MC68HC05J3/D

MC68HC05J3

TECHNICAL DATA

HC05

MC68HC05J3

TECHNICAL
DATA

freescale™
semiconductor

MC68HC05J3

High-density Complementary Metal Oxide Semiconductor (HCMOS) Microcomputer Unit

TABLE OF CONTENTS

Paragraph Number | **TITLE** | Page Number

1
INTRODUCTION

2
MODES OF OPERATION AND PIN DESCRIPTIONS

3
MEMORY AND REGISTERS

4
INPUT/OUTPUT PORTS

5
CORE TIMER

6
16-BIT PROGRAMMABLE TIMER

Paragraph Number	TITLE	Page Number

7
RESETS AND INTERRUPTS

8
CPU CORE AND INSTRUCTION SET

9
ELECTRICAL SPECIFICATIONS

10
MECHANICAL DATA

11
ORDERING INFORMATION

LIST OF FIGURES

THIS PAGE LEFT BLANK INTENTIONALLY

LIST OF TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1 INTRODUCTION

The MC68HC05J3, with 2kbytes of ROM is a member of the Motorola M68HC05 family of HCMOS 8-bit single chip microcomputers. Based on the industry-standard M68HC05 CPU core and its familiar, efficient instruction set, this device provides a cost effective, low pin-count microcomputer solution suitable for use in a wide variety of application areas, including car body electronics. The keyboard interrupt function, which shares 4 I/O lines of port B, provides a simple interface to keypads, switches and other similar input media. In addition, the port pins are capable of sinking a current of 8mA at 0.8V and can therefore be used to drive certain LED's.

1.1 Features

- Fully static design featuring the industry standard M68HC05 core
- On-chip oscillator with crystal, resistor, external clock or ceramic resonator connection
- 2048 bytes of User ROM
- 128 bytes of RAM
- Power saving STOP and WAIT modes
- 16-bit programmable timer with input capture and output compare functions
- 8-bit multi-purpose timer
- Real time interrupt circuit
- Computer operating properly watchdog timer (mask option)
- Interrupt request input ($\overline{\text{IRQ}}$), plus four on-chip hardware interrupt sources
- Keyboard interrupt feature on four port B input/output lines
- One 8-bit and one 6-bit parallel I/O port (two lines on port B are shared with the 16-bit timer)
- All port pins are capable of sinking a current of 8mA at 0.8V
- Available in 20-pin plastic SOIC and 20-pin plastic DIL packages

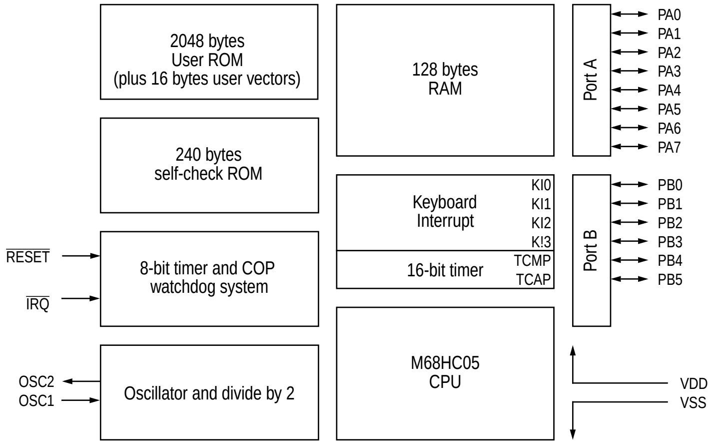


Figure 1-1 MC68HC05J3 block diagram

1.2 MC68HC05J3 mask options

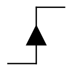
There are four mask options on the MC68HC05J3 which are programmed during manufacture and therefore must be specified on the order form: $\overline{\text{IRQ}}$ sensitivity (edge sensitive or edge-and-level sensitive), COP watchdog enable/disable, STOP instruction enable/disable and RC or crystal clock selection.

2
MODES OF OPERATION AND PIN DESCRIPTIONS

2.1 Modes of operation

The MC68HC05J3 has two modes of operation, single chip and RAM bootloader mode. Table 2-1 shows the conditions required to enter each mode on the rising edge of $\overline{\text{RESET}}$.

Table 2-1 MC68HC05J3 operating mode entry conditions

$\overline{\text{RESET}}$	$\overline{\text{IRQ}}$	TCAP	PA3	Mode	
	V_{SS} to V_{DD}	x	x	Single chip	
	$1.8V_{DD}$	1	0	RAM bootloader	Jump to RAM ($0081)
		1	1		Load RAM & execute ($0081)

x = don’t care

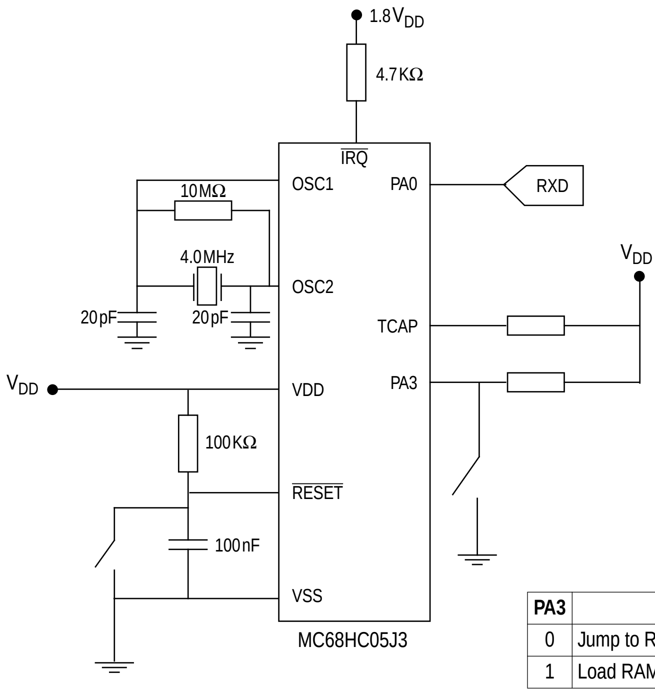
2.1.1 Single chip mode

This is the normal operating mode of the MC68HC05J3. In this mode the device functions as a self-contained microcomputer (MCU) with all on-board peripherals, including the 8-bit I/O port (A) and the 6-bit I/O port (B), 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.

2.1.2 RAM bootloader mode

The RAM bootloader mode for the MC68HC05J3 allows the user to run a program 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 TCAP and PA3, to load a user program into RAM and then to execute it.



PA3	Function
0	Jump to RAM ($0081)
1	Load RAM & execute ($0081)

All resistors are 10 kΩ unless otherwise specified

Figure 2-1 RAM bootloader circuit

The RAM bootloader is selected when the device is put into bootloader mode with TCAP held high. If PA3 is low, the program counter is set to $0081 and a previously loaded RAM program can be executed. If PA3 is high at reset a program is serially loaded from PA0 into the RAM and executed from $0081, once the last byte has been received.

The first byte to be loaded is the count byte which must contain the total number of bytes to be transferred, including the count byte itself. Therefore, for a program length of $30, the count should equal $31. The maximum program size including the count byte is 124 bytes ($7C), since four bytes must be left for the stack during download.

Providing the oscillator is running at 4MHz, 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 2-2). 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.

Table 2-2 RAM bootloader mode jump vectors

Address	Pseudo-vector
0084	Software interrupt
0087	$\overline{\text{IRQ}}$ interrupt
008A	Core timer interrupt
008D	Input capture
0090	Output compare interrupt
0093	Timer overflow interrupt

2.2 Pin descriptions

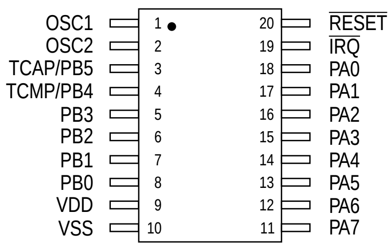


Figure 2-2 20-pin SOIC/DIP single chip and bootloader mode pin assignments

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 bypassing at the MCU. Bypass capacitors should have good high-frequency characteristics and be as close to

2

the MCU as possible. Bypassing 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 selected to be edge sensitive or edge-and-level sensitive (see Section 1.2). The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity.

2.2.3 OSC1, OSC2

These pins provide control input for an on-chip oscillator circuit. A crystal, ceramic resonator, resistor 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}).

