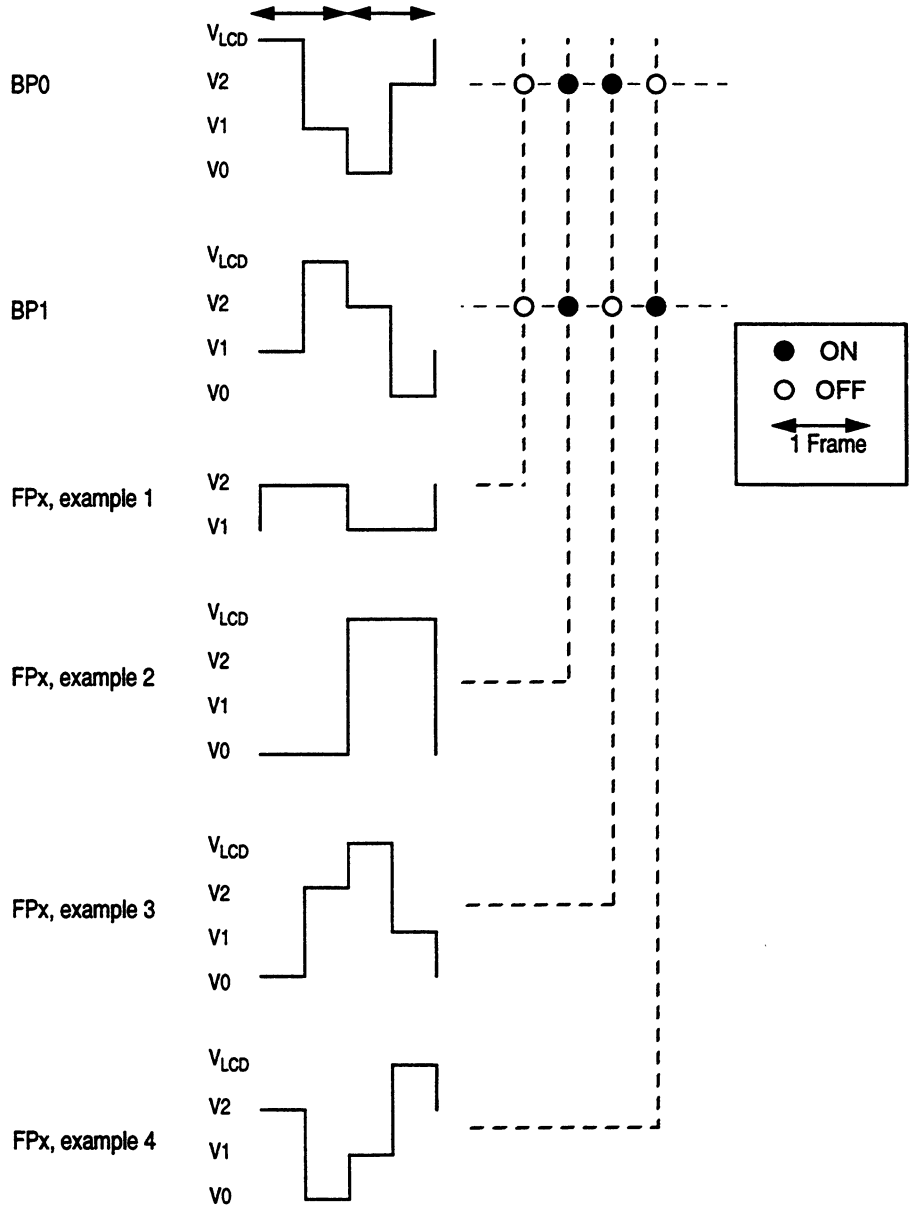


Figure 7-4 LCD waveform with 2 backplanes, 1/3 bias

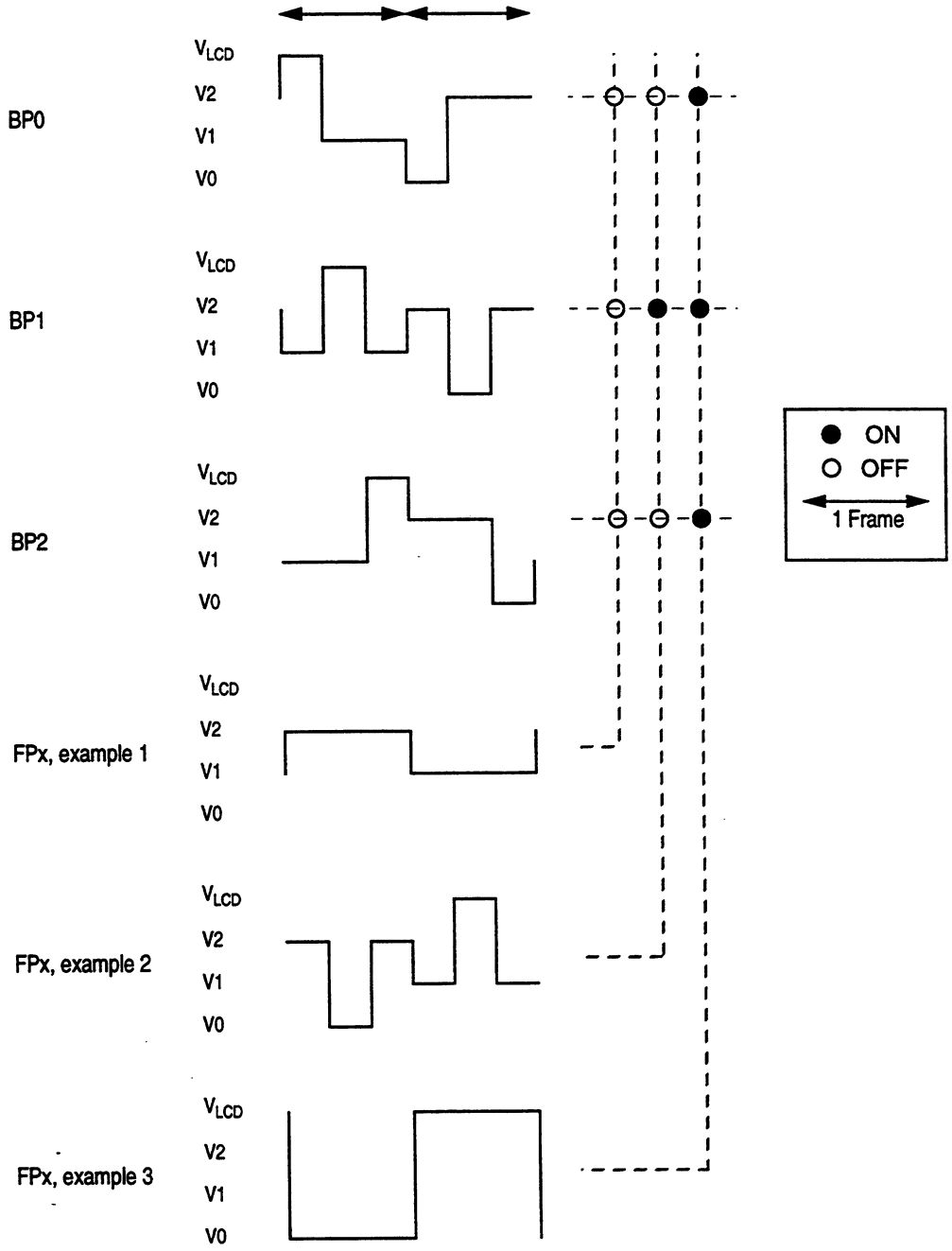


Figure 7-5 LCD waveform with 3 backplanes

7.4 LCD registers

7.4.1 LCD control register

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$001E	0	0	0	0	FDISP	MUX4	MUX3	DISON	0000 0000

7.4.1.1 FDISP — Display frequency

1 (set) – An extra divide by two stage is included in the LCD clock generator to give a reduced frame rate. For example, in the 3-way multiplexing mode, a frame rate of 45.5 Hz instead of 91 Hz can be achieved. Alternatively, the FDISP bit can be used to accommodate a higher bus frequency while maintaining the original frame rates of 61 and 91 Hz.

0 (clear) – No divider included in the LCD clock; default frame rate is used.

7.4.1.2 MUX4, MUX3 — Multiplex ratio

These two bits select the multiplex ratio to be either 2 or 3 backplanes.

Table 7-2 Multiplex ratio/backplane selection

MUX4	MUX3	BACKPLANES	BIAS	FREQUENCY
0	0	2	1/2	61 Hz
0	1	3	1/3	91 Hz
1	0	–	–	–
1	1	2	1/3	61 Hz

7.4.1.3 DISON — LCD ON/OFF

1 (set) – LCD is on.

0 (clear) – LCD is off.

7.5 LCD during WAIT mode

During WAIT mode, the LCD drivers function normally, keeping the display active if the DISON bit in the LCD register is set.

8
ANALOG TO DIGITAL CONVERTER

The analog to digital converter system consists of a single 8-bit successive approximation converter and a 9-channel multiplexer. One of the channels is connected to the ADIN pin of the MC68HC05L2 and the other eight channels are dedicated to internal reference points for test functions. The ADIN pin does not have any internal output driver circuitry connected to it because this circuitry would load the analog input signal due to output buffer leakage current. There is one 8-bit result data register (address $14) and one 8-bit status/control register (address $15).

