

**MC68HC05E6
MC68HC705E6
Rev. 1.0**

**HCMOS Microcontroller Unit
TECHNICAL DATA**

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

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

List of Sections

| | |
|---|-----|
| List of Sections | 3 |
| Table of Contents. | 5 |
| General Description | 9 |
| Modes of Operation and Pin Descriptions | 13 |
| Memory and Registers | 25 |
| Input/Output Ports | 35 |
| Core Timer | 45 |
| Programmable Timer. | 51 |
| A-to-D Converter. | 65 |
| Resets and Interrupts. | 73 |
| Central Processing Unit. | 85 |
| Electrical Specifications | 105 |
| Mechanical Data | 117 |
| Ordering Information. | 121 |
| Glossary | 125 |
| Literature Updates | 137 |

Table of Contents

| | | |
|---|--------------------------------|----|
| General Description | Contents | 9 |
| | Introduction | 9 |
| | Features | 10 |
| | Mask options | 11 |
| Modes of Operation and Pin Descriptions | Contents | 13 |
| | Modes of operation | 13 |
| | Pin descriptions | 17 |
| | Low power modes | 22 |
| Memory and Registers | Contents | 25 |
| | Introduction | 25 |
| | Registers | 25 |
| | RAM | 26 |
| | Bootloader ROM | 26 |
| | ROM (MC68HC05E6 only) | 26 |
| | EPROM (MC68HC705E6 only) | 26 |
| | EEPROM | 28 |
| Input/Output Ports | Contents | 35 |
| | Introduction | 35 |
| | Input/output programming | 36 |
| | Port A | 37 |
| | Port B | 37 |
| | Port C | 37 |
| | Port D | 41 |
| | Port G | 41 |
| | Port registers | 42 |

Table of Contents

| | |
|--------------------------|--|
| Core Timer | <ul style="list-style-type: none"> Contents45 Introduction45 Real time interrupts (RTI)47 Computer operating properly (COP) watchdog timer47 Core timer registers48 Core timer during WAIT50 Core timer during STOP50 |
| Programmable Timer | <ul style="list-style-type: none"> Contents51 Introduction51 Keyboard/timer register (KEY/TIM)52 Counter52 Timer functions55 Timer during WAIT mode60 Timer during STOP mode60 Timer state diagrams60 |
| A-to-D Converter | <ul style="list-style-type: none"> Contents65 Introduction65 A/D converter operation66 A/D registers68 A/D converter during WAIT mode70 A/D converter during STOP mode71 A/D analog input71 |
| Resets and Interrupts | <ul style="list-style-type: none"> Contents73 Resets73 Interrupts75 Nonmaskable software interrupt (SWI)78 Maskable hardware interrupts78 Hardware controlled interrupt sequence83 |

| | | |
|---------------------------|--|-----|
| Central Processing Unit | Contents | 85 |
| | Introduction | 86 |
| | CPU Registers | 86 |
| | Arithmetic/Logic Unit (ALU) | 89 |
| | Instruction Set Overview | 90 |
| | Addressing Modes | 90 |
| | Instruction Types | 93 |
| | Instruction Set Summary | 98 |
| Electrical Specifications | Contents | 105 |
| | Introduction | 105 |
| | Maximum ratings | 106 |
| | Thermal characteristics and power considerations | 107 |
| | DC electrical characteristics | 108 |
| | AC electrical characteristics | 111 |
| | A/D converter electrical characteristics | 114 |
| Mechanical Data | Contents | 117 |
| | Introduction | 117 |
| Ordering Information | Contents | 121 |
| | Introduction | 121 |
| | EPROMS | 123 |
| | Verification media | 123 |
| Literature Updates | Literature Distribution Centers | 137 |
| | Customer Focus Center | 138 |
| | Mfax | 138 |
| | Motorola SPS World Marketing World Wide Web Server | 138 |
| | Microcontroller Division's Web Site | 138 |

General Description

Contents

| | |
|--------------------|----|
| Introduction | 9 |
| Features | 10 |
| Mask options | 11 |

Introduction

The MC68HC05E6 is a member of the M68HC05 family of HCMOS microcomputers. Features of the MC68HC05E6 include 6K bytes of user ROM, 160 bytes of EEPROM, a keyboard interface, an A/D converter and a 16-bit timer. The device is further enhanced by a core timer which enables real time interrupts at, for example, 8ms intervals and a watchdog facility which guards against CPU 'runaway'. The on-board EEPROM facilitates storage of alterable, non-volatile, user specified data, such as personalisation data or control parameters. The features highlighted make the part ideally suited to various control applications, while the low voltage design and low cost option ensure that it can also be used in telecommunications applications such as telephone handsets.

For maximum I/O capability, the 44-pin version of the MC68HC05E6 is recommended, however for cost-sensitive applications a 28-pin version is also available.

NOTE: *The entire data sheet applies to the 44-pin version of the device, however some of the features are not available to the user when the part is bonded in the 28-pin package (see [Figure 1](#)).*

The MC68HC705E6 is an EPROM version of the MC68HC05E6, with the user ROM replaced by a similar amount of EPROM. All references to the MC68HC05E6 apply equally to the MC68HC705E6, unless otherwise noted.

General Description

References specific to the MC68HC705E6 are italicised in the text.

NOTE: *Information given for the MC68HC705E6 cannot be guaranteed. All values are design targets only and may change before the device is qualified.*

Features

- Fully static design featuring the industry-standard M68HC05 core
- On-chip oscillator with crystal or ceramic resonator clock options
- 6000 bytes of user ROM plus 16 bytes of user vectors (MC68HC05E6); *6144 bytes of user EPROM plus 16 bytes of user vectors (MC68HC705E6)*
- 240 bytes of bootloader ROM (MC68HC705E6)
- 160 bytes of EEPROM
- 128 bytes of RAM
- Single supply voltage (no external voltage required for EEPROM programming)
- 4-channel, 8-bit A/D converter
- 16-bit programmable timer with input capture and output compare
- 15-stage multipurpose core timer with overflow, watchdog and real-time interrupt
- 32 programmable bidirectional I/O lines and four input-only lines including eight fixed wire pull-ups (port A), eight fixed wire pull-downs (port B) and 16 software selectable pull-ups (ports C and D)
- Keyboard interrupt facility available on all pins of port C
- Low voltage indicator (LVI)
- Power saving STOP and WAIT modes
- Available in 44-pin QFP and 28-pin SOIC packages; four of the ports are not fully bonded out in the 28-pin package (see [Figure 1](#))

Mask options

There are two mask options on the MC68HC05E6 which are programmed during manufacture and therefore must be specified on the order form; COP watchdog timer enable/disable and STOP instruction enable/disable.

LVI/options register (LVI/OPT)

In addition, the $\overline{\text{IRQ}}$ pin can be configured to be either edge or edge-and-level sensitive. This is done using the IRQ bit in the LVI/options register at \$0F.

There are two options on the MC68HC705E6 which are programmed using configuration bits in the LVI/options register; COP watchdog timer enable/disable and edge or edge-and-level sensitive IRQ triggering. The STOP instruction is permanently enabled on the MC68HC705E6.

| | 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 | | | COP | IRQ | 0u00 0u00 |

COP — Computer operating properly watchdog enable/disable
 1 = COP watchdog disabled.
 0 = COP watchdog enabled.

Reset clears this bit, thus the COP watchdog is enabled automatically after reset. Because of the 'write-once' nature of the COP bit, it is recommended that it be written to immediately after reset to lock the desired state.

IRQ — Interrupt triggering sensitivity
 1 = IRQ is negative edge-and-level sensitive
 0 = IRQ is negative edge sensitive only
 Reset clears the IRQ bit.

NOTE: *The bits in the LVIOPT register which are shaded are not relevant to this section of the data sheet. These bits are described in [Resets and Interrupts](#).*

General Description

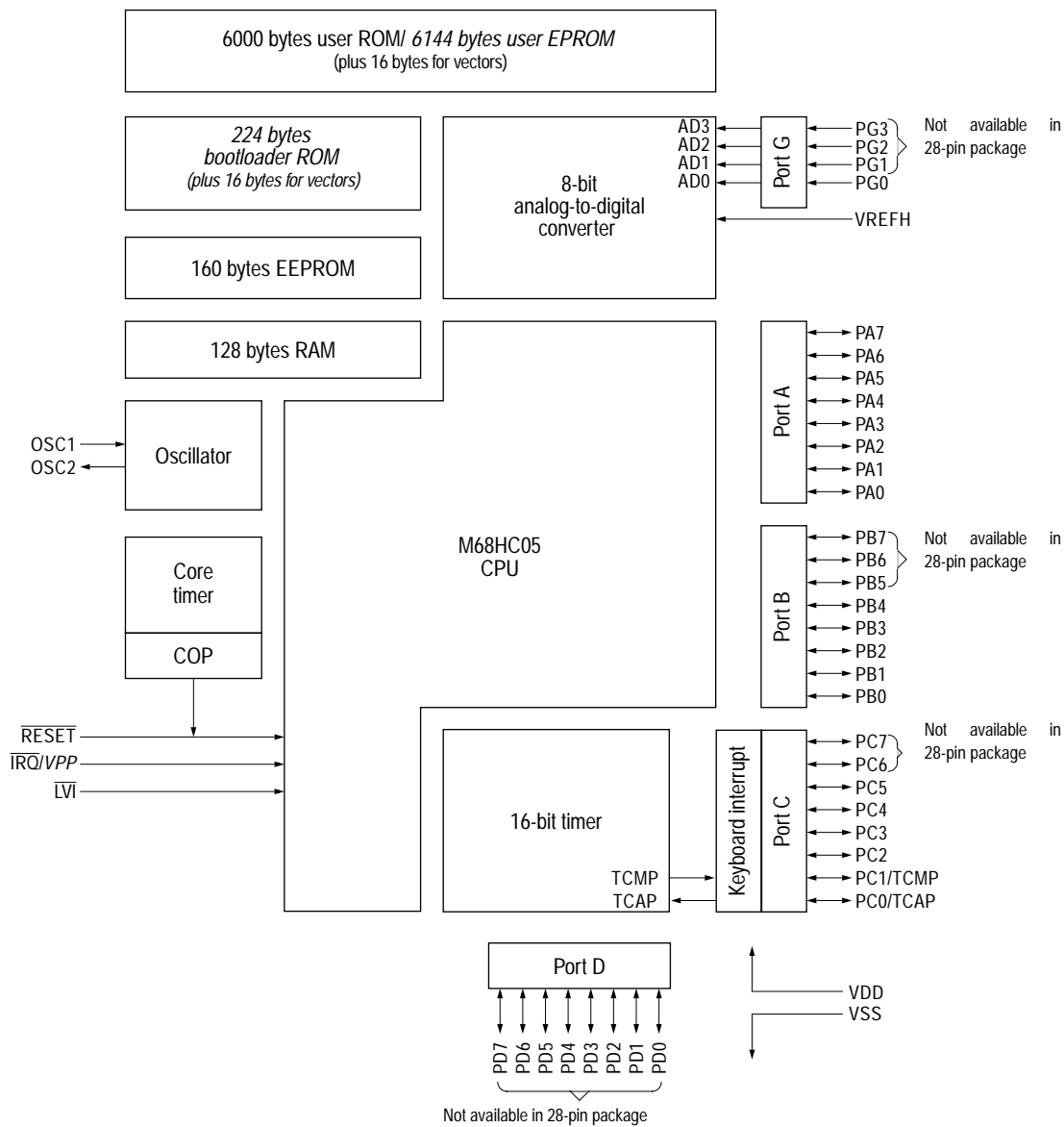


Figure 1 MC68HC05E6 and MC68HC705E6 block diagram

Modes of Operation and Pin Descriptions


Contents

| | |
|--------------------------|----|
| Modes of operation | 13 |
| Pin descriptions | 17 |
| Low power modes | 17 |

Modes of operation

The MC68HC05E6 operates in single chip mode only. *The MC68HC705E6 has two modes of operation; single chip and EPROM bootloader mode. Table 1 shows the conditions required to enter the EPROM modes of operation on the rising edge of RESET.*

Table 1 MC68HC705E6 operating mode entry conditions

| RESET | IRQ/VPP | PC4 | PC3 | PC2 | Mode | |
|---|------------------------------------|-----|-----|-----|------------------|-----------------|
|  | V _{SS} to V _{DD} | x | x | x | Single chip | |
| | 2V _{DD} | 1 | 0 | 1 | EPROM bootloader | Verify only |
| | | 1 | 1 | 0 | | Program 1 byte |
| | | 1 | 1 | 1 | | Program 4 bytes |

x = don't care

Single chip mode

This is the normal operating mode of the MC68HC05E6. In this mode the device functions as a self-contained microcomputer (MCU) with all on-board peripherals, including the 8-bit I/O ports (A, B, C and D) and the 4-bit input-only port (G), available to the user. All address and data activity occurs within the MCU.

Single chip mode is entered on the rising edge of RESET if the voltage level on the IRQ pin is within the normal operating range.

Modes of Operation and Pin Descriptions

NOTE: *For the MC68HC705E6, all vectors are fetched from EPROM in single chip mode; therefore, the EPROM must be programmed (via the bootloader mode) before the device is powered up in single chip mode.*

**EPROM
bootloader mode
for the
MC68HC705E6**

This mode is used for programming the on-board EPROM. The pin assignments are identical to those of single chip mode (see [Figure 3](#)). Because the addresses in the MC68HC705E6 and the external EPROM containing the user code are incremented independently, it is essential that the data layout in the 27256 EPROM conforms exactly to the MC68HC705E6 memory map.

The bootloader uses two external latches to address the memory device containing the code to be copied. [Figure 2](#) shows a suggested EPROM programming circuit.