2.2.3.1 Crystal

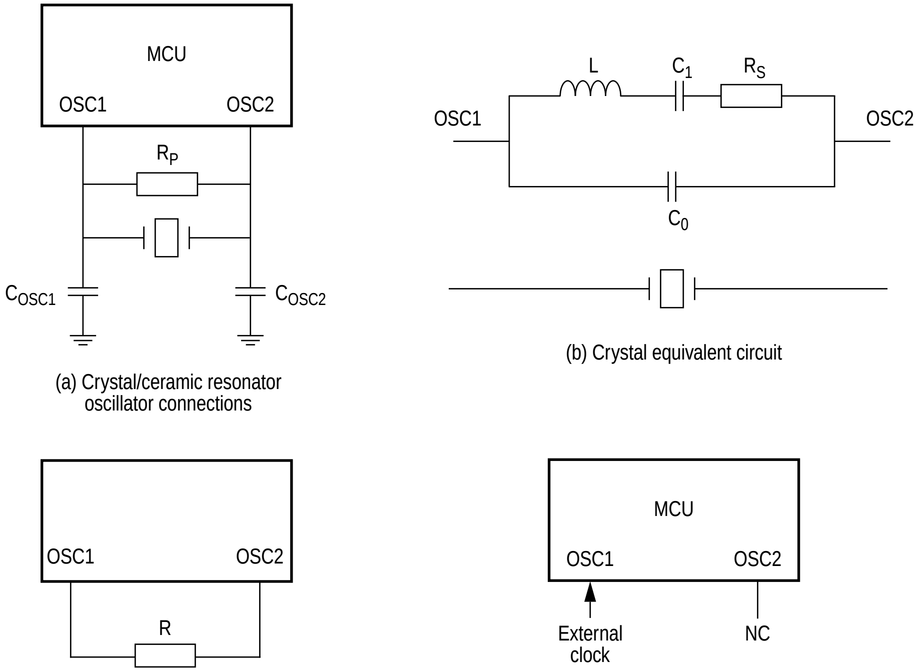
The circuit shown in Figure 2-3(a) is recommended when using either a crystal or a ceramic resonator. Figure 2-3(e) provides the recommended capacitance and feedback resistance values. The internal oscillator is designed to interface with an AT-cut parallel-resonant quartz crystal resonator in the frequency range specified for f_{osc} (see Table 9-5 and Table 9-6). 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 stabilization 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 2-3(a) is recommended when using either a crystal or a ceramic resonator. Figure 2-3(e) lists the recommended capacitance and feedback resistance values. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

2.2.3.3 RC network

With this option, a resistor is connected to the oscillator pins as shown in Figure 2-3(c). The overall accuracy of the RC oscillator is approximately ±25% (contact factory for more accurate details).



(a) Crystal/ceramic resonator oscillator connections

(b) Crystal equivalent circuit

(c) RC oscillator connections

(d) External clock source connections

Crystal

	2MHz	**4MHz**	**Unit**
R_S(max)	400	75	Ω
C_0	5	7	pF
C_1	8	12	ƒF
C_{OSC1}	15 – 40	15 – 30	pF
C_{OSC2}	15 – 30	15 – 25	pF
R_P	10	10	MΩ
Q	30 000	40 000	—

Ceramic resonator

	2 – 4MHz	**Unit**
R_S(typ)	10	Ω
C_0	40	pF
C_1	4.3	pF
C_{OSC1}	30	pF
C_{OSC2}	30	pF
R_P	1 – 10	MΩ
Q	1250	—

(e) Crystal and ceramic resonator parameters

Figure 2-3 Oscillator connections

2

2.2.3.4 External clock

An external clock should be applied to the OSC1 input, with the OSC2 pin left unconnected, as shown in Figure 2-3(d). The t_{OXOV} specification (see Table 9-5 and Table 9-6) 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 input-only 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.

2.2.5 PA0–PA7

These 8 I/O lines comprise ports A. The state of any pin is software programmable, and all the pins are configured as inputs during power-on or reset.

2.2.6 PB0–PB5

These six pins comprise port B. The state of any pin is software programmable, and all the pins are configured as inputs during power-on or reset. In addition to their normal I/O functions, pins PB0–PB3 can be used to generate a keyboard interrupt when configured as an input while PB4 is shared with the output compare function (TCMP) of the programmable timer and PB5 with the input capture function (TCAP) (see Section 4.3). PB0–PB5 contain an internal Schmitt trigger as part of their input to improve noise immunity.

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 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 STOP mode only by an external interrupt, a keyboard interrupt, if enabled, or a reset (see Figure 2-4).

2.3.2 WAIT

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

During 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 interrupts may be enabled to allow a periodic exit from WAIT mode. See Figure 2-4.

2

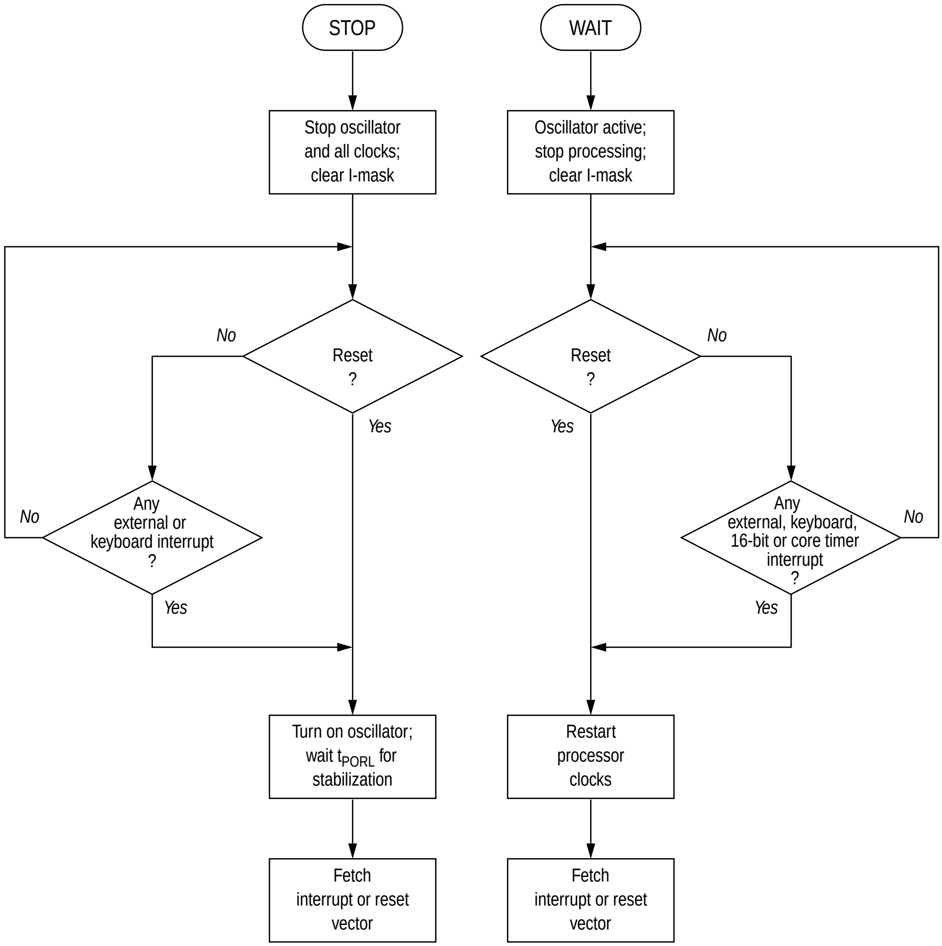


Figure 2-4 STOP and WAIT flowcharts

3

MEMORY AND REGISTERS

The MC68HC05J3 has a 4 kbyte memory map consisting of registers (for I/O, control and status), User RAM, User ROM, 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 MC68HC05J3 are contained within the first 32-byte block of the memory map, as detailed in Table 3-1.

3.2 RAM

The User RAM consists of 128 bytes of memory, from $0080 to $00FF. 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.

3.3 Non-volatile memory (NVM)

The NVM consists of 2048 bytes of ROM (MC68HC05J3) from $0700 to $0EFF, 240 bytes of bootloader ROM and 16 bytes of user vectors ($0FF0 to $0FFF).

