HC05

MC68HC05E16

TECHNICAL
DATA

MC68HC05E16

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

Conventions

Where abbreviations are used in the text, an explanation can be found in the glossary, at the back of this manual. Register and bit mnemonics are defined in the paragraphs describing them.

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

Unless otherwise stated, shaded cells in a register diagram indicate that the bit is either unused or reserved; 'u' is used to indicate an undefined state (on reset).

TABLE OF CONTENTS

6
I²C-BUS

7
A/D CONVERTER

11
PHASE-LOCKED LOOP

12
OSCILLATOR SYSTEM

13
CPU CORE AND INSTRUCTION SET

LIST OF FIGURES

LIST OF TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1
INTRODUCTION

The MC68HC05E16 HCMOS 8-bit microcomputer (MCU) is a member of the M68HC05 family. It has 16256 bytes of user ROM, 320 bytes of EEPROM and 352 bytes of RAM. In addition to its on-board memory, the MC68HC05E16 has a 16-bit programmable timer, a core timer and computer operating properly (COP) watchdog, a multiplexed dual I^2C bus (M-bus) and a two channel A/D converter. The device also incorporates an on-board 32 kHz PLL, which permits the use of, for example, a low cost watch crystal to generate the MCU clock. With these features, and its 49 dedicated I/O lines, the MC68HC05E16 is a versatile general purpose device and is particularily suited to many telecommunications and consumer applications.

1.1 Features

- Fully static design featuring the industry standard M68HC05 core
- Choice of oscillator: internal RC, external crystal/ceramic, or 32 kHz PLL (software selectable)
- 16240 bytes of user ROM and 16 bytes of user vectors: 128 bytes of bootloader ROM
- 320 bytes EEPROM
- 352 bytes of RAM
- 16-bit programmable timer with two input captures and two output compares
- 15-stage multi-functional core timer with overflow, real time interrupt and watchdog
- Two hardware interrupts
- Custom periodic interrupt at 0.25, 0.5 or 1s intervals (software selectable), when using a 32 kHz crystal
- Keyboard interrupt facility
- Multiplexed dual I^2C bus (M-bus)
- 2-channel 8-bit analog-to-digital converter
- 49 dedicated I/O Lines (29 on 44-pin QFP); 47 bidirectional and two input-only lines

- Programmable 32 kHz phase-locked-loop (PLL) synthesizer with programmable bus speed (x16, 32, 64 or 128) and buffered crystal output, plus a low power 16 kHz run mode
- Low voltage detection capability
- Power saving STOP and WAIT modes
- Available in 64-pin QFP package (a 44-pin QFP package is available for the MC68HC05E16 for cost and space sensitive applications) and in 56-pin SDIP package

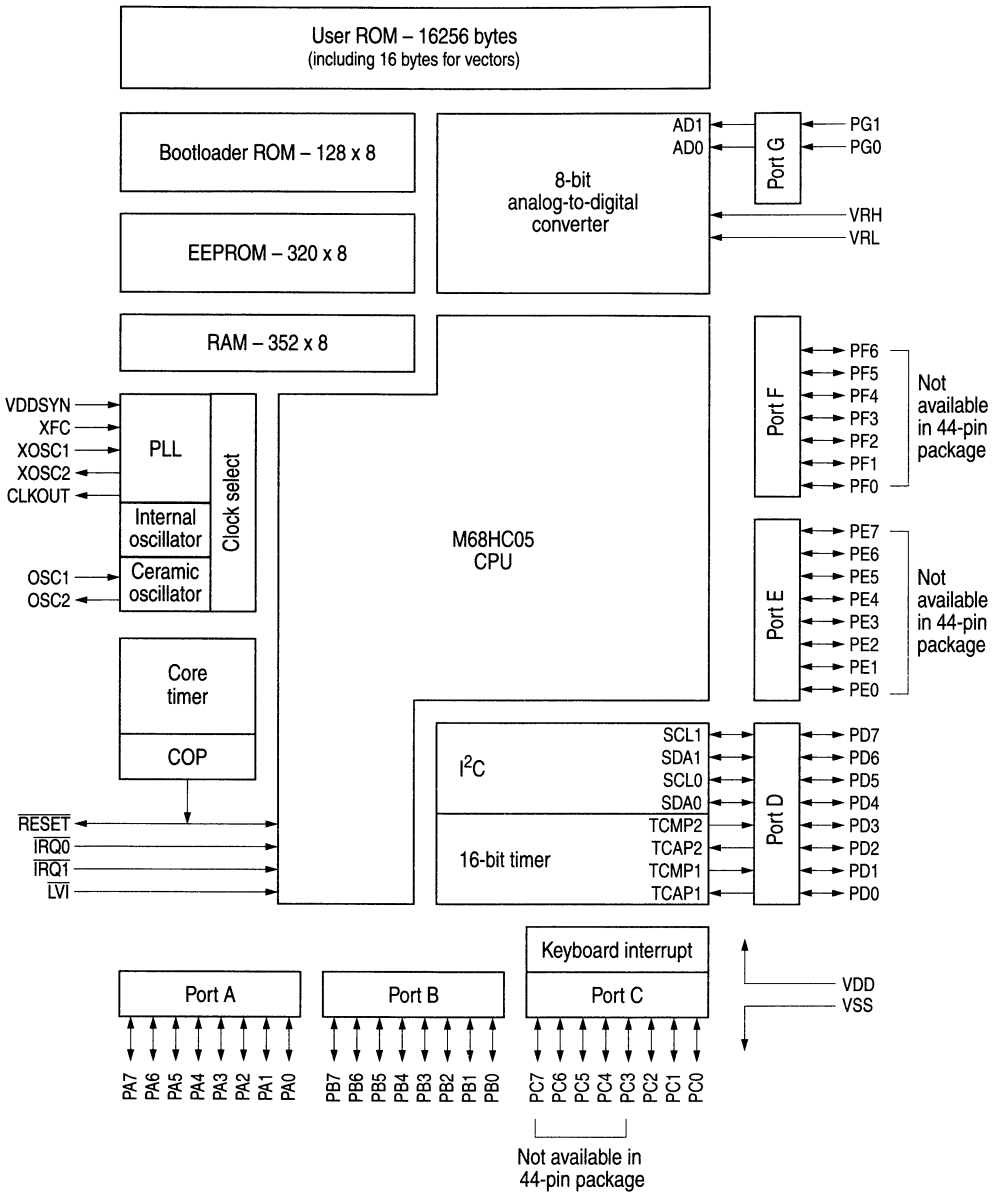


Figure 1-1 MC68HC05E16 block diagram

1.2 Mask options

The MC68HC05E16 has two mask options:

- Stop instruction enable/disable
- COP enable/disable

These mask options are programmed during manufacture and must be specified at the time of ordering.

THIS PAGE LEFT BLANK INTENTIONALLY

2
MODES OF OPERATION AND PIN DESCRIPTIONS

2.1 Modes of operation

The MC68HC05E16 has only one mode of operation available to the user: single-chip mode. Single-chip mode is the normal user operating mode of M68HC05 devices.

2.1.1 Single-chip mode

In single-chip mode, the MCU functions as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions, including the five 8-bit I/O ports, the 7-bit I/O port and the 2-bit input-only port. Some port pins share their functions with the timer, M-bus and the A/D converter. All address and data activity occurs within the MCU.

2.2 Low-power modes

2.2.1 STOP

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

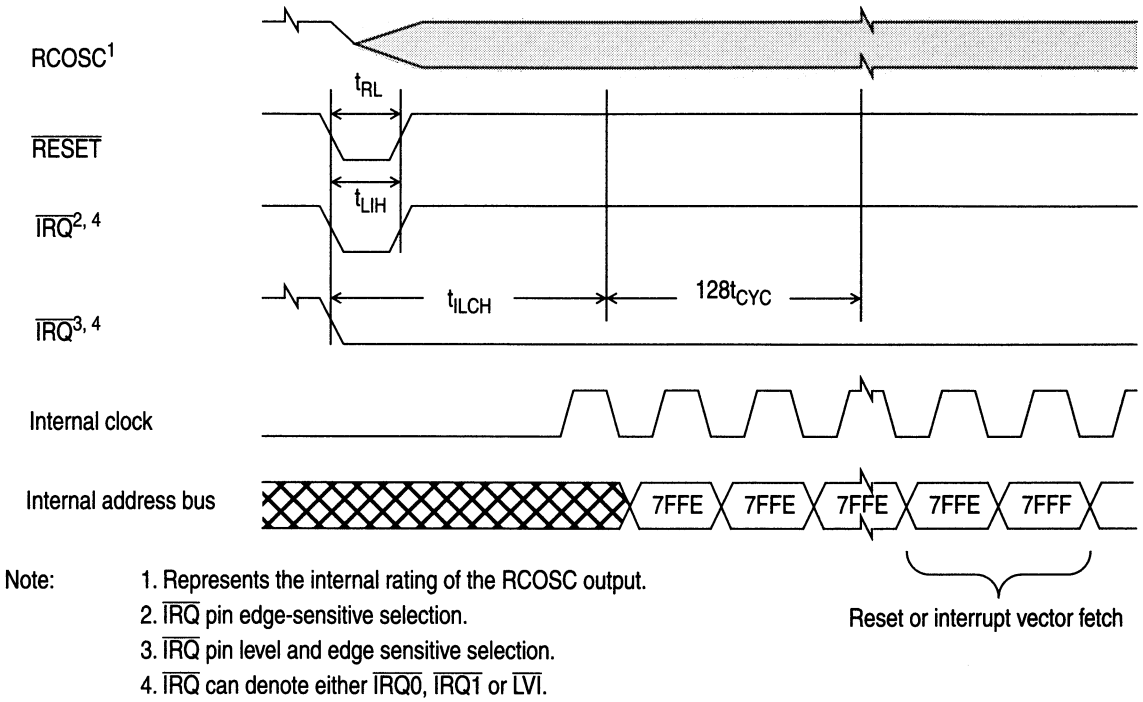
During STOP mode, the core timer interrupt flags (CTOF and RTIF) and interrupt enable bits (CTOE and RTIE) in the CTCSR are cleared by internal hardware. The I-bit in the CCR is cleared to enable external interrupts. All other registers, including the remaining bits in the CTCSR, and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of STOP mode only by an external interrupt, ($\overline{\text{LVI}}$, $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$ or keyboard wake-up) or $\overline{\text{RESET}}$.

Refer also to Section 12 for the behaviour of the different oscillators during stop mode and stop recovery.

The STOP instruction can be disabled by a mask option. When disabled, the STOP instruction is executed as a NOP.

2.2.2 STOP recovery

The processor can be brought out of STOP mode only by an external interrupt ($\overline{\text{LVI}}$, $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$ or keyboard wake-up) or $\overline{\text{RESET}}$. See Figure 2-1.



Note:
1. Represents the internal rating of the RCOSC output.
2. $\overline{\text{IRQ}}$ pin edge-sensitive selection.
3. $\overline{\text{IRQ}}$ pin level and edge sensitive selection.
4. $\overline{\text{IRQ}}$ can denote either $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$ or $\overline{\text{LVI}}$.

Figure 2-1 Stop recovery timing

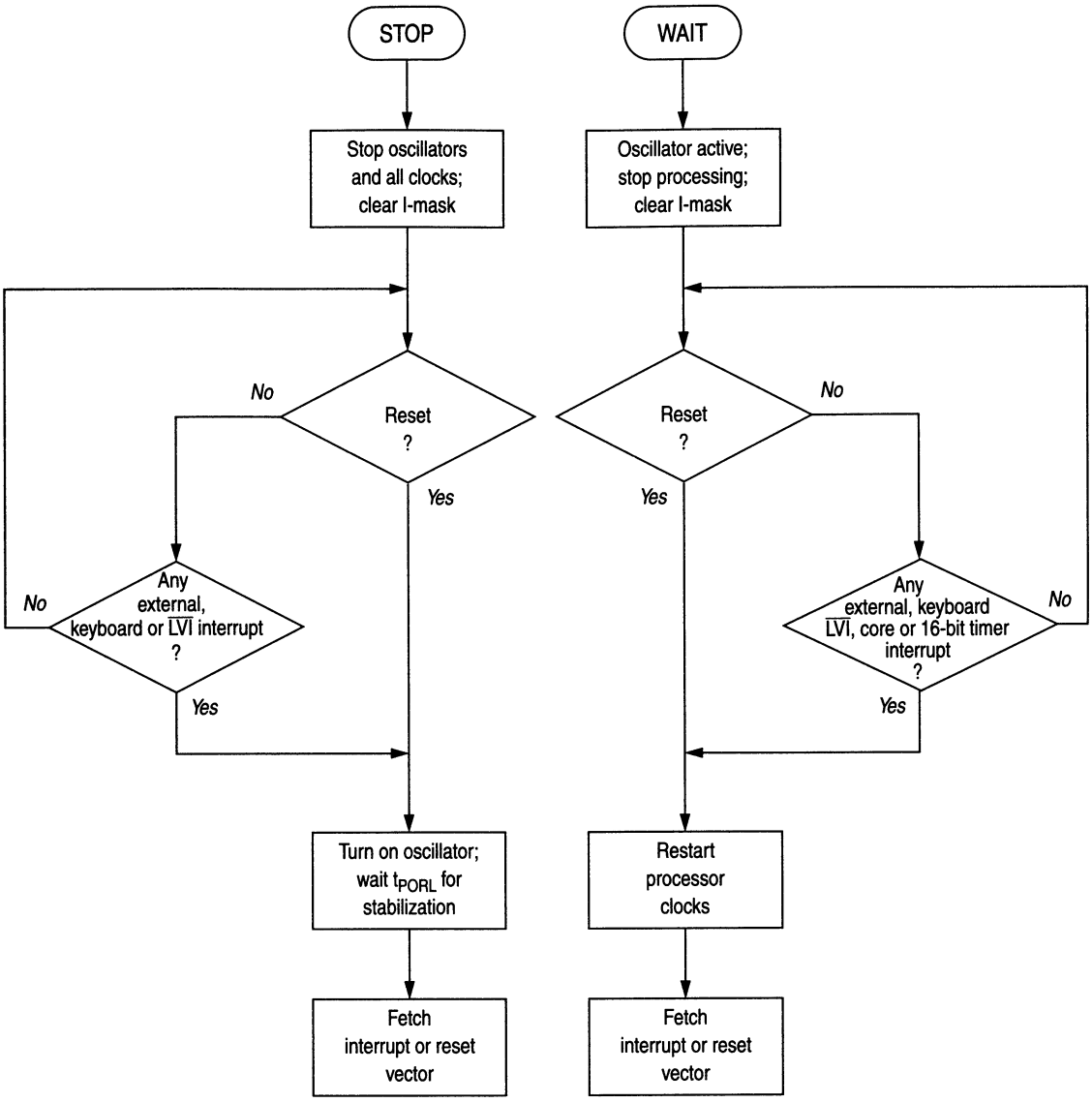


Figure 2-2 STOP/WAIT flowcharts

2.2.3 WAIT

The WAIT instruction places the MCU in a low power consumption mode, though it consumes more power than in STOP mode. All CPU action is suspended, but the core timer and the 16-bit timer remain active. An interrupt from either timer, if enabled, will cause the MCU to exit WAIT mode, as well as the standard external/hardware interrupts.

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 core timer or 16-bit timer may be enabled to allow a periodic exit from the WAIT mode. M-bus in slave mode or a custom periodic interrupt may also cause the device to exit WAIT mode.

2.2.4 Data retention mode

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

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

2.3 Pin descriptions

2.3.1 VDD and VSS

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

2.3.2 $\overline{\text{LVI}}$

This additional interrupt pin activates an interrupt when the supply voltage to the chip falls below a low voltage threshold. The $\overline{\text{LVI}}$ pin contains an internal Schmitt trigger as part of its input circuitry to improve noise immunity. Refer to Section 10.2 for more detail.

2.3.3 $\overline{\text{IRQ0}}$

The interrupt triggering sensitivity of this pin can be programmed as falling-edge or low-level. The $\overline{\text{IRQ0}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to Section 10.2 for more detail.

2.3.4 $\overline{\text{IRQ1}}$

The interrupt triggering sensitivity of this additional interrupt pin can be programmed as falling-edge, rising-edge, low-level and high-level. The $\overline{\text{IRQ1}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. In addition to the $\overline{\text{IRQ0}}$ there is an enable/disable mask-bit and an interrupt flag available for $\overline{\text{IRQ1}}$. Refer to Section 10.2 for more detail.

2.3.5 OSC1, OSC2

These pins provide control input for a ceramic resonator connected to these pins. This oscillator can be chosen instead of the 32 kHz PLL frequency generator to drive the MC68HC05E16 if faster start-up times are required. However, the PLL frequency generator is still required to feed the PLL circuit.

2.3.6 XOSC1, XOSC2, CLKOUT

These pins provide control input for an on-chip clock oscillator PLL circuit. A crystal, a ceramic resonator (usually a 32.768 kHz watch crystal), or an external signal connects to these pins providing a system clock. CLKOUT provides a buffered output of the crystal oscillator.

2.3.7 VDDSYN

This pin provides a separate 'clean' power connection to the PLL synthesizer which should be at the same potential as VDD. This supply should be routed from the source completely separately from the VDD track used for all other components on the PCB, including the microcontroller (see Section 11.2).

2.3.8 XFC

This pin provides a means for connecting an external filter capacitor to the synthesizer phase-locked loop filter. Refer to Section 11 for additional information concerning this capacitor.

2.3.9 $\overline{\text{RESET}}$

This active low pin is used to reset the MCU to a known start-up state by pulling $\overline{\text{RESET}}$ low. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity.

2.3.10 PA0 – PA7, PB0 – PB7

These sixteen I/O lines comprise ports A and B. The state of any pin is software programmable. All port A and B lines are configured as inputs during power-on or reset, with pull-down resistors on port A and pull-up resistors on port B. See Section 4.1 for a detailed description of I/O programming.

2.3.11 PC0 – PC7/keyboard interrupt

The eight I/O lines of port C are shared with the keyboard interrupt function. In addition, a configuration register allows the lines to be defined as inputs with pull-up resistors, or as open-drain outputs. Refer to Section 4.1 for a detailed description of I/O programming. Port C pins contain internal Schmitt triggers as part of their input to improve noise immunity.

2.3.12 PD0/TCAP1, PD1/TCMP1, PD2/TCAP2, PD3/TCMP2, PD4/SDA0, PD5/SCL0, PD6/SDA1 and PD7/SCL1

These eight port lines are shared with other functions to give added flexibility. In addition, a configuration register allows the lines to be defined as inputs with pull up resistors, or open-drain outputs. Port pins PD0, PD1, PD2 and PD3 share pins with timer input capture one (TCAP1), timer output compare one (TCMP1), timer input capture two (TCAP2) and timer output compare two (TCMP2) respectively (see Section 5). Port pins PD4 and PD5 share pins with SDA0 and SCL0, the bidirectional data and clock lines for one I^2C bus connection. PD6 and PD7 pins are shared with SDA1 and SCL1, the bidirectional data and clock lines for the other I^2C bus connection. Internally, the two I^2C buses are multiplexed into one I^2C module. See Section 6. Port D pins contain internal Schmitt triggers as part of their input to improve noise immunity.

2.3.13 PE0 – PE7, PF0 – PF6

These fifteen I/O lines comprise port E and port F and do not share their pins with any other functions. In addition, a configuration register allows the lines to be defined as inputs with pull-up resistors, or as open-drain outputs. See Section 4.1 for a detailed description of I/O programming.

2.3.14 PG0/AD0 – PG1/AD1

These two port lines are shared with the 8-bit A/D converter. These lines can also be used as two digital inputs. See Section 4.7 for more information.

2.3.15 VREFH, VREFL

These pins provide the reference voltages for the A/D converter.

3
MEMORY AND REGISTERS

The MC68HC05E16 has a 32K byte memory map, consisting of registers (for I/O, control and status), user ROM (including user vectors), bootloader ROM, user RAM, and EEPROM, as shown in Figure 3-1.

3.1 Registers

All the I/O, control and status registers of the MC68HC05E16 are contained within the first 64 byte block of the memory map, as shown in Figure 3-1.

3.2 RAM

The user RAM comprises 352 bytes of memory, from $0040 to $019F. This is shared with a 64-byte stack area. The stack begins at $00FF and can extend down to $00C0. See Section 13.1.4.

3.3 ROM

The 16256 bytes of ROM are located from $4000 to $7FFF and are mapped as follows:

- 16240 bytes of user ROM from $4000 to $7F6F
- 16 bytes of user vectors ranging from $7FF0 to $7FFF

3.4 Bootloader ROM

The 112 bytes of bootloader ROM are located from $7F70 to $7FDF, with 16 bytes of bootloader vectors from $7FE0 to $7FDF.

3.5 EEPROM

The 320 bytes of user EEPROM are located at addresses $01C0 to $02FF.

Programming or erasing the EEPROM can be done by the user on a single byte basis; erasing may also be performed on a block or bulk basis. All programming or erasing is accomplished by manipulating the programming register (EEPROG), located at address $001C.

Note: The erased state of an EEPROM byte is '$FF'. This means that a write forces zeros to the bits specified, whilst bits defined as ones are unchanged by a write operation.

3.5.1 EEPROM programming register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPROM programming (EEPROG)	$001C	0	CPEN	0	ER1	ER0	EELATCH	EERC	EEPGM	0000 0000

CPEN — Charge pump enable

1 (set) – Charge pump enabled.

0 (clear) – Charge pump disabled.

When set, CPEN enables the charge pump which produces the internal programming voltage. This bit should be set at the same time as the EELATCH bit. The programming voltage will not be available until EEPGM is set. The charge pump should be disabled when not in use. CPEN is readable and writable and is cleared by reset.

ER1, ER0 — Erase select bits

ER1 and ER0 are used to select either single byte programming or one of three erase modes: byte, block, or bulk. Table 3-1 shows the mode selected for each bit configuration. These bits are readable and writable and are cleared by reset.

Table 3-1 Erase modes

ER1	ER0	Mode
0	0	Program
0	1	Byte erase
1	0	Block erase
1	1	Bulk erase

- In byte erase mode, only the selected byte is erased.
- In block erase mode, a 64-byte block of EEPROM is erased. The EEPROM memory space is divided into five 64-byte blocks ($01C0 – $01FF, $0200 – $023F, $0240 – $027F, $0280 – $02BF and $02C0 – $02FF) and performing a block erase on any address within a block will erase the entire block.
- In bulk erase mode, the entire 320 bytes of EEPROM are erased.

EELATCH — EEPROM latch bit

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

0 (clear) – EEPROM address and data buses are configured for normal operation.

When set, the EELATCH bit configures the EEPROM address and data buses for programming. In addition, writes to the EEPROM array cause the address and data buses to be latched. This bit is readable and writable, but reads from the EEPROM array are inhibited if the EELATCH bit is set and a write to the EEPROM space has taken place. When this bit is clear, address and data buses are configured for normal operation. Reset clears this bit.

EERC — EEPROM RC oscillator control

1 (set) – Use internal RC oscillator for EEPROM.

0 (clear) – Use CPU clock for EEPROM.

When this bit is set, the EEPROM memory array uses the internal RC oscillator instead of the CPU clock. After setting the EERC bit, the user should wait a time t_{RCON} to allow the RC oscillator to stabilize. This bit is readable and writable and should be set by the user when the internal bus frequency falls below 1.0 MHz. Reset clears this bit.

EEPGM — EEPROM programming power enable

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

0 (clear) – Programming power switched off.

EEPGM must be set to enable the EEPGM function. When set, EEPGM turns on the charge pump and enables the programming (or erasing) power to the EEPROM array. When clear, this power is switched off. This will enable pulsing of the programming voltage to be controlled internally. This bit can be read at any time, but can only be written to if EELATCH = 1, i.e. if EELATCH is not set, then EEPGM cannot be set. Reset clears this bit.