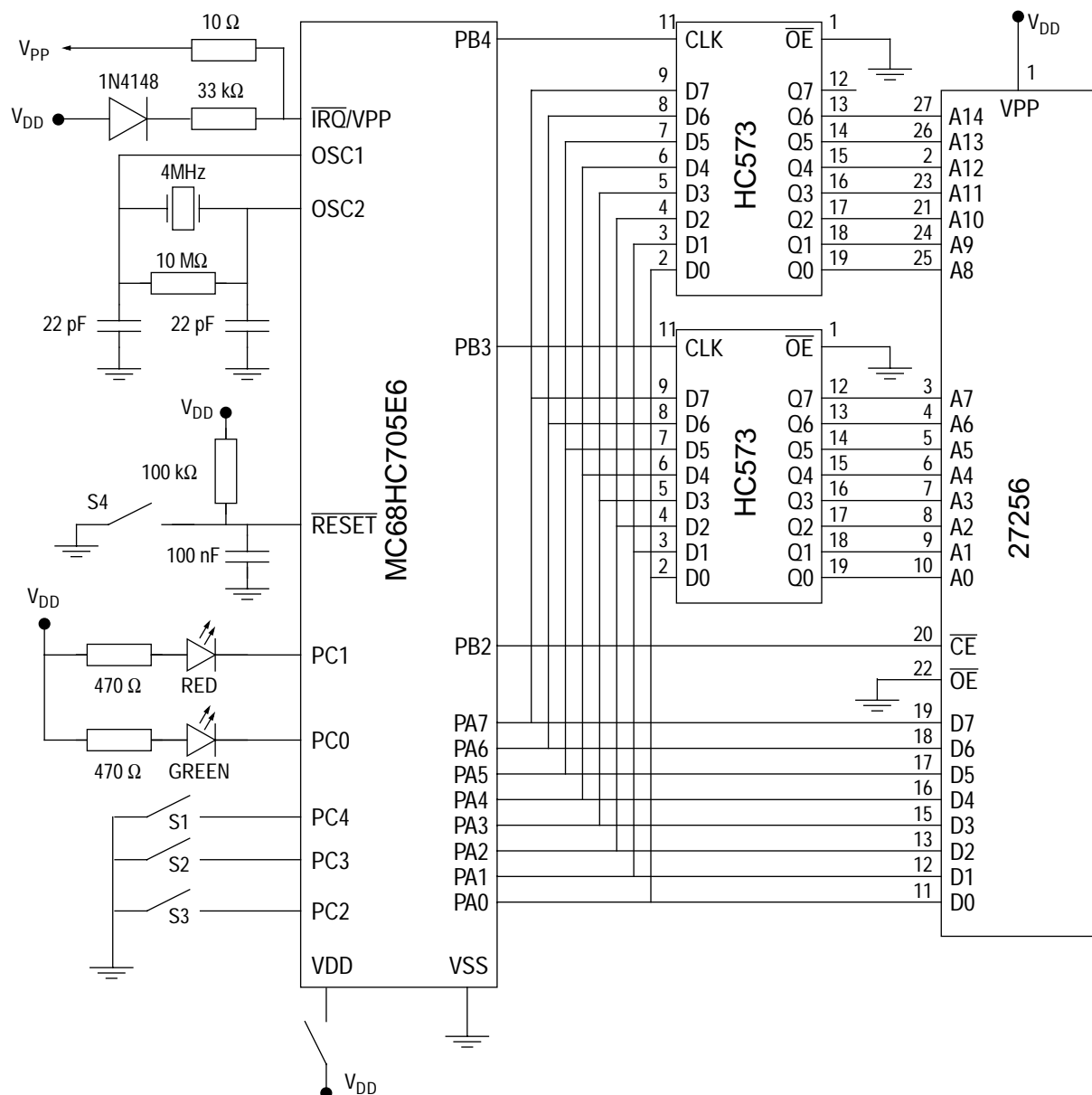


Figure 2 EPROM programming circuit

NOTE: This mode must not be used in the user's application.

Modes of Operation and Pin Descriptions

Bootloader functions

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

Two pins, PC0 and PC1, are used to drive the PROG and VERF LED outputs. While the EPROM is being programmed, the PROG LED lights up; when programming is complete, the internal EPROM contents are compared to that of the external EPROM and, if they match exactly, the VERF LED lights up.

The EPROM must be in the erased state before performing a program cycle. The erased state of the EPROM is \$00.

EPROM programming instructions

The following procedure should be carried out when programming the EPROM. In order to prevent damage to the device, V_{DD} should always be applied to the part before V_{PP} when applying power. Similarly, V_{PP} should be removed from the part before V_{DD} when switching the power off.

1. Turn off power to the circuit.
2. Install the MCU and the EPROM.
3. Select the bootloader function:
 Program 1 byte: Open S1 and S2 and close S3.
 Program 4 bytes: Open S1, S2 and S3.
 Verify only: Open S1 and S3 and close S2.
4. Close switch S4 to hold the MCU in reset.
5. Apply V_{DD} to the circuit.
6. Apply the EPROM programming voltage (V_{PP}) to the circuit.
7. Open switch S4 to take the MCU out of reset.
 During programming the PROG LED turns on and is switched off when the verification routine begins. If verification is successful, the VERF LED turns on. If the bootloader finds an error during verification, it puts the error address on the external address bus and stops running.
8. Close switch S4 to hold the MCU in reset.
9. Remove the V_{PP} voltage.
10. Remove the V_{DD} voltage.

NOTE: *EPROM programming must be carried out at room temperature.*

Pin descriptions

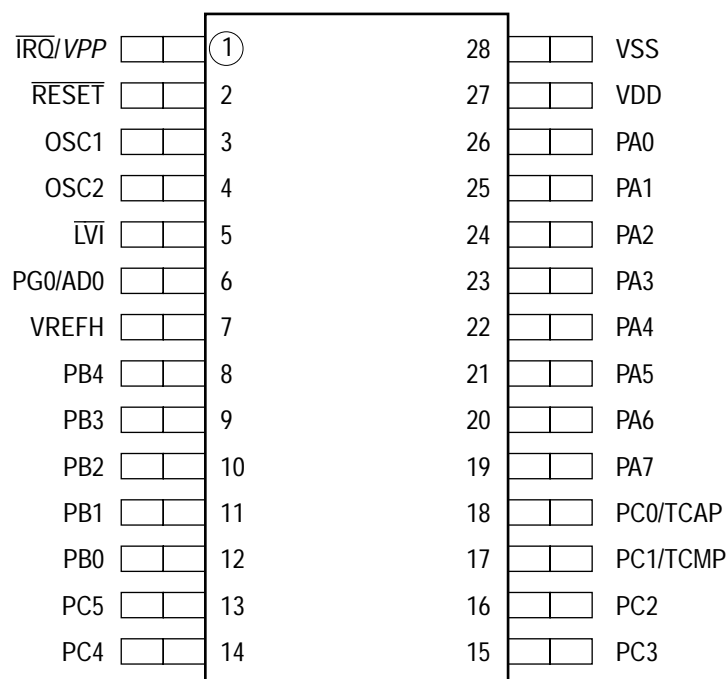


Figure 3 28-pin SOIC pinout

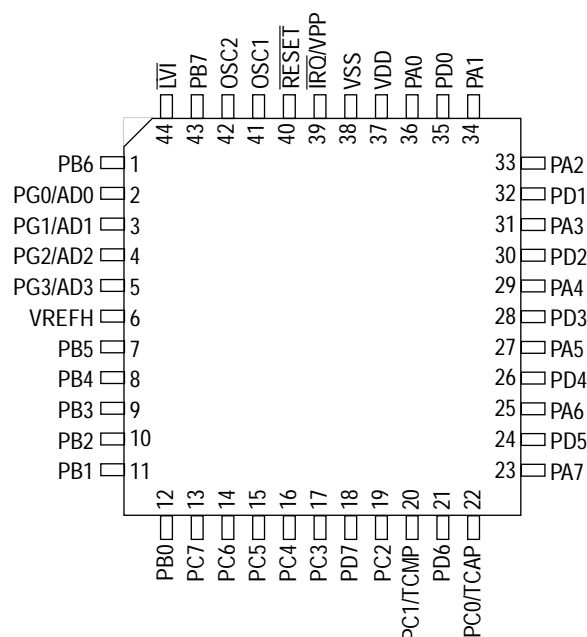


Figure 4 44-pin QFP pinout

Modes of Operation and Pin Descriptions

VDD and VSS

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

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

$\overline{\text{IRQ}}$ / VPP

This is an input-only pin for external interrupt sources. Interrupt triggering can be selected to be negative edge sensitive or negative edge-and-level sensitive by correctly configuring the IRQ bit in the LVIOPT register (see [Mask options](#)). The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. *For the MC68HC705E6, this pin also serves as the input pin for the EPROM programming voltage (VPP).*

OSC1, OSC2

These pins provide control input for an on-chip oscillator circuit. A crystal, ceramic resonator or external clock signal connected to these pins supplies the oscillator clock. The oscillator frequency (f_{OSC}) is divided by two to give the internal bus frequency (f_{OP}).

Crystal

The circuit shown in [Figure 5\(a\)](#) is recommended when using either a crystal or a ceramic resonator. [Figure 5\(e\)](#) provides the recommended capacitance and feedback resistance values. The internal oscillator is designed to interface with an AT-cut parallel-resonant quartz crystal resonator in the frequency range specified for f_{OSC} (see [Table 23](#)). Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and associated components should be mounted as close as possible to the input pins to minimise output distortion and start-up stabilization time. The manufacturer of the particular crystal being considered should be consulted for specific information.

Ceramic resonator

A ceramic resonator may be used instead of a crystal in cost sensitive applications. The circuit shown in [Figure 5\(a\)](#) is recommended when using either a crystal or a ceramic resonator. [Figure 5\(e\)](#) lists the recommended capacitance and feedback resistance values. The manufacturer of the particular ceramic resonator being considered should be consulted for specific information.

External clock

An external clock should be applied to the OSC1 input, with the OSC2 pin left unconnected, as shown in [Figure 5\(c\)](#). The t_{OCoV} specification (see [Table 23](#)) does not apply when using an external clock input. The equivalent specification of the external clock source should be used in lieu of t_{OCoV} .

\overline{LVI}

This active low input pin is used to indicate a drop in supply voltage below a useful operating level. Driving this pin low initiates either a pre-defined software routine or a hardware interrupt (see [Low voltage indicator interrupt](#)). The \overline{LVI} pin contains an internal Schmitt trigger to improve noise immunity.

\overline{RESET}

This active low input-only pin is used to reset the MCU. Applying a logic zero to this pin forces the device to a known start-up state. The \overline{RESET} pin has an internal Schmitt trigger to improve noise immunity.

PA0–PA7

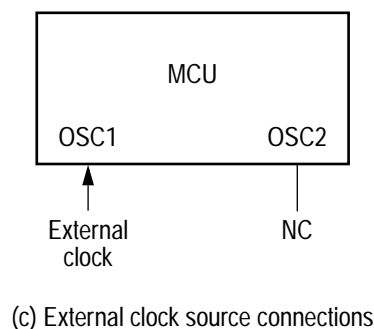
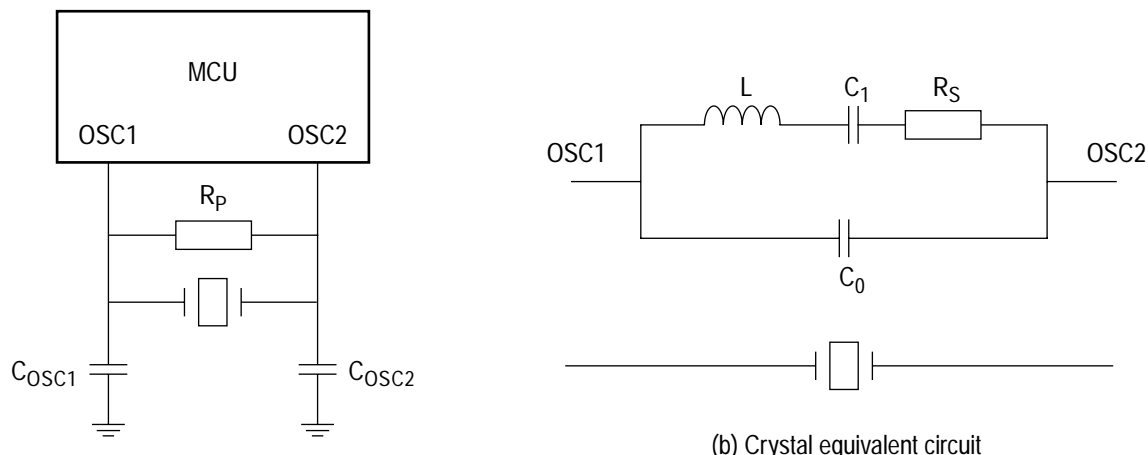
These eight I/O lines comprise port A. The state of any pin is software programmable, and all the pins are configured as inputs with fixed pull-up resistors during power-on or reset.

PB0–PB7

These eight pins comprise port B. The state of any pin is software programmable, and all the pins are configured as inputs with fixed pull-down resistors during power-on or reset.

NOTE: *PB5–PB7 are not bonded out in the 28-pin package.*

Modes of Operation and Pin Descriptions



| Crystal | | | |
|-------------------|---------|---------|------------|
| | 2MHz | 4MHz | Unit |
| $R_S(\text{max})$ | 400 | 75 | Ω |
| C_0 | 5 | 7 | pF |
| C_1 | 8 | 12 | fF |
| C_{OSC1} | 15 – 40 | 15 – 30 | pF |
| C_{OSC2} | 15 – 30 | 15 – 25 | pF |
| R_P | 10 | 10 | M Ω |
| Q | 30 000 | 40 000 | — |

| Ceramic resonator | | |
|-------------------|----------|------------|
| | 2 – 4MHz | Unit |
| $R_S(\text{typ})$ | 10 | Ω |
| C_0 | 40 | pF |
| C_1 | 4.3 | pF |
| C_{OSC1} | 30 | pF |
| C_{OSC2} | 30 | pF |
| R_P | 1 – 10 | M Ω |
| Q | 1250 | — |

(e) Crystal and ceramic resonator parameters

Figure 5 Oscillator connections

PC0–PC7

These eight port pins comprise port C. The state of any pin is software programmable, and all the pins can be configured as inputs with pull-up resistors or open-drain outputs using the data direction register and configuration register. During power-on or reset, all port C registers are cleared, thereby returning the port pins to normal inputs with pull-up resistors. In addition to their normal I/O functions, PC0 is shared with the input capture function (TCAP) of the programmable timer and PC1 with the output compare function (TCMP); see [Port C](#). PC0–PC7 are used to provide a keyboard interrupt facility when configured as inputs and not used for timer functions. They contain an internal Schmitt trigger as part of their input to improve noise immunity.

NOTE: *PC6 and PC7 are not available in the 28-pin package.*

PD0–PD7

These eight port pins comprise port D. The state of any pin is software programmable, and all the pins can be configured as inputs with pull-up resistors or open-drain outputs using the configuration register. During power-on or reset, all port D registers are cleared, thereby returning the port pins to normal inputs with pull-up resistors.

NOTE: *PD0–PD7 are not bonded out in the 28-pin package.*

PG0–PG3

These four input-only port pins comprise port G. In addition to their normal input functions, the pins are shared with the A/D converter subsystem where they are used as the digital input pins AD0–AD3 (see [A-to-D Converter](#)).

NOTE: *PG1–PG3 are not bonded out in the 28-pin package; the A/D converter has only one input channel.*

VREFH

This pin provides the high voltage reference for the A/D converter. The low voltage reference (VREFL) is tied to VSS internally.

Low power modes

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.

The I-bit in the CCR is cleared to enable external interrupts. All other registers, the remaining bits in the CTCSR, and memory contents remain unaltered. All input/output lines remain unchanged. The processor can be brought out of STOP mode only by an external interrupt, a keyboard interrupt (if enabled), an LVI interrupt (if enabled), or a reset (see [Figure 6](#)). On the MC68HC05E6, the STOP instruction can be disabled using a mask option, in which case it is executed as a no operation (NOP).

NOTE: *When exiting STOP mode there is a 4064 cycle delay before normal operation resumes.*

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.

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 interrupts may be enabled to allow a periodic exit from WAIT mode. See [Figure 6](#).