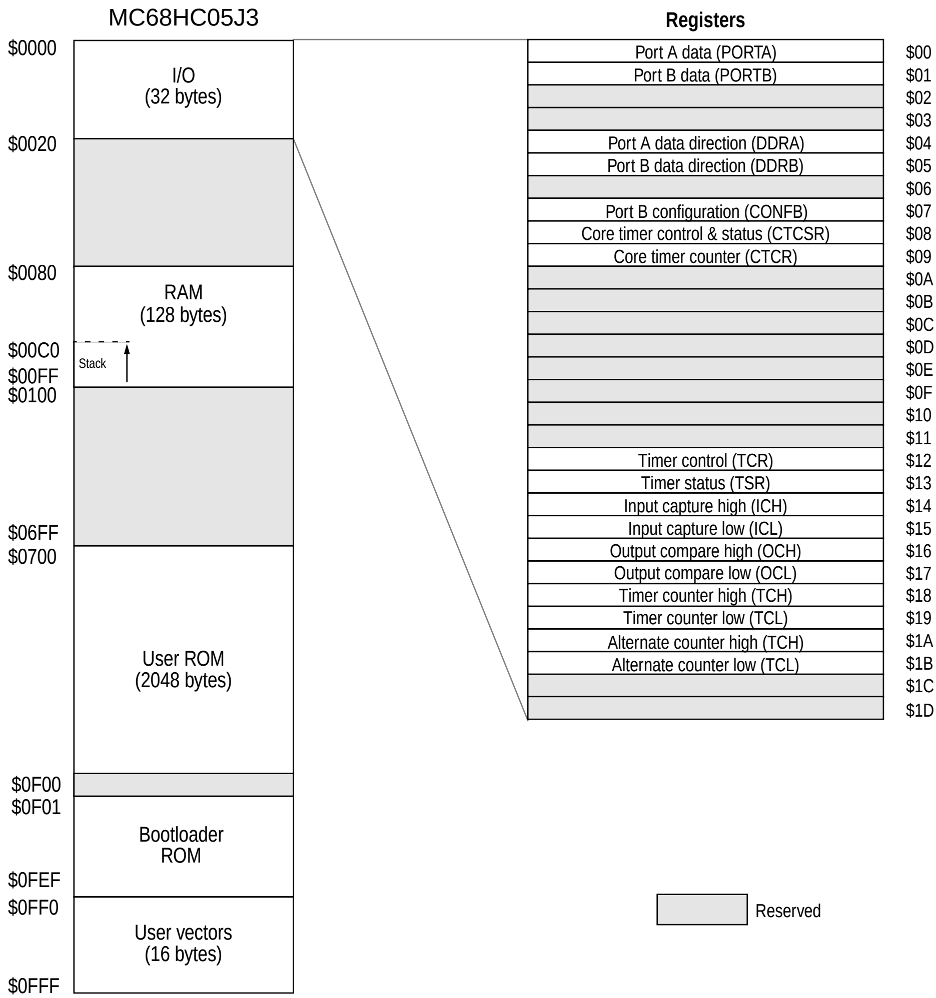


Figure 3-1 Memory map of the MC68HC05J3

3

Table 3-1 Register outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
Port A data (PORTA)	$0000									Undefined
Port B data (PORTB)	$0001	0	0							Undefined
Reserved	$0002									
	$0003									
Port A data direction (DDRA)	$0004									0000 0000
Port B data direction (DDRB)	$0005	0	0							0000 0000
Reserved	$0006									
Port B configuration (CONFB)	$0007	KSF	KIE	TCAP	TCMP	0	0	0	0	0000 0000
Core timer control/status (CTCSR)	$0008	CTOF	RTIF	CTOFE	RTIE	0	0	RT1	RT0	uu00 0011
Core timer counter (CTCR)	$0009									Undefined
Reserved	$000A									
	$000B									
	$000C									
	$000D									
	$000E									
	$000F									
	$0010									
	$0011									
Timer control (TCR)	$0012	ICIE	OCIE	TOIE	0	0	0	IEDG	OLV	0000 00u0
Timer status (TSR)	$0013	ICF	OCF	TOF	0	0	0	0	0	uuu0 0000
Input capture high (ICH)	$0014	(bit 15)							(bit 8)	Undefined
Input capture low (ICL)	$0015									Undefined
Output compare high (OCH)	$0016	(bit 15)							(bit 8)	Undefined
Output compare low (OCL)	$0017									Undefined
Timer counter high (TCH)	$0018	(bit 15)							(bit 8)	1111 1111
Timer counter low (TCL)	$0019									1111 1100
Alternate counter high (ACH)	$001A	(bit 15)							(bit 8)	1111 1111
Alternate counter low (ACL)	$001B									1111 1100
Reserved	$001C to $0EFF									
Reserved	$0F00									

3

THIS PAGE LEFT BLANK INTENTIONALLY

4

INPUT/OUTPUT PORTS

4

In single chip mode, the MC68HC05J3 has a total of 14 I/O lines, arranged as one 8-bit port (A) and one 6-bit port (B). Each I/O line is individually programmable as either input or output, under the software control of the data direction registers. Four of the port B pins can be configured to respond to keyboard interrupts, while the other two are shared with the timer subsystem. The port B configuration register provides the control for all pins of port B.

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. A pin is configured as an input if its corresponding DDR bit is cleared.

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 buffer and not the I/O pin. The operation of the standard port hardware is shown schematically in Figure 4-1.

This is further summarized in Table 4-1, which 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 not available to the user.

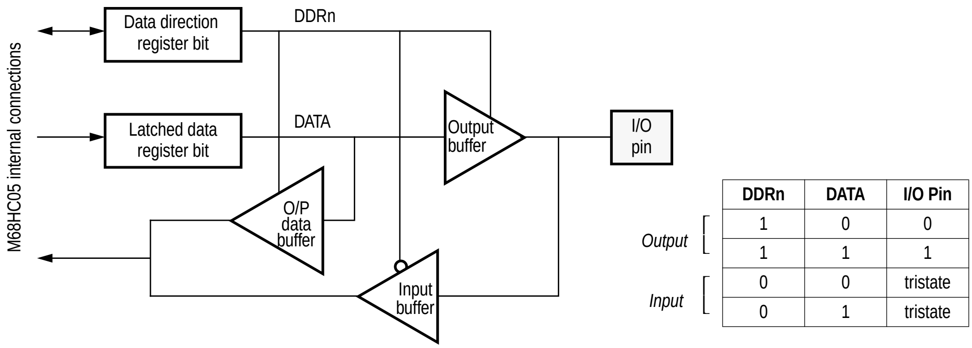


Figure 4-1 Standard I/O port structure

Table 4-1 I/O pin states

R/$\overline{W}$	DDRn	Action of MCU write to/read of data bit
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch, and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in output mode. The output data latch is read.

4.2 Port A

This port is a standard M68HC05 bidirectional I/O port, comprising a data register and a data direction register.

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

4.3 Port B

In addition to the standard port functions, this 6-bit port has a programmable keyboard interrupt feature on pins PB0–PB3 and shares two pins (PB4 and PB5) with the timer subsystem. On reset, this port is configured as a standard I/O port, comprising a data register and a data direction register.

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

Provided that the interrupt mask bit of the condition code register is cleared, the keyboard interrupt facility is enabled by setting the keyboard interrupt enable bit (KIE) in the port B configuration register at location $07. The configuration register is described in Section 4.4.3.

Pins configured as output do not contribute to the wired-or interrupt. The structure of the port pins is shown diagrammatically in Figure 4-2. When a low-to-high transition is detected on any of these port pins, a keyboard interrupt request is generated and the port B interrupt status flag (KSF) is set. The address of the interrupt service routine is specified by the contents of memory locations $0FFA and $0FFB. Since this interrupt vector is shared with the $\overline{\text{IRQ}}$ external interrupt function, the interrupt service routine should check KSF to determine the interrupt source. KSF can be cleared by accessing the port B data register. The keyboard interrupt is edge sensitive. Care must be taken to allow adequate time for switch debounce before clearing the flag.

A keyboard interrupt will force the MCU out of STOP or WAIT mode.

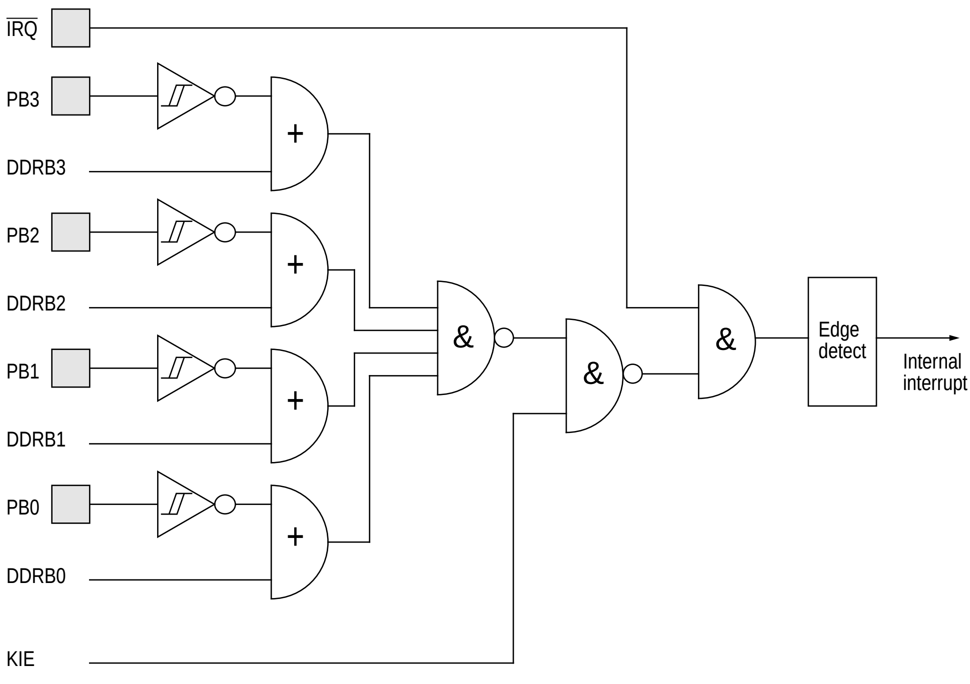


Figure 4-2 Port B keyboard interrupt function

4.4 Port registers

The following sections explain in detail the individual bits in the data and control registers associated with the ports.

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
Port A data (PORTA)	$0000									Undefined

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

The state of the bits in the port data register following reset is undefined.

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
Port B data (PORTB)	$0001	0	0							Undefined

In addition to the normal port functions, port B is equipped with a keyboard interrupt capability as described in Section 4.3. Two of the port pins (PB4 and PB5) are shared with the TCMP and TCAP pins respectively. These functions are controlled using the port B configuration register.

4.4.3 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
Port B configuration (CONFB)	$0007	KSF	KIE	TCAP	TCMP	0	0	0	0	0000 0000

Reset clears this register, thus returning all port B pins to normal I/O lines.

KSF — Keyboard interrupt status flag

1 (set) – Keyboard interrupt has occurred.

0 (clear) – No keyboard interrupt has occurred.

KIE — Keyboard interrupt enable

1 (set) – Keyboard interrupt enabled on port B.

0 (clear) – Keyboard interrupt disabled.

TCAP — Timer input capture function

1 (set) – PB5 acts as the TCAP input to the 16-bit timer. The pin is forced to an input state regardless of the data direction bit.

0 (clear) – PB5 functions as a normal I/O pin.

TCMP — Timer output compare function

1 (set) – PB4 acts as the TCMP output for the 16-bit timer. The pin is forced to an output state regardless of the data direction bit.

0 (clear) – PB4 functions as a normal I/O pin.

Bits 3–0 are not implemented on the configuration register and always read zero. Writing to these bits has no meaning or effect.

4

4.4.4 Data direction register A (DDRA)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data direction (DDRA)	$0004									0000 0000

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

Reset clears these registers, thus configuring all pins as inputs.

4.4.5 Data direction register B (DDRB)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port B data direction (DDRB)	$0005	0	0							0000 0000

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

Reset clears these registers, thus configuring all pins as inputs.