3.5.2 Programming and erasing procedures

To program a byte of EEPROM, set EELATCH = CPEN = 1, set ER1 = ER0 = 0, write data to the desired address and then set EEPGM for a time t_{EPGM}.

There are three possibilities for erasing data from the EEPROM array, depending on how much data is affected.

- To erase a byte of EEPROM, set EELATCH = CPEN = 1, set ER1 = 0 and ER0 = 1, write data to the desired address and then set EEPGM for a time t_{EBYTE}.
- To erase a block of EEPROM, set EELATCH = CPEN = 1, set ER1 = 1 and ER0 = 0, write data to any address in the block and then set EEPGM for a time t_{EBLOCK}.
- To bulk erase the EEPROM, set EELATCH = CPEN = 1, set ER1 = ER0 = 1, write data to any address in the array and then set EEPGM for a time t_{EBULK}.

To terminate the programming or erase sequence, clear EEPGM, wait for a time t_{FPV} to allow the programming voltage to fall, and then clear EELATCH and CPEN to release the buses. Following each erase or programming sequence, clear all programming control bits.

3.5.3 Sample EEPROM programming sequence

The following program is an example of the EEPROM programming sequence, using the timer to implement the required delay and assuming a 1 MHz bus frequency.

```
TCSR        EQU   $0008          TIMER CONTROL AND STATUS REGISTER
TCNT        EQU   $0009          TIMER COUNTER REGISTER
TOF         EQU   7              TOF BIT OF TCSR
PROG        EQU   $001C          EEPROM PROGRAM REGISTER
CPEN        EQU   6              CHARGE PUMP ENABLE BIT
ER1         EQU   4              ERASE SELECT BIT 1
ER0         EQU   3              ERASE SELECT BIT 0
EELATCH     EQU   2              EELATCH BIT
EERC        EQU   1              RC/OSC SELECTOR BIT
EEPGM       EQU   0              EEPROM PROGRAM BIT
EESTART     EQU   $01C0          START ADDRESS OF EEPROM
SUMPIN      EQU   $FF            DUMMY DATA

            ORG   $4000
START       EQU   *
            BSET  EERC, PROG     SELECT RC OSCILLATOR
            BSR   DELAY          RC OSCILLATOR STABILIZATION
            BSET  CPEN, PROG     TURN ON CHARGE PUMP
            BSET  EELATCH, PROG  ENABLE EELATCH BIT
            BCLR  ER1, PROG      SELECT PROGRAM (NOT ERASE)
            BCLR  ER0, PROG      SELECT PROGRAM (NOT ERASE)

            LDA   #SUMPIN        GET DATA
            STA   EESTART
            BSET  EEPGM, PROG    ENABLE PROGRAMMING POWER
            JSR   DELAY          WAIT FOR PROGRAMMING TIME
            BCLR  EEPGM, PROG    CLEAR EEPGM

            JSR   DELAY          WAIT FOR PROG VOLTAGE TO FALL
            BCLR  LATCH, PROG    CLEAR LATCH
            BCLR  CPEN, PROG     DISABLE CHARGE PUMP
            CMP   EESTART        VERIFY
            BNE   OUT1
            CLC                  CLEAR CARRY BIT IF NO ERROR

OUT         RTS

OUT1        SEC                  FLAG AN ERROR
            RTS

*THIS ROUTINE GIVES A 15MS (+/-1MS) DELAY AT 1 MHZ BUS. THE SAME DELAY
* ROUTINE IS USED IN THIS EXAMPLE FOR SIMPLICITY, USING THE LONGEST DELAY
* TIME. USERS WILL WANT TO WRITE SHORTER DELAY ROUTINES FOR APPLICATIONS
*IN WHICH SPEED IS IMPORTANT.

DELAY       EQU   *
            LDX   #15            COUNT OF 15
TIMLP       BCLR  TOF, TCSR      CLEAR TOF
            BRCLR TOF, TCSR      WAIT FOR TOF FLAG
            DECX
            BNE   TIMLP          COUNT DOWN TO 0
            RTS
```

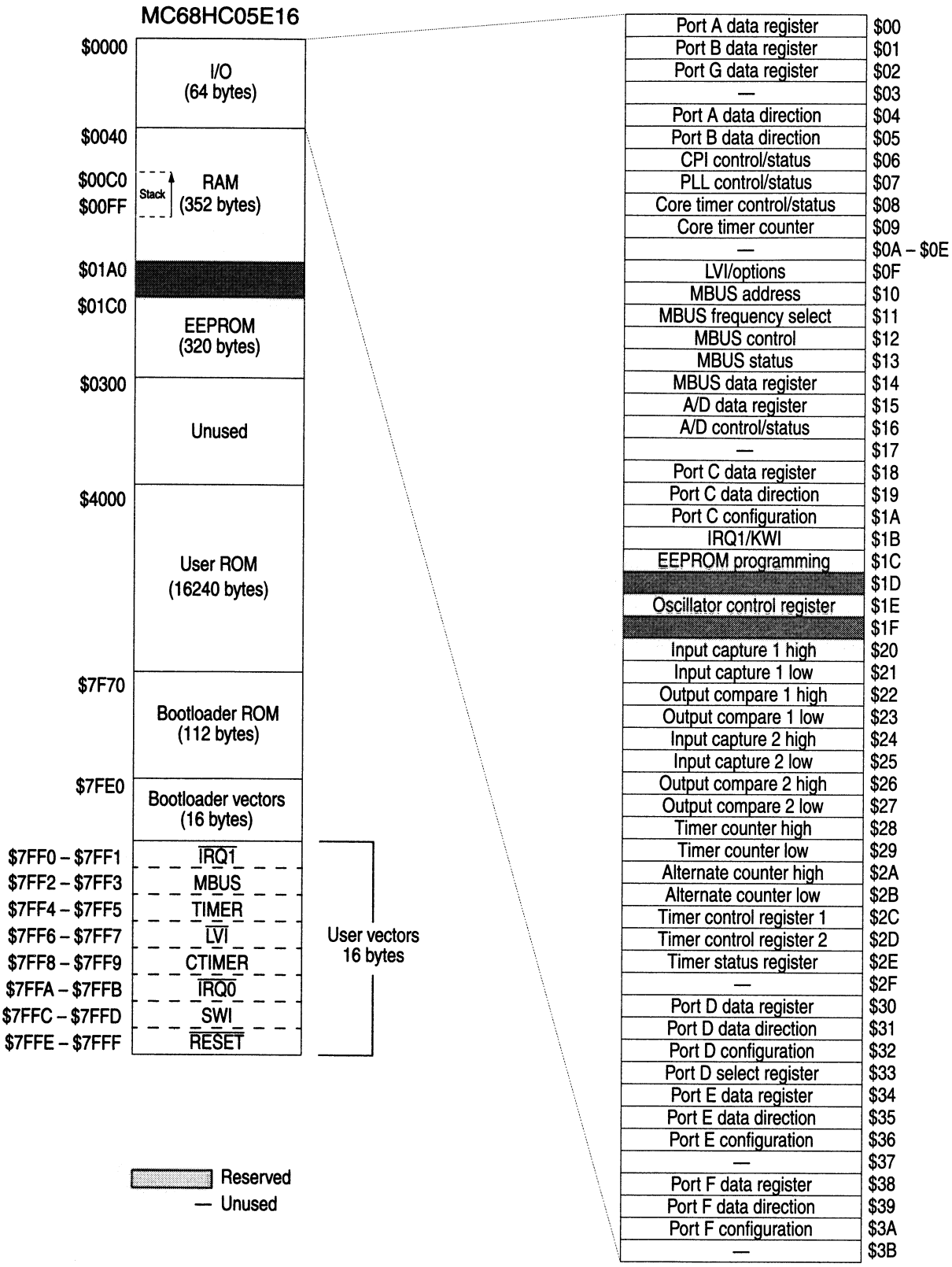


Figure 3-1 Memory map of the MC68HC05E16

Table 3-2 Register outline

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined
Port B data (PORTB)	$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined
Port G data (PORTG)	$0002							PG1	PG0	undefined
Port A data direction (DDRA)	$0004									0000 0000
Port B data direction (DDRB)	$0005									0000 0000
CPI control/status (CPICSR)	$0006	0	CPIF	0	CPIE	0	0	CPI1	CPI0	0000 0000
PLL control/status (PLLCR)	$0007	0	BCS	1	BWC	PLLON	1	PS1	PS0	0010 1100
Core timer control/status (CTCSR)	$0008	CTOF	RTIF	CTOE	RTIE	RCTOF	RRTIF	RT1	RT0	0000 0011
Core timer counter (CTCR)	$0009									0000 0000
LVI/options (LVIOPT)	$000F	LVIINT	LVIVAL	LVIRST	LVIENA	0	0	OPTCOP	OPTIRQ0	0u00 uu00
M-bus address (MADDR)	$0010	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	0	0000 0000
M-bus frequency divider (MFDR)	$0011				MBC4	MBC3	MBC2	MBC1	MBC0	uuu0 0000
M-bus control (MCR)	$0012	MEN	MIEN	MSTA	MTX	TXAK	MMUX	0	0	0000 0000
M-bus status (MSR)	$0013	MCF	MAAS	MBB	MAL	0	SRW	MIF	RXAK	1000 0001
M-bus data register (MDR)	$0014	TRXD7	TRXD6	TRXD5	TRXD4	TRXD3	TRXD2	TRXD1	TRXD0	undefined
A/D data register (ADDATA)	$0015									undefined
A/D control/status (ADSTAT)	$0016	COCO	ADRC	ADON	0	0	CH2	CH1	CH0	0u00 0uuu
Port C data (PORTC)	$0018	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined
Port C data direction (DDRC)	$0019									0000 0000
Port C configuration (CONFC)	$001A									0000 0000
IRQ1/KWI status/control (IRQ1/KWI)	$001B	KSF	KIE	IRQ1INT	IRQ1ENA	IRQ1LV	IRQ1EDG	IRQ1RST	IRQ1VAL	0000 000u
EEPROM programming (EEPROG)	$001C	0	CPEN	0	ER1	ER0	EELATCH	EERC	EEPGM	0000 0000
Oscillator control (OSCREG)	$001E	STOPPLL	STOPQC	STOPRC	SET1	SET0	RUNPLL	RUNQC	RUNRCC	0000 0000
Input capture high 1 (ICH1)	$0020	(bit 15)							(bit 8)	undefined
Input capture low 1 (ICL1)	$0021									undefined
Output compare high 1 (OCH1)	$0022	(bit 15)							(bit 8)	undefined
Output compare low 1 (OCL1)	$0023									undefined
Input capture high 2 (ICH2)	$0024	(bit 15)							(bit 8)	undefined
Input capture low 2 (ICL2)	$0025									undefined

Table 3-2 Register outline (Continued)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare high 2 (OCH2)	$0026	(bit 15)							(bit 8)	undefined
Output compare low 2 (OCL2)	$0027									undefined
Timer counter high (TCH)	$0028	(bit 15)							(bit 8)	$FF
Timer counter low (TCL)	$0029									$FC
Alternate counter high (ACH)	$002A	(bit 15)							(bit 8)	$FF
Alternate counter low (ACL)	$002B									$FC
Timer control 1 (TCR1)	$002C	IC1IE	IC2IE	OC1IE	TOIE	CO1E	IEDG1	IEDG2	OLVL1	0000 0uuu
Timer control 2 (TCR2)	$002D			OC2IE		CO2E			OLVL2	0000 000u
Timer status (TSR)	$002E	IC1F	IC2F	OC1F	TOF	TCAP1	TCAP2	OC2F	0	uuu0 00u0
Port D data (PORTD)	$0030	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	undefined
Port D data direction (DDRD)	$0031									0000 0000
Port D configuration (CONFD)	$0032									0000 0000
Port D select (SELD)	$0033	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	0000 0000
Port E data (PORTE)	$0034	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined
Port E data direction (DDRE)	$0035									0000 0000
Port E configuration (CONFE)	$0036									0000 0000
Port F data (PORTF)	$0038	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined
Port F data direction (DDRF)	$0039									0000 0000
Port F configuration (CONFF)	$003A									0000 0000

u = undefined

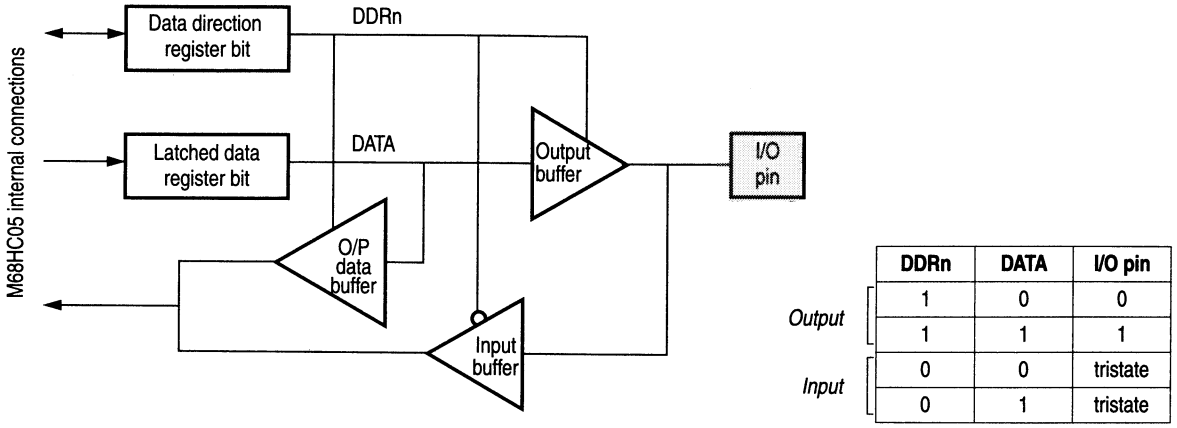
4
INPUT/OUTPUT PORTS

In single-chip mode, the MC68HC05E16 has a total of 49 I/O lines, arranged as four 8-bit I/O ports, one 7-bit I/O port, one 8-bit port which shares various I/O configurations with the programmable timer and M-bus subsystems and one 2-bit input-only port shared with the A/D converter. Each I/O line is individually programmable as either input or output, under the software control of the data direction registers. To avoid glitches on the output pins, data should be written to the I/O port data register before writing ones to the corresponding data direction register bits to set the pins to output mode.

4.1 Input/output programming

The bidirectional port lines may be programmed as inputs or outputs under software control. The direction of each pin is determined by the state of the corresponding bit in the port data direction register (DDR). Each port has an associated DDR. Any I/O port pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared.

At power-on or reset, all DDRs are cleared, thus configuring all port pins as inputs. The data direction registers can be written to or read by the MCU. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin. The operation of the standard port hardware is shown schematically in Figure 4-1.



	DDRn	DATA	I/O pin
Output	1	0	0
	1	1	1
Input	0	0	tristate
	0	1	tristate

Figure 4-1 Standard I/O port structure

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

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 Ports A and B

These ports are standard M68HC05 bidirectional I/O ports, each 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.

There are fixed pull-down resistors on all pins of port A configured as inputs and fixed pull-up resistors on all port B pins configured as inputs.

4.3 Port C

Port C is an 8-bit bidirectional port which is equipped with a keyboard interrupt and does not share any of its pins externally with other subsystems. (It does share with IRQ1 logic internally.) There are four read/write registers associated with the port for selecting the different functions. The register locations are as shown in Table 4-2.

Table 4-2 Port C register addresses

Data register	PORTC	$0018
Data direction register	DDRC	$0019
Configuration register	CONFC	$001A
Keyboard wake-up	IRQ1/KWI	$001B

All registers except the data register are cleared by reset, thereby turning all ports to normal inputs with pull-up resistors. The data direction register and configuration register determine the behaviour of the I/O port according to Table 4-3.

Table 4-3 I/O port configuration functions

DDRC	CONFC	Function
0	0	Input with pull-up
0	1	Input without pull-up
1	0	Push-pull output
1	1	Open-drain output

Every port C pin configured as an input contributes to the wired-OR keyboard interrupt, so a single pin can be disabled by configuring it as an output.

4.3.1 Keyboard interrupt

For the following discussion refer also to Section 10.2 and Section 10.4.8. The keyboard interrupt is enabled by setting the keyboard interrupt enable bit (KIE in the IRQ1/KWI register). These input lines have an internal pull-up. This pull-up can be switched off, setting the corresponding configuration register bit to 1. Once a high to low transition is sensed on any of the lines of PC0 – PC7 configured as input, provided the interrupt mask bit of the condition code register is cleared, a keyboard interrupt is generated and the keyboard status flag (KSF) is set. The interrupt service routine is specified by the contents of the memory locations $7FF0 and $7FF1. The interrupt is cleared by reading the port C data register. The keyboard interrupt is edge-and-level sensitive and will force the processor out of STOP or WAIT mode.

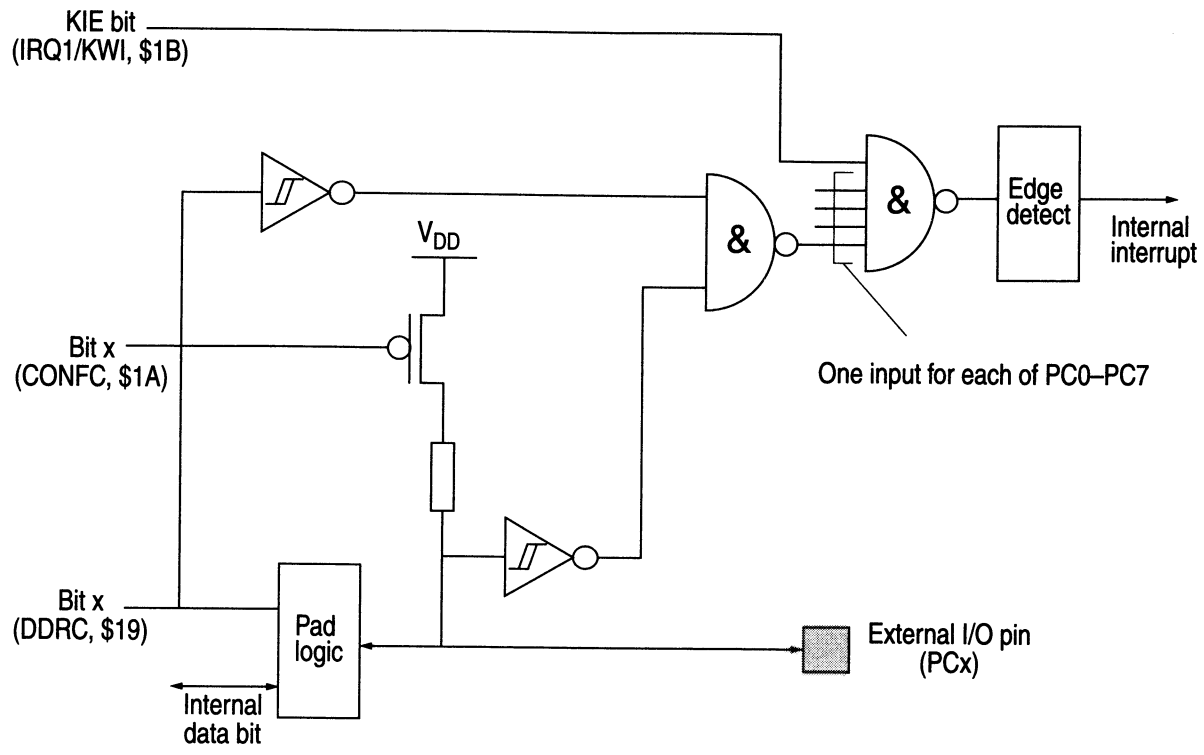


Figure 4-2 Port C keyboard interrupt function

4.3.1.1 IRQ1/KWI status/control register (IRQ1/KWI)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
IRQ1/KWI status/control (IRQ1/KWI)	$001B	KSF	KIE	IRQ1INT	IRQ1ENA	IRQ1LV	IRQ1EDG	IRQ1RST	IRQ1VAL	0000 000u

KSF — KWI interrupt flag

1 (set) – A valid keyboard interrupt has been generated.

0 (clear) – A valid keyboard interrupt has not been generated.

KIE — KWI interrupt enable

1 (set) – Keyboard interrupts are enabled.

0 (clear) – Keyboard interrupts are disabled.

Note: Shaded cells indicate bits that are not described in this section. These bits are described in Section 10.4.8.1.

4.4 Port D

Port D is an 8-bit port which shares its pins with the programmable and M-bus subsystems. There are four read/write registers associated with the port for selecting the different functions. All the ports in port D can be configured as input/output pins or can be used by other systems within the MCU. The register locations are as shown in Table 4-2.

Table 4-4 Port D register addresses

Data register	PORTD	$0030
Data direction register	DDRD	$0031
Configuration register	CONFD	$0032
Function select register	SELD	$0033

All registers except the data register are cleared by reset, thereby turning all ports to normal inputs with pull-up resistors.

The data direction register and the configuration register determine the behaviour of the I/O port according to Table 4-3. Setting bits 7–0 in the port D select register to logical '1' configures the pin to be dedicated to the programmable timer or M-bus subsystems.

Note: This select bit overwrites the corresponding data direction bit.

4.4.1 PD0/TCAP1

This pin is set as PD0 when bit 0 in the port D select register is '0' and becomes timer input capture 1 input when the select bit is set. Setting the configuration bit 0 switches off the pull-up resistor.

4.4.2 PD1/TCMP1

This pin is set as PD1 when bit 1 in the port D select register is '0' and is configured as timer output compare 1 output when the corresponding select bit is set. Setting the configuration bit 0 makes it an open-drain output.

4.4.3 PD2/TCAP2

This pin is set as PD2 when bit 2 in the port D select register is '0' and becomes timer input capture 2 input when the select bit is set. Setting the configuration bit 2 switches off the pull-up resistor.

4.4.4 PD3/TCMP2

This pin is set as PD3 when bit 3 in the port D select register is '0' and is configured as timer output compare 2 output when the corresponding select bit is set. Setting the configuration bit 3 makes it an open-drain output.

4.4.5 PD4/SDA0

This pin is set as PD4 when bit 4 in the port D select register is '0' and is configured as M-bus data pin 0 when the corresponding select bit is set. In this case it is always an open-drain I/O.

4.4.6 PD5/SCL0

This pin is set as PD5 when bit 5 in the port D select register is '0' and is configured as M-bus clock pin 0 when the corresponding select bit is set. In this case it is always an open-drain I/O.

4.4.7 PD6/SDA1

This pin is set as PD6 when bit 6 in the port D select register is '0' and is configured as M-bus data pin 1 when the corresponding select bit is set. In this case it is always an open-drain I/O.

4.4.8 PD7/SCL1

This pin is set as PD7 when bit 7 in the port D select register is '0' and is configured as M-bus clock pin 1 when the corresponding select bit is set. In this case it is always an open-drain I/O.

4.5 Port E

Port E is an 8-bit bidirectional port which does not share any of its pins with other subsystems. There are three read/write registers associated with the port for selecting the different functions. The register locations are shown in Table 4-2.

Table 4-5 Port E register addresses

Data register	PORTE	$0034
Data direction register	DDRE	$0035
Configuration register	CONFE	$0036

All registers except the data register are cleared by reset, thereby turning all ports to normal inputs with pull-up resistors.

The data direction register along with the configuration register determines the behaviour of the I/O port according to Table 4-3.

4.6 Port F

Port F is an 7-bit bidirectional port which does not share any of its pins with other subsystems. There are three read/write registers associated with the port for selecting the different functions. The register locations are shown in Table 4-2.

Table 4-6 Port F register addresses

Data register	PORTF	$0038
Data direction register	DDRF	$0039
Configuration register	CONFF	$003A

All registers except the data register are cleared by reset, thereby turning all ports to normal inputs with pull-up resistors. Unused bits read as 0.

The data direction register along with the configuration register determines the behaviour of the I/O port according to Table 4-3.