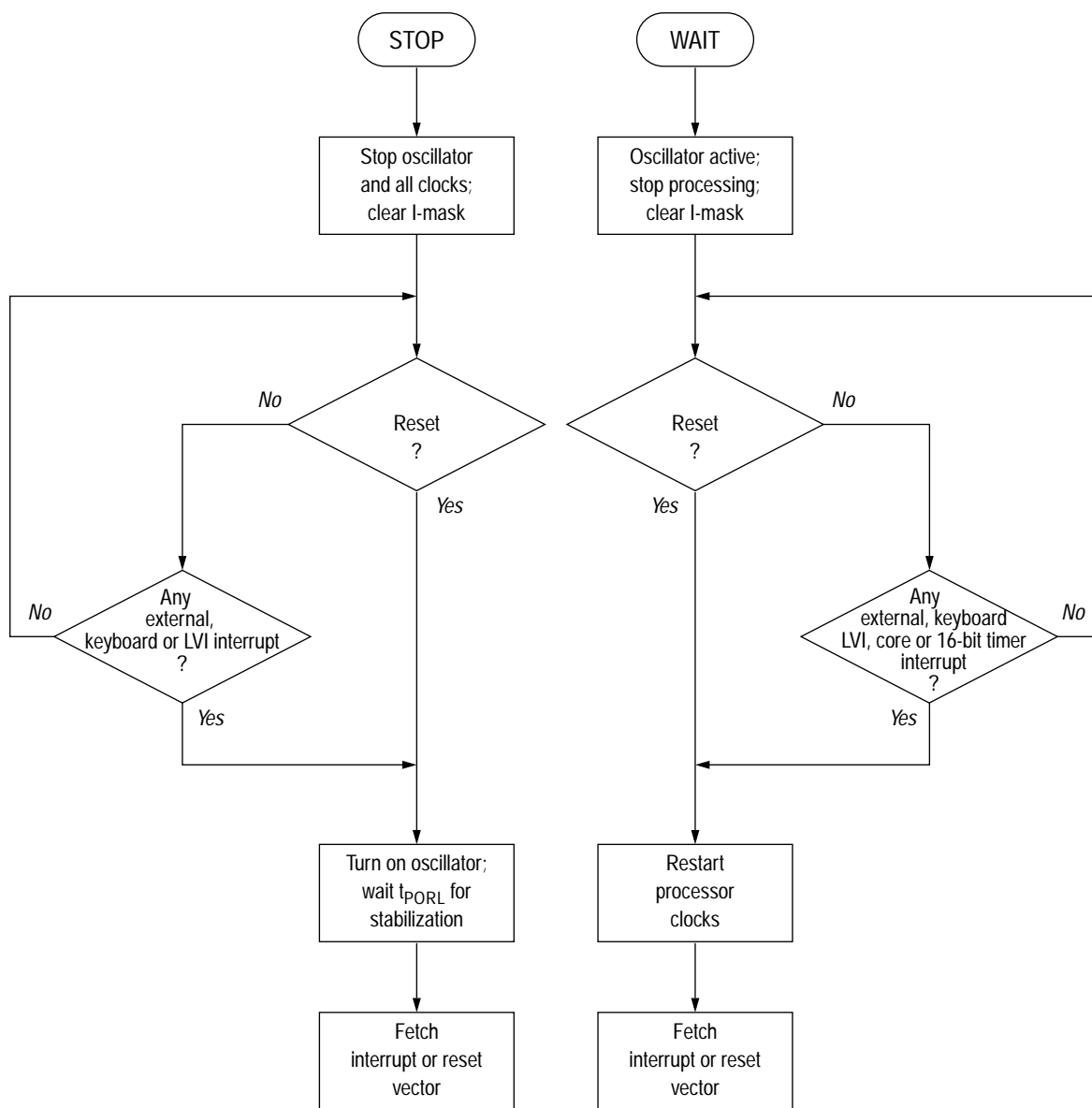


Figure 6 STOP and WAIT flowcharts

Memory and Registers

Contents

Introduction 25

Registers 25

RAM 26

Bootloader ROM 26

ROM (MC68HC05E6 only) 26

EPROM (MC68HC705E6 only) 26

EEPROM 28

Introduction

The MC68HC05E6 has an 8K byte memory map consisting of registers (for I/O, control and status), user RAM, user ROM or EPROM, EEPROM and reset and interrupt vectors as shown in [Figure 7](#). *In addition to the above, the MC68HC705E6 also has an area of bootloader ROM.*

Registers

All the I/O, control and status registers of the MC68HC(7)05E6 are contained within the first 32-byte block of the memory map, as detailed in [Table 3](#).

RAM

The user RAM consists of 128 bytes of memory, from \$0080 to \$00FF. This is shared with a 64-byte stack area. The stack begins at \$00FF and may extend down to \$00C0.

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

Bootloader ROM

The MC68HC705E6 has 224 bytes of bootloader ROM which ranges from \$1F00 to \$1FDF. This program contains code to program the EPROM by copying from a 27256 EPROM master device.

NOTE: *The bootloader ROM is not accessible if the ELATCH bit in the EPROM programming register (\$1D) is set.*

ROM (MC68HC05E6 only)

The MC68HC05E6 has 6000 bytes of ROM located from \$0800 to \$1F6F plus 16 bytes of user vectors from \$1FF0 to \$1FFF.

EPROM (MC68HC705E6 only)

The MC68HC705E6 has 6144 bytes of EPROM located from \$0700 to \$1EFF, plus 16 bytes of user vectors from \$1FF0 to \$1FFF. Four bytes of EPROM can be programmed simultaneously by correctly manipulating the bits in the EPROM programming register.

EPROM programming register (PROG)

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|--------------------------|---------|-------|-------|-------|-------|-------|--------|-------|-------|-------------------|
| EPROM programming (PROG) | \$001D | 0 | 0 | 0 | 0 | 0 | ELATCH | 0 | EPGM | 0000 0000 |

ELATCH — EPROM latch control

- 1 = EPROM address and data buses configured for programming.
- 0 = EPROM address and data buses configured for normal reads.

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

EPGM — EPROM program control

- 1 = Programming power connected to the EPROM array.
- 0 = Programming power disconnected from the EPROM array.

NOTE: *ELATCH and EPGM cannot be set on the same write operation. EPGM can only be set if ELATCH is set. EPGM is automatically cleared when ELATCH is cleared.*

EPROM programming procedure

The following steps should be taken to program a byte of EPROM:

1. Apply the programming voltage V_{PP} to the IRQ/VPP pin.
2. Set the ELATCH bit.
3. Write to the EPROM address.
4. Set the EPGM bit for a time t_{EPGM} to apply the programming voltage.
5. Clear the ELATCH bit.

If the address bytes A15–A2 do not change, i.e. all bytes are located within the same 4 byte address block, then multibyte programming is permitted. The multibyte programming facility allows 4 bytes of data to be written to the desired addresses after the ELATCH bit has been set (see [Table 1](#) for multibyte programming entry conditions).

NOTE: *The erased state of the EPROM is \$00.*

EEPROM

The 160 byte block of EEPROM is located at address \$0100 to \$019F. The EEPROM can be programmed on a single-byte basis by manipulating the EEPROM programming register at \$001C. The voltage necessary for programming and erasing the internal EEPROM is generated by an on-chip charge pump. No external programming voltage is necessary.

NOTE: *The erased state of an EEPROM cell is '1'. This means that a write forces zeroes to the bits specified, whilst bits defined as ones are unchanged by the write operation.*

EEPROM programming register (EPROG)

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

CPEN — Charge pump enable

1 = The charge pump, which produces the internal programming voltage for the EEPROM, is enabled; the programming voltage is available as soon as the EEGPM bit is set. The EELATCH bit should also be set in order that programming can take place.

0 = The charge pump is disabled.

This bit should always be cleared when the charge pump is not in use.

ER1–ER0 — Erase select bits

ER1 and ER0 form a 2-bit field which is used to select one of three erase modes: byte, block or bulk. [Table 2](#) shows the modes selected for each bit configuration.

In byte erase mode, only the selected byte is erased. In block erase mode, a 32-byte block of EEPROM is erased (the EEPROM memory

Table 2 Erase mode select

| ER1 | ER0 | Mode |
|-----|-----|-------------|
| 0 | 0 | No erase |
| 0 | 1 | Byte erase |
| 1 | 0 | Block erase |
| 1 | 1 | Bulk erase |

space is arranged into five 32-byte blocks: \$0100–\$011F, \$0120–\$013F, \$0140–\$015F, \$0160–\$017F and \$0180–\$019F). Performing a bulk erase to any EEPROM address will erase the entire 160 byte EEPROM array.

EELATCH — EEPROM latch control

- 1 = EEPROM address and data buses are configured for programming; reads from the array are inhibited while this bit is set.
- 0 = EEPROM address and data buses are configured for normal reads.

EERC — EEPROM RC oscillator control

- 1 = The EEPROM section uses the internal RC oscillator instead of the CPU clock; a time delay of t_{RCON} should be allowed for the RC oscillator to stabilize (see [Table 23](#)).
- 0 = The EEPROM section uses the CPU clock.

NOTE: The EERC bit should always be set if the bus frequency falls below 1.0 MHz.

EEPGM — EEPROM programming power enable/disable

- 1 = Voltage from the charge pump is supplied to the EEPROM array in order that programming or erasing may take place.
- 0 = Voltage from the charge pump is removed from the EEPROM array.

EEPROM 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 EEGPM for a time t_{EPGM} .

Memory and Registers

There are three possibilities for erasing data from the EEPROM array, depending on the amount of data to be 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} (see [Table 23](#)).
- 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} (see [Table 23](#)).
- 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} (see [Table 23](#)).

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 (see [Table 23](#)). Following each erase or programming sequence, clear all programming control bits.

Example program

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 | LATCH BIT |
| EERC | EQU | 1 | RC/OSC SELECTOR BIT |
| EEPGM | EQU | 0 | EEPROM PROGRAM BIT |
| EESTART | EQU | \$0100 | STARTING ADDRESS OF EEPROM |
| SUMPIN | EQU | \$FF | DUMMY DATA |
| ORG | | \$1000 | |
| START | EQU | * | |
| BSET | EERC, PROG | | SELECT RC OSCILLATOR |
| BSR | DELAY | | RC OSCILLATOR STABILIZATION |
| BSET | CPEN, PROG | | TURN ON CHARGE PUMP |
| BSET | EELATCH, PROG | | ENABLE LATCH BIT |

Freescale Semiconductor, Inc.

Memory and Registers EEPROM

```

                                BCLR    ER1, PROG    SELECT PROGRAM (NOT ERASE)
                                BCLR    ER0, PROG    SELECT PROGRAM (NOT ERASE)

OK                               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    EELATCH, 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    BSET    3, TCSR        CLEAR TOF
                                BRCLR   TOF, TCSR, *  WAIT FOR TOF FLAG
                                DECX
                                BNE     TIMLP        COUNT DOWN TO 0
                                RTS

```

Memory and Registers

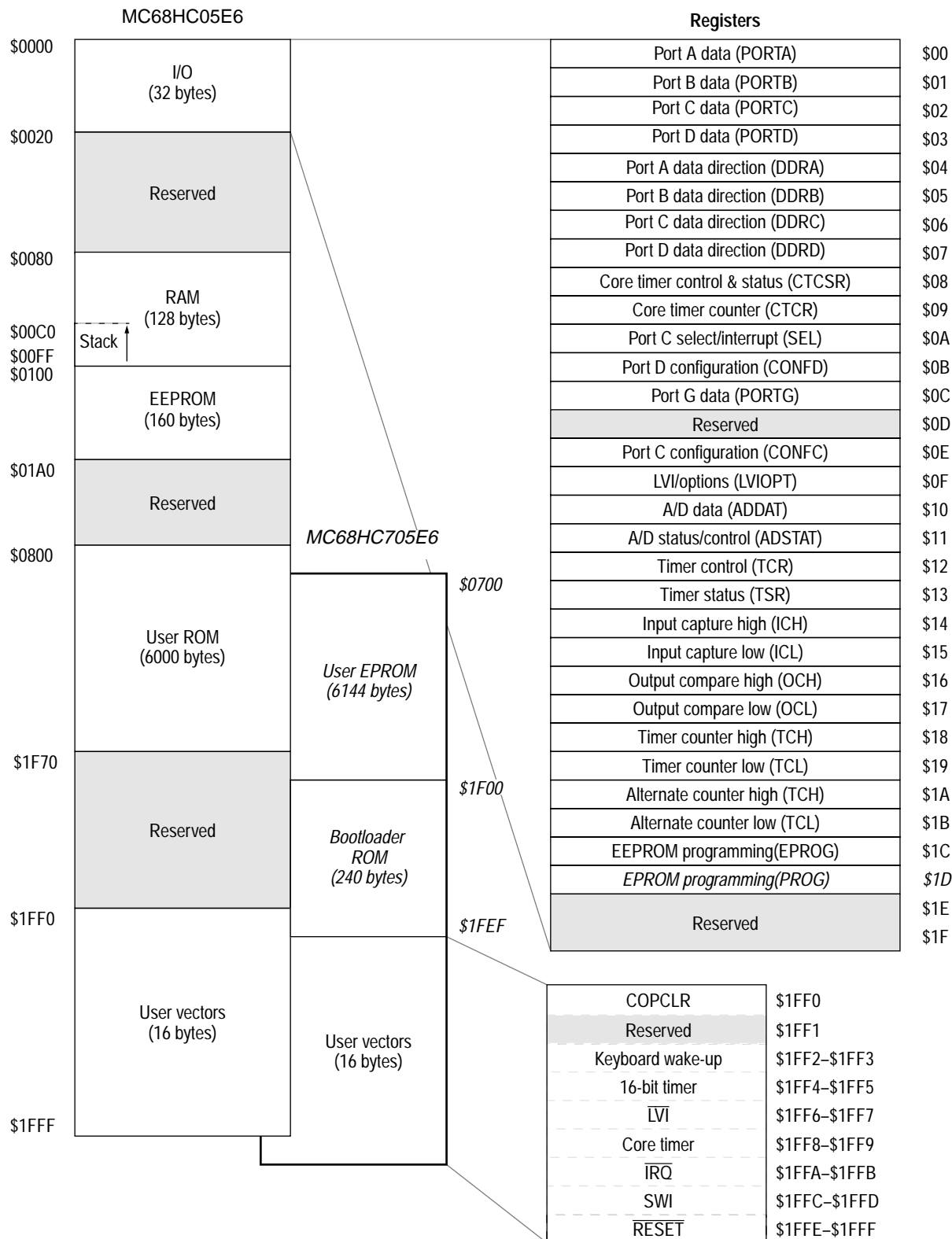


Figure 7 Memory map of the MC68HC05E6 and MC68HC705E6

Table 3 Register outline

| Register name | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on Reset |
|---|---------|----------|--------|--------|-------|-------|---------|--------------------|---------|----------------|
| Port A data (PORTA) | \$0000 | | | | | | | | | Undefined |
| Port B data (PORTB) | \$0001 | | | | | | | | | Undefined |
| Port C data (PORTC) | \$0002 | | | | | | | | | Undefined |
| Port D data (PORTD) | \$0003 | | | | | | | | | Undefined |
| Port A data direction (DDRA) | \$0004 | | | | | | | | | 0000 0000 |
| Port B data direction (DDRB) | \$0005 | | | | | | | | | 0000 0000 |
| Port C data direction (DDRC) | \$0006 | | | | | | | | | 0000 0000 |
| Port D data direction (DDRD) | \$0007 | | | | | | | | | 0000 0000 |
| Core timer control/status (CTCSR) | \$0008 | CTOF | RTIF | CTOFE | RTIE | RTOF | RRTIF | RT1 | RT0 | 0000 0011 |
| Core timer counter (CTCR) | \$0009 | | | | | | | | | 0000 0000 |
| Keyboard/timer (KEY/TIM) | \$000A | KSF | KIE | KIRST | | | | SEL1 | SEL0 | 000u uu00 |
| Port D configuration (CONFD) | \$000B | | | | | | | | | 0000 0000 |
| Port G data (PORTG) | \$000C | | | | | | | | | Undefined |
| Reserved | \$000D | | | | | | | | | |
| Port C configuration (CONFC) | \$000E | | | | | | | | | 0000 0000 |
| LVI/options (LVIOPT) | \$000F | LVIINT | LVIVAL | LVIRST | LVIE | | | COP ⁽¹⁾ | IRQ | 0u00 uu00 |
| A/D data (ADDATA) | \$0010 | | | | | | | | | Undefined |
| A/D status/control (ADSTAT) | \$0011 | COCO | ADRC | ADON | 0 | 0 | CH2 | CH1 | CH0 | 0u00 0uuu |
| Timer control (TCR) | \$0012 | ICIE | OCIE | TOIE | 0 | 0 | 0 | IEDG | OLV | 0000 00u0 |
| Timer status (TSR) | \$0013 | ICF | OCF | TOF | 0 | 0 | 0 | 0 | 0 | 0000 0000 |
| Input capture high (ICH) | \$0014 | (bit 15) | | | | | | | (bit 8) | Undefined |
| Input capture low (ICL) | \$0015 | | | | | | | | | Undefined |
| Output compare high (OCH) | \$0016 | (bit 15) | | | | | | | (bit 8) | Undefined |
| Output compare low (OCL) | \$0017 | | | | | | | | | Undefined |
| Timer counter high (TCH) | \$0018 | (bit 15) | | | | | | | (bit 8) | 1111 1111 |
| Timer counter low (TCL) | \$0019 | | | | | | | | | 1111 1100 |
| Alternate counter high (ACH) | \$001A | (bit 15) | | | | | | | (bit 8) | 1111 1111 |
| Alternate counter low (ACL) | \$001B | | | | | | | | | 1111 1100 |
| EEPROM programming (EPROG) | \$001C | 0 | CPEN | 0 | ER1 | ER0 | EELATCH | EERC | EEPGM | 0000 0000 |
| EPROM programming (PROG) ⁽¹⁾ | \$001D | 0 | 0 | 0 | 0 | 0 | ELATCH | 0 | EPGM | 0000 0000 |
| Reserved | \$001E | | | | | | | | | |
| | \$001F | | | | | | | | | |
| COPCLR | \$1FF0 | | | | | | | | CCLR | 0000 0000 |