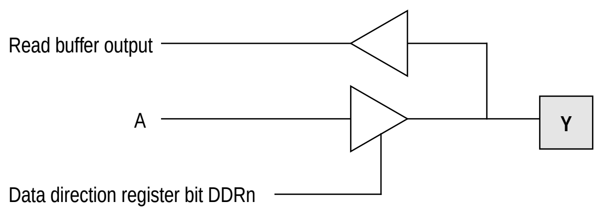
4.5 Other port considerations

All output ports can emulate ‘open-drain’ outputs. 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 4-3.

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.

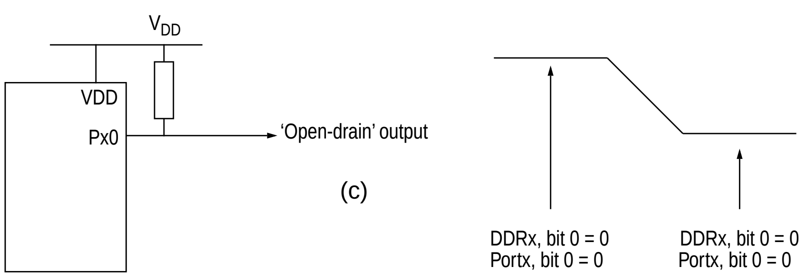
Note: ‘Open-drain’ outputs should not be pulled above V_{DD}.

(a)



(b)

DDRn	A	Y	
1	0	0	Normal operation – tri state
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 4-3 Port logic levels

5
CORE TIMER

5

The MC68HC05J3 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.

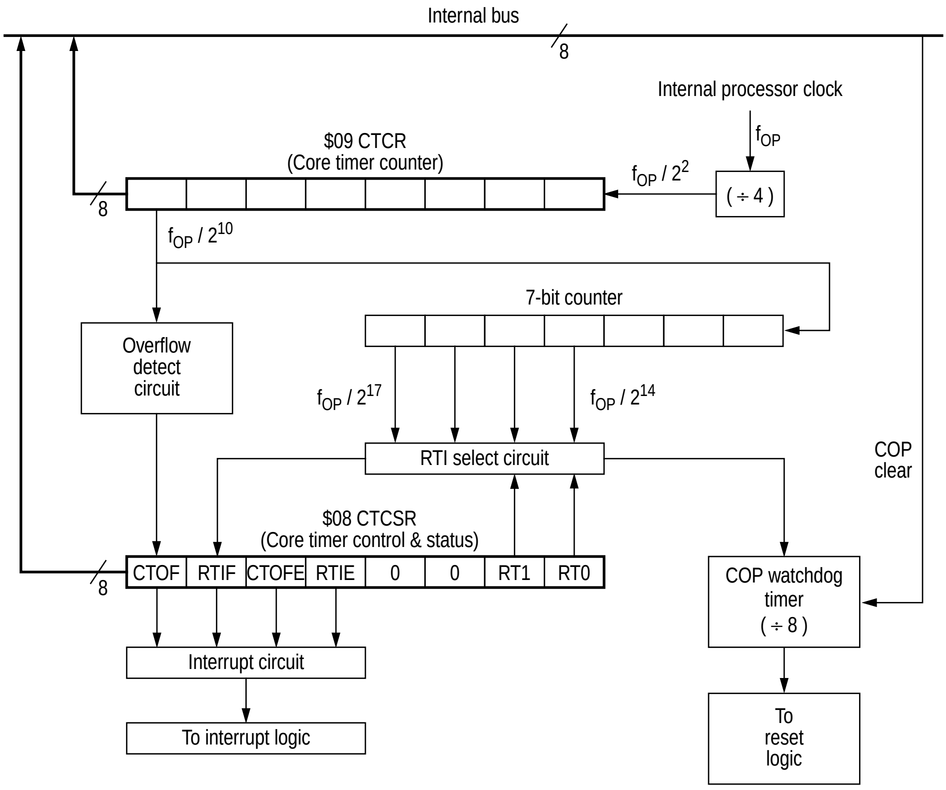


Figure 5-1 Core timer block diagram

5

As shown 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 timer 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 a mask option, enabled or disabled during device manufacture.

If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. COP timeout is prevented 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

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
Core timer control/status (CTCSR)	$0008	CTOF	RTIF	CTOFE	RTIE	0	0	RT1	RT0	uu00 0011

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.

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.

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.

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.

RT1:RT0 — Real time interrupt rate select

5

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
Core timer counter (CTCR)	$0009								

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 stabilization delay (t_{PORL}). The timer is then cleared and operation resumes.

5

THIS PAGE LEFT BLANK INTENTIONALLY

6
16-BIT PROGRAMMABLE TIMER

The programmable timer on the MC68HC05J3 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, the programmed pin action takes place. Refer to Figure 6-1 for a block diagram of the timer. The input capture and output compare functions can only be enabled by setting bit 4 and bit 5 of the port B configuration register as described in Section 4.4.3.

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.

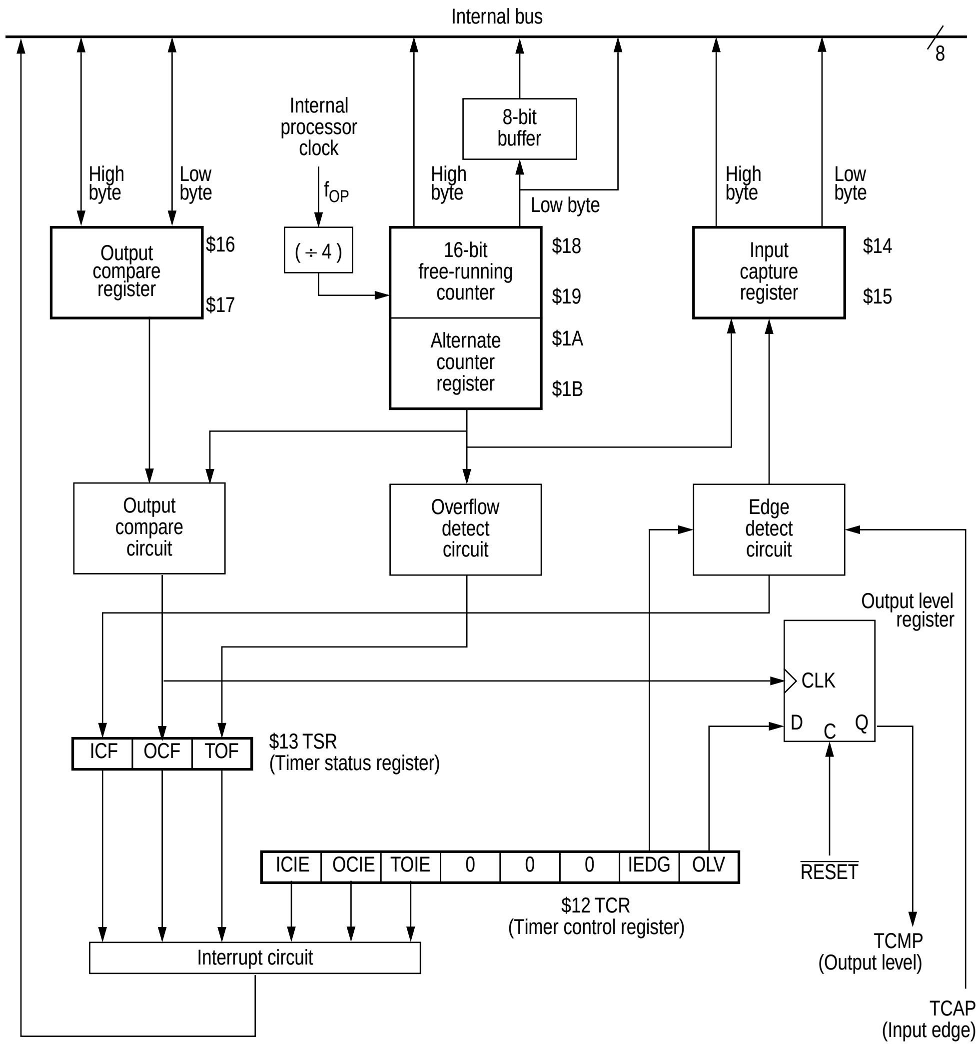


Figure 6-1 16-bit programmable timer block diagram

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 2µs if the internal bus clock is 2MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

6.1.1 Counter high register
Counter low register
Alternate counter high register
Alternate counter low register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer counter high (TCH)	$0018	(bit 15)							(bit 8)	1111 1111
Timer counter low (TCL)	$0019									1111 1100
Alternate counter high (ACH)	$001A	(bit 15)							(bit 8)	1111 1111
Alternate counter low (ACL)	$001B									1111 1100

The double-byte, free-running counter can be read from either of two locations, the counter register at $18 – $19 or the alternate counter register at $1A – $1B. 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), $18 or $1A, 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.

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.

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

6.2.1 Timer control register – TCR

The timer control register at location $12 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) and output level polarity (OLV).

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control (TCR)	$0012	ICIE	OCIE	TOIE	0	0	0	IEDG	OLV	0000 00u0

ICIE — Input capture interrupt enable

1 (set) – Input capture interrupt enabled.

0 (clear) – Input capture interrupt disabled.

OCIE — Output compare interrupt enable

1 (set) – Output compare interrupt enabled.

0 (clear) – Output compare interrupt disabled.

TOIE — Timer overflow interrupt enable

1 (set) – Timer overflow interrupt enabled.

0 (clear) – Timer overflow interrupt disabled.

IEDG — Input edge

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

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

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

OLV — Output level

1 (set) – A high output level will appear on the TCMP pin.

0 (clear) – A low output level will appear on the TCMP pin.

When OLV is set, a high output level will be clocked into the output level register by the next successful output compare, and will appear on the TCMP pin. When clear, it will be a low level that will appear on the TCMP pin.