The A/D converter is ratiometric and two dedicated pins, VRH and VRL, are used to supply the reference voltage levels of each analog input. These pins are used in preference to the system power supply lines because any voltage drops in the bonding wires of the heavily loaded supply pins could degrade the accuracy of the A/D conversion. An input voltage equal to or greater than V_{RH} converts to $FF (full scale) with no overflow indication and an input voltage equal to V_{RL} converts to $00.

The A/D converter can operate from either the bus clock or an internal RC type oscillator. The internal RC type oscillator is activated by the ADRC bit in the A/D status/control register (ADSTAT) and can be used to give a sufficiently high clock rate to the A/D converter when the bus speed is too low to provide accurate results (see Section 8.2.1.2). When the A/D converter is not being used it can be disconnected, using the ADON bit in the ADSTAT register, in order to save power (see Section 8.2.1.3).

For further information on A/D converter operation please refer to the M68HC11 Reference Manual — M68HC11RM/AD.

8.1 A/D converter operation

The A/D converter consists of an analog multiplexer, an 8-bit digital to analog capacitor array, a comparator and a successive approximation register (SAR) (see Figure 8-1).

There are four options that can be selected by the multiplexer; the ADIN input pin, VRH, (VRH+VRL)/2 or VRL. Selection is done via the CHx bits in the ADSTAT register (see Section 8.2.1.4). ADIN is the only input point for A/D conversion operations; the others are reference points which can be used for test purposes.

The A/D reference input (ADIN) is applied to a precision internal digital to analog converter. Control logic drives this D/A converter and the analog output is successively compared with the analog input (ADIN) sampled at the beginning of the conversion. The conversion is monotonic with no missing codes.

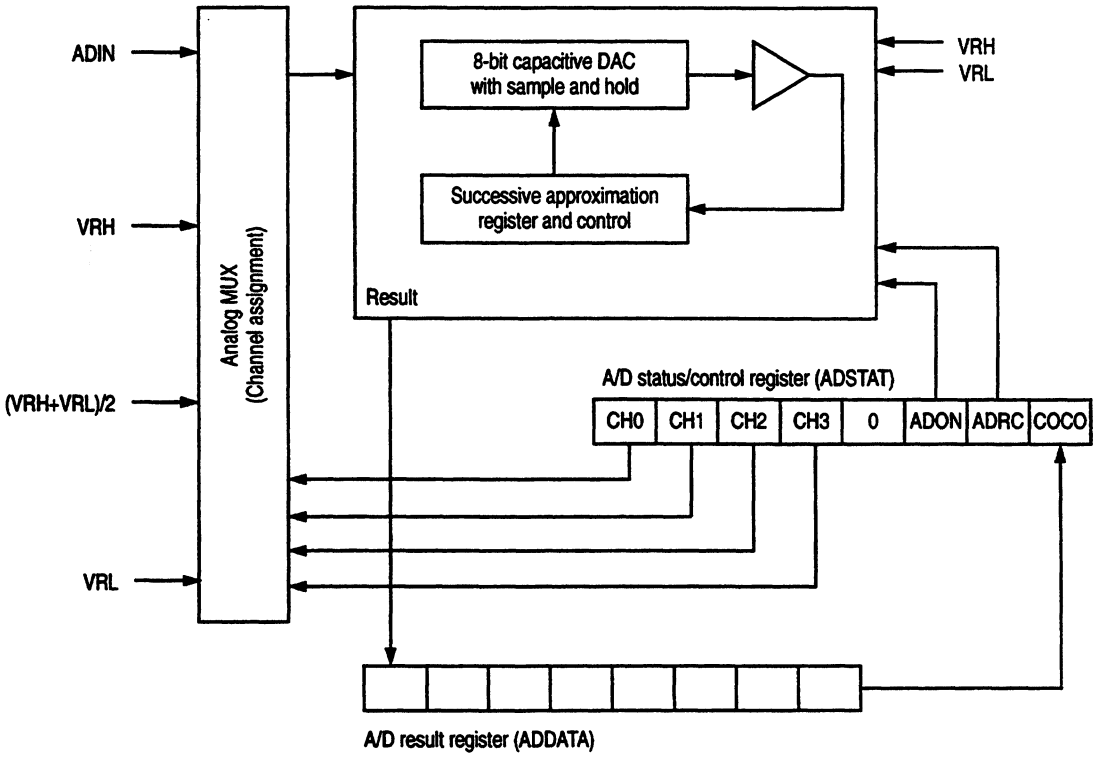


Figure 8-1 A/D converter block diagram

The result of each successive comparison is stored in the SAR and, when the conversion is complete, the contents of the SAR are transferred to the read-only result data register ($14), and the conversion complete flag, COCO, is set in the A/D status/control register ($15).