4.7 Port G

Port G is an 2-bit fixed input-only port which can be read at any time; it reads the two analog inputs to the A/D converter, when it is enabled.

Port G can still be read during an A/D conversion sequence, but this may inject noise on the analog inputs, resulting in reduced accuracy of the A/D. Furthermore, performing a digital read of port G with levels other than VDD or VSS on the port G pins will result in greater power dissipation during the read cycles.

Since port G is input only there is no data direction register (DDR) associated with it. The data register is located at $0002.

5
16-BIT PROGRAMMABLE TIMER

The programmable timer on the MC68HC05E16 consists of a 16-bit read-only free-running counter, with a fixed divide-by-four prescaler, plus the input capture/output compare circuitry. The timer can be used for many purposes including measuring pulse length of two input signals and generating two output signals. Pulse lengths for both input and output signals can vary from several microseconds to many seconds. The timer is also capable of generating periodic interrupts or indicating passage of an arbitrary multiple of four CPU cycles. A block diagram is shown in Figure 5-1, and timing diagrams are shown in Figure 5-2, Figure 5-3, Figure 5-4 and Figure 5-5.

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.

The 16-bit programmable timer is monitored and controlled by a group of fifteen registers, full details of which are contained in this section.

Note: A problem may arise if an interrupt occurs in the time between the high and low bytes being accessed. To prevent this, the I-bit in the condition code register (CCR) should be set while manipulating both the high and low byte register of a specific timer function, ensuring that an interrupt does not occur.

5.1 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 2 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

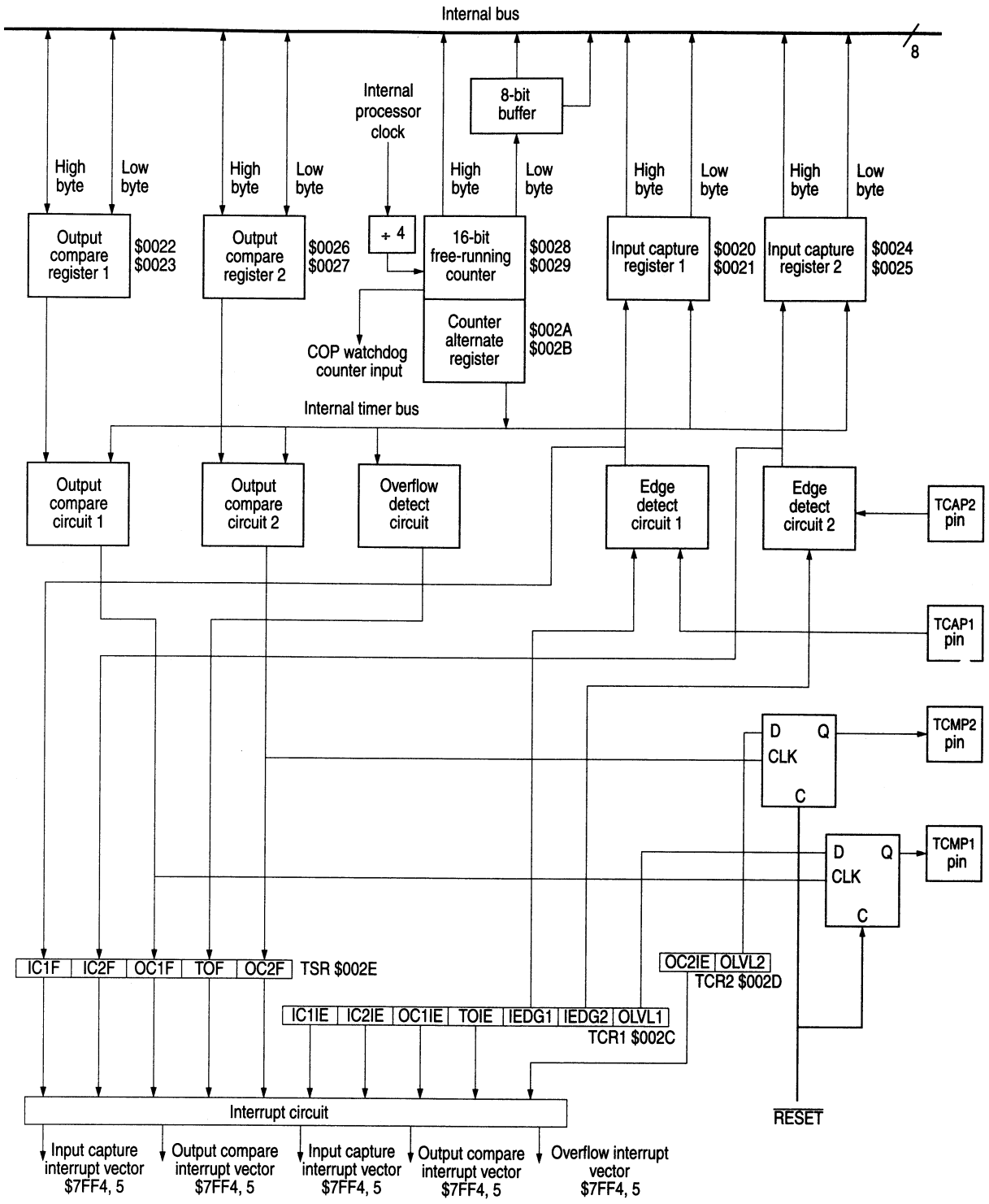


Figure 5-1 16-bit programmable timer block diagram

5.1.1 Counter register and alternate counter register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer counter high	$0028									1111 1111
Timer counter low	$0029									1111 1100

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Alternate counter high	$002A									1111 1111
Alternate counter low	$002B									1111 1100

5

The double-byte, free-running counter can be read from either of two locations, $0028 – $0029 (counter register) or $002A – $002B (alternate counter register). A read from only the less significant byte (LSB) of the free-running counter ($0029 or $002B) 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) ($0028 or $002A), 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 timer status register (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, where it is critical to avoid the possibility of missing timer overflow interrupts due to clearing of TOF, the alternate counter register should be used.

The free-running counter is set to $FFFC during power-on and external reset and is always a read-only register. During a power-on reset, the counter begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-4 prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. TOF is set when the counter overflows (from $FFFF to $0000); this will cause an interrupt if TOIE is set.

The divide-by-4 prescaler is also reset and the counter resumes normal counting operation. All of the flags and enable bits remain unaltered by this operation. If access has previously been made to the high byte of the free-running counter ($0028 or $002A), then the reset counter operation terminates the access sequence.

Caution: This operation may affect the function of the watchdog system (see Section 10.1.3).

5.2 Timer control and status

The various functions of the timer are monitored and controlled using the timer control and status registers described below.

5.2.1 Timer control registers 1 and 2 (TCR1 and TCR2)

The two timer control registers TCR1 and TCR2 ($002C and $002D) are used to enable the input captures (IC1IE and IC2IE), output compares (OC1IE and OC2E), and timer overflow (TOIE) functions as well as enabling the compare outputs (CO1E and CO2E), selecting input edge sensitivity (IEDG1 and IEDG2) and levels of output polarity (OLVL1 and OLVL2).

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 1 (TCR1)	$002C	IC1IE	IC2IE	OC1IE	TOIE	CO1E	IEDG1	IEDG2	OLVL1	0000 0000

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 2 (TCR2)	$002D	0	0	OC2IE	0	CO2E	0	0	OLVL2	0000 0000

IC1IE — Input capture 1 interrupt enable

If this bit is set, a timer interrupt is enabled whenever the IC1F status flag (in the timer status register) is set.

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

IC2IE — Input capture 2 interrupt enable

If this bit is set, a timer interrupt is enabled whenever the IC2F status flag (in the timer status register) is set.

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

OC1IE — Output compare 1 interrupt enable

If this bit is set, a timer interrupt is enabled whenever the OC1F status flag (in the timer status register) is set.

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

TOIE — Timer overflow interrupt enable

If this bit is set, a timer interrupt is enabled whenever the TOF status flag (in the timer status register) is set.

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

CO1E — Timer compare 1 output enable

If this bit is set, the output from timer output compare 1 is enabled.

1 (set) – Output compare 1 enabled.

0 (clear) – Output compare 1 disabled.

IEDG1 — Input edge 1

When IEDG1 is set, a positive-going edge on the TCAP1 pin will trigger a transfer of the free-running counter value to the input capture register 1. When clear, a negative-going edge triggers the transfer.

1 (set) – TCAP1 is positive-going edge sensitive.

0 (clear) – TCAP1 is negative-going edge sensitive.

IEDG2 — Input edge 2

When IEDG2 is set, a positive-going edge on the TCAP2 pin will trigger a transfer of the free-running counter value to the input capture register 2. When clear, a negative-going edge triggers the transfer.

1 (set) – TCAP2 is positive-going edge sensitive.

0 (clear) – TCAP2 is negative-going edge sensitive.

OLVL1 — Output level 1

When OLV1 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 TCMP1 pin. When clear, it will be a low level which will appear on the TCMP1 pin.

1 (set) – A high output level will appear on the TCMP1 pin.

0 (clear) – A low output level will appear on the TCMP1 pin.

OC2IE — Output compare 2 interrupt enable

If this bit is set, a timer interrupt is enabled whenever the OC2F status flag (in the timer status register) is set.

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

CO2E — Timer compare 2 output enable

If this bit is set, the output from timer output compare 2 is enabled.

1 (set) – Output compare 2 enabled.

0 (clear) – Output compare 2 disabled.

OLVL2 — Output level 2

When OLV2 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 TCMP2 pin. When clear, it will be a low level which will appear on the TCMP2 pin.

1 (set) – A high output level will appear on the TCMP2 pin.

0 (clear) – A low output level will appear on the TCMP2 pin.

5.2.2 Timer status register (TSR)

The timer status register ($002E) contains the status bits corresponding to the timer interrupt conditions – IC1F, IC2F, OC1F, TOF, TCAP1, TCAP2 and OC2F.

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)	$002E	IC1F	IC2F	OC1F	TOF	TCAP1	TCAP2	OC2F	0	Undefined

IC1F — Input capture 1 flag

This bit is set when the selected polarity of edge is detected by the input capture edge detector 1 at TCAP1; an input capture interrupt will be generated, if IC1IE is set. IC1F is cleared by reading the TSR and then the input capture 1 low register ($0021).

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

0 (clear) – No input capture has occurred.

IC2F — Input capture 2 flag

This bit is set when a negative edge is detected by the input capture edge detector 2 at TCAP2; an input capture interrupt will be generated if IC2IE is set. IC2F is cleared by reading the TSR and then the input capture 2 low register ($0025).

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

0 (clear) – No input capture has occurred.

OC1F — Output compare 1 flag

This bit is set when the output compare register 1 contents match those of the free-running counter; an output compare interrupt will be generated if OC1IE is set. OC1F is cleared by reading the TSR and then the output compare 1 low register ($0023).

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

0 (clear) – No output compare has occurred.

TOF — Timer overflow status flag

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 ($0029).

1 (set) – Timer overflow has occurred.

0 (clear) – No timer overflow has occurred.

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.

TCAP1 — Timer capture 1 status flag

This bit reflects the status of the timer capture 1 input.

TCAP2 — Timer capture 2 status flag

This bit reflects the status of the timer capture 2 input.

OC2F — Output compare 2 flag

This bit is set when the output compare register 2 contents match those of the free-running counter; an output compare interrupt will be generated if OC2IE is set. OC2F is cleared by reading the TSR and then the output compare 2 low register ($0027).

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

0 (clear) – No output compare has occurred.

5.3 Input capture

'Input capture' is a technique whereby an external signal 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.

There are two input capture registers: input capture register 1 (ICR1) and input capture register 2 (ICR2).

There are two input capture interrupt enable bits (IC1IE and IC2IE).

5.3.1 Input capture register 1 (ICR1)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Input capture 1 high	$0020									Undefined
Input capture 1 low	$0021									Undefined

The two 8-bit registers that make up the 16-bit input capture register 1 are read-only, and are used to latch the value of the free-running counter after the input capture edge detector circuit 1 senses a valid transition at TCAP1. The level transition that triggers the counter transfer is defined by the input edge bit (IEDG1). When an input capture 1 occurs, the corresponding flag IC1F in TSR is set. An interrupt can also accompany an input capture 1 provided the IC1IE bit in TCR1 is set. The 8 most significant bits are stored in the input capture register 1 high at $0020, the 8 least significant bits in the input capture register 1 low at $0021.

The result obtained from an input capture will be one greater than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles. The free-running counter contents are transferred to the input capture register 1 on each valid signal transition whether the input capture 1 flag (IC1F) is set or clear. The input capture register 1 always contains the free-running counter value that corresponds to the most recent input capture 1. After a read of the input capture register 1 MSB ($0020), the counter transfer is inhibited until the LSB ($0021) 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 1 LSB ($0021) does not inhibit the free-running counter transfer since the two actions occur on opposite edges of the internal bus clock.

Reset does not affect the contents of the input capture register 1, except when exiting STOP mode (see Section 5.5).

5.3.2 Input capture register 2 (ICR2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Input capture 2 high	$0024									Undefined
Input capture 2 low	$0025									Undefined

The two 8-bit registers that make up the 16-bit input capture register 2 are read-only, and are used to latch the value of the free-running counter after the input capture edge detector circuit 2 senses a negative transition at pin TCAP2. When an input capture 2 occurs, the corresponding flag IC2F in TSR is set. An interrupt can also accompany an input capture 2 provided the IC2IE bit in TCR1 is set.The 8 most significant bits are stored in the input capture 2 high register at $0024, the 8 least significant bits in the input capture 2 low register at $0025.

The result obtained from an input capture will be one greater than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles. The free-running counter contents are transferred to the input capture register 2 on each valid signal transition whether the input capture 2 flag (IC2F) is set or clear. The input capture register 2 always contains the free-running counter value that corresponds to the most recent input capture 2. After a read of the input capture register 2 MSB ($0024), the counter transfer is inhibited until the LSB ($0025) 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 2 LSB ($0024) does not inhibit the free-running counter transfer since the two actions occur on opposite edges of the internal bus clock.

Reset does not affect the contents of the input capture register 2, except when exiting STOP mode (see Section 5.5).

5.4 Output compare

'Output compare' is a technique which 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.

There are two output compare registers: output compare register 1 (OCR1) and output compare register 2 (OCR2).

There are two output compare interrupt enable bits (OC1IE and OC2IE).

5.4.1 Output compare register 1 (OCR1)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 1 high	$0022									Undefined
Output compare 1 low	$0023									Undefined

The 16-bit output compare register 1 is made up of two 8-bit registers at locations $0022 (MSB) and $0023 (LSB). The contents of the output compare register 1 are compared with the contents of the free-running counter continually and, if a match is found, the corresponding output compare flag (OC1F) in the timer status register is set and the output level (OLVL1) is transferred to pin TCMP1. The output compare register 1 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 (OC1IE) is set. (The free-running counter is updated every four internal bus clock cycles.)

After a processor write cycle to the output compare register 1 containing the MSB ($0022), the output compare function is inhibited until the LSB ($0023) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($0023) will not inhibit the compare 1 function. The processor can write to either byte of the output compare register 1 without affecting the other byte. The output level (OLVL1) bit is clocked to the output level register and hence to the TCMP1 pin whether the output compare flag 1 (OC1F) is set or clear. The minimum time required to update the output compare register 1 is a function of the program rather than the internal hardware. Because the output compare flag 1 and the output compare register 1 are not defined at power on, and not affected by reset, care must be taken when initializing output compare functions with software. The following procedure is recommended:

- Write to output compare 1 high to inhibit further compares;
- Read the timer status register to clear OC1F (if set);
- Write to output compare 1 low to enable the output compare 1 function.

The purpose of this procedure is to prevent the OC1F bit from being set between the time it is read and the write to the corresponding output compare register.

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.

5.4.2 Output compare register 2 (OCR2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 2 high	$0026									Undefined
Output compare 2 low	$0027									Undefined

The 16-bit output compare register 2 is made up of two 8-bit registers at locations $0026 (MSB) and $0027 (LSB). The contents of the output compare register 2 are compared with the contents of the free-running counter continually and, if a match is found, the corresponding output compare flag (OC2F) in the timer status register is set and the output level (OLVL2) is transferred to pin TCMP2. The output compare register 2 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 (OC2IE) is set. (The free-running counter is updated every four internal bus clock cycles.)

After a processor write cycle to the output compare register 2 containing the MSB ($0026), the output compare function is inhibited until the LSB ($0027) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($0027) will not inhibit the compare 2 function. The processor can write to either byte of the output compare register 2 without affecting the other byte. The output level (OLVL2) bit is clocked to the output level register and hence to the TCMP2 pin whether the output compare 2 flag (OC2F) is set or clear. The minimum time required to update the output compare register 2 is a function of the program rather than the internal hardware. Because the output compare 2 flag and the output compare register 2 are not defined at power on, and not affected by reset, care must be taken when initializing output compare functions with software. The following procedure is recommended:

- Write to output compare 2 high to inhibit further compares;
- Read the timer status register to clear OC2F (if set);
- Write to output compare 2 low to enable the output compare 2 function.

The purpose of this procedure is to prevent the OC2F bit from being set between the time it is read and the write to the corresponding output compare register.

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.

5.5 Timer during STOP mode

When the MCU enters STOP mode, the timer counter stops counting and remains at that particular count value until STOP mode is exited by an interrupt. If STOP mode is exited by power-on or external reset, the counter is forced to $FFFC but if it is exited by external interrupt ($\overline{IRQ1}$ or $\overline{IRQ0}$) then the counter resumes from its stopped value.

Another feature of the programmable timer is that if at least one valid input capture edge occurs at one of the TCAP pins while in STOP mode, the corresponding input capture detect circuitry is armed. This action does not wake the MCU or set any timer flags, but when the MCU does wake-up there will be an active input capture flag (and data) from that first valid edge which occurred during STOP mode.

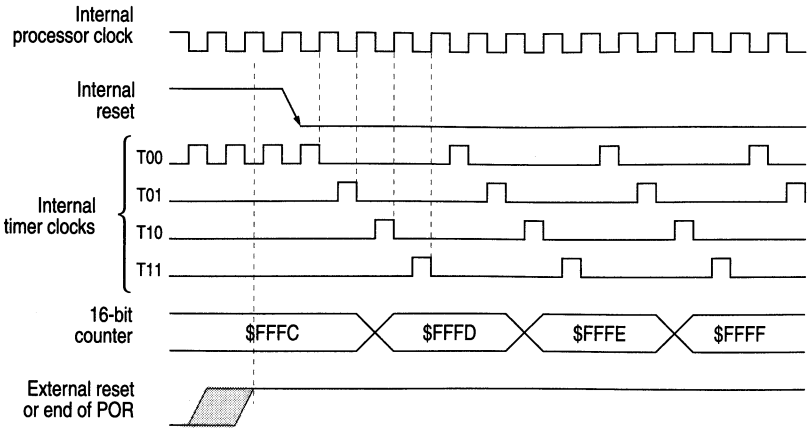
If STOP mode is exited by an external reset then no such input capture flag or data action takes place even if there was a valid input capture edge (at one of the TCAP pins) during STOP mode.

5.6 Timer during WAIT mode

The timer system is not affected by WAIT mode and continues normal operation. Any valid timer interrupt will wake-up the system.

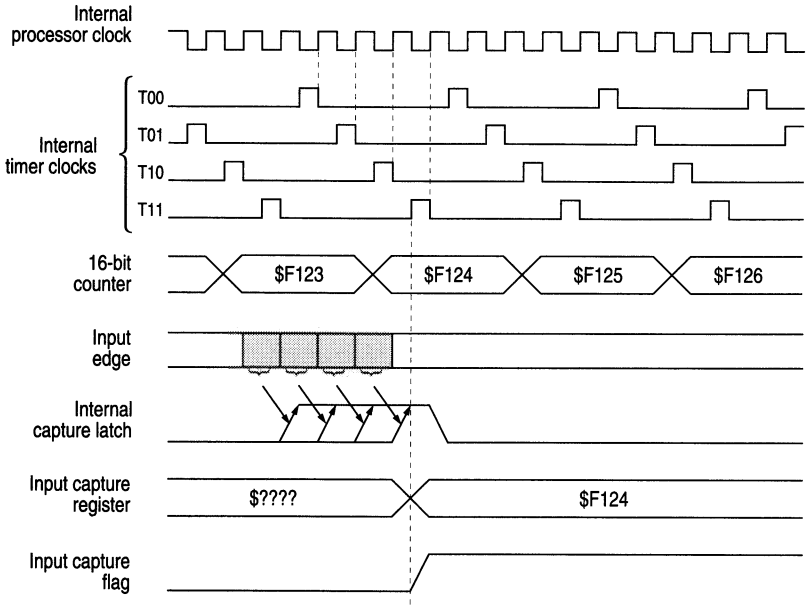
5.7 Timer state diagrams

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



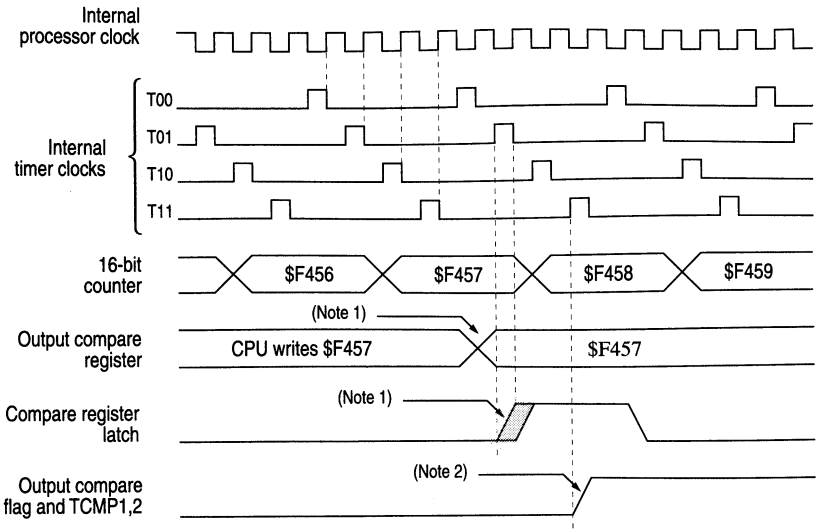
Note: The counter and timer control registers are the only ones affected by power-on or external reset.

Figure 5-2 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.

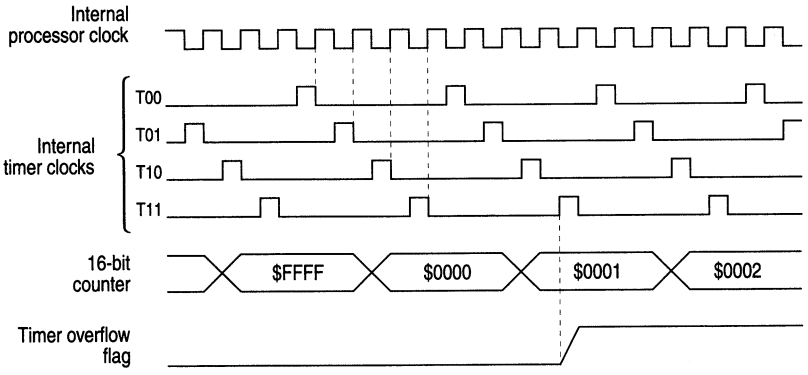
Figure 5-3 Timer state timing diagram for input capture



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

2 The output compare flag is set at the timer state T11 that follows the comparison match ($F457 in this example).

Figure 5-4 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 5-5 Timer state timing diagram for timer overflow

6
I²C-BUS

I²C-bus is a two-wire, bidirectional serial bus that provides a simple, efficient way to exchange data between devices. Being a two-wire device, the I²C-bus minimizes the need for large numbers of connections between devices, and eliminates the need for an address decoder.