6

6.2.2 Timer status register – TSR

The Timer Status register ($13) contains the status bits for the above three interrupt conditions — ICF, OCF, TOF.

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

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

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 at $15.

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 at $17.

TOF — Timer overflow flag

1 (set) – Timer overflow has occurred.

0 (clear) – No timer overflow has occurred.

This bit is set when the free-running counter overflows from $FFFF to $0000; a timer overflow interrupt will occur, if TOIE is set. TOF is cleared by reading the TSR and the counter low register, $19.

6

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
Input capture low register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Input capture high (ICH)	$0014	(bit 15)							(bit 8)	Undefined
Input capture low (ICL)	$0015									Undefined

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 $14, the least significant in the input capture low register at $15.

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 ($14), the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period. A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since the two actions occur on opposite edges of the internal bus clock.

The contents of the input capture register are undefined following reset.

6

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.

6.2.6 Output compare high register
Output compare low register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare high (OCH)	$0016	(bit 15)							(bit 8)	Undefined
Output compare low (OCL)	$0017									Undefined

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (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 and the output level (OLV) bit clocked to the output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt 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 ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB will not inhibit the compare function. The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLV) bit is clocked to the output level register whether the output compare flag (OCF) is set or clear. 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

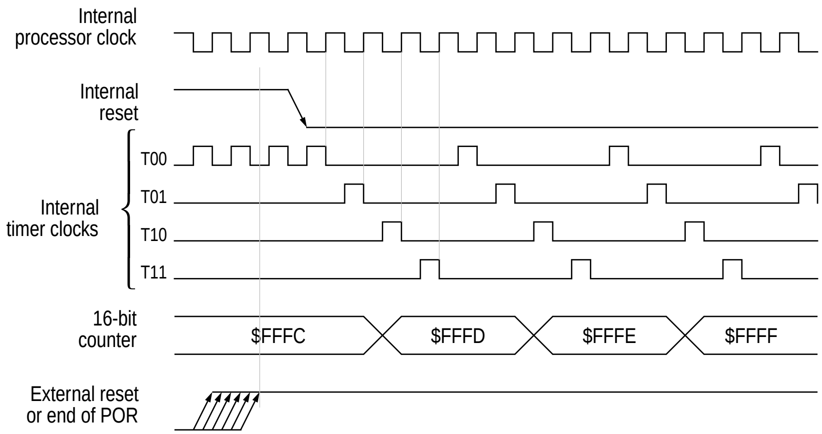
In WAIT mode all CPU action is suspended, but the programmable timer continues counting. An interrupt from an input capture, an output compare or a timer overflow, if enabled, will cause the processor to exit WAIT mode.

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.

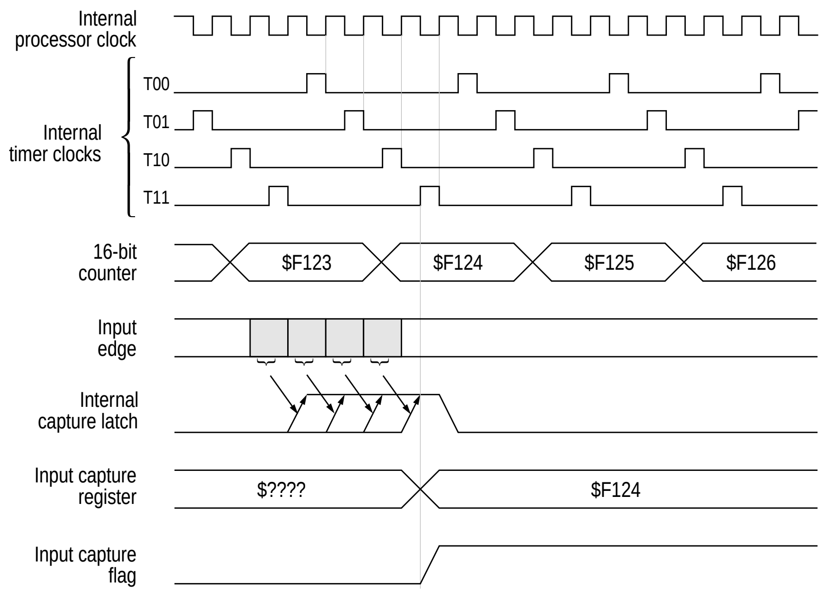
6.5 Timer state diagrams

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.

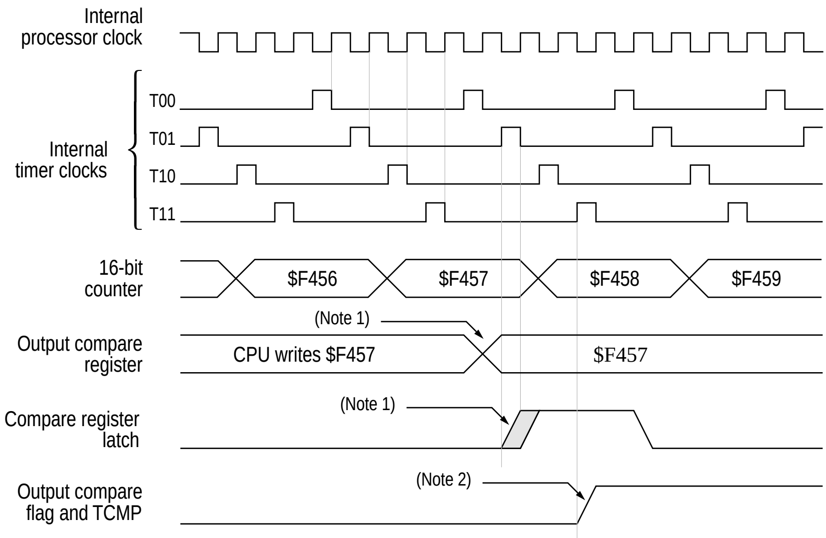
Figure 6-2 Timer state timing diagram for reset



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.

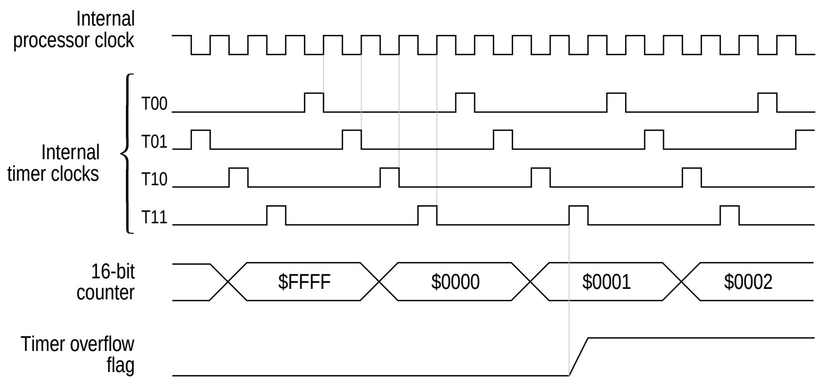
Figure 6-3 Timer state timing diagram for input capture

6



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
RESETS AND INTERRUPTS

7.1 Resets

The MC68HC05J3 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 reset, if the watchdog timer is enabled. Any of these resets will cause the program to go to its starting address, specified by the contents of memory locations $0FFE and $0FFF, and cause the interrupt mask of the condition code register to be set.

7

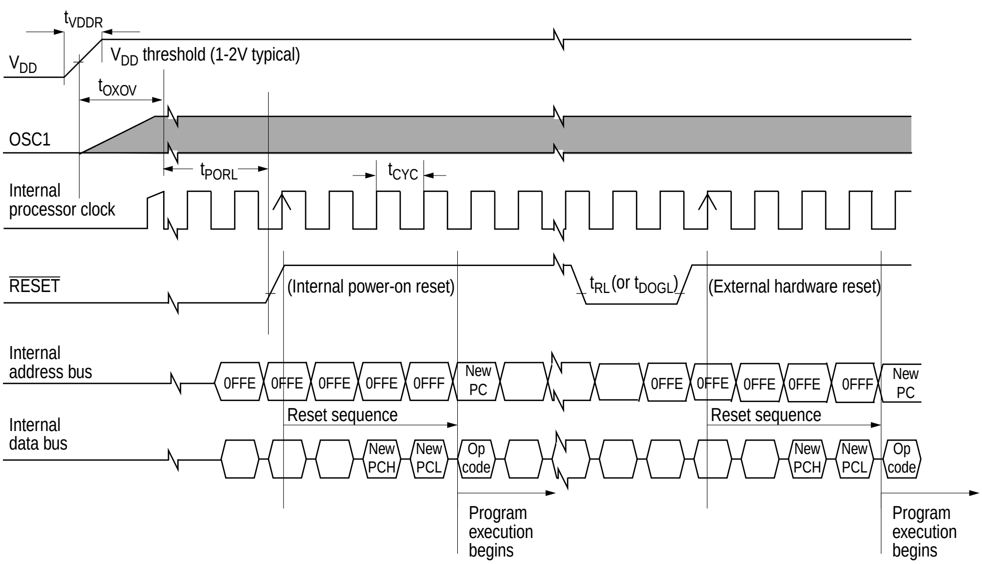


Figure 7-1 Reset timing diagram

7.1.1 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 stabilization 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 stabilize.

7.1.2 $\overline{\text{RESET}}$ 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 the $\overline{\text{RESET}}$ pin goes high, the MCU will resume operation on the following cycle.

7.1.3 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 by a mask option (see Section 1.2).

7.2 Functions affected by reset

When processing stops within the MCU for any reason, i.e. power-on reset, external reset or the execution of a STOP or WAIT instruction, various internal functions of the MCU are affected. Table 7-1 shows the resulting action of any type of system reset, but not necessarily in the order in which they occur.