Caution: Any write to the A/D status/control register will abort the current conversion, reset the conversion complete flag and start a new conversion on the selected channel.

At power-on or external reset, both the ADRC and ADON bits are cleared, thus the A/D is disabled.

8.2 A/D registers

8.2.1 A/D status/control register (ADSTAT)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0015	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0	0000 0000

8.2.1.1 COCO — Conversion complete flag

1 (set) – COCO flag is set each time a conversion is complete, allowing the new result to be read from the A/D result data register ($14). The converter then starts a new conversion.

0 (clear) – COCO is cleared by reading the result data register or writing to the status/control register.

Reset clears the COCO flag.

8.2.1.2 ADRC — A/D RC oscillator control

The ADRC bit allows the user to control the A/D RC oscillator, which is used to provide a sufficiently high clock rate to the A/D to ensure accuracy when the chip is running at low speeds.

1 (set) – When the ADRC bit is set, the A/D RC oscillator is turned on and, if ADON is set, the A/D runs from the RC oscillator clock. See Table 8-1.

0 (clear) – When the ADRC bit is cleared, the A/D RC oscillator is turned-off and, if ADON is set, the A/D runs from the CPU clock.

When the A/D RC oscillator is turned on, it takes a time t_{ADRC} to stabilise (see Section 11.4). During this time A/D conversion results may be inaccurate.

Power-on or external reset clears the ADRC bit.

8.2.1.3 ADON — A/D converter on

The ADON bit allows the user to enable/disable the A/D converter.

1 (set) – A/D converter is switched on.

0 (clear) – A/D converter is switched off.

When the A/D converter is switched on, it takes a time t_{ADON} for the current sources to stabilise (see Section 11.4). During this time A/D conversion results may be inaccurate.

Power-on or external reset will clear the ADON bit, thus disabling the A/D converter.

Table 8-1 A/D clock selection

ADRC	ADON	RC oscillator	A/D converter	Comments
0	0	OFF	OFF	A/D switched off.
0	1	OFF	ON	A/D using CPU clock.
1	0	ON	OFF	Allows the RC oscillator to stabilise.
1	1	ON	ON	A/D using RC oscillator clock.

8.2.1.4 CH3–CH0 — A/D channels 3, 2, 1 and 0

The CH3–CH0 bits allow the user to determine which channel of the A/D converter multiplexer is selected. See Table 8-2 for channel selection.

Reset clears the CH0–CH3 bits.

Table 8-2 A/D channel assignment

CH3	CH2	CH1	CH0	Channel selected
0	0	0	0	ADIN
0	0	0	1	Unused
0	0	1	0	Unused
0	0	1	1	Unused
0	1	0	0	Unused
0	1	0	1	Unused
0	1	1	0	Unused
0	1	1	1	Unused
1	0	0	0	VRH pin (high)
1	0	0	1	(VRH + VRL) / 2
1	0	1	0	VRL pin (low)
1	0	1	1	VRL pin (low)
1	1	0	0	VRL pin (low)
1	1	0	1	VRL pin (low)
1	1	1	0	VRL pin (low)
1	1	1	1	VRL pin (low)

8.2.2 A/D input register (ADIN)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0016	0	0	0	0	0	0	0	ADIN	0000 000u

8.2.2.1 ADIN — A/D input

The ADIN bit allows the A/D input to be read as a static input. Reading this bit during an A/D conversion sequence may inject noise into the analog input and reduce the accuracy of the A/D result.

Note: Performing a digital read of the A/D input with levels other than V_{DD} or V_{SS} on the ADIN pin will result in greater power dissipation during the read cycle. This will also give unpredictable results on the ADIN input.

Reset does not affect the ADIN bit.

8

8.2.3 A/D result data register (ADDATA)

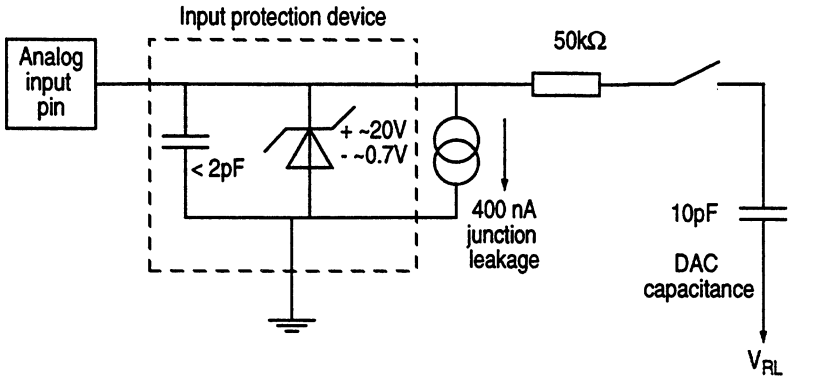
Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0014									0000 0000

ADDATA is a read-only register which is used to store the result of an A/D conversion. The result is loaded into the register from the SAR and the conversion complete flag in the ADSTAT register, COCO, is set.

8.3 ADIN analog input

The external analog voltage value to be processed by the A/D converter is sampled on an internal capacitor through a resistive path, provided by input-selection switches and a sampling aperture time switch, as shown in Figure 8-2. Sampling time is limited to 12 bus clock cycles. After sampling, the analog value is stored on the capacitor and held until the end of conversion. During this hold time, the analog input is disconnected from the internal A/D system and the external voltage source sees a high impedance input.

The equivalent analog input during sampling is an RC low-pass filter with a minimum resistance of 50 kΩ and a capacitance of at least 10pF. (It should be noted that these are typical values measured at room temperature).



Note: The analog switch is closed during the 12 cycle sample time only.

Figure 8-2 Electrical model of an A/D input pin

9
RESETS AND INTERRUPTS

9.1 Resets

The MC68HC05L2 can be reset in three ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin and by a COP watchdog timer reset, if the watchdog timer is enabled.