1. MC68HC705E6 only

u = undefined

Input/Output Ports

Contents

| | |
|------------------------------------|----|
| Introduction | 35 |
| Input/output programming | 36 |
| Port A | 37 |
| Port B | 37 |
| Port C | 37 |
| Port D | 41 |
| Port G | 41 |
| Port registers | 42 |

Introduction

In single chip mode, the MC68HC05E6 has a total of 32 I/O lines, arranged as three 8-bit I/O ports (A, B and D), one 8-bit I/O port (C) which shares two pins with the timer subsystem, and one 4-bit input-only port (G) which is 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. All of the port C pins can be configured to respond to keyboard interrupts.

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.

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 I/O port has an associated DDR. Any I/O port pin is configured as an output if its corresponding DDR bit is set. A pin is configured as an input if its corresponding DDR bit is cleared.

At power-on or reset, all DDRs are cleared, thus configuring all port pins as inputs. The data direction registers can be written to or read by the MCU. During the programmed output state, a read of the data register actually reads the value of the output data buffer and not the I/O pin. The operation of the standard port hardware is shown schematically in Figure 8.

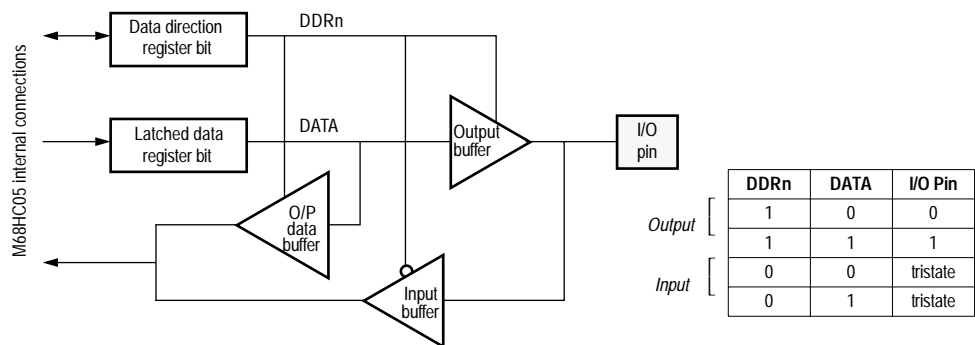


Figure 8 Standard I/O port structure

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

Table 4 I/O pin states

| R/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. |

Port A

This 8-bit port comprises a data register and a data direction register. When the pins of port A are configured as inputs they have fixed pull-up resistors connected to them.

The pull-ups and pull-downs associated with ports A, B and C are MOS transistors which vary with the supply voltage and the loading on the pins.

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

Port B

This 8-bit port comprises a data register and a data direction register. When the pins of port B are configured as inputs they have fixed pull-down resistors connected to them.

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

NOTE: *PB5–PB7 are not available in the 28-pin package.*

Port C

This 8-bit bidirectional port shares two of its pins with the timer subsystem and comprises a data register (PORTC), a data direction register (DDRC) and a configuration register (CONFC). The behaviour of the port C I/O pins is determined by the configuration of the DDRC and the CONFC registers as shown in [Table 5](#). During power-on or reset, all

port C registers are cleared, thereby returning the port pins to normal inputs with pull-up resistors.

Table 5 Port C I/O pin configurations

| DDRC | CONF C | Function |
|------|--------|-----------------------|
| 0 | 0 | Input with pull-up |
| 0 | 1 | Input without pull-up |
| 1 | 0 | Push-pull output |
| 1 | 1 | Open-drain output |

NOTE: Due to an internal clamp diode, the pins cannot be pulled higher than V_{DD} , even when configured as open-drain outputs, that is, maximum ratings for input voltage still apply to open drain outputs.

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 keyboard/timer register (KEY/TIM) 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 CONF C bits as outlined in [Table 5](#).

The structure of the port C pins is shown diagrammatically in [Figure 9](#). Once a high to low transition is sensed on any of the port C lines (PC0–PC7) configured as inputs and not being used by the timer, a keyboard interrupt will be generated and the keyboard status flag will be set, provided the interrupt mask bit of the condition code register is cleared and KIE is set. The interrupt service routine is specified by the contents of the memory locations \$1FF2 and \$1FF3. The interrupt is cleared by writing a '1' to the KIRST bit in the KEY/TIM register. The keyboard interrupt is negative edge-and-level sensitive and will force the processor out of STOP or WAIT mode.

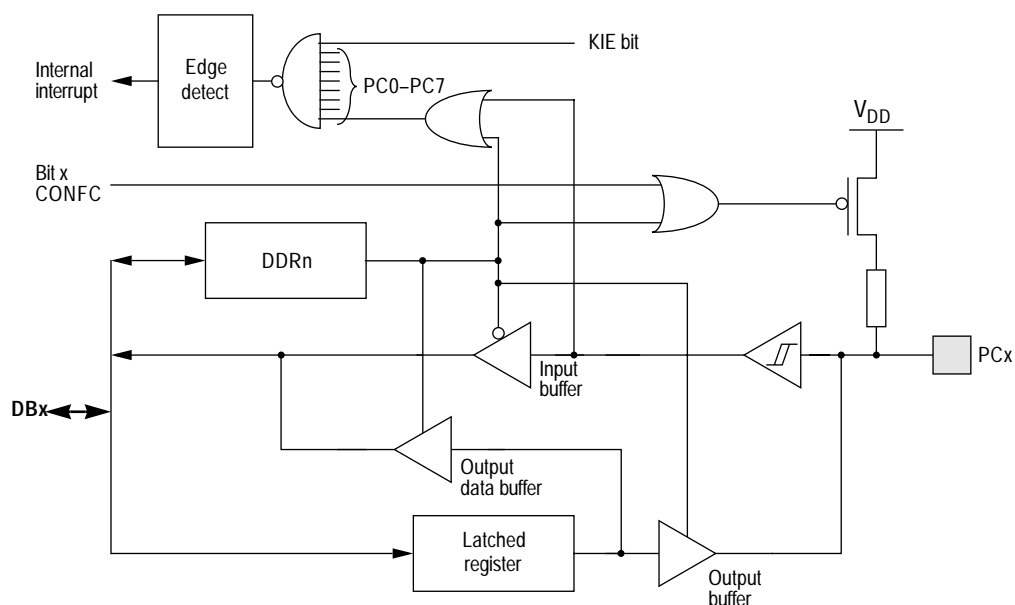


Figure 9 Port C keyboard interrupt function

NOTE: PC6 and PC7 are not available in the 28-pin package.

Keyboard/timer register (KEY/TIM)

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|--------------------------|---------|-------|-------|-------|-------|-------|-------|-------|-------|----------------|
| Keyboard/timer (KEY/TIM) | \$000A | KSF | KIE | KIRST | | | | SEL1 | SEL0 | 0000 0000 |

KSF — Keyboard status flag

1 = This read-only flag is set when there is a high to low transition on any of the port C pins configured as inputs.

0 = No high to low transition has been detected on any of the port C pins configured as inputs.

Writing a '1' to KIRST will clear the interrupt status flag.

KIE — Keyboard interrupt enable

1 = A keyboard interrupt will be generated if KSF is set and the I-bit in the CCR is clear.

0 = No keyboard interrupt will be generated, regardless of the state of the KSF flag and the I-bit.

KIRST — Keyboard interrupt reset

This bit should be set at the end of the interrupt service routine in order to clear the status flag (KSF). KIRST always reads zero.

Clearing this bit has no effect.

SEL1 — Timer select bit 1

In addition to its normal I/O function, PC1 is shared with the output compare function (TCMP) of the programmable timer. The SEL1 bit switches the pin between the two functions.

1 = Port C bit 1 is configured as the TCMP pin of the programmable timer.

0 = Port C bit 1 is configured as a standard I/O pin.

Clearing CONFC bit 1 makes TCMP a push-pull output, while setting this bit creates an open-drain output.

SEL0 — Timer select bit 0

In addition to its normal I/O function, PC0 is shared with the input capture function (TCAP) of the programmable timer. The SEL0 bit switches the pin between the two functions.

1 = Port C bit 0 is configured as the TCAP pin of the programmable timer.

0 = Port C bit 0 is configured as a standard I/O pin.

An internal pull-up resistor can be connected to the pin by clearing bit 0 of the CONFC register; alternatively it can be disconnected by setting this bit.

Port D

Port D is an 8-bit bidirectional port which does not share any of its pins with other subsystems. Port D comprises a data register (PORTD), a data direction register (DDRD) and a configuration register (CONFD). The behaviour of the port D I/O pins is determined by the configuration of the DDRD and the CONFD registers as shown in Table 6. During power-on or reset, all port D registers are cleared, thereby returning the port pins to normal inputs with pull-up resistors.

Table 6 Port D I/O pin configurations

| DDRD | CONFD | Function |
|------|-------|-----------------------|
| 0 | 0 | Input with pull-up |
| 0 | 1 | Input without pull-up |
| 1 | 0 | Push-pull output |
| 1 | 1 | Open-drain output |

NOTE: PD0–PD7 are not available in the 28-pin package.

NOTE: Due to an internal clamp diode, the pins cannot be pulled higher than V_{DD} , even when configured as open-drain outputs, that is, maximum ratings for input voltage still apply to open drain outputs.

Port G

Port G is a 4-bit input-only port which shares all of its pins with the A/D converter. When the A/D converter is enabled, PG0–PG3 read the four analog inputs. Port G can be read at any time, however reading the port during an A/D conversion sequence may inject noise on the analog inputs, resulting in reduced accuracy of the conversion. Because port G is an input-only port, there is no data direction register associated with it.

NOTE: PG1–PG3 are not available in the 28-pin package, therefore the A/D converter has only one input channel.

Input/Output Ports

The A/D converter is enabled by setting the ADON bit in the A/D status/control (ADSTAT) register.

| | 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 |

ADON — A/D converter enable/disable

1 = A/D converter is enabled.

0 = A/D converter is disabled.

The shaded bits in the ADSTAT register are described fully in [A/D status/control register \(ADSTAT\)](#).

NOTE: 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.

Port registers

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

Port data registers (PORTA, PORTB, PORTC and PORTD)

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|---------------------|---------|-------|-------|-------|-------|-------|-------|-------|-------|----------------|
| Port A data (PORTA) | \$0000 | | | | | | | | | Undefined |
| Port B data (PORTB) | \$0001 | | | | | | | | | Undefined |
| Port C data (PORTC) | \$0002 | | | | | | | | | Undefined |
| Port D data (PORTD) | \$0003 | | | | | | | | | Undefined |

For all port registers, each bit can be configured as input or output via the corresponding data direction bit in the port data direction register (DDR_x).

Port G data register (PORTG)

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|---------------------|---------|-------|-------|-------|-------|-------|-------|-------|-------|----------------|
| Port G data (PORTG) | \$000C | | | | | | | | | Undefined |

Port G is a 4-bit input-only port and therefore has no data direction register associated with it.

Data direction registers (DDRA, DDRB, DDRC and DDRD)

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|------------------------------|---------|-------|-------|-------|-------|-------|-------|-------|-------|----------------|
| Port A data direction (DDRA) | \$0004 | | | | | | | | | 0000 0000 |
| Port B data direction (DDRB) | \$0005 | | | | | | | | | 0000 0000 |
| Port C data direction (DDRC) | \$0006 | | | | | | | | | 0000 0000 |
| Port D data direction (DDRD) | \$0007 | | | | | | | | | 0000 0000 |

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

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

Port configuration registers (CONFC and CONFD)

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|------------------------------|---------|-------|-------|-------|-------|-------|-------|-------|-------|----------------|
| Port C configuration (CONFC) | \$000E | | | | | | | | | 0000 0000 |
| Port D configuration (CONFD) | \$000B | | | | | | | | | 0000 0000 |

The port configuration registers (CONFC and CONFD) are used in conjunction with the data direction registers of ports C and D to correctly

Input/Output Ports

configure the input/output pin. The effect of setting or clearing bits in these registers is shown in [Table 5](#).

Table 7 Port C and port D I/O pin configurations

| DDR _x | CONF _x | Function |
|------------------|-------------------|-----------------------|
| 0 | 0 | Input with pull-up |
| 0 | 1 | Input without pull-up |
| 1 | 0 | Push-pull output |
| 1 | 1 | Open-drain output |

Reset clears both of these registers.

Core Timer

Contents

Introduction 45

Real time interrupts (RTI) 47

Computer operating properly (COP) watchdog timer 47

Core timer registers. 48

Core timer during WAIT 50

Core timer during STOP 50

Introduction

The MC68HC05E6 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.

As shown in [Figure 10](#), the timer is driven by the internal bus clock divided by four with a fixed prescaler. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time, by accessing the CTIMER counter register (CTCR) at address \$09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of $f_{OP}/1024$. (The POR signal (t_{PORL}) is also derived from this register, at $f_{OP}/4064$.) The counter register circuit is followed by four more stages, with the resulting clock ($f_{OP}/16384$) driving the real time interrupt circuit. The RTI circuit consists of three divider stages with a 1-of-4 selector. The output of the RTI circuit is further divided by 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.