Table 7-1 Effect of $\overline{\text{RESET}}$, POR, STOP and WAIT

Function/effect	$\overline{\text{RESET}}$	POR	WAIT	STOP
16-bit timer prescaler set to zero	x	x	-	-
16-bit timer counter set to $FFFC	x	x	-	-
All timer enable bits cleared (disable)	x	x	-	-
Data direction registers cleared (inputs)	x	x	-	-
Stack pointer set to $00FF	x	x	-	-
Force internal address bus to restart	x	x	-	-
Vector $0FFE, $0FFF	x	x	-	-
Interrupt mask bit (I-bit in CCR) set	x	x	-	-
Interrupt mask bit (I-bit in CCR) cleared	-	-	x	x
Reset STOP latch	x	x	-	-
Reset $\overline{\text{IRQ}}$ latch	x	x	-	-
Reset WAIT latch	x	x	-	-
Oscillator disabled for 4064 cycles	-	x	-	x
Timer clock cleared	-	x	-	x
Watchdog counter reset	x	x	x	x

7.3 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. While executing the RTI instruction, the interrupt mask bit (I-bit) will be cleared providing the corresponding enable bit stored on the stack is zero, i.e. the interrupt is disabled.

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.

Note: Power-on or external reset clear all interrupt enable bits thus preventing interrupts during the reset sequence.

7.3.1 Interrupt priorities

Each potential interrupt source is assigned a priority which means that if more than one interrupt is pending at the same time, the processor will service the one with the highest priority first. For example, if both an external interrupt and a timer interrupt are pending after an instruction execution, the external interrupt is serviced first.

Table 7-2 shows the relative priority of all the possible interrupt sources. Figure 7-2 shows the interrupt processing flow.

Table 7-2 Interrupt priorities

Source	Register	Flags	Vector address	Priority
Reset	—	—	\$0FFE, \$0FFF	highest
Software interrupt (SWI)	—	—	\$0FFC, \$0FFD	
External interrupt ($\overline{\text{IRQ}}$)/ keyboard interrupt	— CONFB	— PTBIF	\$0FFA, \$0FFB	
Core timer	CTCSR	CTOF, RTIF	\$0FF8, \$0FF9	
16-bit timer – input capture	TSR	ICF	\$0FF6, \$1FF7	
16-bit timer – output compare	TSR	OCF	\$0FF4, \$0FF5	
16-bit timer – overflow	TSR	TOF	\$0FF2, \$0FF3	
Reserved	—	—	\$0FF0, \$0FF1	

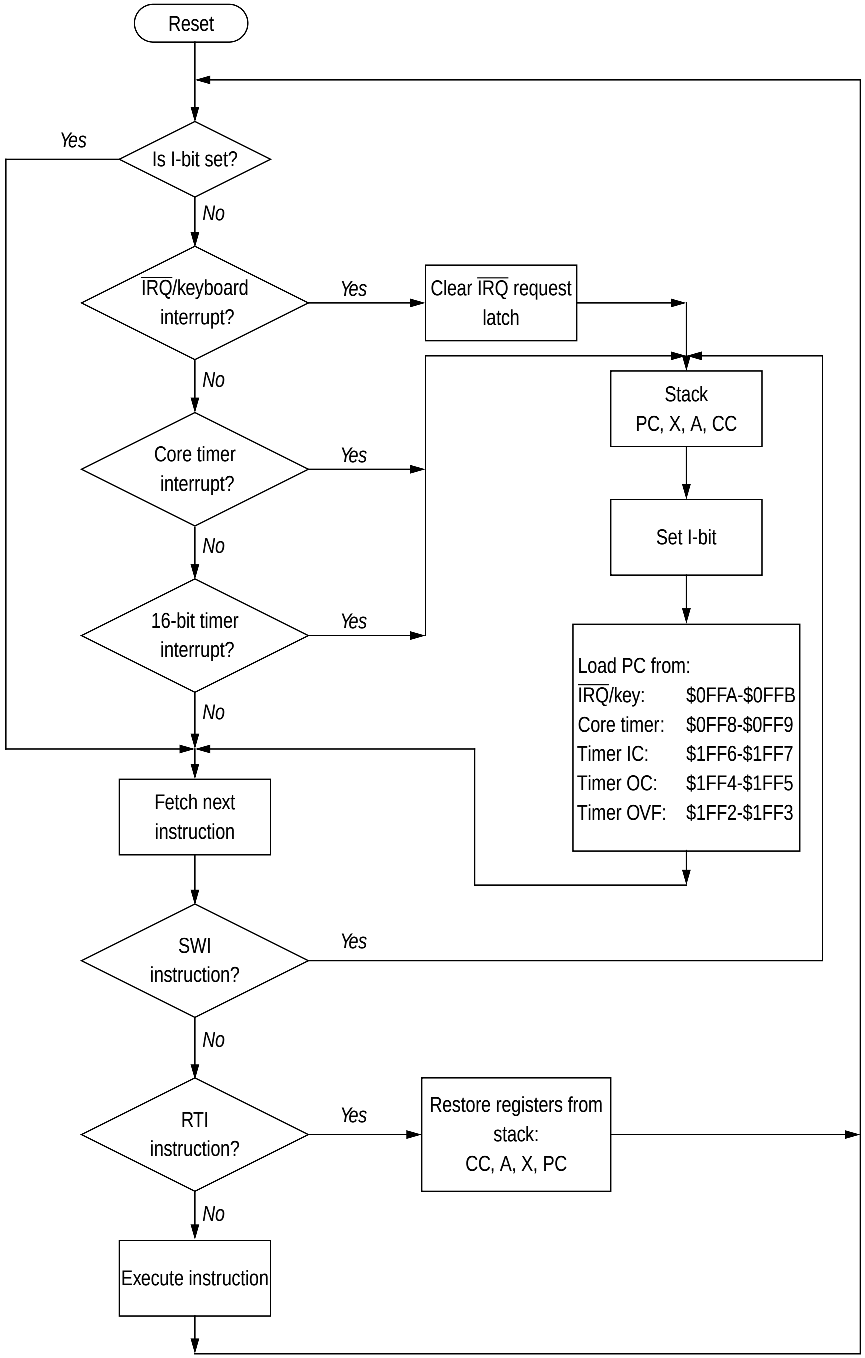


Figure 7-2 Interrupt flow chart

7.4 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.

7.5 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.

7

7.5.1 External interrupt ($\overline{\text{IRQ}}$ or keyboard)

These external interrupt sources will vector to the same interrupt service routine, whose start address is contained in memory locations $0FFA and $0FFB. $\overline{\text{IRQ}}$ can be selected to be either edge sensitive or edge-and-level sensitive. Further details of the keyboard interrupt facility can be found in Section 4.3.

7.5.2 Core timer interrupts

There are two core timer interrupt flags that cause an interrupt whenever an interrupt is enabled and its flag becomes set (RTIF and CTOF). The interrupt flags and enable bits are located in the core timer 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. Full details of the core timer can be found in Section 5.

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.

7.5.3 16-bit timer interrupts

There are three different timer interrupt flags (ICF, OCF and TOF) that will cause a timer interrupt whenever they are set and enabled. These three interrupt flags are found in the three most significant bits of the timer status register (TSR) at location \$13. ICF will vector to the service routine defined by \$0FF6 - \$0FF7, OCF will vector to the service routine defined by \$0FF4 - \$0FF5 and TOF will vector to the service routine defined by \$0FF2 - \$0FF3 as shown in Table 7-2.

There are three corresponding enable bits; ICIE, OCIE and TOIE which are located in the timer control register (TCR) at address \$12. Full details of the programmable timer can be found in Section 6.

7.6 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 2-4.

RESET: A reset condition causes the program to vector to its starting address, which is contained in memory locations \$1FFE (MSB) and \$1FFF (LSB). The I-bit in the condition code register is also set, to disable interrupts.

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

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

7

7

THIS PAGE LEFT BLANK INTENTIONALLY

8

CPU CORE AND INSTRUCTION SET

This section provides a description of the CPU core registers, the instruction set and the addressing modes of the MC68HC05J3.

8.1 Registers

The MCU contains five registers, as shown in the programming model of Figure 8-1. The interrupt stacking order is shown in Figure 8-2.

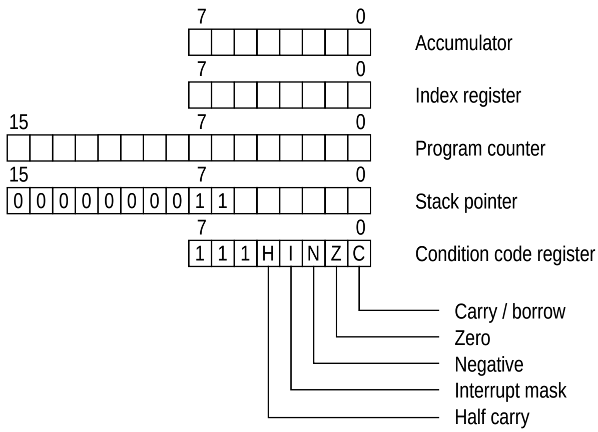


Figure 8-1 Programming model

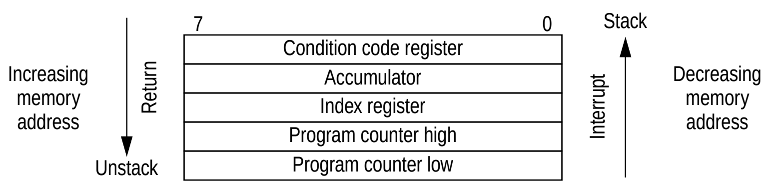


Figure 8-2 Stacking order

8.1.1 Accumulator (A)

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

8.1.2 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.