The bus is suitable for applications involving frequent communications between a number of devices over short distances. The number of devices connected to the I²C-bus is limited only by a maximum bus capacitance of 400pF; it has a maximum data rate of 100 kbits per second.

The I²C-bus system is a true multi-master bus including collision detection and arbitration to prevent data corruption if two or more masters attempt to control the bus simultaneously. This feature provides the capability for complex applications with multiprocessor control. It may also be used for rapid testing and alignment of end products via external connections to an assembly line computer.

The I²C-bus function is enabled by the MEN bit in the I²C-bus control register (MCR).

6.1 I²C-bus features

- Multi-master operation
- Software-programmable for one of 32 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven, byte-by-byte data transfer
- Arbitration-lost-driven interrupt with automatic switching from master to slave mode
- Calling address identification interrupt
- Generates/detects the START or STOP signal
- Repeated START signal generation
- Generates/recognizes the acknowledge bit
- Bus busy detection

6.2 I²C-bus system configuration

The I²C-bus system uses a serial data line and a serial clock line for data transfer. All the devices connected to it must have open drain or open collector outputs. A logic 'AND' function is used on both lines with two pull-up resistors.

6.3 I²C-bus protocol

A standard communication is normally composed of four parts: START signal, slave address transmission, data transfer, and STOP signal. These signals are described in the following sections and illustrated in Figure 6-1.

6.3.1 START signal

When the bus is free (no master device engaging the bus; SCL and SDA lines at a logic high), a master may initiate communication by sending a START signal, which is defined as being a high to low transition of SDA with SCL high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and wakes up all slaves.

6.3.2 Transmission of the slave address

The first byte of data transferred after the START signal is the slave address transmitted by the master. This address is seven bits long, followed by a R/$\overline{W}$ bit which tells the slave the desired direction of transfer of all the following bytes (until a STOP or repeated start).

Only the slave with the calling address that matches the one transmitted by the master responds by sending back an acknowledge bit. This is done by pulling the SDA low at the ninth clock (see Figure 6-1).

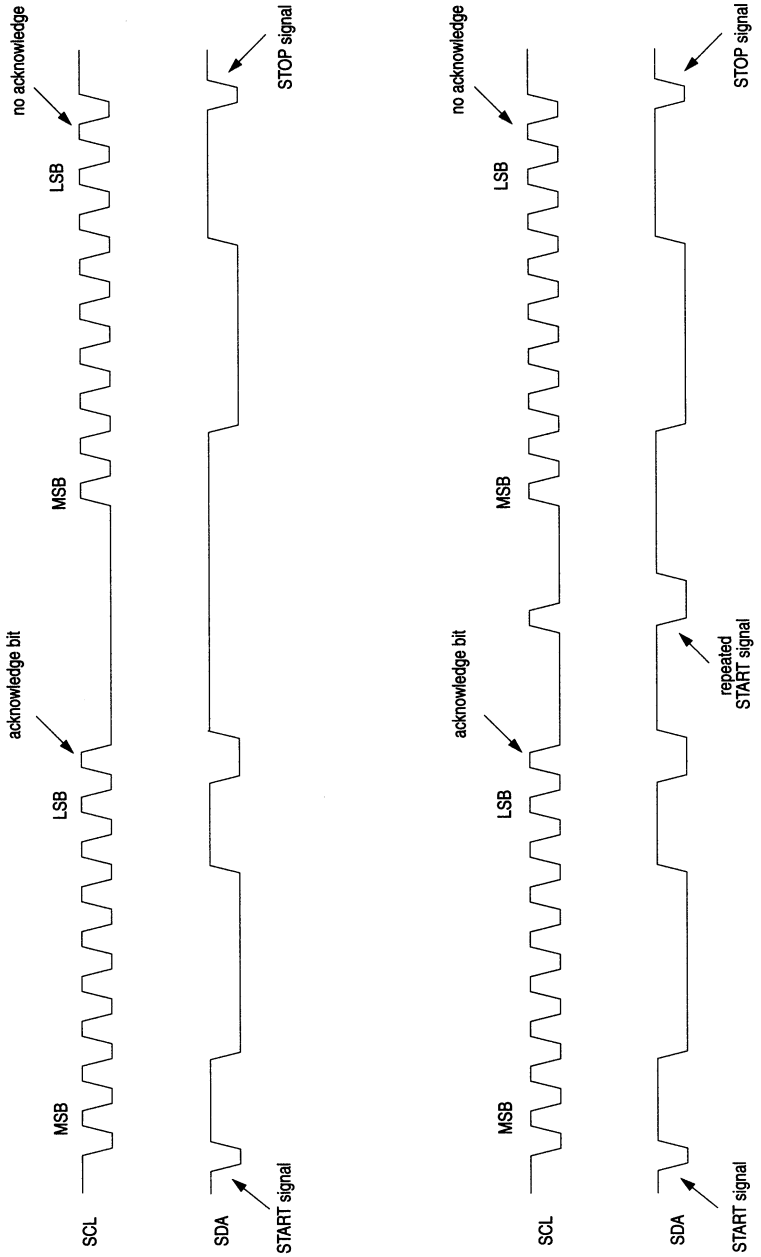


Figure 6-1 I^2C bus transmission signal diagrams

6.3.3 Data transfer

Once successful slave addressing has been achieved, the data transfer can proceed byte by byte, in the direction specified by the R/$\overline{W}$ bit.

Data can be changed only when SCL is low and must be held stable while SCL is high. The MSB is transmitted first. Each data byte is eight bits long, and there is one clock pulse on SCL for each data bit. Every byte of data must be followed by an acknowledge bit, which the receiving device signals by pulling SDA low at the ninth clock. Therefore, one complete data byte transfer needs nine clock pulses.

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

If the master receiver does not acknowledge the slave transmitter after one byte of transmission, it means 'end of data' to the slave, which then releases the SDA line so that the master can generate the STOP or START signal.

6.3.4 STOP signal

The master can terminate the communication by generating a STOP signal to free the bus. A STOP signal is defined as a low to high transition of SDA while SCL is high (see Figure 6-1).

6.3.5 Repeated START signal

A repeated START signal generates a START signal without first generating a STOP signal to terminate the communication. This is used by the master to communicate with another slave, or with the same slave in a different mode (transmit/receive mode), without releasing the bus.

6.3.6 Arbitration procedure

The I^2C-bus is a true multi-master system that allows more than one master to be connected to it. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock, for which the low period is equal to the longest clock low period and the high period is equal to the shortest clock high period among the masters. A data arbitration procedure determines the relative priority of the contending masters; a master loses arbitration if it transmits logic 1 while another transmits logic 0. The losing masters then immediately switch to slave receive mode and stop driving SDA outputs. The transition from master to slave mode does not generate a STOP condition in this case. At this point, the MAL bit in the I^2C-bus status register (MSR) is set by hardware to indicate loss of arbitration.

6.3.7 Clock synchronization

Since wired-AND logic is performed on the SCL line, a high to low transition on SCL affects all the devices connected on the bus. The devices start counting their low period and once a device's clock has gone low, it holds the SCL line low until the clock high state is reached. However, the change of low to high in this device clock may not change the state of the SCL line if another device clock is still within its low period. Therefore, synchronized clock SCL is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time (see Figure 6-2). When all devices concerned have counted off their low period, the SCL line is released and pulled high. There is then no difference between the device clocks and the state of the SCL line, and all of them start counting their high periods. The first device to complete its high period pulls the SCL line low again.

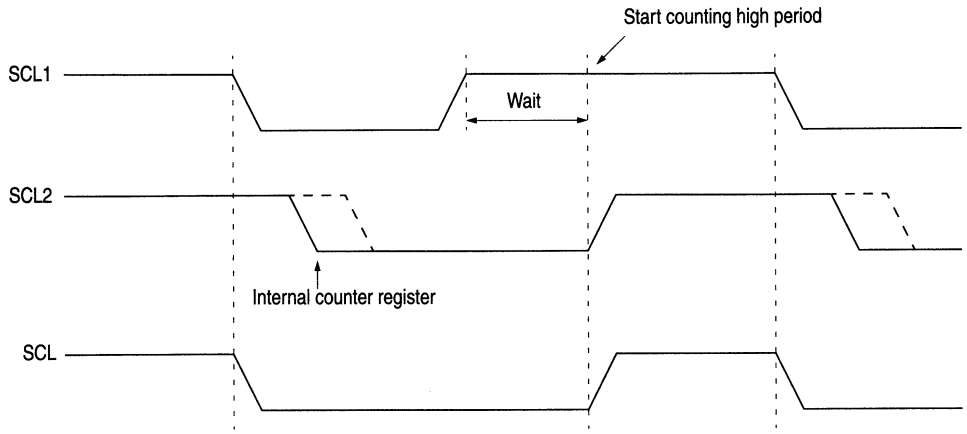


Figure 6-2 Clock synchronization

6.3.8 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfer. The slave device may hold SCL low after the completion of one byte of data transfer (nine bits). In such cases, it halts the bus clock and forces the master clock into a wait state until the slave releases the SCL line.

6.4 Registers

6.4.1 M-bus address register (MADR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
M-bus address (MADR)	$0010	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	0	0000 0000

ADR7 – ADR1 — Slave address bits

These bits define the slave address of the M-bus, and are used in slave mode in conjunction with the MAAS bit in the MSR register (see Section 6.4.4). These bits can be read and written at any time.

Bit 0 – not implemented; always reads zero.

6.4.2 I²C-bus frequency divider register (FDR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
I²C-bus frequency divider register (FDR)	$0011				MBC4	MBC3	MBC2	MBC1	MBC0	uuu0 0000

MBC4 – MBC0 — Clock rate select bits

These bits can be read and written at any time.

These bits are used to prescale the clock for bit rate selection. Due to the potential slow rise and fall times of the SCL and SDA signals the bus signals are sampled at the prescaler frequency. This sampling incurs an overhead of six clocks per SCL pulse. The serial bit clock frequency is equal to the CPU clock divided by the divider shown in Table 6-1, plus the sampling overhead of six clocks per cycle.

For a 4 MHz external crystal operation, the serial bit clock frequency of the I²C-bus ranges from 460 Hz to 90909 kHz.

Table 6-1 I²C-bus prescaler

MCB4-0	Divider	MCB4-0	Divider	MCB4-0	Divider	MCB4-0	Divider
00000	22	01000	88	10000	352	11000	1408
00001	24	01001	96	10001	384	11001	1536
00010	28	01010	112	10010	448	11010	1792
00011	34	01011	136	10011	544	11011	2176
00100	44	01100	176	10100	704	11100	2816
00101	48	01101	192	10101	768	11101	3072
00110	56	01110	224	10110	896	11110	3584
00111	68	01111	272	10111	1088	11111	4352

6.4.3 I²C-bus control register (MCR)

These bits can be read and written at any time.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
I²C-bus control register (MCR)	$0012	MEN	MIEN	MSTA	MTX	TXAK				0000 0uuu

MEN — I²C-bus enable

1 (set) – I²C-bus interface system is enabled. This bit must be set before any other MCR bits can be set.

0 (clear) – I²C-bus interface system is disabled and reset. This is the power-on reset case. When low, the interface is held in reset, but registers can be accessed.

If the module is enabled in the middle of a byte transfer, the interface behaves as follows:

slave mode ignores the current transfer on the bus and starts operating when a subsequent start condition is detected.

Master mode is not aware that the bus is busy, so if a start cycle is initiated the current bus cycle may become corrupt. This results in either the current bus master or the I²C-bus losing arbitration, after which bus operation returns to normal.

MIEN — I²C-bus interrupt enable

1 (set) – I²C-bus interrupt is requested when MIF is set.

0 (clear) – I²C-bus interrupt is disabled.

MSTA — Master/slave mode select

1 (set) – Master mode; send START signal when set.

0 (clear) – Slave mode; send STOP signal when cleared.

This bit is cleared on reset. When MSTA is changed from 0 to a 1, a START signal is generated on the bus and the master mode is selected. When this bit changes from a 1 to a 0, a STOP signal is generated and the slave mode is selected. In master mode, clearing MSTA and then immediately setting it generates a repeated START signal without generating a STOP signal (see Figure 6-1).

MTX — Transmit/receive mode select

1 (set) – Transmit mode.

0 (clear) – Receive mode.

TXAK — Transmit acknowledge bit

1 (set) – No acknowledge signal response.

0 (clear) – An acknowledge signal will be sent to the bus at the ninth clock bit after receiving one byte of data.

This bit only has meaning in master receive mode.

Bits 2–0 — not implemented; always read zero.

6.4.4 I²C-bus status register (MSR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
I²C-bus status register (MSR)	$0013	MCF	MAAS	MBB	MAL		SRW	MIF	RXAK	1000 u001

Bits in this register can be read at any time; Bits 4 and 1 can be cleared at any time.

MCF — Data transferring

1 (set) – Data transmit complete.

0 (clear) – Data is being transferred.

MAAS — I²C-bus addressed as a slave

1 (set) – I²C-bus is addressed as a slave.

0 (clear) – I²C-bus is not addressed.

This bit is set when the address of the I²C-bus (specified in MADR) matches the calling address. An interrupt is generated providing the MIEN bit in the MCR register is set; the CPU then selects its transmit/receive mode according to the state of the SRW bit. Writing to the MCR register clears this bit.

MBB — Bus busy

1 (set) – Bus is busy.

0 (clear) – Bus is idle.

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

MAL — Arbitration lost

1 (set) – Arbitration lost.

0 (clear) – Default state.

MAL is set by hardware when the arbitration procedure is lost during a master transmission mode. This bit must be cleared by software.

Bit 3 — Not implemented; always reads zero.

SRW — Read/write command

1 (set) – R/W command bit is set (read).

0 (clear) – R/W command bit is clear (write).

When MAAS is set, the R/W command bit of the calling address sent from a master is latched into this bit. On checking this bit, the CPU can select slave transmit/receive mode according to the command of the master.

MIF — I^2C-bus interrupt flag

1 (set) – An I^2C-bus interrupt is pending.

0 (clear) – No I^2C-bus interrupt is pending.

When this bit is set, an I^2C-bus interrupt is generated provided the MIEN bit in the MCR register is set. MIF is set when one of the following events occurs:

1) The transfer of one byte of data is complete; MIF is set at the falling edge of the ninth clock after the byte has been received.
2) A calling address is received which matches the address of the I^2C-bus in slave receive mode.
3) Arbitration is lost.

MIF must be cleared by software in the interrupt routine.

RXAK — Received acknowledge bit

1 (set) – No acknowledge signal has been detected at the ninth clock after the transmission of a byte of data.

0 (clear) – An acknowledge bit has been received at the ninth clock after the transmission of a byte of data.

6.4.5 I²C-bus data register (MDR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
I²C-bus data register (MDR)	$0014	TRXD7	TRXD6	TRXD5	TRXD4	TRXD3	TRXD2	TRXD1	TRXD0	Undefined

These bits can be read and written at any time.

In master transmit mode, a write to this register will cause the data in it to be sent to the bus automatically, MSB first. In master receive mode, a read of this register initiates the transfer of the next incoming byte of data into the register. See Figure 6-3.

In slave transmit mode, the SCL line is forced low until data is written into this register, to prevent transmission. Similarly, in slave receive mode, the data bus must be read before a transmission can occur.

6

6.5 Programming

6.5.1 Initialization

After a reset, the I²C-bus control register (MCR) is in a default state. Before the I²C-bus can be used, it must be initialized as follows:

1) Configure the frequency divider register for the desired SCL frequency.
2) Configure the I²C-bus address register (MADR) to define the slave address of the I²C-bus.
3) Set the MEN bit in the I²C-bus control register (MCR) to enable the I²C-bus system.
4) Configure the other bits in the MCR register.

6.5.2 START signal and the first byte of data

After the initialization procedure has been completed, serial data can be transmitted by selecting the 'master transmitter' mode. If the device is connected to a multi-master bus system, the state of the I²C-bus busy bit (MBB) must be tested to check whether the serial bus is free. If the bus is free (MBB = 0), the START condition and the first byte (the slave address) can be sent. An example of a program that does this is shown below:

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

6.5.3 Software response

The transmission or reception of a byte sets the data transferring bit, MCF, which indicates that one byte of communication is finished. Also, the I²C-bus interrupt bit, MIF, is set to generate an I²C-bus interrupt (if MIEN is set). Figure 6-3 shows an example of a typical I²C-bus interrupt routine. In the interrupt routine, the first step is for software to clear the MIF bit. The MCF bit can be cleared by reading from the I²C-bus data I/O register (MDR) in receive mode, or by writing to MDR in transmit mode. Software may service the I²C-bus I/O in the main program by monitoring the MIF bit if the interrupt function is disabled. The following is an example of a software response by a 'master transmitter' in the interrupt routine:

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

6.5.4 Generation of a STOP signal

A data transfer ends with a STOP signal generated by the master device. A master transmitter can simply generate a STOP signal after all the data has been transmitted; for example:

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

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

```
MASR    DEC     RXCNT
        BEQ     ENMASR      ;LAST BYTE TO BE READ
        LDA     RXCNT
        DECA                ;CHECK SECOND LAST BYTE TO BE
                            ;READ
        BNE     NXMAR       ;NOT LAST ONE OR SECOND LAST
LAMAR   BSET    3,MCR       ;SECOND LAST, DISABLE
                            ;ACKNOWLEDGEMENT TRANSMITTING
        BRA     NXMAR       ;NXMAR
ENMASR  BCLR    5,MCR       ;LAST  ONE, GENERATE  STOP
 SIGNAL
NXMAR   LDA     MDR         ;READ DATA AND STORE
        STA     RXBUF
        RTI
```

6.5.5 Generation of a repeated START signal

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

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

6.5.6 Slave mode

In the slave interrupt service routine, the MAAS bit should be tested to check if a calling of its own address has just been received. If MAAS is set, software should set the transmit/receive mode select bit (MTX) according to the R/W command bit, SRW. Writing to the MCR clears the MAAS bit automatically. A data transfer may then be initiated by writing to MDR or by performing a dummy read from MDR.

In the slave transmitter routine, the received acknowledge bit (RXAK) must be tested before transmitting the next byte of data. If RXAK is set, this means an 'end of data' signal from the master receiver, which must then switch from transmitter mode to receiver mode by software. This is followed by a dummy read, which releases the SCL line so that the master can generate a STOP signal.

6.5.7 Arbitration lost

Only one master can engage the device at one time. Those devices wishing to engage the bus, but having lost arbitration, are immediately switched to slave receive mode by hardware. Their data output to the SDA line is stopped, but the internal transmitting clock is still generated until the end of the byte during which arbitration was lost. An interrupt occurs at the falling edge of the ninth clock of this transfer with MAL = 1 and MSTA = 0. If one master attempts to start transmission while the bus is being engaged by another master, the hardware inhibits the transmission; the MSTA bit is cleared without generating a STOP condition, an interrupt is generated, and MAL is set to indicate that the attempt to engage the bus has failed. In these cases, the slave interrupt service routine should test MAL first; if MAL is set, it should be cleared by software.

6.5.8 Operation during STOP and WAIT modes

During STOP mode, the I^2C-bus is disabled.

During WAIT mode, the I^2C-bus is idle, but 'wakes up' when it receives a valid start condition in slave mode. If the interrupt is enabled, the CPU comes out of WAIT mode after the end of a byte of transmission.

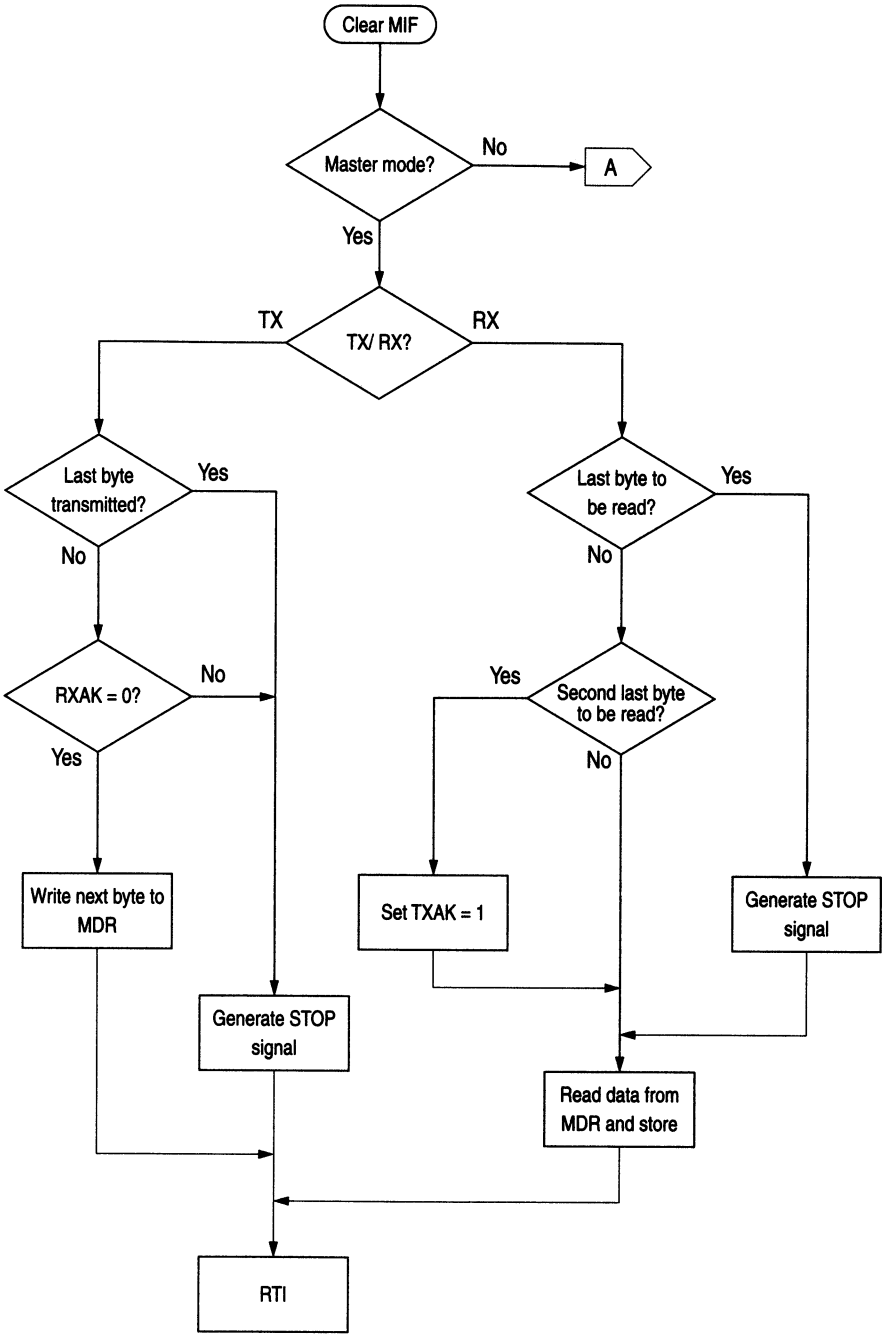


Figure 6-3 Example of a typical I²C-bus interrupt routine

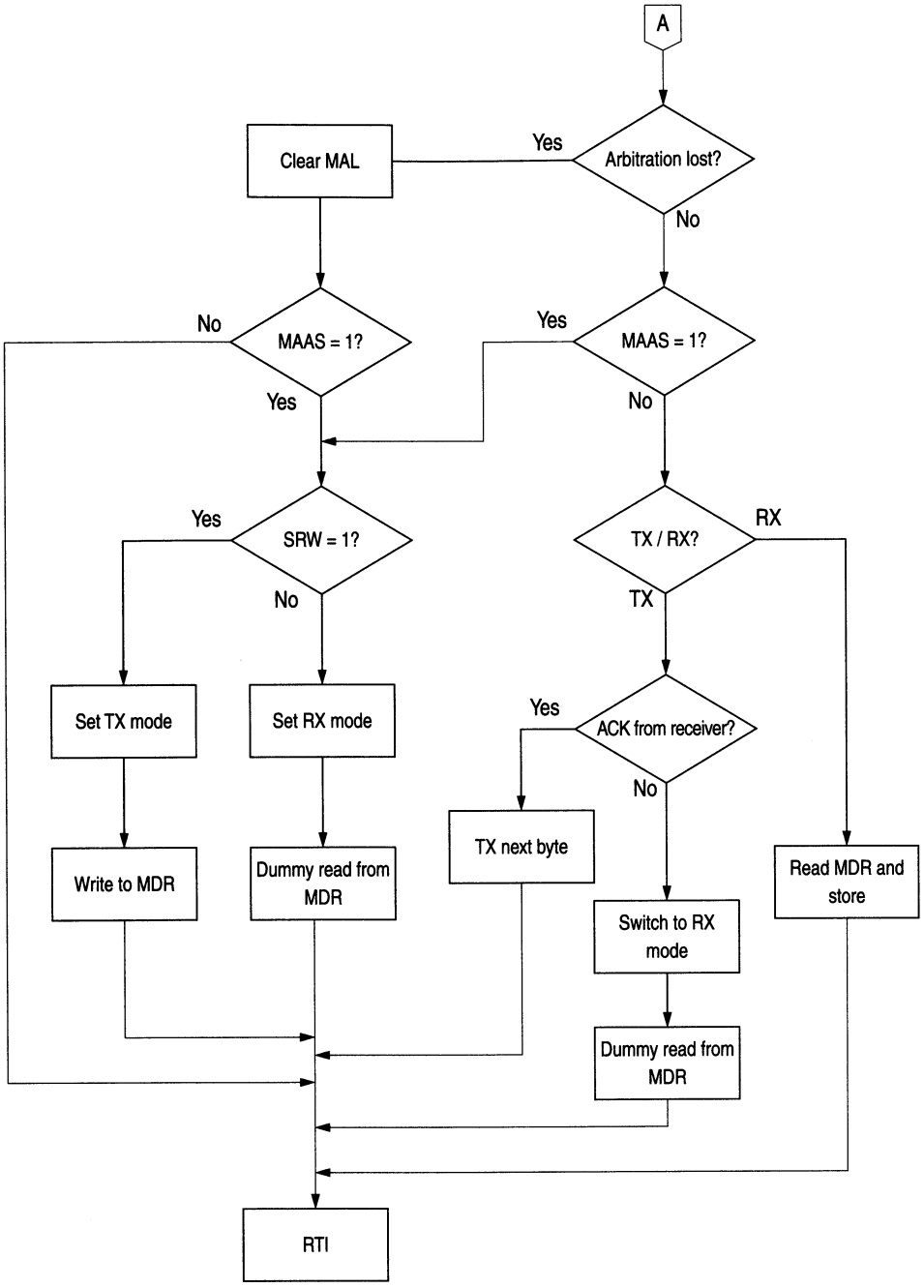


Figure 6-3 Example of a typical I²C-bus interrupt routine (Continued)

THIS PAGE LEFT BLANK INTENTIONALLY

7
A/D CONVERTER

The analog to digital converter system consists of a two-channel, multiplexed input to a successive approximation A/D converter. The two A/D input channels are connected to pins PG0–PG1 and the particular input to be selected is determined by the setting/clearing of the CHx bits in the A/D status/control register at $16 (ADSTAT). A further four channels are available internally for test purposes. In addition to the ADSTAT register there is one 8-bit result data register at address $15 (ADDATA).