9.1.1 Power-on reset

A power-on reset occurs when a positive transition is detected on VDD. The power-on reset function is strictly for power turn-on conditions and should not be used to detect drops in the power supply voltage. The power-on circuitry provides a stabilisation delay (t_{PORL}) from when the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of this delay then the processor remains in the reset state until $\overline{\text{RESET}}$ goes high. The user must ensure that the voltage on VDD has risen to a point where the MCU can operate properly by the time t_{PORL} has elapsed. If there is doubt, the external $\overline{\text{RESET}}$ pin should remain low until the voltage on VDD has reached the specified minimum operating voltage. This may be accomplished by connecting an external RC-circuit to this pin to generate a power-on reset (POR). In this case, the time constant must be great enough (at least 100ms) to allow the oscillator circuit to stabilise.

At power-up, the $\overline{\text{RESET}}$ pin is pulled active low by an internal open drain N-channel device driven from the power-on reset signal. This pin can be used as a reset output.

9.1.2 $\overline{\text{RESET}}$ pin

When the oscillator is running in a stable state, the MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a minimum period of 1.5 machine cycles (t_{CYC}). This pin contains an internal Schmitt Trigger as part of its input to improve noise immunity. When a reset condition occurs internally, i.e. from the COP watchdog, the $\overline{\text{RESET}}$ pin provides an active-low open drain output signal which may be used to reset external hardware.

9.1.3 Computer Operating Properly (COP) reset

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

Note: COP timeout is prevented by periodically writing a '0' to bit 0 of address $0FF0.

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

The COP reset function is enabled or disabled using the COPE bit in the core timer control and status register (see Section 5.3.1).

9.2 Interrupts

The MCU can be interrupted by four different sources, three maskable hardware interrupts and one non-maskable software interrupt:

- External signal on the $\overline{\text{IRQ}}$ pin
- Core timer
- 16-bit programmable timer
- Software interrupt instruction (SWI)

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 (return from interrupt) 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) and the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending after an instruction execution, the external interrupt is serviced first.

Table 9-1 shows the relative priority of all the possible interrupt sources. Figure 9-1 shows the interrupt processing flow.

Table 9-1 Interrupt priorities

Source	Register	Flags	Vector Address	Priority
Reset	—	—	$0FFE, $0FFF	highest
Software Interrupt (SWI)	—	—	$0FFC, $0FFD	↑
External Interrupt ($\overline{IRQ}$)	—	—	$0FFA, $0FFB	
Core timer	CTCSR	CTOF, RTIF	$0FF8, $0FF9	
Programmable timer	TSR	ICF, OCF, TOF	$0FF6, $0FF7	lowest

9.2.1 Non-maskable software interrupt (SWI)

The software interrupt (SWI) is an executable instruction and a non-maskable interrupt; it is executed regardless of the state of the I-bit in the CCR. If the I-bit is zero (interrupts 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 $0FFC and $0FFD.

9.2.2 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.

9.2.2.1 External interrupt ($\overline{IRQ}$)

The external interrupt will vector to the interrupt service routine, whose start address is contained in memory locations $0FFA and $0FFB. $\overline{IRQ}$ can be selected in software to be either edge sensitive or edge-and-level sensitive (see Section 5.3.1).