Core Timer

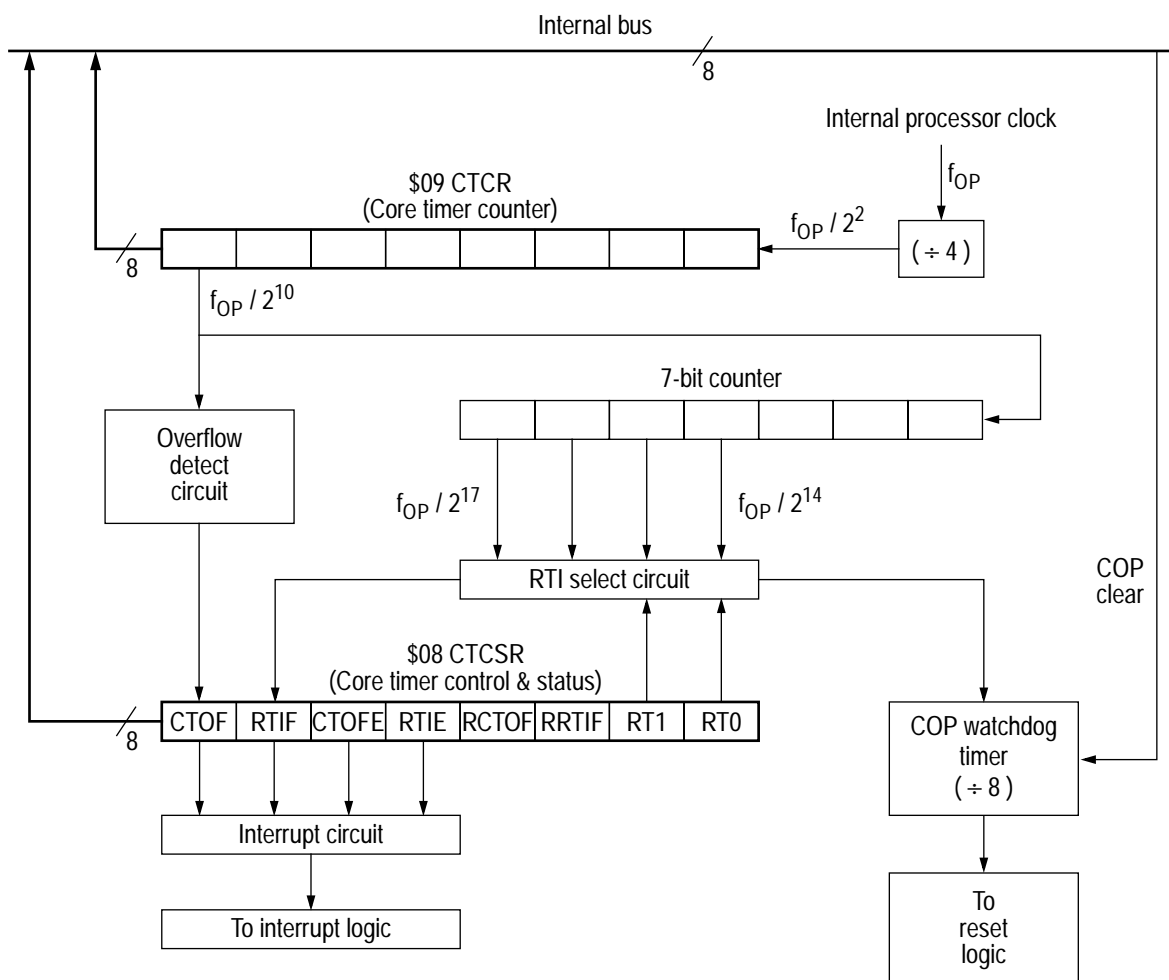


Figure 10 Core timer block diagram

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 CTOFE is set. Clearing CTOF is done by writing a '1' to the RCTOF bit (bit 3) in the CTCSR. Reset clears CTOF.

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

The core timer counter register (CTCR) is a read-only register that contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked at $f_{OP}/4$ and can be used for various functions including a software input capture. Extended time

periods can be attained using the CTIMER overflow function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter.

The power-on cycle clears the entire counter chain and begins clocking the counter. After t_{PORL} cycles, the power-on reset circuit is released, which again clears the counter chain and allows the device to come out of reset. At this point, if \overline{RESET} is not asserted, the timer will start counting up from zero and normal device operation will begin. When \overline{RESET} is asserted at any time during operation (other than POR), the counter chain will be cleared.

Real time interrupts (RTI)

The real time interrupt circuit consists of a three stage divider and a 1-of-4 selector. The clock frequency that drives the RTI circuit is $f_{OP}/2^{14}$ (or $f_{OP}/16384$), with three additional divider stages. Register details are given in [Core timer registers](#).

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 10](#). Note that the minimum COP timeout period is seven times the RTI period. This is because the COP will be cleared asynchronously with respect to the value in the core timer counter register/RTI divider, hence the actual COP timeout period will vary between 7x and 8x the RTI period.

The COP function is a mask option, enabled or disabled during device manufacture. *On the MC68HC705E6, the COP function is controlled using the COP bit in the LVI/options register (see [Mask options](#)).*

Core Timer

COP clear register (COPCLR) 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 (CCLR) of the COPCLR register (address \$1FF0). When the COP is cleared, only the final divide-by-eight stage is cleared (see [Figure 10](#)).

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|--------|---------|-------|-------|-------|-------|-------|-------|-------|-------|----------------|
| COPCLR | \$1FF0 | | | | | | | | CCLR | 0000 0000 |

Core timer registers

Core timer control and status register (CTCSR)

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|-----------------------------------|---------|-------|-------|-------|-------|-------|-------|-------|-------|----------------|
| Core timer control/status (CTCSR) | \$0008 | CTOF | RTIF | CTOFE | RTIE | RCTOF | RRTIF | RT1 | RT0 | uu00 0011 |

CTOF — Core timer overflow

1 = This read-only flag is set whenever a core timer overflow occurs.

0 = 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 CTOFE is set. When set, the bit may be cleared by writing a '1' to the RCTOF bit.

RTIF — Real time interrupt flag

1 = 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](#).

0 = 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.

CTOFE — Core timer overflow interrupt enable

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

0 = 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 = A real time interrupt will be generated if RTIF is set and the I-bit in the CCR is clear.

0 = 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, T0 — 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](#) for some example RTI periods.

Core Timer

Table 8 Example RTI periods

| RT1 | RT0 | Division ratio | Bus frequency $f_{OP} = 500 \text{ kHz}$ | | Bus frequency $f_{OP} = 1 \text{ MHz}$ | | Bus frequency $f_{OP} = 2 \text{ MHz}$ | |
|-----|-----|----------------|---|--------------------|---|--------------------|---|--------------------|
| | | | RTI period | Minimum COP period | RTI period | Minimum COP period | RTI period | Minimum COP period |
| 0 | 0 | 2^{14} | 32.8ms | 229.38ms | 16.4ms | 114.69ms | 8.2ms | 57.34ms |
| 0 | 1 | 2^{15} | 65.5ms | 458.75ms | 32.8ms | 229.38ms | 16.4ms | 114.69ms |
| 1 | 0 | 2^{16} | 131.1ms | 917.5ms | 65.5ms | 458.75ms | 32.8ms | 229.38ms |
| 1 | 1 | 2^{17} | 262.1ms | 1.835s | 131.1ms | 917.50ms | 65.5ms | 458.75ms |

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.

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.

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.

Programmable Timer

Contents

| | |
|---|----|
| Introduction | 51 |
| Keyboard/timer register (KEY/TIM) | 52 |
| Counter | 52 |
| Timer functions | 55 |
| Timer during WAIT mode | 60 |
| Timer during STOP mode | 60 |
| Timer state diagrams | 60 |

Introduction

The programmable timer on the MC68HC05E6 consists of a 16-bit read-only free-running counter, with a fixed divide-by-four prescaler, plus the input capture/output compare circuitry. Selected input edges cause the current counter value to be latched into a 16-bit input capture register so that software can later read this value to determine when the edge occurred. When the free running counter value matches the value in the output compare registers, the programmed pin action takes place. Refer to [Figure 11](#) for a block diagram of the timer. The input capture and output compare functions can only be enabled by setting bit 0 and bit 1 of the keyboard/timer register.

Programmable Timer

Keyboard/timer register (KEY/TIM)

| | Address | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | State on reset |
|--------------------------|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------------------|
| Keyboard/timer (KEY/TIM) | \$000A | KSF | KIE | KIRST | | | | SEL1 | SEL0 | 0000 0000 |

SEL1 — Timer select bit 1

1 = Port C bit 1 is configured as the TCMP pin of the programmable timer.

0 = Port C bit 1 is configured as a standard I/O pin.

SEL0 — Timer select bit 0

1 = Port C bit 0 is configured as the TCAP pin of the programmable timer.

0 = Port C bit 0 is configured as a standard I/O pin.

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

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

Counter

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

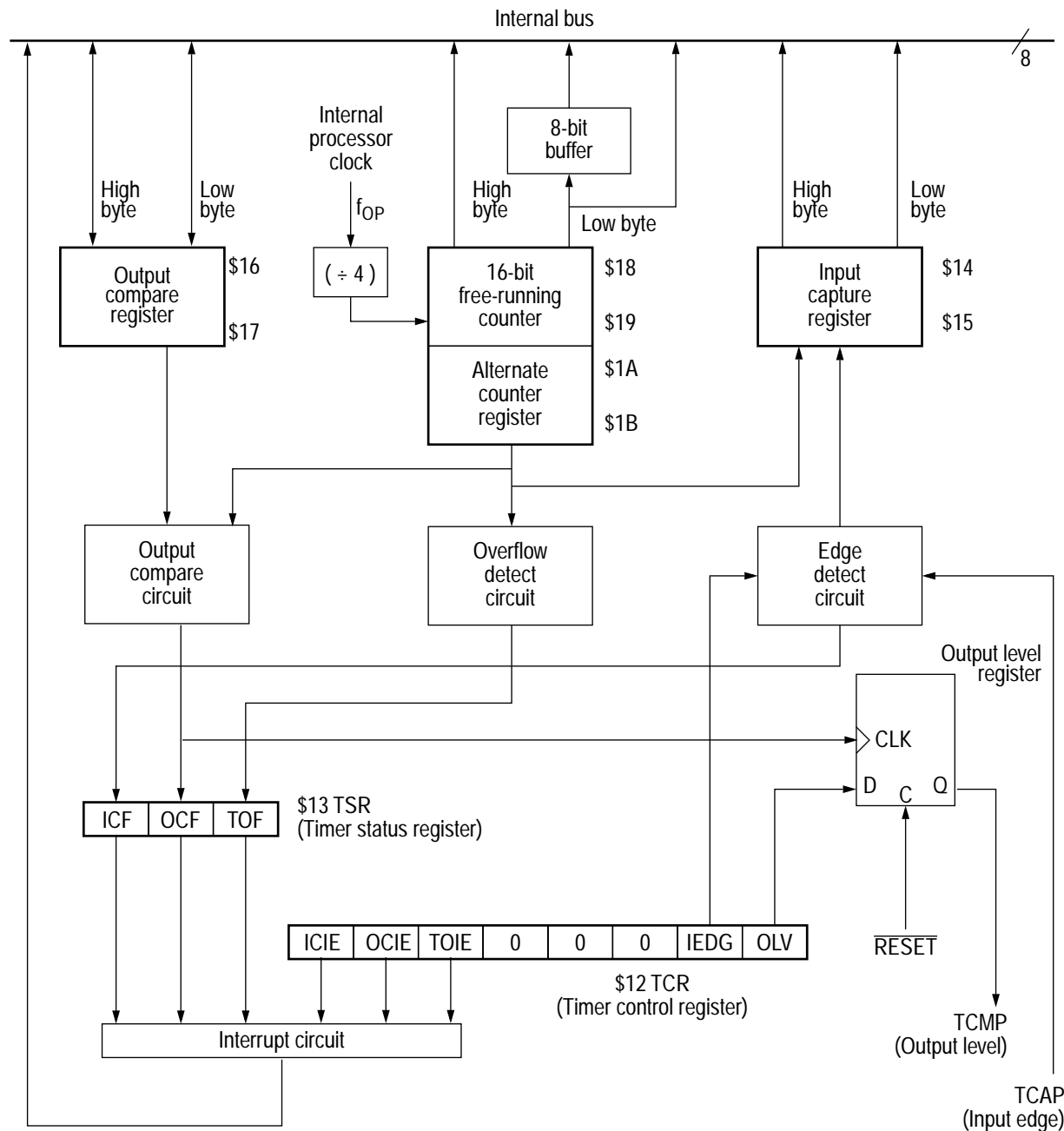


Figure 11 16-bit programmable timer block diagram

Programmable Timer

Counter high
register, Counter
low register,
Alternate counter
high register,
Alternate counter
low register

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

The double-byte, free-running counter can be read from either of two locations, the counter register at \$18 – \$19 or the alternate counter register at \$1A – \$1B. A read from only the less significant byte (LSB) of the free-running counter, \$19 or \$1B, receives the count value at the time of the read. If a read of the free-running counter or alternate counter register first addresses the more significant byte (MSB), \$18 or \$1A, the LSB is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or alternate counter register LSB and thus completes a read sequence of the total counter value. In reading either the free-running counter or alternate counter register, if the MSB is read, the LSB must also be read to complete the sequence. If the timer overflow flag (TOF) is set when the counter register LSB is read, then a read of the TSR will clear the flag.

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

The free-running counter is set to \$FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to \$FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the free-running counter repeats every 262144

internal bus clock cycles. TOF is set when the counter overflows (from \$FFFF to \$0000); this will cause an interrupt if TOIE is set.

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

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

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

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

Timer functions

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

Timer control register – TCR

The timer control register at location \$12 is used to enable the input capture (ICIE), output compare (OCIE), and timer overflow (TOIE) interrupt enable functions as well as selecting input edge sensitivity (IEDG) and output level polarity (OLV).

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

ICIE — Input capture interrupt enable

1 = Input capture interrupt enabled.

0 = Input capture interrupt disabled.

OCIE — Output compare interrupt enable

1 = Output compare interrupt enabled.

0 = Output compare interrupt disabled.

Programmable Timer

TOIE — Timer overflow interrupt enable
 1 = Timer overflow interrupt enabled.
 0 = Timer overflow interrupt disabled.

IEDG — Input edge
 1 = TCAP is positive-going edge sensitive.
 0 = TCAP is negative-going edge sensitive.

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

OLV — Output level
 1 = A high output level will appear on the TCMP pin.
 0 = A low output level will appear on the TCMP pin.

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

Timer status register – TSR

The timer status register at location (\$13) contains the status bits for the input capture, output compare and timer overflow interrupt conditions.

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

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

ICF — Input capture flag
 1 = A valid input capture has occurred.
 0 = No input capture has occurred.

This bit is set when the selected polarity of edge is detected by the input capture edge detector; an input capture interrupt will be generated if ICIE is set. ICF is cleared by reading the TSR and then the input capture low register at \$15.

OCF — Output compare flag

1 = A valid output compare has occurred.

0 = No output compare has occurred.

This bit is set when the output compare register contents match those of the free-running counter; an output compare interrupt will be generated if OCIE is set. OCF is cleared by reading the TSR and then the output compare low register at \$17.

TOF — Timer overflow flag

1 = Timer overflow has occurred.

0 = No timer overflow has occurred.

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

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

1. the timer status register is read or written when 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.