The A/D converter is ratiometric and a dedicated pin, VREFH, is used to supply the upper reference voltage level of each analog input. The lower voltage reference point, V_{REFL}, is internally connected to the VSS pin. An input voltage equal to or greater than V_{RH} converts to $FF (full scale) with no overflow indication. For ratiometric conversions, the source of each analog input should use V_{REFH} as the supply voltage and be referenced to V_{REFL}.

The A/D converter can operate from either the bus clock or an internal RC type oscillator. The internal RC type oscillator is activated by the ADRC bit in the A/D status/control register (ADSTAT) and can be used to give a sufficiently high clock rate to the A/D converter when the bus speed is too low to provide accurate results (see Section 7.2.1). When the A/D converter is not being used it can be disconnected using the ADON bit in the ADSTAT register, in order to save power (see Section 7.2.1).

7.1 A/D converter operation

The A/D converter consists of an analog multiplexer, an 8-bit digital-to-analog capacitor array, a comparator and a successive approximation register (SAR). See Figure 7-1.

There are two A/D input options that can be selected by the multiplexer: AD0/PG0 and AD1/PG1. Selection is made via the CHx bits in the ADSTAT register (see Section 7.2.1). These bits can also be used to select one of the internal test channels.

The A/D reference input (AD0–AD1) is applied to a precision internal digital-to-analog converter. Control logic drives this D/A converter and the analog output is successively compared with the analog input sampled at the beginning of the conversion. The conversion is monotonic with no missing codes.

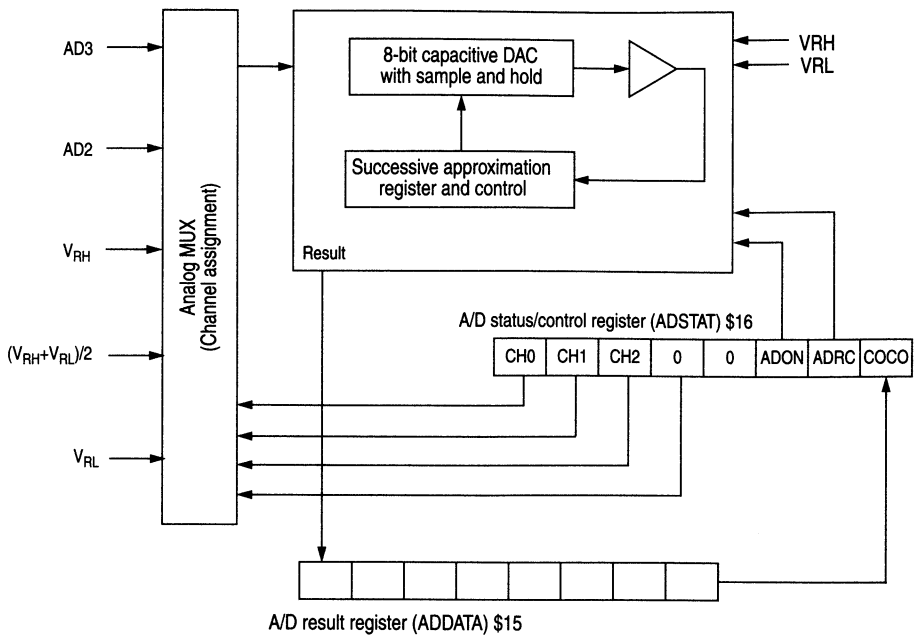


Figure 7-1 A/D converter block diagram

The result of each successive comparison is stored in the SAR and, when the conversion is complete, the contents of the SAR are transferred to the read-only result data register ($15), and the conversion complete flag, COCO, is set in the A/D status/control register ($16).

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

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

7.2 A/D registers

7.2.1 A/D status/control register (ADSTAT)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D status/control (ADSTAT)	$0011	COCO	ADRC	ADON	0	0	CH2	CH1	CH0	0u00 0uuu

COCO — Conversion complete flag

Each channel conversion takes 32 clock cycles at f_{OP}, where f_{OP} is equal to or greater than 1 MHz.

1 (set) – COCO flag is set each time a conversion is complete, allowing the new result to be read from the A/D result data register ($15). The converter then starts a new conversion.

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

Reset clears the COCO flag.

ADRC — A/D RC oscillator control

If the MCU bus frequency is less than 1 MHz, an internal RC oscillator must be used for the A/D conversion clock. This selection is made by setting the ADRC bit in ADSTAT. The ADRC bit allows the user to control the A/D RC oscillator.

1 (set) – The A/D RC oscillator is turned on and, if ADON is set, the A/D runs from the internal RC oscillator clock (see Table 7-1).

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

When the A/D RC oscillator is turned on, it takes a time t_{RCON} to stabilize (see Table 14-7). During this time A/D conversion results may be inaccurate.

Table 7-1 A/D clock selection

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

When the internal RC oscillator is being used as the conversion clock, the following limitations apply.

1) Due to the frequency tolerance of the RC oscillator and its asynchronism with regard to the MCU bus clock, the conversion complete flag (COCO) must be used to determine when a conversion sequence has been completed.
2) The conversion process runs at the nominal 1.5MHz rate but the conversion results must be transferred to the MCU result registers synchronously with the MCU bus clock in order that conversion time is limited to a maximum of one channel per bus clock cycle.
3) If the system clock is running faster than the RC oscillator, the RC oscillator should be switched off and the system clock used as the conversion clock.

ADON — A/D converter on

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

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

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

When the A/D converter is switched on, it takes a time t_{ADON} for the current sources to stabilize (see Table 14-7). During this time A/D conversion results may be inaccurate.

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

CH2 – CH0 — A/D channel selection

The CH2–CH0 bits allow the user to determine which channel of the A/D converter multiplexer is selected (see Table 7-2).

Table 7-2 A/D channel assignment

CH2	CH1	CH0	Channel	Signal
0	0	0	0	AD0/PG0
0	0	1	1	AD1/PG1
0	1	0	2	unused
0	1	1	3	unused
1	0	0	4	V_{RH}
1	0	1	5	$(V_{RH}+V_{RL})/2$
1	1	0	6	V_{RL}
1	1	1	7	Factory test

7.2.2 A/D result data register (ADDATA)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D data (ADDATA)	$0010									Undefined

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

Caution: Performing a digital read of port G with levels other than V_{DD} or V_{SS} on the pins will result in greater power dissipation during the read cycles.

7.3 A/D converter during WAIT mode

The A/D converter continues to operate normally during WAIT mode. To decrease power consumption during WAIT, it is recommended that both the ADON and ADRC bits in the ADSTAT register are cleared, if the A/D converter is not being used. If the A/D converter is being used and the system clock frequency is above 1MHz, the ADRC bit should be cleared to disable the internal RC oscillator.

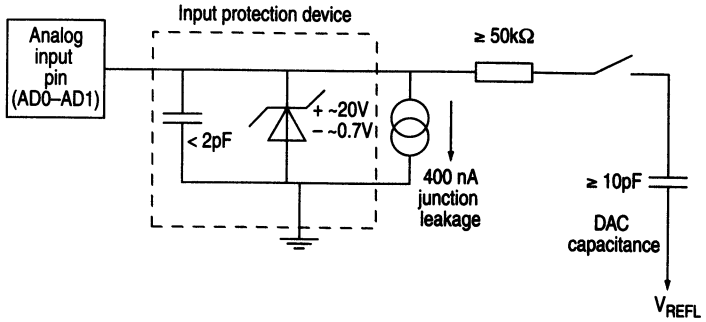
7.4 A/D converter during STOP mode

In STOP mode the comparator and charge pump are turned off and the A/D converter ceases to operate. Any pending conversion is aborted.

7.5 A/D analog input

The external analog voltage value to be processed by the A/D converter is sampled on an internal capacitor through a resistive path, provided by input-selection switches and a sampling aperture time switch, as shown in Figure 7-2. Sampling time is limited to 12 bus clock cycles. After sampling, the analog value is stored on the capacitor and held until the end of conversion. During this hold time, the analog input is disconnected from the internal A/D system and the external voltage source sees a high impedance input.

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



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

Figure 7-2 Electrical model of an A/D input pin

8
CORE TIMER

The MC68HC05E16 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.

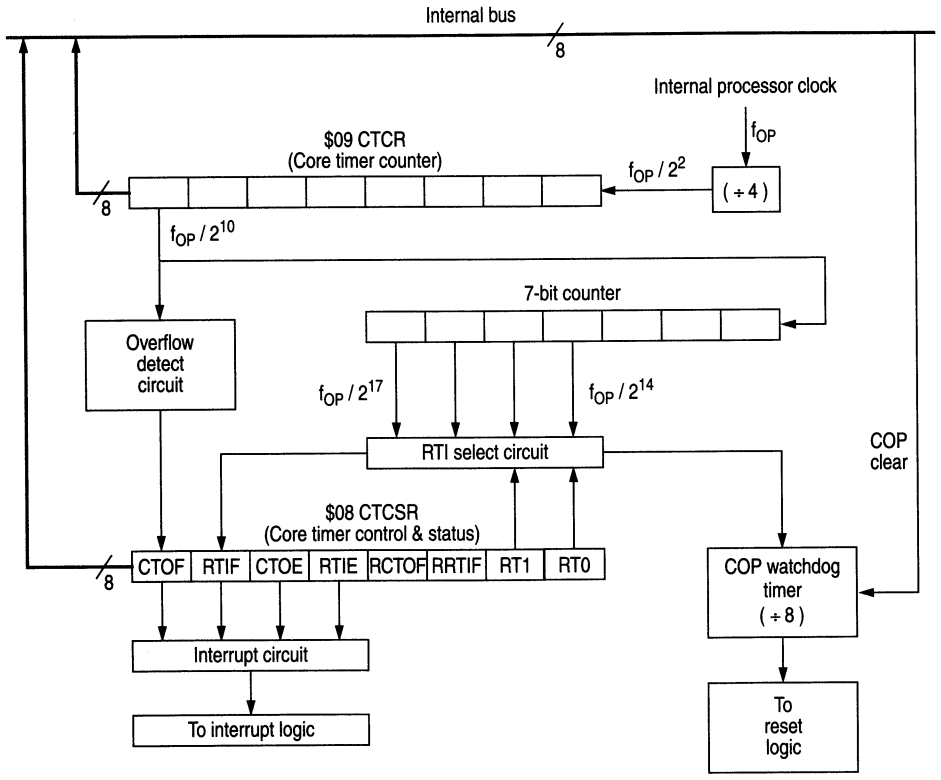


Figure 8-1 Core timer block diagram

As shown in Figure 8-1, 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 $f_{OP}/32$.) 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 eight 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 read-only status bit which is set when the 8-bit ripple counter rolls over from $FF to $00. A CPU interrupt request will be generated if CTOE is set. Clearing CTOF is done by writing a '1' to the RCTOF bit (bit 3) in the CTCSR. Reset clears CTOF.

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

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.

8.1 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 $f_{OP}/2^{13}$ (or $f_{OP}/8192$), with three additional divider stages, giving a maximum interrupt period of 4 seconds at a bus frequency (f_{OP}) of 32.768kHz. Register details are given in Section 8.3.

8.2 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 8-1. 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.

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 $7FF0. When the COP is cleared, only the final divide-by-eight stage is cleared (see Figure 8-1).

The COP function is a mask option, enabled or disabled during device manufacture.

8.3 Core timer registers

8.3.1 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	CTOE	RTIE	RCTOF	RRTIF	RT1	RT0	uu00 0011

CTOF — Core timer overflow

1 (set) – This read-only flag is set whenever a core timer overflow occurs.

0 (clear) – No core timer overflow has occurred.

This bit is set when the core timer counter register rolls over from $FF to $00; an interrupt request will be generated if CTOE is set. When set, the bit may be cleared by writing a '1' to the RCTOF bit.

RTIF — Real time interrupt flag

1 (set) – This read-only flag is set when the pre-selected RTI period has elapsed. The RTI period is selected using the RT0 and RT1 bits as shown in Table 8-1.

0 (clear) – The pre-selected RTI period has not elapsed.

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 '1' to the RRTIF bit.

CTOE — Core timer overflow interrupt enable

1 (set) – A core timer overflow interrupt will be generated if CTOF is set and the I-bit in the CCR is clear.

0 (clear) – No core timer overflow interrupt will be generated regardless of the state of the CTOF flag and the I-bit.

RTIE — Real time interrupt enable

1 (set) – A real time interrupt will be generated if RTIF is set and the I-bit in the CCR is clear.

0 (clear) – No real time interrupt will be generated regardless of the state of the RTIF flag and the I-bit.

RCTOF — Reset core timer overflow flag

This bit always reads as zero. Writing a '1' to the RCTOF bit clears the timer overflow flag. Writing a '0' to it has no effect.

RRTIF — Reset real time interrupt flag

This bit always reads as zero. Writing a '1' to the RRTIF bit clears the real time interrupt flag. Writing a '0' to it has no effect.

RT1, RT0 — Real time interrupt rate select

These two bits select one of four taps from the real time interrupt circuitry. Reset sets both RT0 and RT1 to one, selecting the lowest periodic rate and therefore the maximum time in which to alter them if necessary. The COP reset times are also determined by these two bits. Care should be taken when altering RT0 and RT1 if a timeout is imminent, or if 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 8-1 for some example RTI periods.

Table 8-1 Example RTI periods

			Bus frequency f_{OP} = 500 kHz		Bus frequency f_{OP} = 1 MHz		Bus frequency f_{OP} = 2 MHz	
RT1	**RT0**	**Division ratio**	**RTI period**	**Minimum COP period**	**RTI period**	**Minimum COP period**	**RTI period**	**Minimum COP period**
0	0	2^{14}	31.3ms	218.8ms	15.6ms	109.4ms	7.8ms	54.7ms
0	1	2^{15}	62.5ms	437.5ms	31.3ms	218.8ms	15.6ms	109.4ms
1	0	2^{16}	125ms	875.0ms	62.5ms	437.5ms	31.3ms	218.8ms
1	1	2^{17}	250ms	1.75s	125.1ms	875.0ms	62.5ms	437.5ms

8.3.2 Core timer counter register (CTCR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Core timer counter (CTCR)	$0009									0000 0000

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

8.4 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.

8.5 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.

8

THIS PAGE LEFT BLANK INTENTIONALLY

9
CUSTOM PERIODIC INTERRUPT

9.1 Custom periodic interrupt (CPI)

The custom periodic interrupt is software programmable to a 0.25 second, 0.5 second, or 1 second interrupt. The interrupt is generated from the 32 kHz XOSC1 input by a 15-bit counter. This interrupt is under the control of the custom periodic interrupt control/status register located at $0006.

9.1.1 CPI control/status register (CPICSR)

The CPICSR contains the CPI flag and enable bits. The custom periodic interrupt control/status register (CPICSR) below shows the location of these bits and their values after reset.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
CPI control/status (CPICSR)	$0006	0	CPIF	0	CPIE	0	0	CPI1	CPI0	0000 0000

CPIF — Custom periodic interrupt flag

1 (set) – A valid CPI interrupt has been generated.

0 (clear) – No valid CPI interrupt has been generated.

CPIF is a clearable, read-only status bit and is set when the 15-bit counter changes from $7FFF to $0000. A CPU interrupt request will be generated if CPIE is set. Clearing the CPIF is done by writing a '0' to it. Writing a '1' to CPIF has no effect on the bit's value. Reset clears CPIF.

CPIE — Custom periodic interrupt enable

1 (set) – CPI interrupts are enabled.

0 (clear) – CPI interrupts are disabled.

When this bit is cleared, the counter is cleared and CPI interrupts are disabled. When this bit is set, the counter starts from $0000 and a CPU interrupt request is generated when the CPIF bit is set. Reset clears this bit.

CPI1, CPI0 — Custom periodic interrupt rate select

These two read/write bits select one of three taps from the 15-stage counter to provide an interrupt rate according to Table 9-1. Reset clears these bits, selecting the lowest periodic rate. Care should be taken when altering CPI1 and CPI0 if the timeout period is imminent or uncertain. If the selected tap is modified during a cycle in which the counter is switching, a CPIF could be missed or an additional one could be generated.

Table 9-1 CPI rates (32.768kHz crystal)

CPI1	CPI0	CPI interrupt rate
0	0	1s (reset condition)
0	1	1s (reserved)
1	0	0.5s
1	1	0.25s

9.2 CPI during STOP mode

The timer system is cleared and the CPI counter is halted when going into STOP mode. When STOP is exited by an external interrupt or an external reset, the internal oscillator resumes operation, followed by a 1024 internal processor oscillator stabilization delay. The timer system counter is then cleared and operation resumes. The CPI continues counting once the oscillator resumes operation and does not wait for the oscillator to stabilize.

9.3 CPI during WAIT mode

The CPU clock halts during WAIT mode, but the timer and CPI remain active. If interrupts are enabled, a timer interrupt or custom periodic interrupt will cause the processor to exit WAIT mode.

10
RESETS AND INTERRUPTS

10.1 Resets

The MC68HC05E16 can be reset in four ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by a COP watchdog reset (if the watchdog timer is enabled) and by an opcode fetch from an illegal address. Any of these resets will cause the program to go to its starting address, specified by the contents of memory locations $7FFE and $7FFF, and cause the interrupt mask of the condition code register to be set.

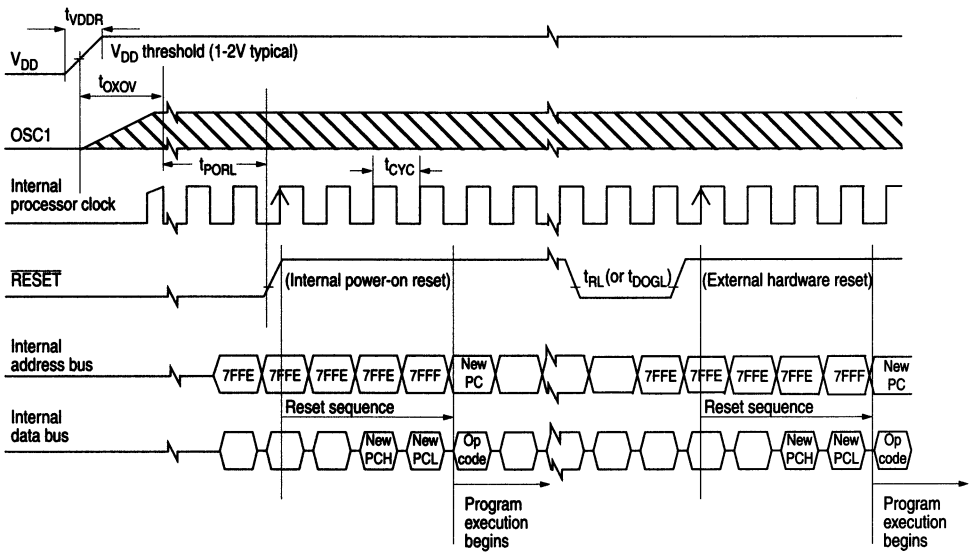


Figure 10-1 Reset timing diagram

10.1.1 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 the $\overline{\text{RESET}}$ pin to generate a power-on reset (POR). In this case, the time constant must be great enough to allow the oscillator circuit to stabilize.

10.1.2 $\overline{\text{RESET}}$ 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.

10.1.3 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 $7FF0.

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.

10.1.4 Illegal address reset

When an opcode fetch occurs from an address which is not part of the RAM ($0040–$019F), ROM ($4000–$7FFF) or EEPROM ($01C0–$02FF), the device is automatically reset.

Note: No JMP,X, RTS or RTI instruction should be placed at the end of a memory block, i.e. at address $019F and $2FF, since this would result in an illegal address reset.

10.2 Interrupts

The MCU can be interrupted by eight different sources (seven maskable hardware interrupts and one nonmaskable software interrupt):

- External signal on the $\overline{\text{IRQ1}}$ pin
- External signal on the $\overline{\text{IRQ0}}$ pin
- Core timer interrupt
- Low voltage indication interrupt (LVI)
- 16-bit programmable timer interrupt
- Keyboard interrupt
- M-bus interrupt
- 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.

10.2.1 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 10-1 shows the relative priorities of all the possible interrupt sources. Figure 10-2 shows the interrupt processing flow.