8.1.3 Program counter (PC)

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

8.1.4 Stack pointer (SP)

8

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.

8.1.5 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.

8.2 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 8-1.

8.2.1 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 8-2 for a complete list of register/memory instructions.

8.2.2 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 8-3.

8.2.3 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 8-4.

8.2.4 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 8-5 for a complete list of read/modify/write instructions.

8.2.5 Control instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 8-6 for a complete list of control instructions.

8.2.6 Tables

Tables for all the instruction types listed above follow. In addition there is a complete alphabetical listing of all the instructions (see Table 8-7 and Table 8-8), and an opcode map for the instruction set of the M68HC05 MCU family (see Table 8-9).

Table 8-1 MUL instruction

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

Table 8-2 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 8-3 Branch instructions

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

Table 8-4 Bit manipulation instructions

Function	Mnemonic	Addressing Modes					
		Bit set/clear			Bit test and branch		
		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 8-5 Read/modify/write instructions

Function	Mnemonic	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 8-6 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

8

Table 8-7 Instruction set (1 of 2)

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

Address mode abbreviations

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

Not implemented

Condition code symbols

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

Table 8-8 Instruction set (2 of 2)

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

Address mode abbreviations

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

(shaded box) 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

8

Table 8-9 M68HC05 opcode map

Cell entries are written as: bytes MNEMONIC cycles address mode. Empty cells are shaded on the page (not implemented).

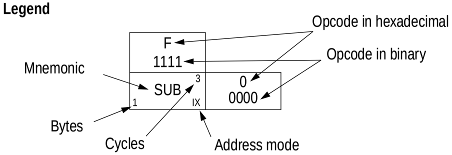
	Bit manipulation		Branch	Read/modify/write					Control		Register/memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
Low \ High	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	High / Low
0 0000	3 BRSET0 5 BTB	2 BSET0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

Abbreviations for address modes and registers

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

Not implemented

Legend



8.3 Addressing modes

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

The term ‘effective address’ (EA) is used in describing the various addressing modes. The effective address is defined as the address from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate ‘contents of’ the location or register referred to. For example, (PC) indicates the contents of the location pointed to by the PC (program counter). An arrow indicates ‘is replaced by’ and a colon indicates concatenation of two bytes. For additional details and graphical illustrations, refer to the ***M6805 HMOS/M146805 CMOS Family Microcomputer/ Microprocessor User's Manual*** or to the ***M68HC05 Applications Guide***.

8.3.1 Inherent

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

8

8.3.2 Immediate

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

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

8.3.3 Direct

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

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

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

8

8.3.4 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)

8.3.5 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

8.3.6 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)

8.3.7 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)

8.3.8 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.

$$EA = PC+2+(PC+1);\ PC \leftarrow EA \text{ if branch taken;}$$
$$\text{otherwise } EA = PC \leftarrow PC+2$$

8.3.9 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.

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

8

8.3.10 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.

$$EA1 = (PC+1);\ PC \leftarrow PC+2$$
$$\text{Address bus high} \leftarrow 0;\ \text{Address bus low} \leftarrow (PC+1)$$
$$EA2 = PC+3+(PC+2);\ PC \leftarrow EA2 \text{ if branch taken;}$$
$$\text{otherwise } PC \leftarrow PC+3$$

8

THIS PAGE LEFT BLANK INTENTIONALLY

9
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the MC68HC05J3.

9.1 Maximum ratings

Table 9-1 Maximum ratings

Rating	Symbol	Value	Unit
Supply voltage[1]	V_{DD}	– 0.3 to +7.0	V
Input voltage –ports, OSC1, $\overline{\text{RESET}}$	V_{IN}	V_{SS} – 0.3 to V_{DD} + 0.5	V
Input voltage – $\overline{\text{IRQ}}$ (bootloader mode)	V_{IN}	V_{SS} – 0.3 to 1.8 x V_{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 (excluding VDD and VSS) –Source[2] –Sink[3]	 I_D I_S	 25 8	 mA 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.

(3) Applicable to PA0–7 and PB0–5; maximum 45mA per device.

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}.

9.2 Thermal characteristics and power considerations

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

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

where:

T_A = Ambient temperature (°C)

θ_{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:

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

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

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

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 9-2.

Table 9-2 Package thermal characteristics

Characteristics	Symbol	Value	Unit
Thermal resistance			
– 20-pin SOIC package	θ_{JA}	60	°C/W
– 20-pin DIL package	θ_{JA}	60	°C/W

Pins	R1	R2	C
PA0–7, PB0–5	3.26kΩ	2.38kΩ	50pF

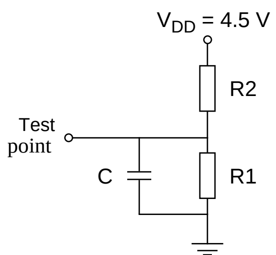


Figure 9-1 Equivalent test load

9.3 DC electrical characteristics

Table 9-3 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 high voltage (I_{LOAD} = – 25 µA)	V_{OH}	V_{DD} – 0.1	—	—	V
Output low voltage (I_{LOAD} = +25 µA)	V_{OL}	—	—	0.1	V
Output high voltage (I_{LOAD} = – 0.8 mA) PA0–7, PB0–5	V_{OH}	V_{DD} – 0.8	V_{DD} – 0.1	—	V
Output low voltage (I_{LOAD} = +1.6 mA) PA0–7, PB0–5	V_{OL}	—	0.2	0.4	V
Output low voltage (I_{LOAD} = +8 mA) PA0–7, PB0–5	V_{OL}	—	0.4	0.8	V
Input high voltage PA0–7, PB0–5, OSC1, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	0.7V_{DD}	—	V_{DD}	V
Input low voltage PA0–7, PB0–5, OSC1, $\overline{IRQ}$, $\overline{RESET}$	V_{IL}	V_{SS}	—	0.2V_{DD}	V
Supply current[3] RUN (at 2.1 MHz bus frequency) WAIT (at 2.1 MHz bus frequency) STOP (oscillators off, 0 to 70°)	I_{DD}	 — — —	 5.0 2.5 1	 10 4 100	 mA mA µA
I/O ports high-Z leakage current PA0–7, PB0–5	I_{OZ}	—	—	±10	µA
Inputs high-Z leakage current $\overline{IRQ}$, $\overline{RESET}$, OSC1	I_{OZ}	—	—	±1	µA
Capacitance Ports (as input or output) $\overline{IRQ}$, $\overline{RESET}$	 C_{OUT} C_{IN}	 — —	 — —	 12 TBD	 pF pF

(1) All I_{DD} 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} = 4.2 MHz); all inputs 0.2V from rail; no DC loads; maximum load on outputs 50pF (except OSC2 load 20pF).
WAIT I_{DD}: only the timer system active; current varies linearly with the OSC2 capacitance.
WAIT and STOP I_{DD}: all ports configured as inputs; V_{IL} = 0.2V and V_{IH} = V_{DD} – 0.2V.
STOP I_{DD}: measured with OSC1 = V_{DD}.

9

Table 9-4 DC electrical characteristics for 3.3V operation

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Characteristic[(1)]	Symbol	Min	Typ[(2)]	Max	Unit
Output high voltage (I_{LOAD} = – 25 µA)	V_{OH}	V_{DD} – 0.1	—	—	V
Output high voltage (I_{LOAD} = + 25 µA)	V_{OH}	—	—	0.1	V
Output high voltage (I_{LOAD} = – 0.2 mA) PA0–7, PB0–5	V_{OH}	V_{DD} – 0.3	V_{DD} – 0.3	—	V
Output low voltage (I_{LOAD} = +0.4 mA) PA0–7, PB0–5	V_{OL}	—	0.2	0.3	V
Output low voltage (I_{LOAD} = +4 mA) PA0–7, PB0–5	V_{OL}	—	0.4	0.5	V
Input high voltage PA0–7, PB0–5, OSC1, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	$0.7V_{DD}$	—	V_{DD}	V
Input low voltage PA0–7, PB0–5, OSC1, $\overline{IRQ}$, $\overline{RESET}$	V_{IL}	V_{SS}	—	$0.2V_{DD}$	V
Supply current[(3)] RUN (at 1.05 MHz bus frequency) WAIT (at 1.05 MHz bus frequency) STOP (oscillators off, 0 to 70°)	I_{DD}	 — — —	 2.0 1.0 1	 5 2 50	 mA mA µA
I/O ports high-Z leakage current PA0–7, PB0–5	I_{OZ}	—	—	±10	µA
Inputs high-Z leakage current $\overline{IRQ}$, $\overline{RESET}$, OSC1	I_{OZ}	—	—	±1	µA
Capacitance Ports (as input or output) $\overline{IRQ}$, $\overline{RESET}$	 C_{OUT} C_{IN}	 — —	 — —	 12 TBD	 pF pF

(1) All I_{DD} 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} = 2.1MHz); all inputs 0.2V from rail; no DC loads; maximum load on outputs 50pF (except OSC2 load 20pF).
WAIT I_{DD}: only the timer system active; current varies linearly with the OSC2 capacitance.
WAIT and STOP I_{DD}: all ports configured as inputs; V_{IL} = 0.2V and $V_{IH} = V_{DD}$ – 0.2V.
STOP I_{DD}: measured with OSC1 = V_{DD}.