Input capture function

'Input capture' is a technique whereby an external signal (connected to the TCAP pin) is used to trigger a read of the free-running counter. In this way it is possible to relate the timing of an external signal to the internal counter value, and hence to elapsed time.

Programmable Timer

Input capture high register, Input capture low register

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

The two 8-bit registers that make up the 16-bit input capture register are read-only, and are used to latch the value of the free-running counter after the input capture edge detector senses a valid transition. The level transition that triggers the counter transfer is defined by the input edge bit (IEDG). The most significant 8 bits are stored in the input capture high register at \$14, the least significant in the input capture low register at \$15.

The result obtained from an input capture will be one greater than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronisation. Resolution is one count of the free-running counter, which is four internal bus clock cycles. The free-running counter contents are transferred to the input capture register on each valid signal transition whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture. After a read of the input capture register MSB (\$14), the counter transfer is inhibited until the LSB (\$15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period. A read of the input capture register LSB (\$15) does not inhibit the free-running counter transfer since the two actions occur on opposite edges of the internal bus clock.

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

Output compare function

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

**Output compare
high register,
Output compare
low register**

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

The 16-bit output compare register is made up of two 8-bit registers at locations \$16 (MSB) and \$17 (LSB). The contents of the output compare register are continually compared with the contents of the free-running counter and, if a match is found, the output compare flag (OCF) in the timer status register is set and the output level (OLV) bit clocked to the output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set. (The free-running counter is updated every four internal bus clock cycles.)

After a processor write cycle to the output compare register containing the MSB (\$16), the output compare function is inhibited until the LSB (\$17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB will not inhibit the compare function. The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLV) bit is clocked to the output level register whether the output compare flag (OCF) is set or clear. The minimum time required to update the output compare register is a function of the program rather than the internal hardware. Because the output compare flag and the output compare register are not defined at power on, and are not affected by reset, care must be taken when initialising output compare functions with software. The following procedure is recommended:

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

Programmable Timer

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.

Timer during WAIT mode

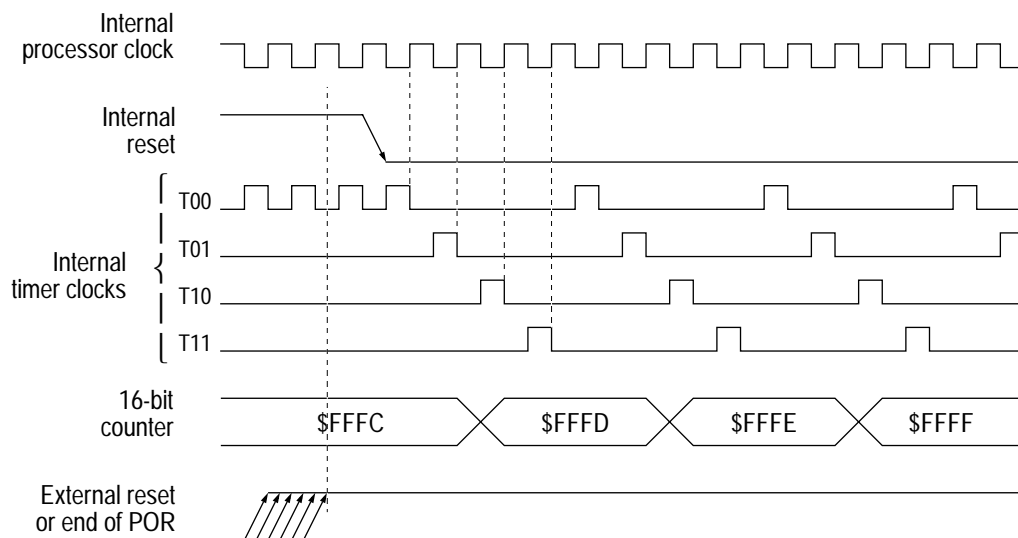
In WAIT mode all CPU action is suspended, but the programmable timer continues counting. An interrupt from an input capture, an output compare or a timer overflow, if enabled, will cause the processor to exit WAIT mode.

Timer during STOP mode

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

Timer state diagrams

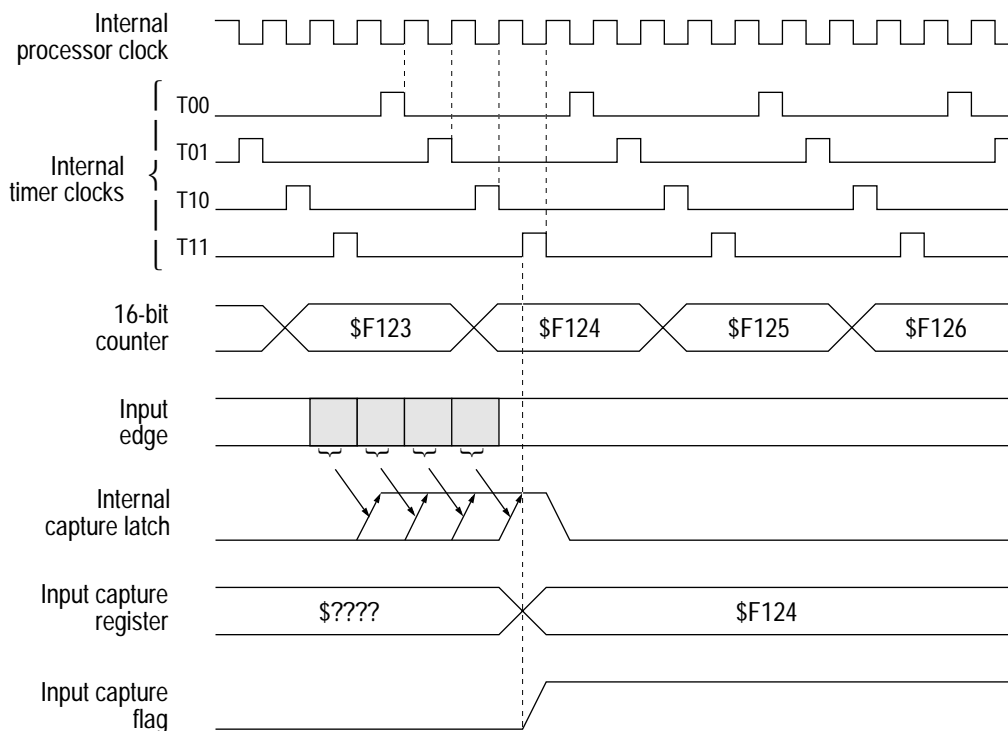
The relationships between the internal clock signals, the counter contents and the status of the flag bits are shown in the following diagrams. It should be noted that the signals labelled 'internal' (processor clock, timer clocks and reset) are not available to the user.



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

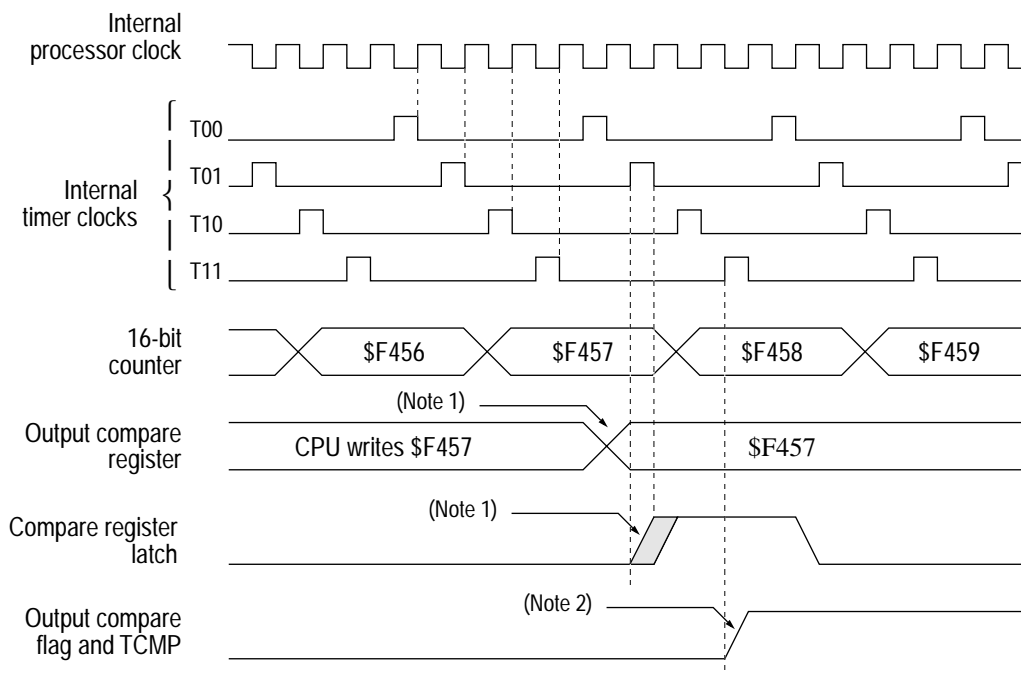
Figure 12 Timer state timing diagram for reset

Programmable Timer



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 13 Timer state timing diagram for input capture

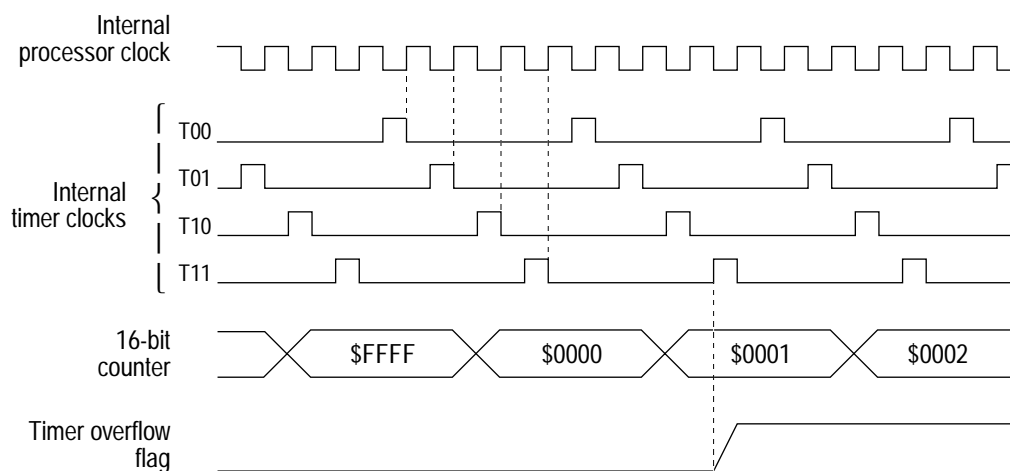


(1) The CPU write to the compare registers may take place at any time, but a compare only occurs at timer state T01. Thus a four cycle difference may exist between the write to the compare register and the actual compare.

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

Figure 14 Timer state timing diagram for output compare

Programmable Timer



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 15 Timer state timing diagram for timer overflow

A-to-D Converter

Contents

| | |
|--|----|
| Introduction | 65 |
| A/D converter operation | 66 |
| A/D registers | 68 |
| A/D converter during WAIT mode | 70 |
| A/D converter during STOP mode | 71 |
| A/D analog input | 71 |

Introduction

The analog to digital converter system consists of a four-channel, multiplexed input and a successive approximation A/D converter. The four A/D input channels are connected to pins PG0–PG3 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 \$11 (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 \$10 (ADDATA).

NOTE: PG1–PG3 are not available in the 28-pin package; in this package the A/D converter has only one external input line.

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 [A/D status/control register \(ADSTAT\)](#)). 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 [A/D status/control register \(ADSTAT\)](#)).

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 [A/D converter block diagram](#).

There are four A/D input options that can be selected by the multiplexer: AD0/PG0, AD1/PG1, AD2/PG2 or AD3/PG3. Selection is made via the CHx bits in the ADSTAT register (see [A/D status/control register \(ADSTAT\)](#)). These bits can also be used to select one of the internal test channels.

The A/D reference input (AD0–AD3) 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.

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 (\$10), and the conversion complete flag, COCO, is set in the A/D status/control register (\$11).

NOTE: *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.*

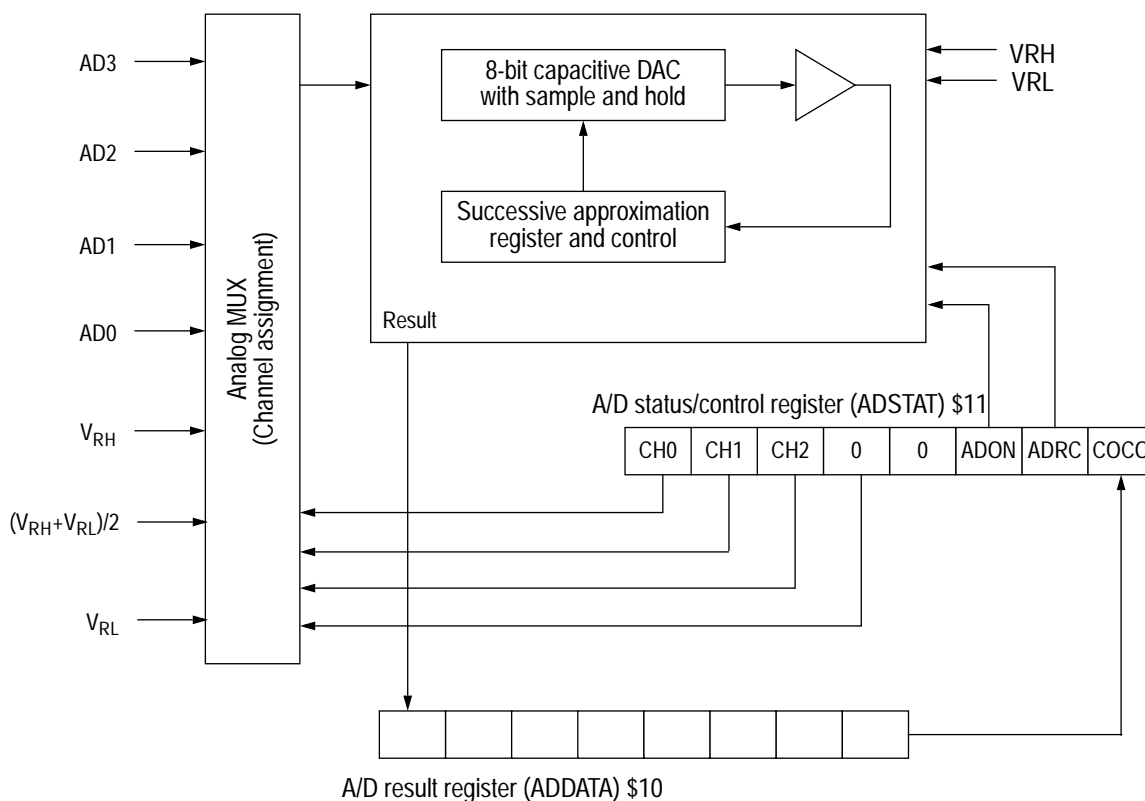


Figure 16 A/D converter block diagram

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

A-to-D Converter

A/D registers

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

1 = COCO flag is set each time a conversion is complete, allowing the new result to be read from the A/D result data register (\$10). The converter then starts a new conversion.

0 = 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 = The A/D RC oscillator is turned on and, if ADON is set, the A/D runs from the RC oscillator clock (see [Table 9](#)).

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

When the A/D RC oscillator is turned on, it takes a time t_{ADRC} to stabilize (see [Table 23](#)). During this time A/D conversion results may be inaccurate.