Table 10-1 Interrupt priorities

Source	Register	Flags	Vector address	Priority
Reset	—	—	$7FFE, $7FFF	highest
Software interrupt	—	—	$7FFC, $7FFD	
External interrupt 0	—	—	$7FFA, $7FFB	
Core timer overflow	CTCSR	TOF	$7FF8, $7FF9	
Real time interrupt	CTCSR	RTI	$7FF8, $7FF9	
Custom periodic interrupt	CPICSR	CPIF	$7FF8, $7FF9	
Low voltage interrupt	LVICSR	LVIINT	$7FF6, $7FF7	
16-bit timer input capture 1	TSR	ICF1	$7FF4, $7FF5	
16-bit timer output compare 1	TSR	OCF1	$7FF4, $7FF5	
16-bit timer input capture 2	TSR	ICF2	$7FF4, $7FF5	
16-bit timer output compare 2	TSR	OCF2	$7FF4, $7FF5	
16-bit timer overflow	TSR	TOF	$7FF4, $7FF5	
M-bus I/O completed	MSR	MIF	$7FF2, $7FF3	
External interrupt 1	IRQ1/KWI	IRQ1INT	$7FF0, $7FF1	
Keyboard interrupt	IRQ1/KWI	KWIINT	$7FF0, $7FF1	

10.3 Nonmaskable software interrupt (SWI)

The software interrupt (SWI) is an executable instruction and a nonmaskable 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 $7FFC and $7FFD.

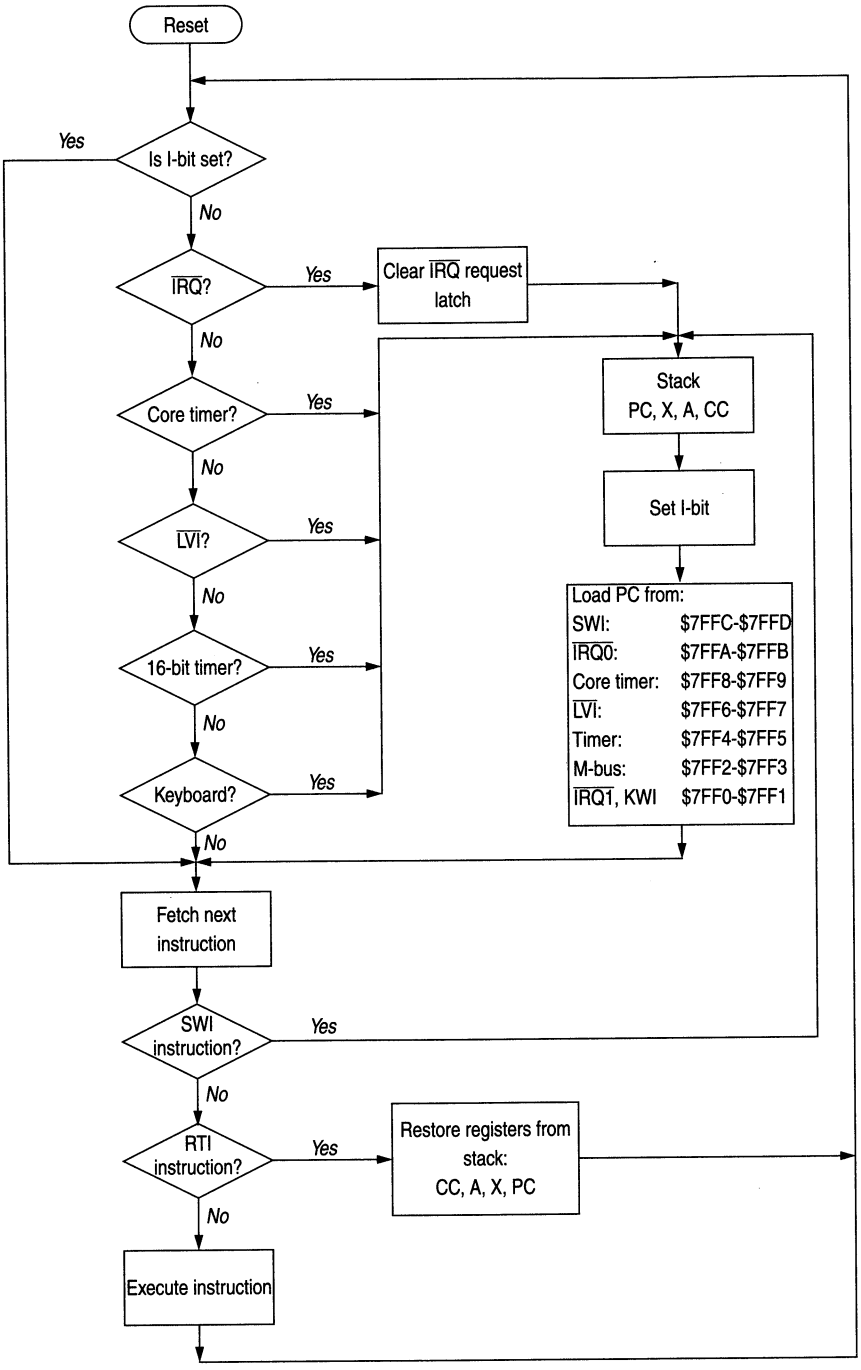


Figure 10-2 Interrupt flowchart

10.4 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.

10.4.1 External interrupt ($\overline{\text{IRQ0}}$)

This external interrupt source will vector to the start address contained in memory locations $7FFA and $7FFB. The trigger sensitivity of the $\overline{\text{IRQ0}}$ interrupt is selected using the OPTIRQ0 bit in the LVI/options register (see Section 10.4.4.1).

10.4.2 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 $7FF8 and $7FF9. 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 CTOE bit must first be set. The CTOF bit will then be set when the core timer counter register overflows from $FF to $00.

10.4.3 Custom periodic (CPI) interrupt

There is a timer interrupt flag that causes a CPI interrupt whenever it is set and enabled. The interrupt flag and the enable bits are located in the CPI control and status register (CPICSR). An interrupt will vector to the same interrupt service routine as the core timer interrupts, whose start address is contained in memory locations $7FF8 and $7FF9.

10.4.4 Low voltage indicator interrupt

The low voltage indicator on the MC68HC05E16 can be configured to respond to a drop in supply voltage in two different ways: it can be serviced by the user software or it can be set up to automatically generate a system interrupt.

In both cases, the power supply could be connected to a low voltage detection circuit which is in turn connected to the $\overline{LVI}$ pin of the device. This allows the voltage from the power supply to be monitored and, if the voltage being supplied to the device falls below a useful operating voltage, the $\overline{LVI}$ pin will be driven low and the LVIVAL bit in the LVI/options register (LVIOPT) will read zero.

It is at this point that the user can decide which way the system should respond. The first method is one in which the user program continually checks the LVIVAL bit for a '0', at which point it enters a particular routine whereby all useful information is saved and the device enters a predefined operating state, e.g. WAIT or STOP mode.

The second method is one whereby a '0' in the LVIVAL bit of LVIOPT automatically generates a system interrupt, provided LVIE is set. The occurrence of a valid LVI interrupt can be detected by reading the LVIINT bit of the LVI/options register. The LVI interrupt has a dedicated vector at $7FF6 – $7FF7.

Note: The interrupt service routine must reset the interrupt by writing a '1' to the LVIRST bit in LVIOPT.

The main feature of these methods of LVI handling is that the user can shut down the micro and the application in an orderly manner before the voltage drops below a useful operating voltage.

If the CPU performs a power-on reset due to a supply voltage below the power-on trip level, no interrupt will be performed and the CPU start-up will be delayed until $\overline{LVI}$ becomes high (see Figure 10-3).

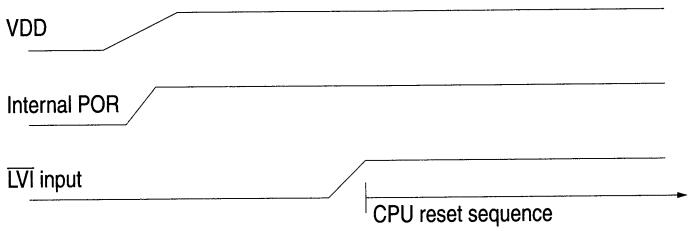


Figure 10-3 LVI power-on sequence

Alternatively, during power-down, when the power to VDD has fallen below a useful operating level, the LVI interrupt will not be generated until the $\overline{\text{LVI}}$ input pin has been driven low (see Figure 10-4).

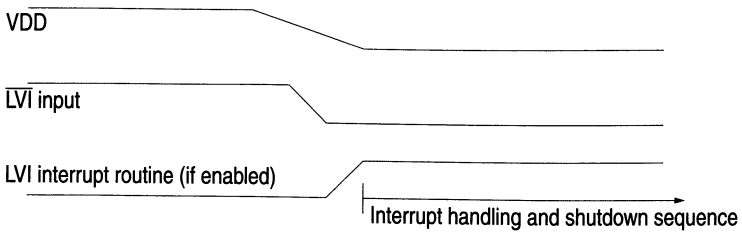


Figure 10-4 LVI power-down sequence

Having dropped to a level which drives the $\overline{\text{LVI}}$ pin low, while staying above the data retention level, the power supply then rises again to the normal operating range along with a low to high transition of $\overline{\text{LVI}}$, an interrupt will be generated to wake-up the CPU. The system clock is restarted if it was halted using the STOP instruction (see Figure 10-5).

Note: All interrupts which should not wake-up the CPU should be disabled prior to entering the low power mode.

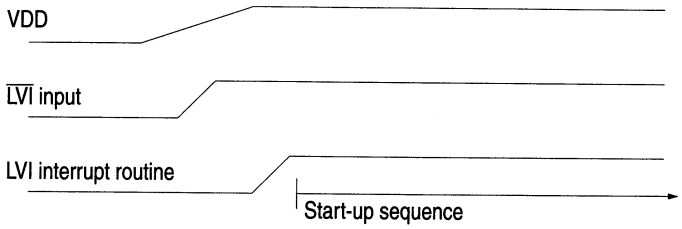


Figure 10-5 LVI recovery sequence

Note: The $\overline{\text{LVI}}$ pin can be used as an additional one bit input by testing the LVIVAL bit in the LVIOPT register. It can also be used as a falling and rising edge sensitive interrupt input. The only restrictions which apply are that the pin must be held high during power-on.

10.4.4.1 LVI/options register (LVIOPT)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
LVI/options (LVIOPT)	$000F	LVIINT	LVIVAL	LVIRST	LVIENA	0	0	0	OPTIRQ0	0u00 uu00

LVIINT — LVI interrupt flag

1 (set) – A valid LVI interrupt has been generated.

0 (clear) – No valid LVI interrupt has been generated.

This flag bit is cleared by writing a '1' to the LVIRST bit and by reset.

LVIVAL — $\overline{\text{LVI}}$ pin level

This bit reflects the level on the $\overline{\text{LVI}}$ input pin and can be used by the user to check the voltage being supplied to VDD.

1 (set) – The $\overline{\text{LVI}}$ pin is high; power supply is above a useful operating level, as determined by voltage detector circuit external to device.

0 (clear) – The $\overline{\text{LVI}}$ pin is low; power supply has fallen below a useful operating level, as determined by voltage detector circuit external to device.

LVIRST — LVI interrupt reset

This bit is write-only; any read will always return zero. Writing a '1' to this bit resets the LVI interrupt routine.

LVIE — LVI interrupt enable

1 (set) – LVI interrupts enabled; an interrupt will be generated on each transition of the $\overline{\text{LVI}}$ pin.

0 (clear) – LVI interrupts disabled.

OPTIRQ0 — $\overline{\text{IRQ0}}$ edge sensitivity

This bit is used to select the sensitivity of the $\overline{\text{IRQ0}}$ interrupt trigger as either edge-and-level sensitive or edge sensitive only. On reset this bit is cleared so that the $\overline{\text{IRQ0}}$ interrupt is edge-sensitive only.

1 (set) – $\overline{\text{IRQ0}}$ is negative edge-and-level sensitive.

0 (clear) – $\overline{\text{IRQ0}}$ is negative edge sensitive only.

10.4.5 16-bit timer interrupts

There are five different timer interrupt flags (IC1F, IC2F, OC1F, OC2F and TOF) that will cause a timer interrupt whenever they are set and enabled. These interrupt flags are found in the five most significant bits of the timer status register (TSR) at location $002E. The interrupt enable bits are found in the timer control registers TCR1 and TCR2 ($002C and $002D). IC1F, IC2F, OC1F, OC2F and TOF will vector to the service routine defined by $7FF4 - $7FF5 as shown in Table 10-1.

Full details of the programmable timer can be found in Section 5.

10.4.6 M-bus interrupt

There is an interrupt flag (together with three status flags) that causes an M-bus interrupt whenever it is set and enabled. Such an interrupt will vector to the interrupt service routine whose start address is contained in memory locations $7FF2 and $7FF3.

10.4.7 $\overline{\text{IRQ1}}$ interrupt

The $\overline{\text{IRQ1}}$ interrupt is much more flexible than the $\overline{\text{IRQ0}}$ interrupt, in that, it can be enabled/disabled independently, and it has a choice of four trigger sensitivities:

- falling edge
- falling edge and low level
- rising edge
- rising edge and high level

The interrupt vector location for this interrupt is at $7FF0 and $7FF1. This is shared with the keyboard interrupt. Therefore, in order to show that $\overline{\text{IRQ1}}$ has caused the interrupt, the IRQ1INT flag in the IRQ1/KWI becomes set (see Section 10.4.8.1).

10.4.8 Keyboard interrupt

When configured as input pins, port C bits 0–7 provide a wired-OR keyboard interrupt facility and will generate an interrupt provided the keyboard interrupt enable bit (KIE) in the IRQ1/KWI register is set. When configured as inputs with keyboard interrupt capability, the pins can also have pull-ups connected to them by correctly configuring the DDRC and CONFC bits as outlined in Table 4-2.

The interrupt vector for this interrupt is located at $7FF2, $7FF3. Further information on the keyboard interrupt facility can be found in Section 4.3.1.

10.4.8.1 IRQ1/KWI status/control register (IRQ1/KWI)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
IRQ1/KWI status/control (IRQ1/KWI)	$001B	KSF	KIE	IRQ1INT	IRQ1ENA	IRQ1LV	IRQ1EDG	IRQ1RST	IRQ1VAL	0000 000u

KSF — KWI interrupt flag

1 (set) – A valid keyboard interrupt has been generated.

0 (clear) – No valid keyboard interrupt has been generated

KIE — KWI interrupt enable

1 (set) – Keyboard interrupts are enabled.

0 (clear) – Keyboard interrupts are disabled.

IRQ1INT — IRQ1 interrupt flag

1 (set) – A valid IRQ1 interrupt has been generated.

0 (clear) – No valid IRQ1 interrupt has been generated.

IRQ1ENA — IRQ1 interrupt enable

1 (set) – IRQ1 interrupts are enabled.

0 (clear) – IRQ1 interrupts are disabled.

IRQ1LV, IRQ1EDG — IRQ1 interrupt sensitivity bits

These two bits are used to select the sensitivity of the IRQ1 interrupt trigger according to Table 10-2.

Table 10-2 $\overline{\text{IRQ1}}$ interrupt sensitivity

IRQ1LV	IRQ1EDG	Interrupt sensitivity
0	0	Falling edge(1)
0	1	Rising edge
1	0	Falling edge and low level
1	1	Rising edge and high level

(1) reset condition

IRQ1RST — IRQ1 reset

The IRQ1 interrupt is cleared by writing a '1' to this bit. This bit is write-only and always returns zero.

IRQ1VAL — $\overline{\text{IRQ1}}$ pin status

The IRQ1VAL bit reflects current status of the $\overline{\text{IRQ1}}$ pin.

10.5 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 2-2.

RESET: A reset condition causes the program to vector to its starting address, which is contained in memory locations $7FFE (MSB) and $7FFF (LSB). The I-bit in the condition code register is also set, to disable interrupts.

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.

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 core timer, the 16-bit timer and the A/D converter remain active. An external, keyboard or LVI interrupt or an interrupt from the core timer or 16-bit timer, if enabled, will cause the MCU to exit WAIT mode.

11
PHASE-LOCKED LOOP

11.1 Phase-locked loop synthesizer

The phase-locked loop (PLL) consists of a variable bandwidth loop filter, a voltage controlled oscillator (VCO), a feedback frequency divider, and a digital phase detector. A small external capacitor (typically 0.1μF) is used by the loop filter. V_{DDSYN} is the supply source for the PLL and should be decoupled to minimize noise.

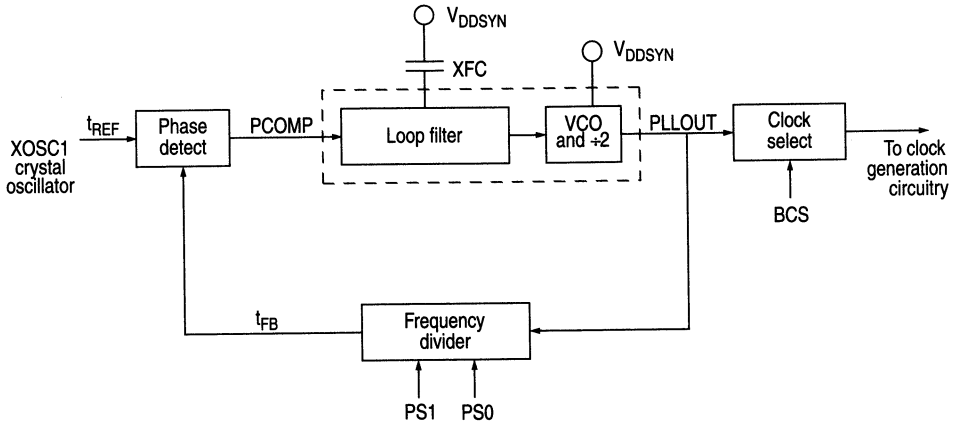


Figure 11-1 PLL circuit

The phase detector compares the frequency and phase of the feedback frequency (t_{FB}) and the crystal oscillator reference frequency (t_{REF}) and generates the output, PCOMP, as shown in Figure 11-1. The output waveform is then integrated and amplified. The resultant dc voltage is applied to the voltage controlled oscillator. The output of the VCO is divided by a variable frequency divider of 256, 128, 64, or 32 to provide the feedback frequency for the phase detector.

To change PLL frequencies, follow the procedure outlined below to modify the bits in the PLL control register:

1) Clear BCS to enable the low frequency bus rate
2) Clear PLLON to disable the PLL
3) Select the speed using PS1 and PS0
4) Set PLLON to enable the PLL
5) Wait for the BWC bit to go low
6) Set BCS to switch to the high frequency bus rate

The user should not switch among the high speeds with the BCS bit set. Following the procedure above will prevent possible bursts of high frequency operation during the re-configuration of the PLL. Using a 32KHz oscillator, the start-up time of the PLL is less than 10 milliseconds.

The PLL loop filter has two bandwidths which are automatically selected by the PLL. Whenever the PLL is first enabled, the wide bandwidth mode is used. This enables the PLL frequency to ramp up quickly. When the output frequency is near the desired frequency, the filter is switched to the narrow bandwidth mode to make the final frequency more stable.

11.2 V_{DD} supply

V_{DDSYN} is used to power the clock circuits when either an internal or an external reference frequency is supplied. A separate power source increases MCU noise immunity and can be used to run the clock when the MCU is powered down. A quiet power supply must be used as the V_{DDSYN} source. Adequate external bypass capacitors should be placed as close as possible to the VDDSYN pin to assure stable operating frequency. When an external system clock signal is applied and the PLL is disabled, V_{DDSYN} should be connected to the V_{DD} supply.

The integrity of the V_{DD} supply is critical to the functionality and stability of the PLL system. This supply should be routed from the source completely separately from the V_{DD} track used for all other components on the PCB, including the microcontroller. An example of this can be seen in Figure 11-2.

The track used for VDDSYN should be routed away from potentially noisy areas, such as fast switching clocks etc., and should be decoupled as close to the VDDSYN pin as possible using the capacitor values shown in Figure 11-3.

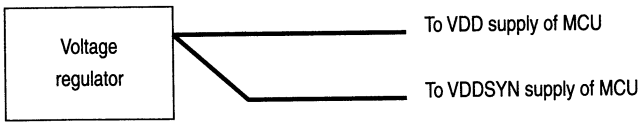


Figure 11-2 Routing of PCB tracks from supply to microcontroller

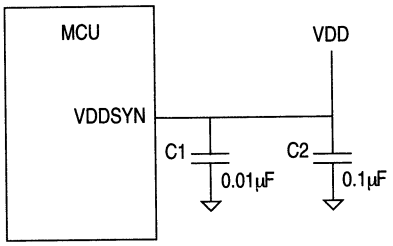


Figure 11-3 Decoupling of V_{DDSYN}

11.2.1 Phase-locked loop control register (PLLCR)

This read/write register contains the control bits to select the PLL frequency and enable/disable the synthesizer.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PLL control (PLLCR)	$0007	0	BCS	1	BWC	PLLON	1	PS1	PS0	0010 1100

BCS — Bus clock select

1 (set) – PLL output is used in internal processor clock generation.

0 (clear) – Processor clock is generated from XOSC1 ÷ 2.

When this bit is set, the output of the PLL is used to generate the internal processor clock. When clear, the internal bus clock is driven by the crystal (XOSC1 ÷ 2). Once BCS has been changed, it may take up to 1.5 XOSC1 cycles + 1.5 PLLOUT cycles to make the transition. During the transition, the clock select output is held low and all CPU and timer activity ceases until the transition is complete. Reset clears this bit.

BWC — Bandwidth control

This bit is primarily intended for Motorola use, to test and characterise the PLL, and should always be cleared by the user. However, BWC can also act as a read-only status bit, indicating the mode selected by the internal circuitry. On PLL start-up, a high bandwidth driver is selected (BWC is set) by internal circuitry until the PLL has locked onto the specified frequency. The high bandwidth driver is then disabled and BWC is cleared. The user can poll BWC to determine when the PLL output is stable and BCS can be set. Reset clears this bit.

PLLON — PLL on

1 (set) – PLL circuit is activated.

0 (clear) – PLL circuit is shut off.

This bit activates the synthesizer circuit without connecting it to the control circuit. This allows the synthesizer to stabilize before it can drive the CPU clocks. When this bit is cleared, the PLL is shut off. Reset sets this bit.

Note: PLLON should not be cleared while using the PLL to drive the internal processor clock, i.e. when BCS is high. If the internal processor clock is driven by the PLL, clearing the PLLON bit would cause the internal processor clock to stop. Exercise caution when using these bits.

PS1, PS0 — PLL synthesizer speed select

These two bits select one of four taps from the PLL to drive the CPU clocks. These bits are used in conjunction with PLLON and BCS bits in the PLL control register. These bits should not be written to if BCS is at a logic high. Reset clears PS1 and PS0, choosing a bus clock frequency of 524kHz.

Table 11-1 PS1 and PS0 speed selects (32.768kHz crystal)

PS1	PS0	CPU bus clock frequency (f_{OP})
0	0	524kHz (reset condition)
0	1	1.049MHz
1	0	2.097MHz
1	1	4.194MHz (to be determined)

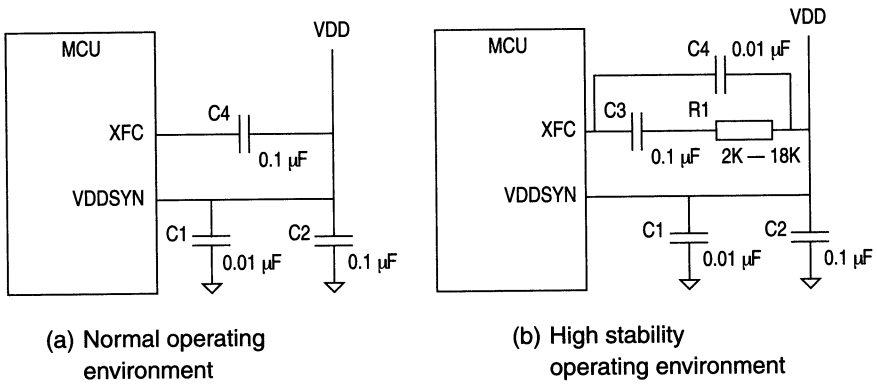
11.2.2 PLL filter operation

A voltage controlled oscillator (VCO) generates the system clock signal. A portion of the clock signal is fed back to a divider/counter. The divider controls the frequency of one input to a phase comparator. The other phase comparator input is a reference signal, either from the crystal oscillator or from an external source. The comparator generates a control signal proportional to the difference in phase between the two inputs. The signal is low-pass filtered and used to correct VCO output frequency.