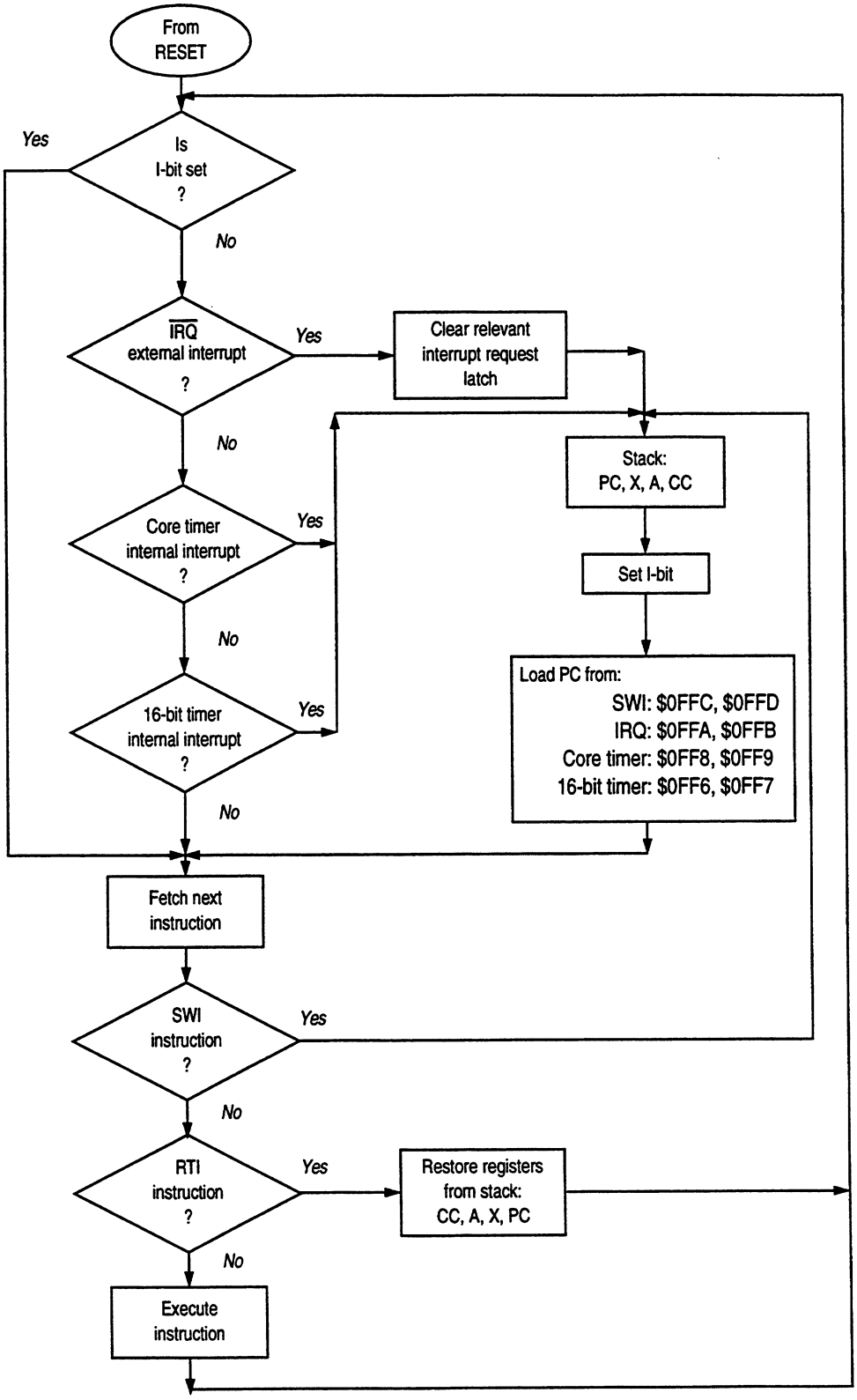


Figure 9-1 Interrupt flow chart

9.2.2.2 Real time and core timer (CTIMER) interrupts

There are two core timer interrupt flags that cause a CTIMER interrupt whenever an interrupt is enabled and its flag becomes set (RTIF and CTOF). The interrupt flags and enable bits are located in the CTIMER control and status register (CTCSR). These interrupts vector to the same interrupt service routine, whose start address is contained in memory locations $0FF8 and $0FF9 (see Section 5.3.1 and Figure 5-1).

To make use of the real time interrupt the RTIE bit must first be set. The RTIF bit will then be set after the specified number of counts.

To make use of the core timer overflow interrupt, the CTOFE bit must first be set. The CTOF bit will then be set when the core timer counter register overflows from $FF to $00.

9.2.2.3 16-bit programmable timer interrupt

There are three different timer interrupt flags (ICF, OCF, TOF) that cause a timer interrupt whenever they are set and enabled. The timer interrupt enable bits (ICIE, OCIE, TOIE) are located in the timer control register (TCR) and the timer interrupt flag is located in the timer status register (TSR). All three interrupts vector to the same service routine, whose start address is contained in memory locations $0FF6 and $0FF7 (see Section 6.2.1 and Section 6.2.2).

9.2.3 Hardware controlled interrupt sequence

The following three functions (RESET, STOP, and WAIT) are not in the strictest sense interrupts. However, they are acted upon in a similar manner. Flowcharts for STOP and WAIT are shown in Figure 2-5.

RESET: A reset condition causes the program to vector to its starting address, which is contained in memory locations $0FFE (MSB) and $0FFF (LSB). The I-bit in the condition code register is also set, to disable maskable interrupts.

STOP: The STOP instruction causes the oscillator to be turned off and the processor to 'sleep' until an external interrupt ($\overline{IRQ}$) occurs, or the device is reset.

WAIT: The WAIT instruction causes all processor clocks to stop, but leaves the core timer and the 16-bit timer clocks running. This 'rest' state of the processor can be cleared by reset, an external interrupt ($\overline{IRQ}$), a core timer interrupt or a 16-bit timer interrupt. There are no special WAIT vectors for these interrupts.

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 MC68HC05L2.

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.

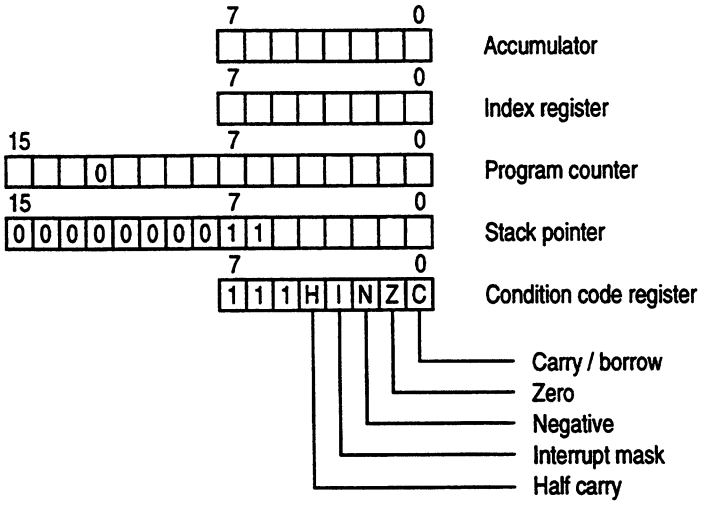


Figure 10-1 Programming model

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.

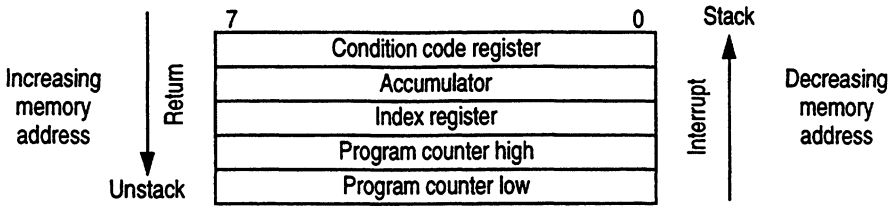


Figure 10-2 Stacking order

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. Although the M68HC05 CPU core can address 64K bytes of memory, the actual address range of the MC68HC05L2 is limited to 4K bytes. The four most significant bits of the program counter are therefore not used and are permanently set to zero.

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 Inherent	Cycles 11	Bytes 1	Opcode $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

10

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

Function	Mnemonic	Addressing modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (no offset)			Indexed (8-bit offset)		
		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

Function	Mnemonic	Inherent addressing mode		
		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

10

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

10.3 Addressing modes

Ten different addressing modes provide programmers with the flexibility to optimize their all situations. The various indexed addressing modes make it possible to locate data tabl conversion tables and scaling tables anywhere in the memory space. Short indexed acce single byte instructions; the longest instructions (three bytes) enable access to tables th memory. Short absolute (direct) and long absolute (extended) addressing are also inclu or two byte direct addressing instructions access all data bytes in most applications. E addressing permits jump instructions to reach all memory locations.

The term 'effective address' (EA) is used in describing the various addressing mo effective address is defined as the address from which the argument for an instruction i or stored. The ten addressing modes of the processor are described below. Parentheses to indicate 'contents of' the location or register referred to. For example, (PC) indi contents of the location pointed to by the PC (program counter). An arrow indicates 'is by' and a colon indicates concatenation of two bytes. For additional details and illustrations, refer to the ***M6805 HMOS/M146805 CMOS Family Microc 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 inst contained in the opcode. Operations specifying only the index register or accumulator, a the control instruction, with no other arguments are included in this mode. These instru one byte long.

10.3.2 Immediate

In the immediate addressing mode, the operand is contained in the byte immediately foll opcode. The immediate addressing mode is used to access constants that do not char 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 s following the opcode byte. Direct addressing allows the user to directly address the l bytes in memory with a single two-byte instruction.