Table 9 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 = A/D converter is switched on.

0 = 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 23](#)). 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.

A-to-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 10](#)).

Table 10 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 | AD2/PG2 |
| 0 | 1 | 1 | 3 | AD3/PG3 |
| 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 |

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.

NOTE: 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.

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.

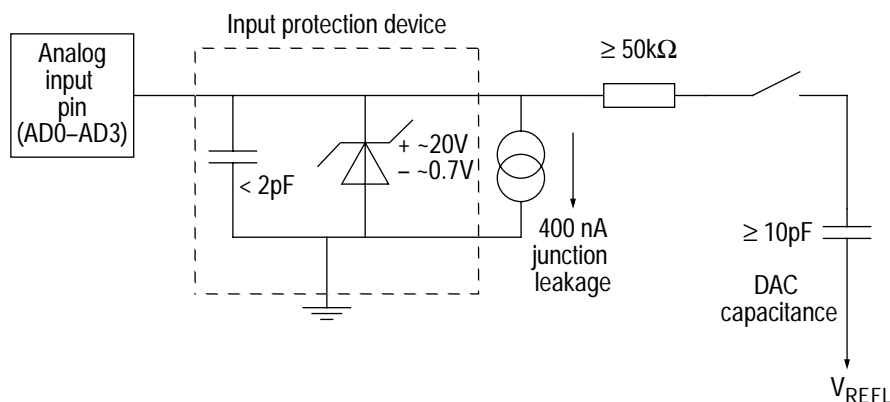
A/D converter during STOP mode

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

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 17. 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 k Ω and a capacitance of at least 10pF. (It should be noted that these are typical values measured at room temperature).



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

Figure 17 Electrical model of an A/D input pin

Resets and Interrupts

Contents

| | |
|--|----|
| Resets | 73 |
| Interrupts | 75 |
| Nonmaskable software interrupt (SWI) | 78 |
| Maskable hardware interrupts | 78 |
| Hardware controlled interrupt sequence | 83 |

Resets

The MC68HC05E6 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 \$1FFE and \$1FFF, and cause the interrupt mask of the condition code register to be set.

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

Resets and Interrupts

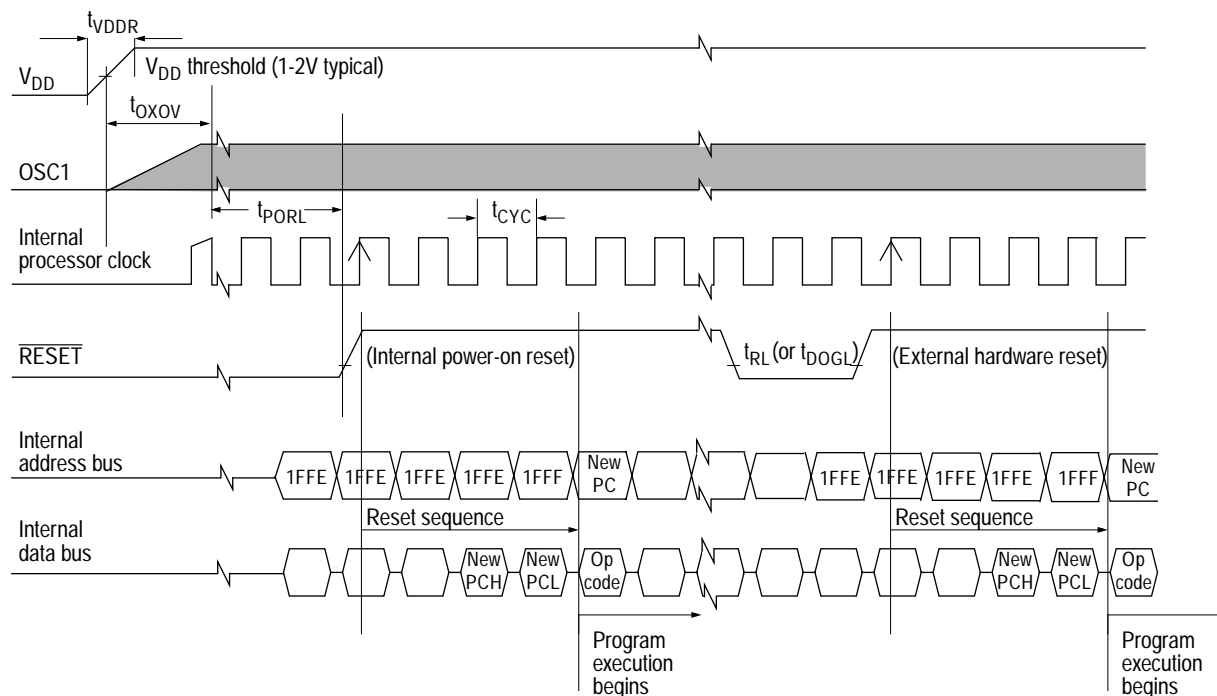


Figure 18 Reset timing diagram

$\overline{\text{RESET}}$ pin to generate a power-on reset (POR). In this case, the time constant must be great enough (at least 100ms) to allow the oscillator circuit to stabilize.

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

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 \$1FF0.*

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

The COP reset function is enabled or disabled by a mask option on the MC68HC05E6 or by the COP bit in the LVIOPT register on the MC68HC705E6 (see [Mask options](#)).

Illegal address reset

When an opcode fetch occurs from an address which is not part of the RAM (\$0080–\$00FF), ROM/EPROM (\$0800–\$1FFF)/(\$0700–\$1FFF) or EEPROM (\$0100–\$019F), the device is automatically reset.

NOTE: *No RTS or RTI instruction should be placed at the end of a memory block, i.e. at address \$017F, since this would result in an illegal address reset.*

Interrupts

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

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

Resets and Interrupts

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.*

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 11 shows the relative priorities of all the possible interrupt sources. Figure 19 shows the interrupt processing flow.

Table 11 Interrupt priorities

| Source | Register | Flags | Vector address | Priority |
|--------------------------|----------|---------------|----------------|---------------------------------|
| Reset | — | — | \$1FFE, \$1FFF | <div>highest</div> <div>↑</div> |
| Software interrupt (SWI) | — | — | \$1FFC, \$1FFD | |
| External interrupt (IRQ) | — | — | \$1FFA, \$1FFB | |
| Core timer | CTCSR | CTOF, RTIF | \$1FF8, \$1FF9 | |
| Low voltage interrupt | LVIPT | LVIINT | \$1FF6–\$1FF7 | |
| 16-bit timer | TSR | ICF, OCF, TOF | \$1FF4, \$1FF5 | |
| Keyboard interrupt | KEY/TIM | KSF | \$1FF2, \$1FF3 | |

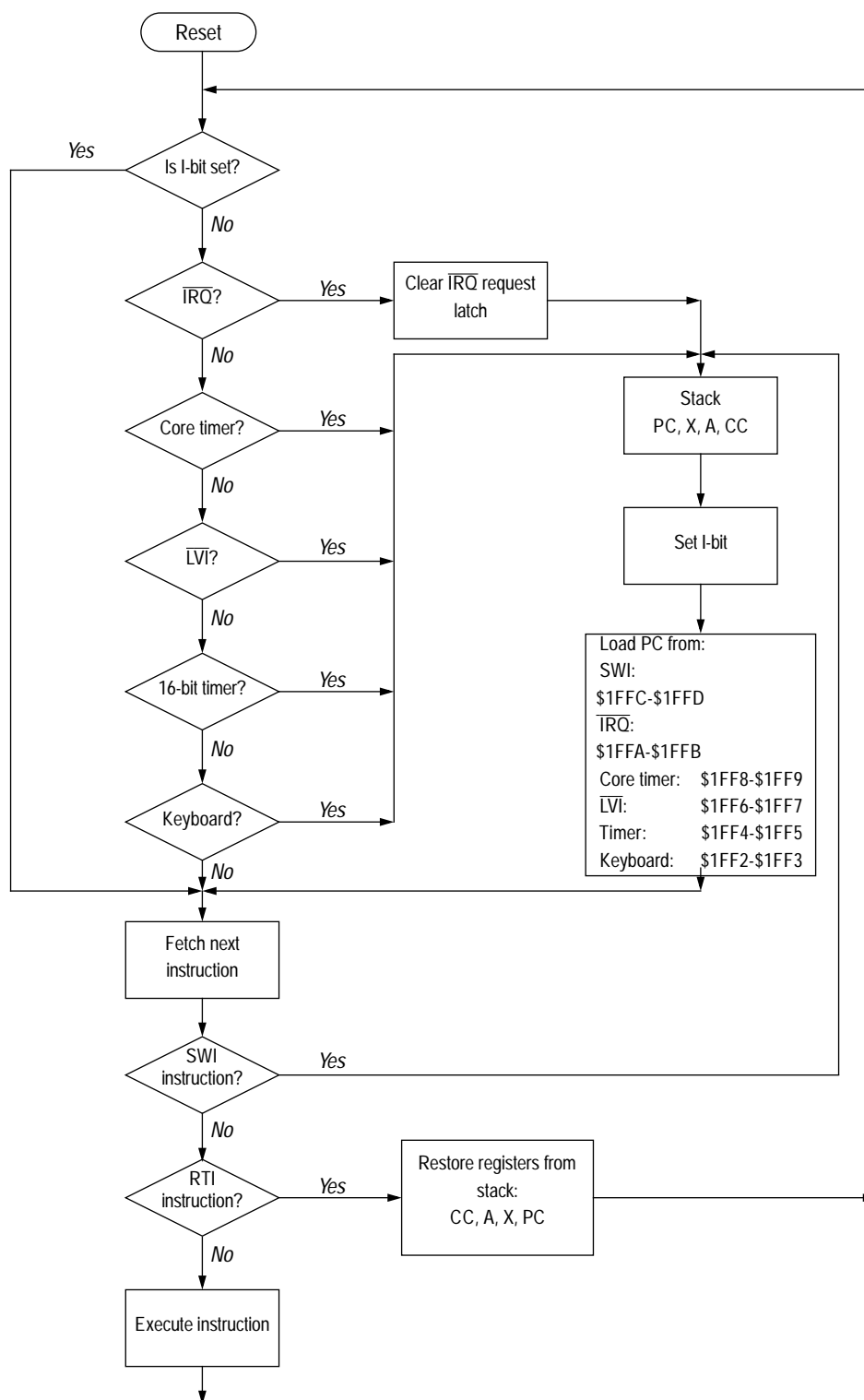


Figure 19 Interrupt flow chart

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 \$1FFC and \$1FFD.

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.*

External interrupt (IRQ)

This external interrupt source will vector to the start address contained in memory locations \$1FFA and \$1FFB. $\overline{\text{IRQ}}$ can be selected to be either edge sensitive or edge-and-level sensitive (see [Mask options](#)) by the IRQ bit in the LVIOPT register.

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 \$1FF8 and \$1FF9. Full details of the core timer can be found in [Core Timer](#).

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

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

Low voltage indicator interrupt

The low voltage indicator on the MC68HC05E6 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 $\overline{\text{LVI}}$ 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 $\overline{\text{LVI}}$ pin will be driven low and the LVIVAL bit in the LVI/options register (LVIOPT) will be cleared.

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 \$1FF6–\$1FF7.

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 $\overline{\text{LVI}}$ becomes high (see [Figure 20](#)).

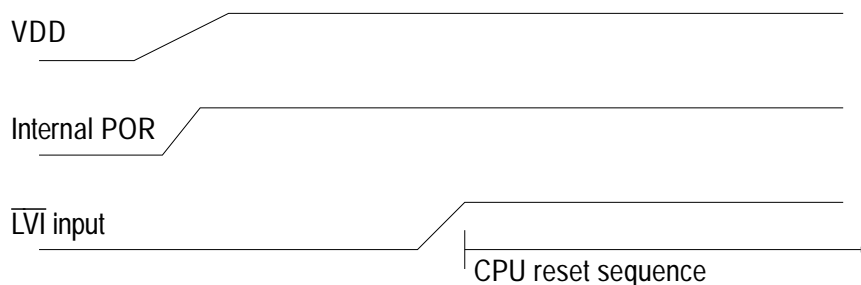


Figure 20 LVI power-on sequence

Alternatively, during power-down, when the power to VDD has fallen below a useful operating level, the $\overline{\text{LVI}}$ interrupt will not be generated until the $\overline{\text{LVI}}$ input pin has been driven low (see Figure 21).

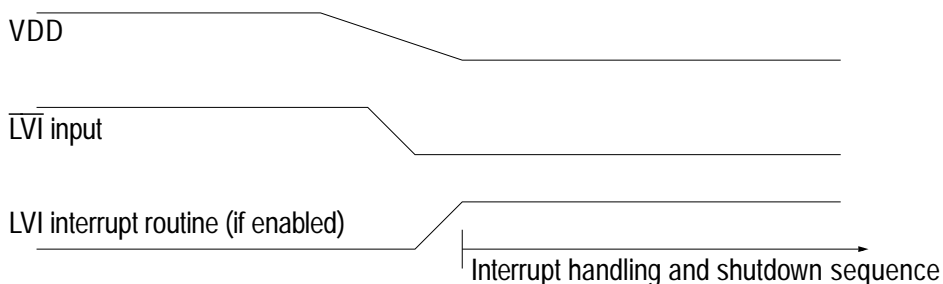


Figure 21 LVI power-down sequence

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 22).

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

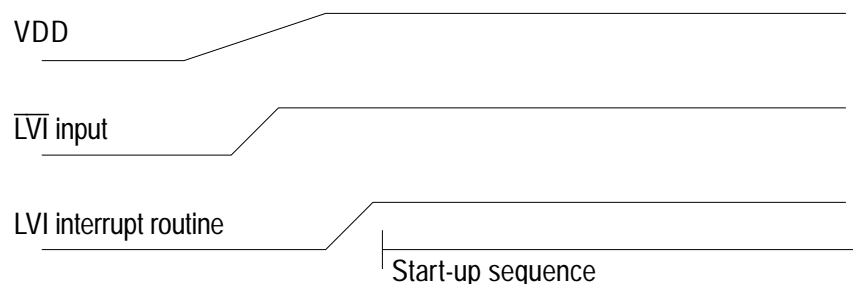


Figure 22 LVI recovery sequence

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.

LVI/options
register

| | 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 | LVIE | | | COP | IRQ | 0u00 0u00 |

LVIINT — LVI interrupt flag

1 = A valid LVI interrupt has been generated.

0 = 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 is used by the user to check the voltage being supplied to VDD.

1 = 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 = 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.

Resets and Interrupts

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 = LVI interrupts enabled; an interrupt will be generated on each high to low transition and each low to high transition of the $\overline{\text{LVI}}$ pin (but not the first low to high transition during, or after a power-on reset).

0 = LVI interrupts disabled; a high to low transition on the $\overline{\text{LVI}}$ pin will be handled by the user software.

NOTE: *The bits which are shown shaded in the LVI/options register are described in [Mask options](#).*

16-bit timer interrupts

There are three different timer interrupt flags (ICF, OCF and TOF) that will cause a timer interrupt whenever they are set and enabled. These three interrupt flags are found in the three most significant bits of the timer status register (TSR) at location \$13. ICF, OCF and TOF will vector to the service routine defined by \$1FF4 - \$1FF5 as shown in [Table 11](#).

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

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 keyboard/timer register (KEY/TIM) 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 5](#).

The interrupt vector for this interrupt is located at \$1FF2, \$1FF3. Further information on the keyboard interrupt facility can be found in [Port C](#).