Filter geometry can vary, depending on the external environment and required clock stability. Figure 11-4 shows two recommended filters.

The filter configuration in Figure 11-4(b) shows an additional series resistor from the circuit in Figure 11-4(a) and also an additional capacitor, C3, which will help to eliminate higher frequency signals. The value of R1 is dependent very much on the application and a higher value may provide a more stable PLL while affecting the time taken for the application to stabilize. Ideally these values should be characterized for the application with the conditions as close as possible to the final operating conditions.

An external filter network connected to the XFC pin is not required when an external system clock signal is applied and the PLL is disabled. The XFC pin must be left floating in this case. The VDDSYN pin must be connected in all cases.



(a) Normal operating environment

(b) High stability operating environment

Figure 11-4 System clock filter networks

11.2.3 Noise immunity

The MCU should be insulated as much as possible from noise in the system. We recommend the following steps be taken to help prevent problems due to noise injection.

1) The application environment should be designed so that the MCU is not near signal traces which switch often, such as a clock signal
2) The oscillator circuit for the MCU should be placed as close as possible to the XOSC1 and XOSC2 pins and the OSC1 and OSC2 pins respectively on the MCU
3) All power pins should be filtered (to minimize noise on these signals) by using bypass capacitors placed as close as possible to the MCU

See the Application Note Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers, available through the Motorola Literature Distribution Centre, document number AN1050/D.

12
OSCILLATOR SYSTEM

12.1 Introduction

The MC68HC05E16 can use three different clock sources: an external crystal/ceramic oscillator (high frequency oscillator), a 32 kHz PLL oscillator, or an internal RC oscillator. All three oscillators have their individual advantages and disadvantages; the user selects (under software control) the oscillator which best fits his needs. In order to reduce power consumption, oscillators that are not needed should be stopped. Figure 12-1 shows an block diagram of the oscillator system.

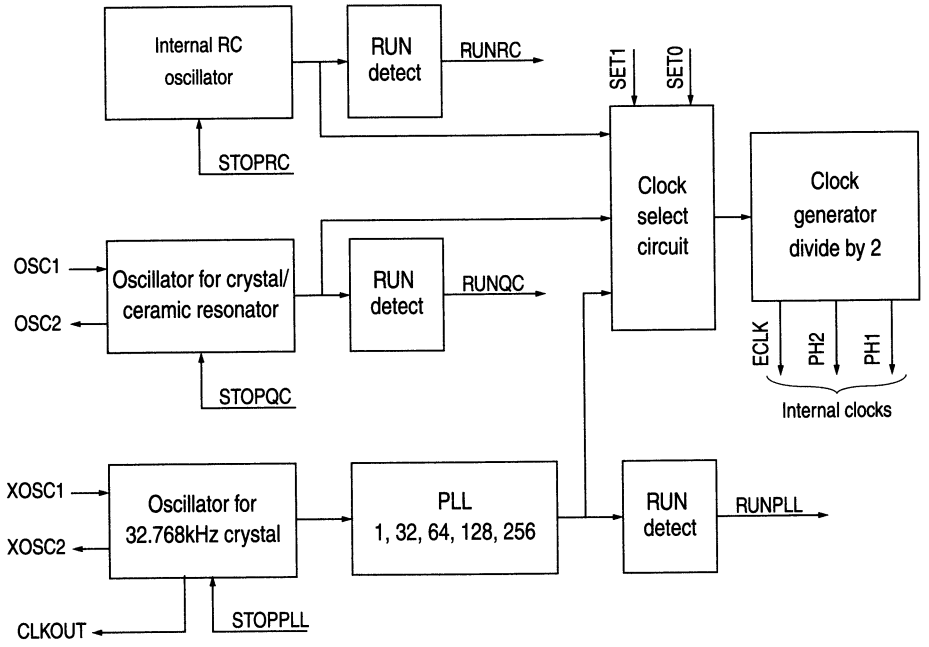


Figure 12-1 Block diagram of the oscillator system

12.2 Crystal/ceramic oscillator

The circuit shown in Figure 12-3 (a) is recommended when using a crystal or a ceramic resonator. The internal oscillator is designed to interface with an AT-cut crystal or ceramic resonator in the frequency range specified for f_{OSC}. Use of an external CMOS oscillator connected to OSC1 is recommended when crystals outside the specified ranges are used. In this case OSC2 must be unconnected, with a maximum load of 20pF on OSC1. The crystal and associated components must be mounted as close as possible to the input pins (OSC1 and OSC2) to minimize output distortion and start-up stabilization time. If short start-up times are needed, the use of a ceramic resonator is recommended, whereas a crystal is more appropriate if high accuracy is required. If this oscillator is not used OSC1 must be tied to VSS.

Recommended values for the crystal/ceramic resonator are shown in the tables in Figure 12-3(c).

12.3 PLL oscillator

The circuit shown in Figure 12-2 is recommended when using a typical 32.768kHz watch crystal. An external CMOS oscillator connected to XOSC1 can also be used to drive the oscillator circuit. In this case, XOSC2 must be unconnected with a maximum load of 5pF. The crystal and components must be mounted as close as possible to the XOSC1 and XOSC2 pins to minimize output distortion and start-up stabilization time. Special care must also be taken for the VSS connection of the external filter capacitor and the decoupling of the VDDSYN power. If this oscillator is not used, XOSC1 must be tied to VSS.

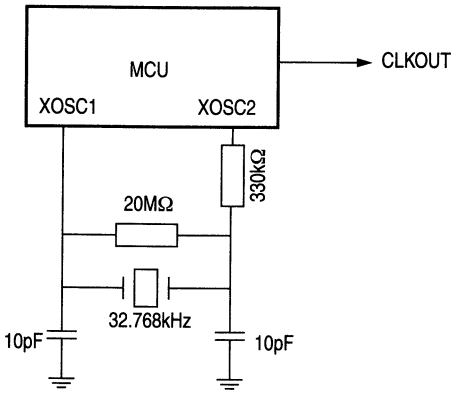
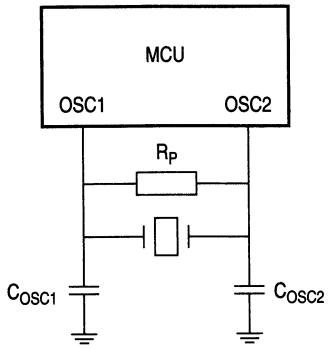
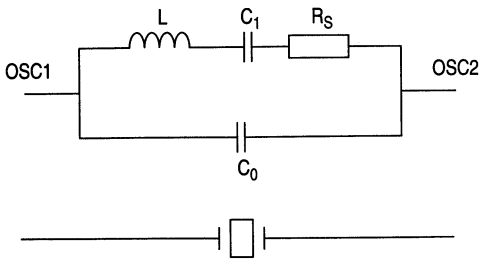


Figure 12-2 Recommended watch crystal connections



(a) Crystal/ceramic resonator oscillator connections



(b) Crystal equivalent circuit

Crystal

	2MHz	4MHz	Unit
R_S(max)	400	75	Ω
C_0	5	7	pF
C_1	8	12	*f*F
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	—

(c) Crystal and ceramic resonator parameters

Figure 12-3 Oscillator connections

12

12.4 Internal RC oscillator

This purely internal RC oscillator is used in order to provide a very fast start-up. Since the accuracy of this oscillator is very poor (±25%), the user should switch to one of the other oscillators as soon as possible to achieve better accuracy, after having completed the simple software initialization procedures.

12.5 Oscillator select register (OSCREG)

This read/write register contains the control bits to start/stop the oscillators, to select the active oscillator and to obtain status information about the oscillators.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Oscillator select (OSCREG)	$001E	STOPPLL	STOPQC	STOPRC	SET1	SET0	RUNPLL	RUNQC	RUNRC	0000 0000

STOPPLL — PLL stop bit

1 (set) – PLL oscillator is stopped provided it is not selected as the active clock.

0 (clear) – Normal PLL oscillator operation.

When this read/write bit is set, the oscillator connected to the XOSC1 and XOSC2 pins, providing the clock for the PLL and the CPI, is stopped. It is not possible to set this bit (i.e. stop the oscillator), if the PLL clock is selected as active clock. Reset and STOP clear this bit, so that this oscillator is started on recovery from reset or STOP.

STOPQC — High frequency oscillator (crystal/ceramic) stop bit

1 (set) – High frequency oscillator is stopped provided it is not selected as the active clock.

0 (clear) – Normal high frequency oscillator operation.

When this read/write bit is set, the oscillator connected to the OSC1 and OSC2 pins is stopped. It is not possible to set this bit (i.e. stop the oscillator), if the ceramic resonator clock is selected as the active clock. Reset and STOP clear this bit, so that this oscillator is started on recovery from reset or STOP.

STOPRC — Internal RC oscillator stop bit

1 (set) – Internal RC oscillator is stopped provided it is not selected as the active clock.

0 (clear) – Normal RC oscillator operation.

When this read/write bit is set, the internal RC oscillator is stopped. It is not possible to set this bit (i.e. stop the oscillator), if the RC oscillator clock is selected as active clock. Reset and STOP clear this bit, so that this oscillator is started recovering from reset or STOP.

SET1, SET0 — Active oscillator select bits

These two read/write bits select the current active oscillator acting as a clock source for the whole system. An oscillator that is not running cannot be selected as active oscillator and the current active oscillator will not be suspended. The SET1 and SET0 bits always reflect the current active

oscillator. Reset and STOP clear these bits, so that the internal RC oscillator is the active one on recovery from reset or STOP.

Table 12-1 Oscillator selection

SET1	SET0	Selected oscillator
0	0	Internal RC oscillator
0	1	High frequency oscillator (crystal/ceramic)
1	0	Low frequency oscillator (PLL)
1	1	Illegal

RUNPLL — PLL stabilization flag

1 (set) – PLL oscillator is stabilized.

0 (clear) – PLL oscillator is not stabilized.

This read-only bit is set if the PLL oscillator (low-frequency oscillator) is properly stabilized. The stabilization period for this oscillator is 1024 cycles. Reset and STOP clear this bit.

RUNQC — High frequency oscillator stabilization flag

1 (set) – High frequency oscillator is stabilized.

0 (clear) – High frequency oscillator is not stabilized.

This read-only bit is set if the high frequency oscillator is properly stabilized. The stabilization period for this oscillator is 1024 cycles. Reset and STOP clear this bit.

RUNRC — Internal RC oscillator stabilization flag

1 (set) – RC oscillator is stabilized.

0 (clear) – RC oscillator is not stabilized.

This read-only bit is set if the internal RC oscillator is properly stabilized. The stabilization period for this oscillator is 128 cycles. Reset and STOP clear this bit.

THIS PAGE LEFT BLANK INTENTIONALLY

13
CPU CORE AND INSTRUCTION SET

This section provides a description of the CPU core registers, the instruction set and the addressing modes of the MC68HC05E16.

13.1 Registers

The MCU contains five registers, as shown in the programming model of Figure 13-1. The interrupt stacking order is shown in Figure 13-2.

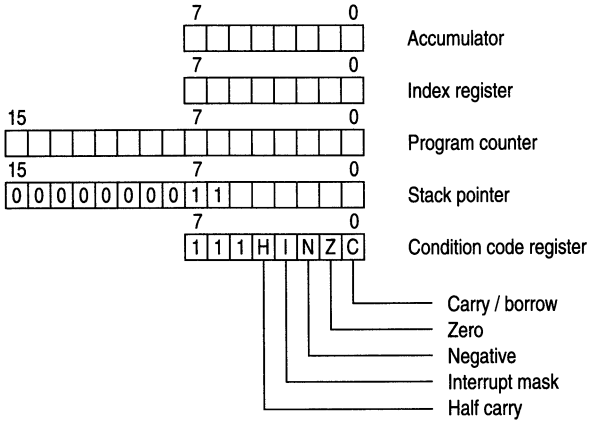


Figure 13-1 Programming model

13.1.1 Accumulator (A)

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

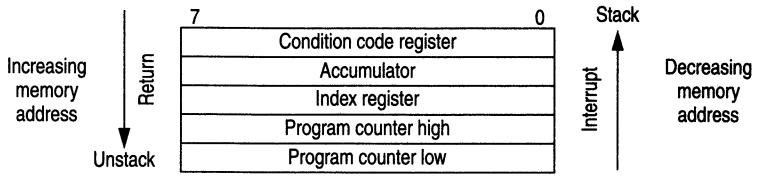


Figure 13-2 Stacking order

13.1.2 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.

13.1.3 Program counter (PC)

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

13.1.4 Stack pointer (SP)

The stack pointer is a 16-bit register, which contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

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

13.1.5 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.

13.2 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 13-1.

13.2.1 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 13-2 for a complete list of register/memory instructions.

13.2.2 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 13-3.

13.2.3 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 13-4.

13.2.4 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 13-5 for a complete list of read/modify/write instructions.

13.2.5 Control instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 13-6 for a complete list of control instructions.

13.2.6 Tables

Tables for all the instruction types listed above follow. In addition there is a complete alphabetical listing of all the instructions (see Table 13-7), and an opcode map for the instruction set of the M68HC05 MCU family (see Table 13-8).

Table 13-1 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 13-2 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

13

Table 13-3 Branch instructions

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

Table 13-4 Bit manipulation instructions

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

Table 13-5 Read/modify/write instructions

Function	Mnemonic	Addressing modes: Inherent (A) Opcode	Inherent (A) # Bytes	Inherent (A) # Cycles	Inherent (X) Opcode	Inherent (X) # Bytes	Inherent (X) # Cycles	Direct Opcode	Direct # Bytes	Direct # Cycles	Indexed (no offset) Opcode	Indexed (no offset) # Bytes	Indexed (no offset) # Cycles	Indexed (8-bit offset) Opcode	Indexed (8-bit offset) # Bytes	Indexed (8-bit offset) # 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 13-6 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

13

Table 13-7 Instruction set

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

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

Not implemented

Table 13-7 Instruction set (Continued)

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

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

Not implemented

13

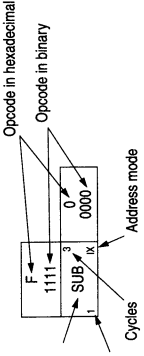
Table 13-8 M68HC05 opcode map

	Bit manipulation		Branch	Read/modify/write					Control		Register/memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
High / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	High / Low
0 0000	BRSET0 5 3 BTB 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 1		SUB 2 IMM 2	SUB 3 DIR 3	SUB 4 EXT 3	SUB 5 IX2 3	SUB 4 IX1 1	SUB 3 IX 1	0 0000
1 0001	BRCLR0 5 3 BTB 2	BCLR0 5 BSC 2	BRN 3 REL 2						RTS 6 INH 1		CMP 2 IMM 2	CMP 3 DIR 3	CMP 4 EXT 3	CMP 5 IX2 3	CMP 4 IX1 1	CMP 3 IX 1	1 0001
2 0010	BRSET1 5 3 BTB 2	BSET1 5 BSC 2	BHI 3 REL 2		MUL 11 INH 1						SBC 2 IMM 2	SBC 3 DIR 3	SBC 4 EXT 3	SBC 5 IX2 3	SBC 4 IX1 1	SBC 3 IX 1	2 0010
3 0011	BRCLR1 5 3 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 1		CPX 2 IMM 2	CPX 3 DIR 3	CPX 4 EXT 3	CPX 5 IX2 3	CPX 4 IX1 1	CPX 3 IX 1	3 0011
4 0100	BRSET2 5 3 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 1			AND 2 IMM 2	AND 3 DIR 3	AND 4 EXT 3	AND 5 IX2 3	AND 4 IX1 1	AND 3 IX 1	4 0100
5 0101	BRCLR2 5 3 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 3	BIT 4 IX1 1	BIT 3 IX 1	5 0101
6 0110	BRSET3 5 3 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 1			LDA 2 IMM 2	LDA 3 DIR 3	LDA 4 EXT 3	LDA 5 IX2 3	LDA 4 IX1 1	LDA 3 IX 1	6 0110
7 0111	BRCLR3 5 3 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 1			STA 3 DIR 3	STA 4 EXT 3	STA 5 IX2 3	STA 4 IX1 1	STA 3 IX 1	7 0111
8 1000	BRSET4 5 3 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 1		EOR 2 IMM 2	EOR 3 DIR 3	EOR 4 EXT 3	EOR 5 IX2 3	EOR 4 IX1 1	EOR 3 IX 1	8 1000
9 1001	BRCLR4 5 3 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 1		ADC 2 IMM 2	ADC 3 DIR 3	ADC 4 EXT 3	ADC 5 IX2 3	ADC 4 IX1 1	ADC 3 IX 1	9 1001
A 1010	BRSET5 5 3 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 1		ORA 2 IMM 2	ORA 3 DIR 3	ORA 4 EXT 3	ORA 5 IX2 3	ORA 4 IX1 1	ORA 3 IX 1	A 1010
B 1011	BRCLR5 5 3 BTB 2	BCLR5 5 BSC 2	BMI 3 REL 2						SEI 2 INH 1		ADD 2 IMM 2	ADD 3 DIR 3	ADD 4 EXT 3	ADD 5 IX2 3	ADD 4 IX1 1	ADD 3 IX 1	B 1011
C 1100	BRSET6 5 3 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 1			JMP 2 DIR 3	JMP 3 EXT 3	JMP 4 IX2 3	JMP 3 IX1 1	JMP 2 IX 1	C 1100
D 1101	BRCLR6 5 3 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 1	BSR 6 REL 2		JSR 3 DIR 3	JSR 5 EXT 3	JSR 7 IX2 3	JSR 6 IX1 1	JSR 5 IX 1	D 1101
E 1110	BRSET7 5 3 BTB 2	BSET7 5 BSC 2	BIL 3 REL 2						STOP 2 INH* 1		LDX 2 IMM 2	LDX 3 DIR 3	LDX 4 EXT 3	LDX 5 IX2 3	LDX 4 IX1 1	LDX 3 IX 1	E 1110
F 1111	BRCLR7 5 3 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 1		STX 3 DIR 3	STX 4 EXT 3	STX 5 IX2 3	STX 4 IX1 1	STX 3 IX 1	F 1111

Abbreviations for address modes and registers

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

Legend



Not implemented

13.3 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**.

13.3.1 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.

13.3.2 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).

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

13.3.3 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.

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

13.3.4 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)

13.3.5 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

13.3.6 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)

13.3.7 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)

13.3.8 Relative

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode are added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from –126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

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

13.3.9 Bit set/clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode. The byte following the opcode specifies the address of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

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

13.3.10 Bit test and branch

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit to be tested and its condition (set or clear) is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte (EA1). The signed relative 8-bit offset in the third byte (EA2) is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branch is from –125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

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

THIS PAGE LEFT BLANK INTENTIONALLY

14
ELECTRICAL SPECIFICATIONS

14.1 Maximum ratings

Table 14-1 Maximum ratings

Rating	Symbol	Value	Unit
Supply voltage(1)	V_{DD}	– 0.3 to +7.0	V
Input voltage	V_{IN}	V_{SS} – 0.3 to V_{DD} + 0.3	V
Input voltage: Bootloader mode ($\overline{IRQ0}$ pin only)	V_{IN}	V_{SS} – 0.3 to $2V_{DD}$ + 0.3	V
Operating temperature range (standard plastic package)	T_A	T_L to T_H –40 to +85	°C
Storage temperature range	T_{STG}	– 65 to +150	°C
Current drain per pin (2) – excluding VDD and VSS	I_D	25	mA

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

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

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

14.2 Thermal characteristics and power considerations

Table 14-2 Package thermal characteristics

Characteristics	Symbol	Value	Unit
Thermal resistance: – Plastic 64-pin QFP package – Plastic 44-pin QFP package – Plastic 56-pin SDIP package	θ_{JA}	 50 60 50	 °C/W °C/W °C/W

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

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

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:

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

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

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

where K is a constant for a particular part. K can be determined by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained for any value of T_A by solving the above equations. The package thermal characteristics are shown in Table 14-2.

Pins	R1	R2	C
PA0–7, PB0–7, PC0–7, PD0–7, PE0–7, PF0–6, PG0–1	3.26kΩ	2.38kΩ	50pF

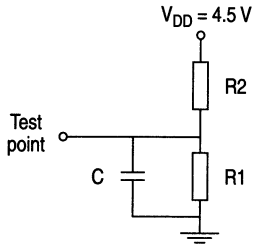


Figure 14-1 Equivalent test load

14.3 DC electrical characteristics for 5V operation

Table 14-3 DC electrical characteristics (V_{DD} = 5 V)

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

Characteristic	Symbol	Min.	Typ.(1)	Max.	Unit
Output voltage(2) I_{LOAD} = –10 µA I_{LOAD} = +10 µA	 V_{OH} V_{OL}	 V_{DD} – 0.1 —	 — —	 — 0.1	 V V
Output high voltage (I_{LOAD} = –0.8 mA) Ports (excluding port G)	V_{OH}	V_{DD} – 0.8	V_{DD} – 0.4	—	V
Output low voltage (I_{LOAD} = +1.6 mA) Ports (excluding port G)	V_{OL}	—	0.1	0.4	V
Input high voltage Ports, OSC1, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{RESET}$, $\overline{LVI}$	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage Ports, OSC1, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{RESET}$, $\overline{LVI}$	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Pull-up source current (V_{IN} = V_{DD}–1.0V) Ports B, C, D, E and F (if enabled)	I_{PU}	10	60	120	µA
Pull-down sink current (V_{IN} = 1.0V) Port A	I_{PD}	10	30	90	µA
I/O ports hi-Z leakage current(3) Ports C, D, E and F, $\overline{RESET}$, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{LVI}$	I_{OZ}	—	±0.2	±1	µA
Capacitance(2) All I/Os	C_{OUT}	—	—	12	pF
Supply current (4) RUN (@ 2MHz f_{BUS}) RUN (@ 16kHz f_{BUS}) WAIT (@ 2MHz f_{BUS}) WAIT (@16kHz f_{BUS}) STOP (oscillators off)	 I_{DD} 	 — — — — —	 5 0.65 1 0.4 1	 12 1.5 4 0.7 8	 mA mA mA mA µA

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

(2) Guaranteed by design

(3) Not applicable to port A or port B due to pullups/pulldowns.

(4) Wait IDD: Only timer system active. Run (Operating) IDD, Wait IDD: Measured using external square wave clock source (fosc = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50pF on all outputs, CL = 20 pF on OSC2, CL = 5pF on XOSC2. Wait, Stop IDD: All ports configured as inputs, VIL = 0.2 V, VIH = VDD–0.2 V. Stop IDD measured with XOSC1, OSC1 = VSS. Wait IDD is affected linearly by the XOSC2, OSC2 capacitance.