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

Mne
C
C
D
E
I
J
J
L
L
L
L
M
N
N
C
F
R
F
F
F
S
S
S
S
S
S
S
S
T
T
T
W

BSC
BTB
DIR
EXT
INH

Table 10-8 M68HC05 opcode map

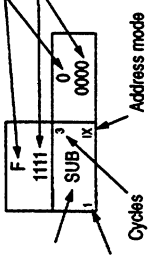
	Bit manipulation		Branch	Read/modify/write					Control			Register/memory							
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	High / Low	
	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111		
0 0000	BRSET0 5 BTB 3	BSET0 5 BSC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



10.3.8 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

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)

11
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the MC68HC05L2.

11.1 Maximum ratings

Table 11-1 Maximum ratings

Rating	Symbol	Value	Unit
Supply voltage(1)	V_{DD}	– 0.3 to +7.0	V
Input voltage	V_{IN}	V_{SS} – 0.3 to V_{DD} + 0.3	V
Input voltage: $\overline{IRQ}$/VPP – MC68HC05L2 and MC68HC705L2 bootloader modes	V_{IN}	V_{SS} – 0.3 to 1.8V_{DD} + 0.3	V
Operating temperature range	T_A	T_L to T_H 0 to +70	°C
Storage temperature range	T_{STG}	– 65 to +150	°C
Current drain per pin (2) – excluding VDD and VSS	I_D	25	mA

(1) All voltages are with respect to V_{SS}.

(2) Maximum current drain per pin is for one pin at a time, limited by an external resistor.

Note: This device contains circuitry designed to protect against damage due to high electrostatic voltages or electric fields. However, it is recommended that normal precautions be taken to avoid the application of any voltages higher than those given in the Maximum Ratings table to this high impedance circuit. For maximum reliability all unused inputs should be tied to either V_{SS} or V_{DD}.

THIS PAGE LEFT BLANK INTENTIONALLY

11.3 DC electrical characteristics

Table 11-3 DC electrical characteristics for 5V operation

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage $I_{LOAD} \leq -10\ \mu A$ $I_{LOAD} \leq +10\ \mu A$	 V_{OH} V_{OL}	 $V_{DD} - 0.1$ —	 — —	 — 0.1	 V V
Output high voltage ($I_{LOAD} = -0.8$ mA) PA0–7, PB0–4	V_{OH}	$V_{DD} - 0.8$	—	—	V
Output low voltage ($I_{LOAD} = +1.6$ mA) PA0–7, PB0–4	V_{OL}	—	—	0.4	V
Input high voltage PA0–7, PB0–4, OSC1, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	$0.7V_{DD}$	—	V_{DD}	V
Input low voltage PA0–7, PB0–4, OSC1, $\overline{IRQ}$, $\overline{RESET}$	V_{IL}	V_{SS}	—	$0.3V_{DD}$	V
Supply current[3] RUN WAIT STOP	I_{DD}	 — — —	 TBD TBD TBD	 10 4 50	 mA mA µA
High-Z leakage current PA0–7, PB0–4	I_{IL}	—	—	TBD	µA
Input current $\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP	I_{IN}	—	—	TBD	µA
Port A and B pull-up resistors PA0–7, PB0–4	—	10	—	30	kΩ
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, TCAP	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
LCD voltage input	V_{LCD}	V_{SS}	—	V_{DD}	V
LCD average DC offset voltage	V_{DC}	—	—	100	mV
MC68HC705L2 EPROM: *Programming voltage* *Programming current* *Programming time*	 V_{PP} I_{PP} t_{PROG}	 *15* — —	 *16* *TBD* *TBD*	 *17* *TBD* *TBD*	 *V* *mA* *ms*

(1) All IDD measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs (see Section 2.2.1).

(2) Typical values are at mid point of voltage range and at 25 °C only

(3) RUN and WAIT I_{DD}: measured using an external square-wave clock source (f_{OSC} = 2.0 MHz); all inputs 0.2 V from rail; no DC loads; maximum load on outputs 50pF (20pF on OSC2).
STOP and WAIT I_{DD}: all ports configured as inputs; V_{IL} = 0.2 V and $V_{IH} = V_{DD} - 0.2$ V.
STOP I_{DD}: measured with OSC1 = V_{DD}.
WAIT I_{DD} is affected linearly by the OSC2 capacitance.

11.2 Thermal characteristics and power considerations

Table 11-2 Package thermal characteristics

Characteristics	Symbol	Value	Unit
Thermal resistance	θ_{JA}		°C/W
– Plastic 42-pin shrink DIL package		40	
– Ceramic 42-pin shrink DIL package		TBD	

The average chip junction temperature, T_J, in degrees Celcius can be obtained from the following equation:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad [1]$$

where:

T_A = Ambient Temperature (°C)

q_{JA} = Package Thermal Resistance, Junction-to-ambient (°C/W)

$P_D = P_{INT} + P_{I/O}$ (W)

P_{INT} = Internal Chip Power = $I_{DD} \bullet V_{DD}$ (W)

$P_{I/O}$ = Power Dissipation on Input and Output pins (User determined)

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = \frac{K}{T_J + 273} \quad [2]$$

Solving equations [1] and [2] for K gives:

$$K = P_D \bullet (T_A + 273) + \theta_{JA} \bullet P_D^2 \quad [3]$$

where K is a constant for a particular part. K can be determined by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained for any value of T_A by solving the above equations. The package thermal characteristics are shown in Table 11-2.

Voltage	Pins	R1	R2	C
4.5V	PA0–7, PB0–4	3.26kΩ	2.38kΩ	50pF
3.0V	PA0–7, PB0–4	10.91kΩ	6.32kΩ	50pF

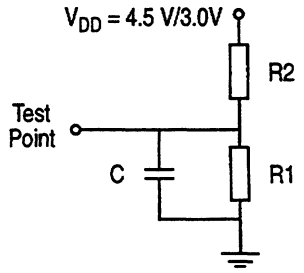


Figure 11-1 Equivalent test load (V_{OH}, V_{OL})

11.5 A/D converter electrical characteristics

Table 11-5 A/D converter electrical characteristics for 5V operation

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)				
Characteristic	**Parameter**	**Min**	**Max**	**Unit**
Resolution	Number of bits resolved by A/D converter	8	—	Bits
Non-linearity	Maximum deviation from best straight line through the A/D transfer characteristics ($V_{RH} = V_{DD}$ and V_{RL} = 0V)	—	±0.5	LSB
Quantization error	Uncertainty due to converter resolution	—	±0.5	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale equivelent of the binary output code for all errors	—	±1	LSB
Conversion range	Analog input voltage range	V_{RL}	V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	V_{DD} + 0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS} - 0.1	V_{RH}	V
Conversion time	Total time to perform a single analog to digital conversion E clock Internal RC oscillator	 — —	 32 32	 t_{CYC} µs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	Guaranteed		
Zero input reading	Conversion result when $V_{IN} = V_{RL}$	$00	—	Hex
Full scale reading	Conversion result when $V_{IN} = V_{RH}$	—	$FF	Hex
Sample acquisition time(1)	Analog input acquisition sample time E clock Internal RC oscillator	 — —	 12 12	 t_{CYC} µs
Sample/hold capacitance	Input capacitance during sample ADIN	—	12	pF
Input leakage(2)	Input leakage on A/D pins ADIN, V_{RL} and V_{RH}	—	1	µA

(1) Source impedences greater than 10kΩ will adversely affect internal RC charging time during input sampling.