9.4 AC electrical characteristics

Table 9-5 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				
Crystal	f_{OSC}	—	4.2	MHz
External clock	f_{OSC}	dc	4.2	MHz
Internal operating frequency				
Crystal (f_{OSC} /2)	f_{OP}	—	2.1	MHz
External clock (f_{OSC} /2)	f_{OP}	dc	2.1	MHz
Processor cycle time	t_{CYC}	480	—	ns
Ceramic resonator start-up time	t_{OXOV}	—	10	ms
Ceramic resonator STOP recovery start-up time	t_{ILCH}	—	10	ms
OSC1 pulse width	t_{OH}, t_{OL}	90	—	ns
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{CYC}
16-bit timer				
Resolution[1]	t_{RESL}	—	4	t_{CYC}
Input capture pulse width	t_{TLTH}	250	—	ns
Input capture pulse period	t_{TLTL}	(2)	—	t_{CYC}
Power-on reset delay	t_{PORL}	4064	4064	t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	125	—	ns
Interrupt pulse period (see Figure 9-2)	t_{ILIL}	(3)	—	t_{CYC}

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

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

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

9

Table 9-6 AC electrical characteristics for 3.3V operation

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Characteristic	Symbol	Min	Max	Unit
Frequency of operation				
Crystal	f_{OSC}	—	2.1	MHz
External clock	f_{OSC}	dc	2.1	MHz
Internal operating frequency				
Crystal (f_{OSC} /2)	f_{OP}	—	1.05	MHz
External clock (f_{OSC} /2)	f_{OP}	dc	1.05	MHz
Processor cycle time	t_{CYC}	1000	—	ns
Ceramic resonator start-up time	t_{OXOV}	—	20	ms
Ceramic resonator STOP recovery start-up time	t_{ILCH}	—	20	ms
OSC1 pulse width	t_{OH}, t_{OL}	200	—	ns
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{CYC}
16-bit timer				
Resolution(1)	t_{RESL}	—	4	t_{CYC}
Input capture pulse width	t_{TLTH}	500	—	ns
Input capture pulse period	t_{TLTL}	(2)	—	t_{CYC}
Power-on reset delay	t_{PORL}	4064	4064	t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	250	—	ns
Interrupt pulse period (see Figure 9-2)	t_{ILIL}	(3)	—	t_{CYC}

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

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

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

Edge-sensitive trigger — The minimum t_{ILIH} is either 125ns (V_{DD}=5V) or 250ns (V_{DD}=3.3V). The minimum period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 21 t_{CYC}.

Edge and level sensitive trigger — If $\overline{\text{IRQ}}$ remains low after the initial interrupt is serviced, the MCU recognises the interrupt until the $\overline{\text{IRQ}}$ line returns to a high level.

Figure 9-2 External interrupt timing

10
MECHANICAL DATA

The MC68HC05J3 is available in both 20-pin SOIC and 20-pin PDIP packages.

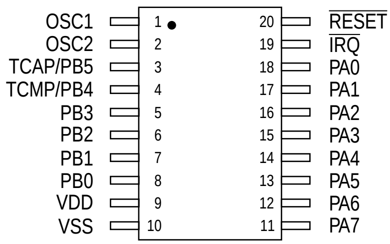


Figure 10-1 20-pin SOIC pinout

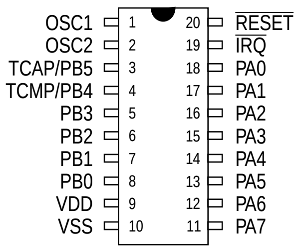
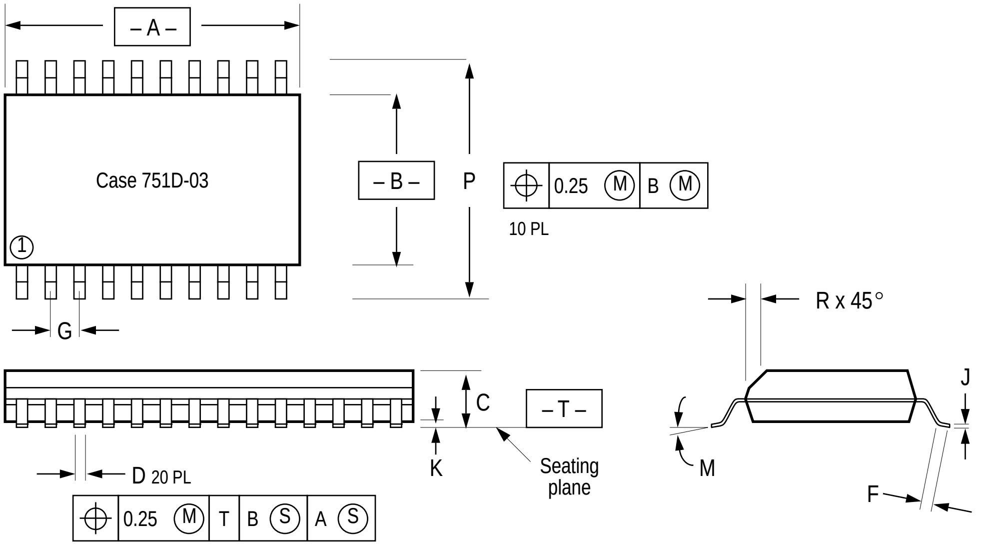


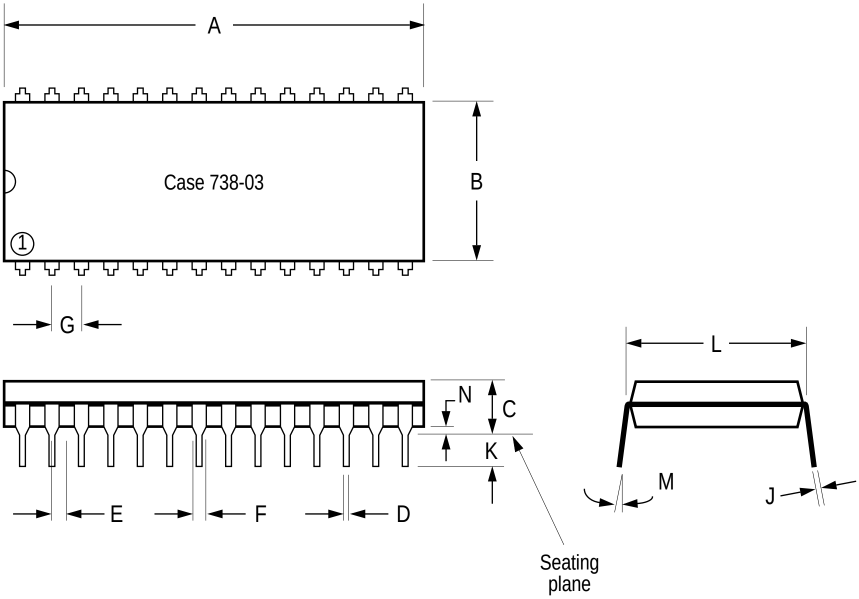
Figure 10-2 20-pin PDIP pinout



Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	12.65	12.95	1. Dimensions 'A' and 'B' are datums and 'T' is a datum surface. 2. Dimensioning and tolerancing per ANSI Y14.5M, 1982. 3. All dimensions in mm. 4. Dimensions 'A' and 'B' do not include mould protrusion. 5. Maximum mould protrusion is 0.15 mm per side.	J	0.25	0.32
B	7.40	7.60		K	0.10	0.25
C	2.35	2.65		M	0°	7°
D	0.35	0.49		P	10.05	10.55
F	0.50	0.90		R	0.25	0.75
G	1.27 BSC			—	—	—

10

Figure 10-3 20-pin SOIC mechanical dimensions



<table>
<tr><th>Dim.</th><th>Min.</th><th>Max.</th><th>Notes</th><th>Dim.</th><th>Min.</th><th>Max.</th></tr>
<tr><td>A</td><td>25.66</td><td>27.17</td><td rowspan="6">1. All dimensions in mm.
2. Positional tolerance of leads (‘D’) shall be within 0.25 mm at maximum material condition, in relation to seating plane and to each other.
3. Dimension ‘L’ is to centre of leads when formed parallel.
4. Dimension ‘B’ does not include mould protrusion.</td><td>G</td><td colspan="2">2.54 BSC</td></tr>
<tr><td>B</td><td>6.10</td><td>6.60</td><td>J</td><td>0.21</td><td>0.38</td></tr>
<tr><td>C</td><td>3.81</td><td>4.57</td><td>K</td><td>2.80</td><td>3.55</td></tr>
<tr><td>D</td><td>0.39</td><td>0.55</td><td>L</td><td colspan="2">7.62 BSC</td></tr>
<tr><td>E</td><td colspan="2">1.27 BSC</td><td>M</td><td>0°</td><td>15°</td></tr>
<tr><td>F</td><td>1.27</td><td>1.77</td><td>N</td><td>0.51</td><td>1.01</td></tr>
</table>

Figure 10-4 20-pin PDIP mechanical dimensions

10

THIS PAGE LEFT BLANK INTENTIONALLY

11
ORDERING INFORMATION

This section describes the information needed to order the MC68HC05J3.

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 11-1 for appropriate part numbers.

Table 11-1 MC order numbers

Device title	Package type	Temperature	Part number
MC68HC05J3	20-pin plastic PDIP	0 to +70°C	MC68HC05J3P
		–40 to + 85°C	MC68HC05J3CP
	20-pin SOIC	0 to +70°C	MC68HC05J3DW
		–40 to + 85°C	MC68HC05J3CDW

11.1 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.

11

11.2 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.

11.3 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.

11

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.
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.

G

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.
$\overline{\textbf{IRQ}}$	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.

$\overline{\mathbf{XIRQ}}$ Non-maskable interrupt request. The overline indicates that this has an active-low signal format.

G

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

freescale™ semiconductor