14.4 DC electrical characteristics for 3.3V operation

Table 14-4 DC electrical characteristics (V_{DD} = 3.3 V)

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

Characteristic	Symbol	Min.	Typ.(1)	Max.	Unit
Output voltage(2) I_{LOAD} = –10 µA I_{LOAD} = +10 µA	 V_{OH} V_{OL}	 V_{DD} – 0.1 —	 — —	 — 0.1	 V V
Output high voltage (I_{LOAD} = –0.4 mA) Ports (excluding port G)	V_{OH}	V_{DD} – 0.3	V_{DD} – 0.3	—	V
Output low voltage (I_{LOAD} = 0.8 mA) Ports (excluding port G)	V_{OL}	—	0.1	0.3	V
Input high voltage Ports, OSC1, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{RESET}$, $\overline{LVI}$	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input low voltage Ports, OSC1, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{RESET}$, $\overline{LVI}$	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Pull-up source current (V_{IN} = V_{DD}–1.0V) Ports B, C, D, E and F (if enabled)	I_{PU}	5	30	100	µA
Pull-down sink current (V_{IN} = 1.0V) Port A	I_{PD}	5	20	80	µA
I/O ports hi-Z leakage current(3) Ports C, D, E and F, $\overline{RESET}$, $\overline{IRQ0}$, $\overline{IRQ1}$, $\overline{LVI}$	I_{OZ}	—	±0.2	±1	µA
Capacitance(2) All I/Os	C_{OUT}	—	—	12	pF
Supply current (4) RUN (@ 1MHz f_{BUS}) RUN (@ 16kHz f_{BUS}) WAIT (@ 1MHz f_{BUS}) WAIT (@16kHz f_{BUS}) STOP (oscillators off)	 I_{DD} 	 — — — — —	 2 0.5 0.5 200 0.5	 6 1 2 400 4	 mA mA mA µA µA

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

(2) Guaranteed by design

(3) Not applicable to port A or port B due to pullups/pulldowns.

(4) Wait IDD: Only timer system active. Run (Operating) IDD, Wait IDD: Measured using external square wave clock source (fosc = 2.1 MHz), all inputs 0.2 V from rail; no dc loads, less than 50pF on all outputs, CL = 20 pF on OSC2, CL = 5pF on XOSC2. Wait, Stop IDD: All ports configured as inputs, VIL = 0.2 V, VIH = VDD–0.2 V. Stop IDD measured with XOSC1, OSC1 = VSS. Wait IDD is affected linearly by the XOSC2, OSC2 capacitance.

14.5 A/D converter characteristics for 5V operation

Table 14-5 A/D converter electrical characteristics for 5V operation

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

Characteristic	Parameter	Min	Max	Unit
Resolution	Number of bits resolved by A/D converter	8	—	Bits
Non-linearity	Maximum deviation from best straight line through the A/D transfer characteristics ($V_{RH} = V_{DD}$ and V_{RL} = 0V)	—	±0.5	LSB
Quantization error	Uncertainty due to converter resolution	—	±0.5	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale equivalent of the binary output code for all errors	—	±1.0	LSB
Zero error	Difference between the output of an ideal and actual A/D for zero input voltage	—	±0.5	LSB
Full scale error	Difference between the output of an ideal and actual A/D for full scale input voltage	—	±0.5	LSB
Conversion range	Analog input voltage range	V_{RL}	V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	V_{DD} + 0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS} - 0.1	V_{RH}	V
Conversion time	Total time to perform a single analog to digital conversion External clock Internal RC oscillator	 — —	 32 32	 t_{CYC} µs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	Guaranteed		
Zero input reading	Conversion result when $V_{IN} = V_{RL}$	$00	—	Hex
Full scale reading	Conversion result when $V_{IN} = V_{RH}$	—	$FF	Hex
Sample acquisition time(1)	Analog input acquisition sample time External clock Internal RC oscillator	 — —	 12 12	 t_{CYC} µs
Sample/hold capacitance	Input capacitance on AD0 and AD1	—	12	pF
Input leakage(2)	Input leakage on AD0 and AD1 Input leakage on V_{RL} and V_{RH}	—	1	µA

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

(2) The external system error caused by input leakage current is approximately equal to the product of R_{SOURCE} and input current. Input current to A/D channel will be dependent on external source impedance (see Figure 7-2).

14

14.6 A/D converter characteristics for 3.3V operation

Table 14-6 A/D converter electrical characteristics for 3.3V operation

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

Characteristic	Parameter	Min	Max	Unit
Resolution	Number of bits resolved by A/D converter	8	—	Bits
Non-linearity	Maximum deviation from best straight line through the A/D transfer characteristics ($V_{RH} = V_{DD}$ and V_{RL} = 0V)	—	±1.0	LSB
Quantization error	Uncertainty due to converter resolution	—	±0.5	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale equivalent of the binary output code for all errors	—	±2.0	LSB
Zero error	Difference between the output of an ideal and actual A/D for zero input voltage	—	±1.0	LSB
Full scale error	Difference between the output of an ideal and actual A/D for full scale input voltage	—	±1.0	LSB
Conversion range	Analog input voltage range	V_{RL}	V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	V_{DD} + 0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS} - 0.1	V_{RH}	V
Conversion time	Total time to perform a single analog to digital conversion External clock Internal RC oscillator	 — —	 32 32	 t_{CYC} µs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	Guaranteed		
Zero input reading	Conversion result when $V_{IN} = V_{RL}$	$00	—	Hex
Full scale reading	Conversion result when $V_{IN} = V_{RH}$	—	$FF	Hex
Sample acquisition time(1)	Analog input acquisition sample time External clock Internal RC oscillator	 — —	 12 12	 t_{CYC} µs
Sample/hold capacitance	Input capacitance on AD0 and AD1	—	12	pF
Input leakage(2)	Input leakage on AD0 and AD1 Input leakage on V_{RL} and V_{RH}	—	1	µA

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

(2) The external system error caused by input leakage current is approximately equal to the product of R_{SOURCE} and input current. Input current to A/D channel will be dependent on external source impedance (see Figure 7-2).

14.7 Control timing for 5V operation

Table 14-7 Control timing

(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 Watch crystal External oscillator	 f_{OSC} f_{OSC}	 — dc	 32.768 4.2	 kHz MHz
Internal operating frequency Watch crystal External oscillator	 f_{OP} f_{OP}	 — dc	 16.384 2.1	 kHz MHz
Cycle time	t_{CYC}	480	—	ns
Ceramic resonator start-up time(1)	t_{OCOV}	—	20	ms
Ceramic resonator stop recovery time(1)	t_{ICCH}	—	20	ms
Crystal oscillator start-up time (32 kHz)(1)	t_{OXOV}	—	1000	ms
Crystal oscillator stop recovery time (32 kHz)(1)	t_{ILCH}	—	1000	ms
PLL start-up time	t_{PLLS}	—	10	ms
PLL stability (V_{DD} = 2.7 – 6.0 V, f = 1.049 MHz)		3.0	3.3	%
OSC1 pulse width	t_{OH}, t_{OL}	90	—	ns
RESET input pulse width	t_{RL}	1.5	—	t_{CYC}
EEPROM byte erase time	t_{EERA}	—	10	ms
EEPROM block erase time	t_{EERA}	—	100	ms
EEPROM bulk erase time	t_{EERA}	—	400	ms
EEPROM byte program time(2)	t_{EEPGM}	—	10	ms
RC oscillator stabilization time	t_{RCON}	—	5	µs
A/D on current stabilization time	t_{ADON}	—	500	µs
Timer Resolution(3) Input capture pulse width Input capture pulse period	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4 250 —(4)	 — — —	 t_{CYC} ns t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	250	—	ns
Interrupt pulse period	t_{ILIL}	—(5)	—	t_{CYC}

(1) These values are out with Motorola's control and are given purely as a guide. For actual values, please refer to the specifications supplied by the manufacturer of the oscillator used.

(2) For bus frequencies less than 1 MHz, the internal RC oscillator should be used when programming the EEPROM.

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

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

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

14.8 Control timing for 3.3V operation

Table 14-8 Control timing

(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 Watch crystal External oscillator	 f_{OSC} f_{OSC}	 — dc	 32.768 2.1	 kHz MHz
Internal operating frequency Watch crystal External oscillator	 f_{OP} f_{OP}	 — dc	 16.384 1.05	 kHz MHz
Cycle time	t_{CYC}	1950	—	ns
Ceramic resonator start-up time(1)	t_{OCOV}	—	20	ms
Ceramic resonator stop recovery time(1)	t_{ICCH}	—	20	ms
Crystal oscillator start-up time (32 kHz)(1)	t_{OXOV}	—	1000	ms
Crystal oscillator stop recovery time (32 kHz)(1)	t_{ILCH}	—	1000	ms
PLL start-up time	t_{PLLS}	—	10	ms
PLL stability (V_{DD} = 2.7 – 6.0 V, f = 1.049 MHz)		3.0	3.3	%
OSC1 pulse width	t_{OH}, t_{OL}	200	—	ns
RESET input pulse width	t_{RL}	1.5	—	t_{CYC}
EEPROM byte erase time	t_{EERA}	—	20	ms
EEPROM block erase time	t_{EERA}	—	200	ms
EEPROM bulk erase time	t_{EERA}	—	400	ms
EEPROM byte program time(2)	t_{EEPGM}	—	20	ms
RC oscillator stabilization time	t_{RCON}	—	5	µs
A/D on current stabilization time	t_{ADON}	—	100	µs
Timer Resolution(3) Input capture pulse width Input capture pulse period	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4 500 —(4)	 — — —	 t_{CYC} ns t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	500	—	ns
Interrupt pulse period	t_{ILIL}	—(5)	—	t_{CYC}

(1) These values are out with Motorola's control and are given purely as a guide. For actual values, please refer to the specifications supplied by the manufacturer of the oscillator used.

(2) For bus frequencies less than 1 MHz, the internal RC oscillator should be used when programming the EEPROM.

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

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

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

15
MECHANICAL DATA

15.1 64-pin quad flat pack (QFP) pinout

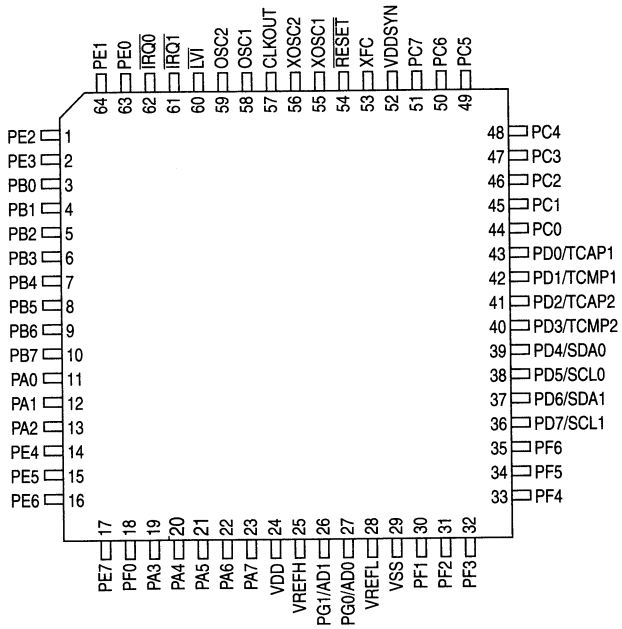
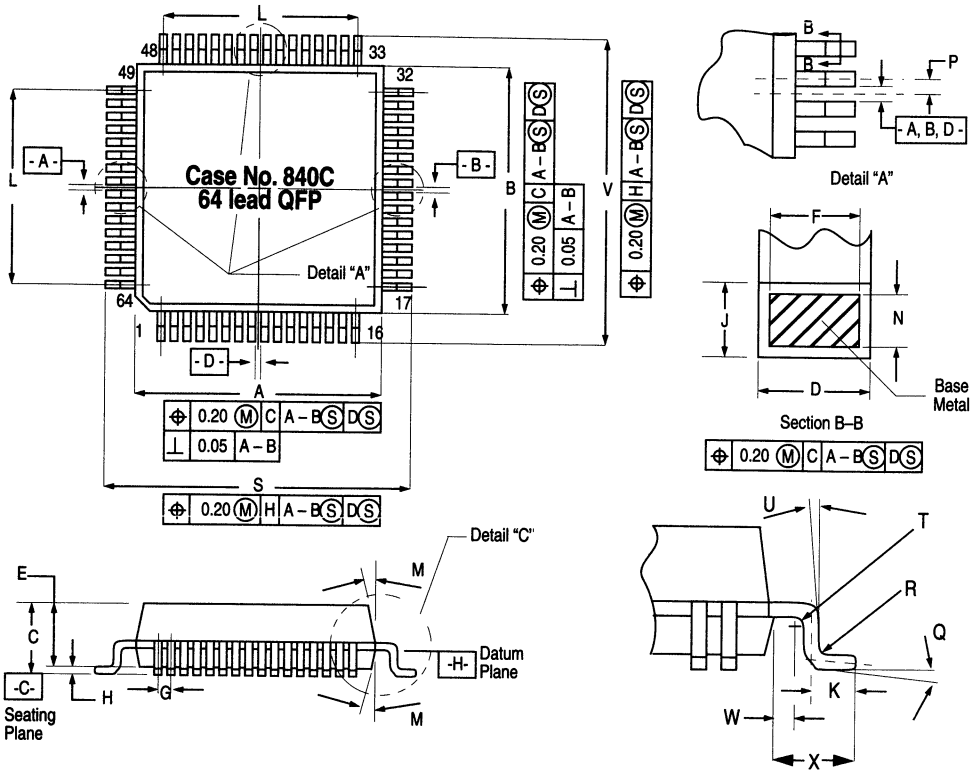


Figure 15-1 64-pin QFP pinout

15

15.2 64-pin quad flat pack (QFP) mechanical dimensions



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

Notes

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

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

Figure 15-2 64-pin QFP mechanical dimensions

15.3 44-pin quad flat pack (QFP) pinout

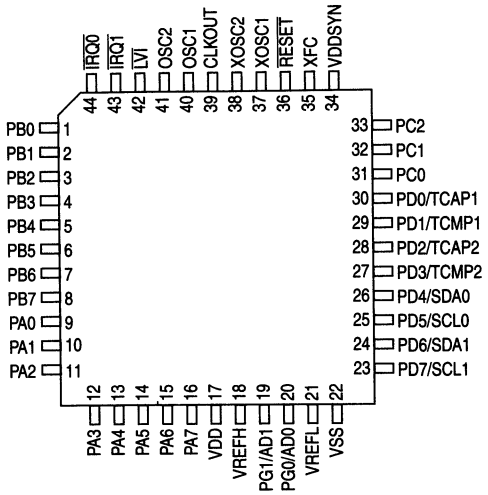


Figure 15-3 44-pin QFP pinout

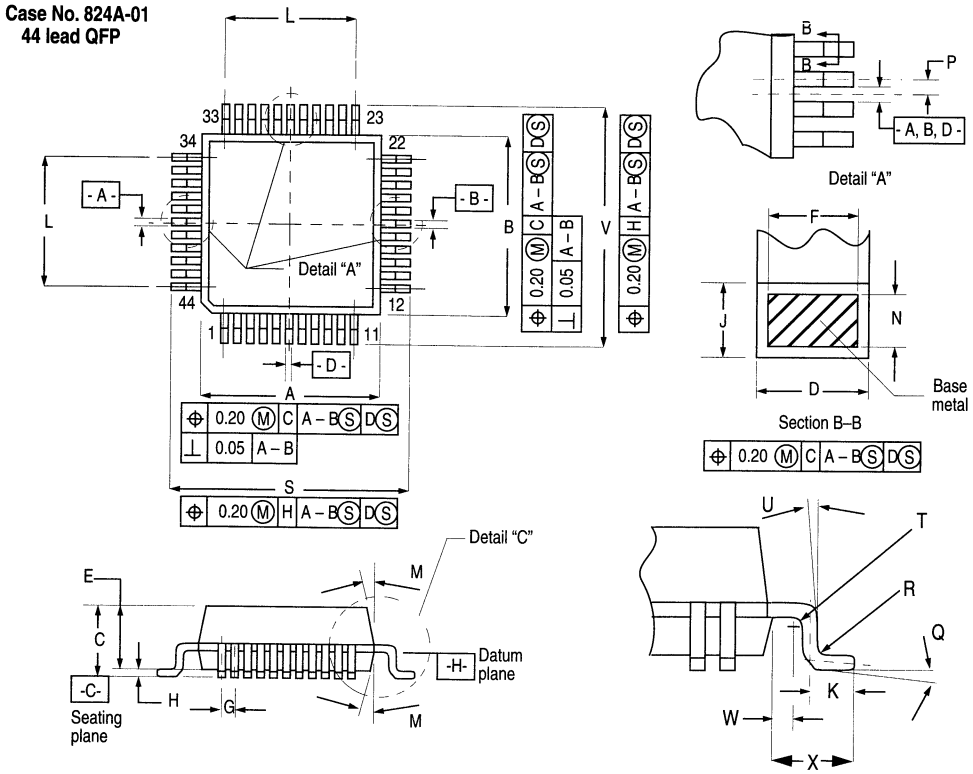
The following pins are not available in the 44-pin QFP version:

PE7 – PE0

PF6 – PF0

PC7 – PC3

15.4 44-pin quad flat pack (QFP) mechanical dimensions



Dim.	Min.	Max.
A	9.90	10.10
B	9.90	10.10
C	2.10	2.45
D	0.30	0.45
E	2.00	2.10
F	0.30	0.40
G	0.80 BSC	
H	—	0.250
J	0.130	0.230
K	0.65	0.95
L	8.00 REF	

Notes

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

Dim.	Min.	Max.
M	5°	10°
N	0.130	0.170
Q	0°	7°
R	0.13	0.30
S	12.95	13.45
T	0.13	—
U	0°	—
V	12.95	13.45
W	0.40	—
X	1.6 REF	

Figure 15-4 44-pin QFP mechanical dimensions

15.5 56-pin shrink dual in line package (SDIP)

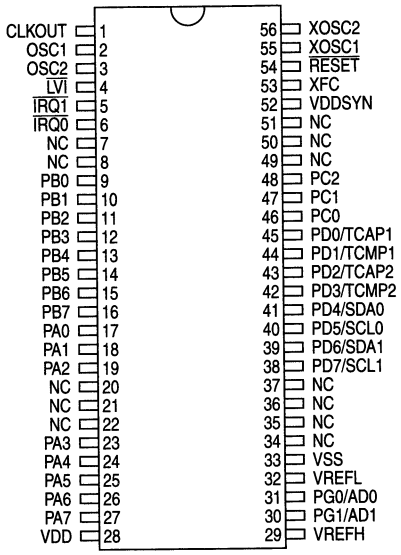


Figure 15-5 56-pin SDIP pinout

The following pins are not available in the 56-pin SDIP version:

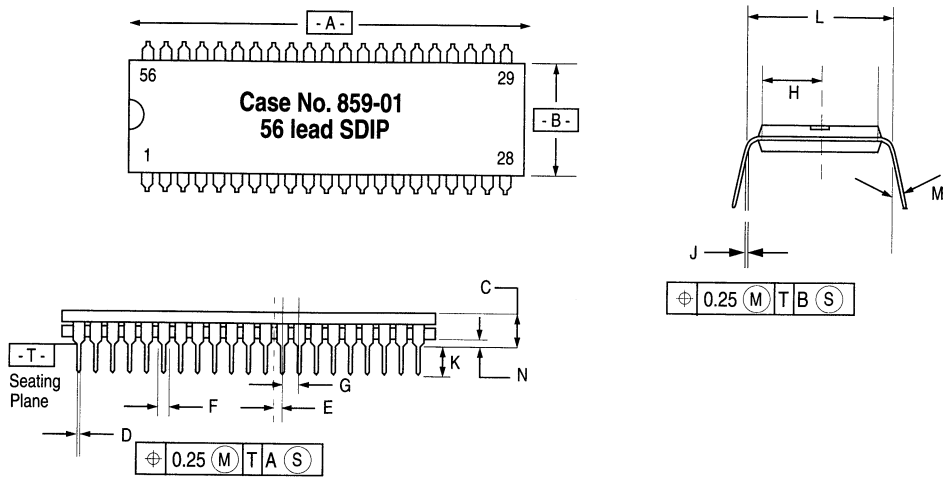
PE7 – PE0

PF6 – PF0

PC7 – PC3

15

15.6 56-pin shrink dual in line package (SDIP)



Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	51.69	52.45	1. **Due to space limitations, this case shall be represented by a general case outline, rather than one showing all the leads.** 2. Dimensions and tolerancing per ANSI Y 14.5 1982. 3. All dimensions in mm. 4. Dimension L to centre of lead when formed parallel. 5. Dimensions A and B do not include mould flash. Allowable mould flash is 0.25 mm.	H	7.62 BSC	
B	13.72	14.22		J	0.20	0.38
C	3.94	5.08		K	2.92	3.43
D	0.36	0.56		L	15.24 BSC	
E	0.89 BSC			M	0°	15°
F	0.81	1.17		N	0.51	1.02
G	1.778 BSC					

Figure 15-6 56-pin SDIP mechanical dimensions

16
ORDERING INFORMATION

This section describes the information needed to order the MC68HC05E16.

Table 16-1 MC order numbers

Device title	Package type	Temperature	Part number
MC68HC05E16	64-pin QFP	0 to +70°C	MC68HC05E16FU
		–40 to +85°C	MC68HC05E16CFU
	44-pin QFP	0 to +70°C	MC68HC05E16FB
		–40 to +85°C	MC68HC05E16CFB
	56-pin SDIP	0 to +70°C	MC68HC05E16B
		–40 to +85°C	MC68HC05E16CB

To initiate a ROM pattern for the MC68HC05E16, it is necessary to first 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 16-1 for appropriate part numbers.

16.1 EPROM

A 16K byte 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.

16.2 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.

16.3 ROM verification units (RVU)

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

GLOSSARY

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

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

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

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

THIS PAGE LEFT BLANK INTENTIONALLY

INDEX

In this index numeric entries are placed first; page references in *italics* indicate that the reference is to a figure.

J

K

L

M

N

O

P

R

S

T

V

W

X

Z

THIS PAGE LEFT BLANK INTENTIONALLY

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC05E16/D)

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

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

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

Comments: _____

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

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

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

Completely ☐ ☐ ☐ ☐ Not at all
Comments: _____

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

Easy ☐ ☐ ☐ ☐ Difficult
Comments: _____

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

	Too little detail				Too much detail
SECTION 1 INTRODUCTION	☐	☐	☐	☐	☐
SECTION 2 MODES OF OPERATION AND PIN DESCRIPTIONS	☐	☐	☐	☐	☐
SECTION 3 MEMORY AND REGISTERS	☐	☐	☐	☐	☐
SECTION 4 INPUT/OUTPUT PORTS	☐	☐	☐	☐	☐
SECTION 5 16-BIT PROGRAMMABLE TIMER	☐	☐	☐	☐	☐
SECTION 6 I^2C-BUS	☐	☐	☐	☐	☐
SECTION 7 A/D CONVERTER	☐	☐	☐	☐	☐
SECTION 8 CORE TIMER	☐	☐	☐	☐	☐
SECTION 9 CUSTOM PERIODIC INTERRUPT	☐	☐	☐	☐	☐
SECTION 10 RESETS AND INTERRUPTS	☐	☐	☐	☐	☐
SECTION 11 PHASE-LOCKED LOOP	☐	☐	☐	☐	☐
SECTION 12 OSCILLATOR SYSTEM	☐	☐	☐	☐	☐
SECTION 13 CPU CORE AND INSTRUCTION SET	☐	☐	☐	☐	☐
SECTION 14 ELECTRICAL SPECIFICATIONS	☐	☐	☐	☐	☐
SECTION 15 MECHANICAL DATA	☐	☐	☐	☐	☐
SECTION 16 ORDERING INFORMATION	☐	☐	☐	☐	☐

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

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

8. How could we improve this document? _____

_____

_____

9. How would you rate Motorola's documentation?

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

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

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

12. How many years have you worked with microprocessors?

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

– Second fold back along this line –

By air mail
Par avion

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

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

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

F.A.O. Technical Publications Manager
(re: MC68HC05E16/D)

MOTOROLA
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

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

_____

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

Name: _____ Phone No: _____

Position: _____ FAX No: _____

Department: _____

Company: _____

Address: _____

_____

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

– Finally, tuck this edge into opposite flap –

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@freescalè.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

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

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

1ATX35765-0 PRINTED IN USA 8/97 IMPERIAL LITHO 28803 1,500 CSIC YGALAA UCCL

MC68HC05E16/D