(2) The external system error caused by input leakage current is approximately equal to the product of R source and input current. Input current to A/D channel will be dependent on external source impedance (see Figure 8-2).

11.4 AC electrical characteristics

Table 11-4 AC electrical characteristics for 5V operation

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of operation Oscillator frequency	f_{OSC}	0	4.2	MHz
Internal operating frequency Crystal External clock	 f_{OP} f_{OP}	 dc dc	 2.1 2.1	 MHz MHz
Processor cycle time	t_{CYC}	250	—	ns
Oscillator clock pulse width	t_{OH}, t_{OL}	90	—	ns
Interrupt pulse width low (edge-triggered)	t_{ILIH}	125	—	ns
Interrupt pulse period(1)	t_{ILIL}	see note	—	t_{CYC}
STOP recovery start-up time(crystal oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{cyc}
Power-on reset delay	t_{PORL}	4064	4064	t_{cyc}
Crystal oscillator start-up time	t_{OXOV}	—	100	ms
RC oscillator stabilisation time	t_{ADRC}	—	100	ms
A/D converter stabilisation time	t_{ADON}		500	µs

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

12
MECHANICAL DATA

12.1 MC68HC05L2 pin configurations

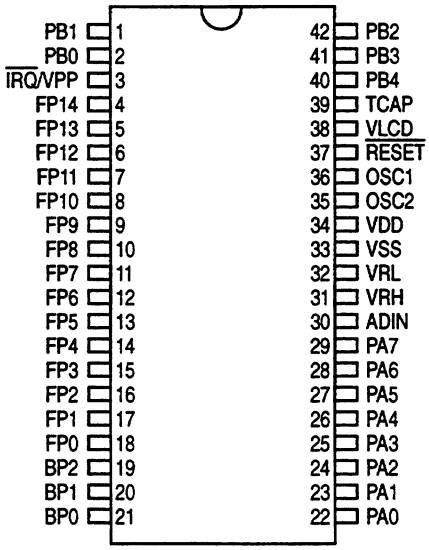


Figure 12-1 42-pin plastic/ceramic shrink dual-in-line (SDIP) pin assignments

THIS PAGE LEFT BLANK INTENTIONALLY

13
ORDERING INFORMATION

This section describes the information needed to order the MC68HC05L2 or *MC68HC705L2.*

To initiate a ROM pattern for the MCU, it is necessary to contact your local field service office, local sales person or Motorola representative. Please note that you will need to supply details such as: mask option selections; temperature range; oscillator frequency; package type; electrical test requirements; and device marking details so that an order can be processed, and a customer specific part number allocated. Refer to Table 13-1 for appropriate part numbers.

Table 13-1 MC order numbers

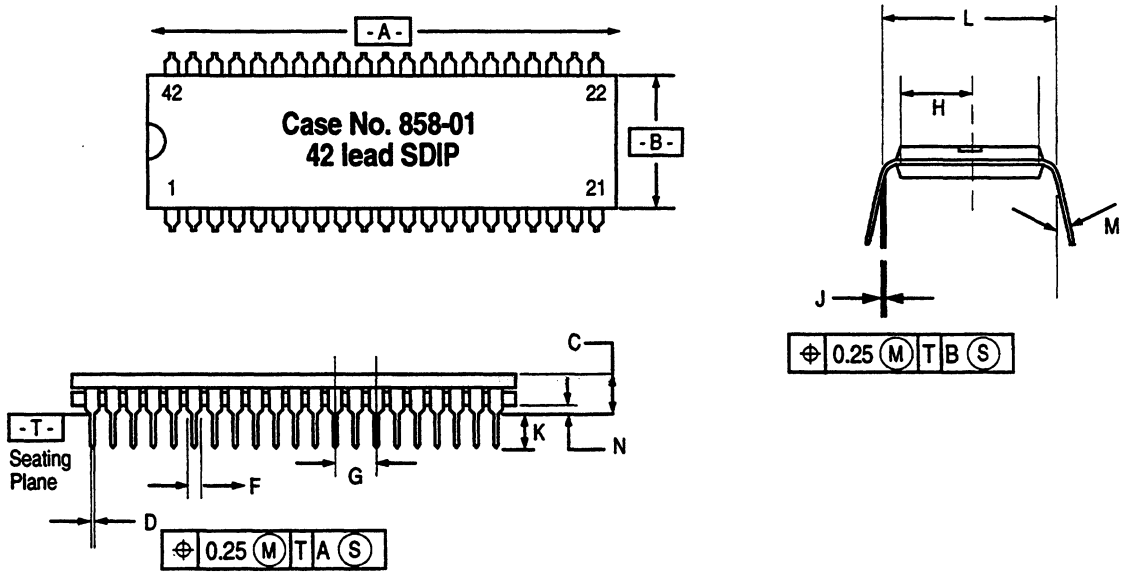
Device Title	Package Type	Temperature	Part Number
MC68HC05L2	42-pin plastic SDIP	0 to +70°C	MC68HC05L2B
		-40 to + 85°C	MC68HC05L2CB
MC68HC705L2	*42-pin windowed ceramic SDIP*	*0 to +70°C*	*MC68HC705L2K*
	42-pin plastic SDIP (OTP)	*0 to +70°C*	*MC68HC705L2B*
		-40 to + 85°C	*MC68HC705L2CB*

13.1 EPROMS

A 4 kbyte EPROM programmed with the customer's software (positive logic for address and data) should be submitted for pattern generation. All unused bytes should be programmed to $00.

The EPROM should be clearly labelled, placed in a conductive IC carrier and securely packed.

12.2 MC68HC05L2 mechanical dimensions



Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	36.45	37.21		H	7.62 BSC	
B	13.72	14.22	1. Dimensions and tolerancing per ANSI Y 14.5 1982.	J	0.20	0.38
C	3.94	5.08	2. Controlling dimension is mm.	K	2.92	3.43
D	0.36	0.56	3. Dimension L to centre of lead when formed parallel.	L	15.24 BSC	
F	0.81	1.17	4. Dimensions A and B do not include mould flash. Allowable mould flash is 0.25 mm.	M	0°	15°
G	1.77 BSC			N	0.51	1.02

Figure 12-2 Mechanical dimensions for 42-pin plastic shrink dual-in-line (PSDIP)

GLOSSARY

This section contains abbreviations and specialist words used in this data sheet and throughout the industry. Further information on many of the terms may be gleaned from Motorola's ***M68HC11 Reference Manual, M68HC11RM/AD***, or from a variety of standard electronics text books.

$xxxx	The digits following the '$' are in hexadecimal format.
%xxxx	The digits following the '%' are in binary format.
A/D, ADC	Analog-to-digital (converter).
Bootstrap mode	In this mode the device automatically loads its internal memory from an external source on reset and then allows this program to be executed.
Byte	Eight bits.
CAN	Controller area network.
CCR	Condition codes register; an integral part of the CPU.
CERQUAD	A ceramic package type, principally used for EPROM and high temperature devices.
Clear	'0' — the logic zero state; the opposite of 'set'.
CMOS	Complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.
COP	Computer operating properly. *aka* 'watchdog'. This circuit is used to detect device runaway and provide a means for restoring correct operation.
CPU	Central processing unit.
D/A, DAC	Digital-to-analog (converter).
EEPROM	Electrically erasable programmable read only memory. *aka* 'EEROM'.
EPROM	Erasable programmable read only memory. This type of memory requires exposure to ultra-violet wavelengths in order to erase previous data. *aka* 'PROM'.
ESD	Electrostatic discharge.

13.2 Verification media

All original pattern media (EPROMs) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned with a listing verification form. The listing should be thoroughly checked and the verification form completed, signed and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the custom mask. If desired, Motorola will program blank EPROMs (supplied by the customer) from the data file used to create the custom mask, to aid in the verification process.

13.3 ROM verification units (RVU)

Ten MCUs containing the customer's ROM pattern will be provided for program verification. These units will have been made using the custom mask but are for ROM verification only. For expediency, they are usually unmarked and are tested only at room temperature (25°C) and at 5 Volts. These RVUs are included in the mask charge and are not production parts. They are neither backed nor guaranteed by Motorola Quality Assurance.

GLOSSARY

This section contains abbreviations and specialist words used in this data sheet and throughout the industry. Further information on many of the terms may be gleaned from Motorola's ***M68HC11 Reference Manual, M68HC11RM/AD***, or from a variety of standard electronics text books.

$xxxx	The digits following the '$' are in hexadecimal format.
%xxxx	The digits following the '%' are in binary format.
A/D, ADC	Analog-to-digital (converter).
Bootstrap mode	In this mode the device automatically loads its internal memory from an external source on reset and then allows this program to be executed.
Byte	Eight bits.
CAN	Controller area network.
CCR	Condition codes register; an integral part of the CPU.
CERQUAD	A ceramic package type, principally used for EPROM and high temperature devices.
Clear	'0' — the logic zero state; the opposite of 'set'.
CMOS	Complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.
COP	Computer operating properly. *aka* 'watchdog'. This circuit is used to detect device runaway and provide a means for restoring correct operation.
CPU	Central processing unit.
D/A, DAC	Digital-to-analog (converter).
EEPROM	Electrically erasable programmable read only memory. *aka* 'EEROM'.
EPROM	Erasable programmable read only memory. This type of memory requires exposure to ultra-violet wavelengths in order to erase previous data. *aka* 'PROM'.
ESD	Electrostatic discharge.

Expanded mode	In this mode the internal address and data bus lines are connected to external pins. This enables the device to be used in much more complex systems, where there is a need for external memory for example.
EVS	Evaluation system. One of the range of platforms provided by Motorola for evaluation and emulation of their devices.
HCMOS	High-density complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.
I/O	Input/output; used to describe a bidirectional pin or function.
Input capture	(IC) This is a function provided by the timing system, whereby an external event is 'captured' by storing the value of a counter at the instant the event is detected.
Interrupt	This refers to an asynchronous external event and the handling of it by the MCU. The external event is detected by the MCU and causes a predetermined action to occur.
$\overline{\text{IRQ}}$	Interrupt request. The overline indicates that this is an active-low signal format.
K byte	A kilo-byte (of memory); 1024 bytes.
LCD	Liquid crystal display.
LSB	Least significant byte.
M68HC05	Motorola's family of 8-bit MCUs.
MCU	Microcontroller unit.
MI BUS	Motorola interconnect bus. A single wire, medium speed serial communications protocol.
MSB	Most significant byte.
Nibble	Half a byte; four bits.
NRZ	Non-return to zero.
Opcode	The opcode is a byte which identifies the particular instruction and operating mode to the CPU.
Operand	The operand is a byte containing information the CPU needs to execute a particular instruction.
Output compare	(OC) This is a function provided by the timing system, whereby an external event is generated when an internal counter value matches a predefined value.
PLCC	Plastic leaded chip carrier package.
PLL	Phase-locked loop circuit. This provides a method of frequency multiplication, to enable the use of a low frequency crystal in a high frequency circuit.

Pull-down, pull-up	These terms refer to resistors, sometimes internal to the device, which are permanently connected to either ground or V_{DD}.
PWM	Pulse width modulation. This term is used to describe a technique where the width of the high and low periods of a waveform is varied, usually to enable a representation of an analog value.
QFP	Quad flat pack package.
RAM	Random access memory. Fast read and write, but contents are lost when the power is removed.
RFI	Radio frequency interference.
RTI	Real-time interrupt.
ROM	Read-only memory. This type of memory is programmed during device manufacture and cannot subsequently be altered.
RS-232C	A standard serial communications protocol.
SAR	Successive approximation register.
SCI	Serial communications interface.
Set	'1' — the logic one state; the opposite of 'clear'.
Silicon glen	An area in the central belt of Scotland, so called because of the concentration of semiconductor manufacturers and users found there.
Single chip mode	In this mode the device functions as a self contained unit, requiring only I/O devices to complete a system.
SPI	Serial peripheral interface.
Test mode	This mode is intended for factory testing.
TTL	Transistor-transistor logic.
UART	Universal asynchronous receiver transmitter.
VCO	Voltage controlled oscillator.
Watchdog	*see* 'COP'.
Wired-OR	A means of connecting outputs together such that the resulting composite output state is the logical OR of the state of the individual outputs.
Word	Two bytes; 16 bits.
XIRQ	Non-maskable interrupt request. The overline indicates that this has an active-low signal format.

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.