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 [STOP and WAIT flowcharts](#).

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

STOP: The STOP instruction 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.

Central Processing Unit

Contents

Introduction 86

CPU Registers 86

Arithmetic/Logic Unit (ALU) 89

Instruction Set Overview 90

Addressing Modes 90

Instruction Types 93

Instruction Set Summary 98

Introduction

This chapter describes the CPU registers and the HC05 instruction set.

CPU Registers

Figure 23 shows the five CPU registers. CPU registers are not part of the memory map.

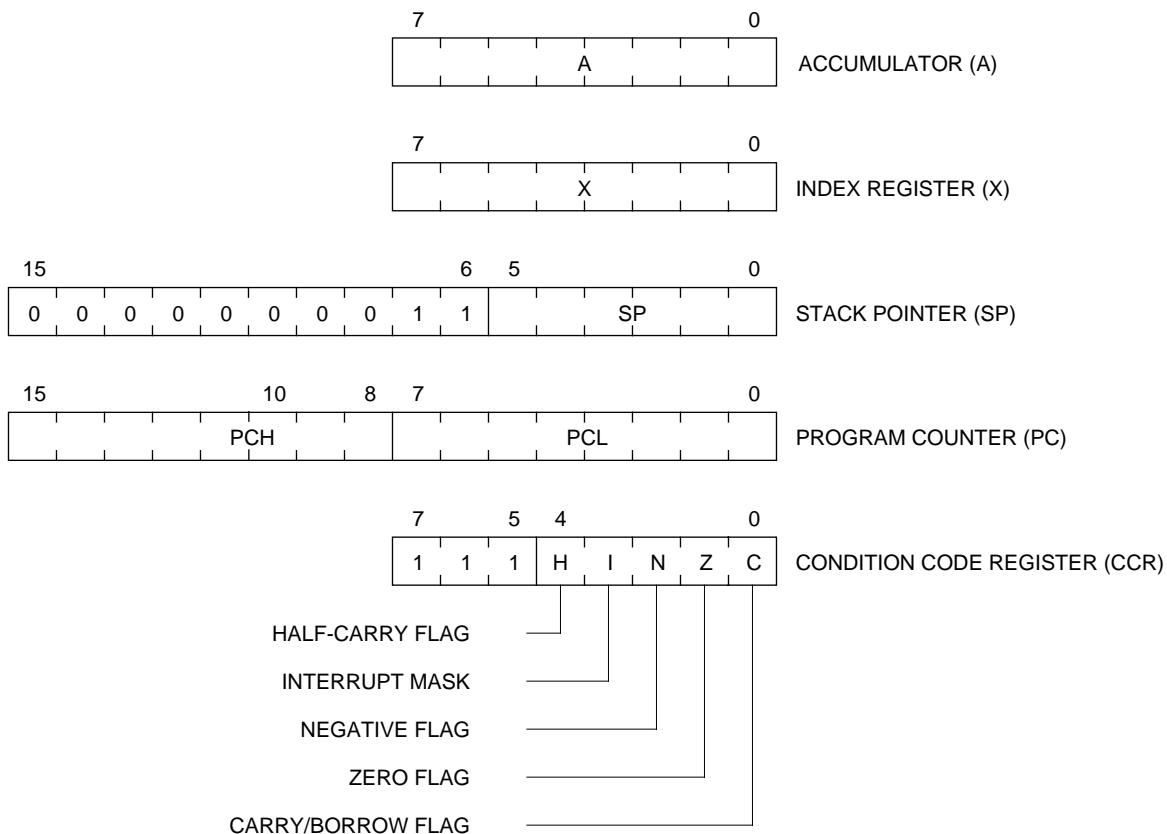


Figure 23 Programming Model

Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and results of arithmetic and non-arithmetic operations.

| | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|--------|---------------------|-------|-------|-------|-------|-------|-------|-------|
| RESET: | Unaffected by reset | | | | | | | |

Figure 24 Accumulator

Index Register

In the indexed addressing modes, the CPU uses the byte in the index register to determine the conditional address of the operand.

| | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|--------|---------------------|-------|-------|-------|-------|-------|-------|-------|
| RESET: | Unaffected by reset | | | | | | | |

Figure 25 Index Register

The 8-bit index register can also serve as a temporary data storage location.

Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next location on the stack. During a reset or after the reset stack pointer (RSP) instruction, the stack pointer is preset to \$00FF. The address in the stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

| | BIT 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | BIT 0 |
|--------|--------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-------|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | | | | | | |
| RESET: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Figure 26 Stack Pointer

The ten most significant bits of the stack pointer are permanently fixed at 000000011, so the stack pointer produces addresses from \$00C0 to \$00FF. If subroutines and interrupts use more than 64 stack locations, the stack pointer wraps around to address \$00FF and begins writing over the previously stored data. A subroutine uses two stack locations. An interrupt uses five locations.

registers on the stack, sets the interrupt mask, and then fetches the interrupt vector. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the CPU processes the latched interrupt as soon as the interrupt mask is cleared again.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its cleared state. After any reset, the interrupt mask is set and can be cleared only by a software instruction.

Negative Flag

The CPU sets the negative flag when an arithmetic operation, logical operation, or data manipulation produces a negative result.

Zero Flag

The CPU sets the zero flag when an arithmetic operation, logical operation, or data manipulation produces a result of \$00.

Carry/Borrow Flag

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some logical operations and data manipulation instructions also clear or set the carry/borrow flag.

Arithmetic/Logic Unit (ALU)

The ALU performs the arithmetic and logical operations defined by the instruction set.

The binary arithmetic circuits decode instructions and set up the ALU for the selected operation. Most binary arithmetic is based on the addition algorithm, carrying out subtraction as negative addition. Multiplication is not performed as a discrete operation but as a chain of addition and shift operations within the ALU. The multiply instruction (MUL) requires 11 internal clock cycles to complete this chain of operations.

Instruction Set Overview

The MCU instruction set has 62 instructions and uses eight addressing modes. The instructions include all those of the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is stored in the index register, and the low-order product is stored in the accumulator.

Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes provide eight different ways for the CPU to find the data required to execute an instruction. The eight addressing modes are:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset
- Relative

Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry flag (SEC) and increment accumulator (INCA). Inherent instructions require no operand address and are one byte long.

| | |
|------------------------------|---|
| Immediate | <p>Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.</p> |
| Direct | <p>Direct instructions can access any of the first 256 memory locations with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses \$00 as the high byte of the operand address.</p> |
| Extended | <p>Extended instructions use three bytes and can access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.</p> <p>When using the Motorola assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.</p> |
| Indexed, No Offset | <p>Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses \$00 as the high byte, so these instructions can address locations \$0000–\$00FF.</p> <p>Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.</p> |
| Indexed, 8-Bit Offset | <p>Indexed, 8-bit offset instructions are 2-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the effective address of the operand. These instructions can access locations \$0000–\$01FE.</p> <p>Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 (\$01FE). The</p> |

Central Processing Unit

k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

Indexed, 16-Bit Offset

Indexed, 16-bit offset instructions are 3-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the effective address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the effective branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two's complement byte that gives a branching range of -128 to $+127$ bytes from the address of the next location after the branch instruction.

When using the Motorola assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

Instruction Types

The MCU instructions fall into the following five categories:

- Register/Memory Instructions
- Read-Modify-Write Instructions
- Jump/Branch Instructions
- Bit Manipulation Instructions
- Control Instructions

Register/Memory Instructions

These instructions operate on CPU registers and memory locations. Most of them use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory.

Table 12 Register/Memory Instructions

| Instruction | Mnemonic |
|---|----------|
| Add Memory Byte and Carry Bit to Accumulator | ADC |
| Add Memory Byte to Accumulator | ADD |
| AND Memory Byte with Accumulator | AND |
| Bit Test Accumulator | BIT |
| Compare Accumulator | CMP |
| Compare Index Register with Memory Byte | CPX |
| EXCLUSIVE OR Accumulator with Memory Byte | EOR |
| Load Accumulator with Memory Byte | LDA |
| Load Index Register with Memory Byte | LDX |
| Multiply | MUL |
| OR Accumulator with Memory Byte | ORA |
| Subtract Memory Byte and Carry Bit from Accumulator | SBC |
| Store Accumulator in Memory | STA |
| Store Index Register in Memory | STX |
| Subtract Memory Byte from Accumulator | SUB |

Central Processing Unit

Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register.

NOTE: Do not use read-modify-write operations on write-only registers.

Table 13 Read-Modify-Write Instructions

| Instruction | Mnemonic |
|-------------------------------------|---------------------|
| Arithmetic Shift Left (Same as LSL) | ASL |
| Arithmetic Shift Right | ASR |
| Bit Clear | BCLR ⁽¹⁾ |
| Bit Set | BSET ⁽¹⁾ |
| Clear Register | CLR |
| Complement (One's Complement) | COM |
| Decrement | DEC |
| Increment | INC |
| Logical Shift Left (Same as ASL) | LSL |
| Logical Shift Right | LSR |
| Negate (Two's Complement) | NEG |
| Rotate Left through Carry Bit | ROL |
| Rotate Right through Carry Bit | ROR |
| Test for Negative or Zero | TST ⁽²⁾ |

1. Unlike other read-modify-write instructions, BCLR and BSET use only direct addressing.

2. TST is an exception to the read-modify-write sequence because it does not write a replacement value.

**Jump/Branch
Instructions**

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The unconditional jump instruction (JMP) and the jump-to-subroutine instruction (JSR) have no register operand. Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed.

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the effective branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from -128 to $+127$ from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register.

Table 14 Jump and Branch Instructions

| Instruction | Mnemonic |
|-------------------------------------|----------|
| Branch if Carry Bit Clear | BCC |
| Branch if Carry Bit Set | BCS |
| Branch if Equal | BEQ |
| Branch if Half-Carry Bit Clear | BHCC |
| Branch if Half-Carry Bit Set | BHCS |
| Branch if Higher | BHI |
| Branch if Higher or Same | BHS |
| Branch if \overline{IRQ} Pin High | BIH |
| Branch if \overline{IRQ} Pin Low | BIL |
| Branch if Lower | BLO |
| Branch if Lower or Same | BLS |
| Branch if Interrupt Mask Clear | BMC |
| Branch if Minus | BMI |
| Branch if Interrupt Mask Set | BMS |
| Branch if Not Equal | BNE |
| Branch if Plus | BPL |
| Branch Always | BRA |
| Branch if Bit Clear | BRCLR |
| Branch Never | BRN |
| Branch if Bit Set | BRSET |
| Branch to Subroutine | BSR |
| Unconditional Jump | JMP |
| Jump to Subroutine | JSR |

Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory, which includes I/O registers and on-chip RAM locations. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations.

Table 15 Bit Manipulation Instructions

| Instruction | Mnemonic |
|---------------------|----------|
| Bit Clear | BCLR |
| Branch if Bit Clear | BRCLR |
| Branch if Bit Set | BRSET |
| Bit Set | BSET |

Control Instructions

These instructions act on CPU registers and control CPU operation during program execution.

Table 16 Control Instructions

| Instruction | Mnemonic |
|--|----------|
| Clear Carry Bit | CLC |
| Clear Interrupt Mask | CLI |
| No Operation | NOP |
| Reset Stack Pointer | RSP |
| Return from Interrupt | RTI |
| Return from Subroutine | RTS |
| Set Carry Bit | SEC |
| Set Interrupt Mask | SEI |
| Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin | STOP |
| Software Interrupt | SWI |
| Transfer Accumulator to Index Register | TAX |
| Transfer Index Register to Accumulator | TXA |
| Stop CPU Clock and Enable Interrupts | WAIT |

Central Processing Unit

Instruction Set Summary

Table 17 Instruction Set Summary

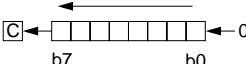
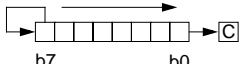
| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|---------------------------------------|--|---------------------|---|---------------------|---------------------|---------------------|--|--|--|--------------------------------------|
| | | | H | I | N | Z | C | | | | |
| ADC #opr ADC opr ADC opr ADC opr,X ADC opr,X ADC ,X | Add with Carry | $A \leftarrow (A) + (M) + (C)$ | $\uparrow \diamond$ | — | $\uparrow \diamond$ | $\uparrow \diamond$ | $\uparrow \diamond$ | IMM DIR EXT IX2 IX1 IX | A9 B9 C9 D9 E9 F9 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| ADD #opr ADD opr ADD opr ADD opr,X ADD opr,X ADD ,X | Add without Carry | $A \leftarrow (A) + (M)$ | $\uparrow \diamond$ | — | $\uparrow \diamond$ | \uparrow | \uparrow | IMM DIR EXT IX2 IX1 IX | AB BB CB DB EB FB | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| AND #opr AND opr AND opr AND opr,X AND opr,X AND ,X | Logical AND | $A \leftarrow (A) \wedge (M)$ | — | — | $\uparrow \diamond$ | \uparrow | — | IMM DIR EXT IX2 IX1 IX | A4 B4 C4 D4 E4 F4 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| ASL opr ASLA ASLX ASL opr,X ASL ,X | Arithmetic Shift Left (Same as LSL) |  | — | — | $\uparrow \diamond$ | \uparrow | \uparrow | DIR INH INH IX1 IX | 38 48 58 68 78 | dd ff | 5 3 3 6 5 |
| ASR opr ASRA ASRX ASR opr,X ASR ,X | Arithmetic Shift Right |  | — | — | $\uparrow \diamond$ | \uparrow | \uparrow | DIR INH INH IX1 IX | 37 47 57 67 77 | dd ff | 5 3 3 6 5 |
| BCC rel | Branch if Carry Bit Clear | $PC \leftarrow (PC) + 2 + rel ? C = 0$ | — | — | — | — | — | REL | 24 | rr | 3 |
| BCLR n opr | Clear Bit n | $M_n \leftarrow 0$ | — | — | — | — | — | DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7) | 11 13 15 17 19 1B 1D 1F | dd dd dd dd dd dd dd dd | 5 5 5 5 5 5 5 5 |
| BCS rel | Branch if Carry Bit Set (Same as BLO) | $PC \leftarrow (PC) + 2 + rel ? C = 1$ | — | — | — | — | — | REL | 25 | rr | 3 |
| BEQ rel | Branch if Equal | $PC \leftarrow (PC) + 2 + rel ? Z = 1$ | — | — | — | — | — | REL | 27 | rr | 3 |
| BHCC rel | Branch if Half-Carry Bit Clear | $PC \leftarrow (PC) + 2 + rel ? H = 0$ | — | — | — | — | — | REL | 28 | rr | 3 |
| BHCS rel | Branch if Half-Carry Bit Set | $PC \leftarrow (PC) + 2 + rel ? H = 1$ | — | — | — | — | — | REL | 29 | rr | 3 |
| BHI rel | Branch if Higher | $PC \leftarrow (PC) + 2 + rel ? C \vee Z = 0$ | — | — | — | — | — | REL | 22 | rr | 3 |

Table 17 Instruction Set Summary (Continued)

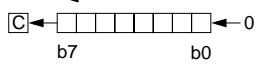
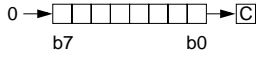
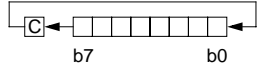
| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|---------------------------------------|--|---------------|---|---|---|---|--|--|--|--------------------------------------|
| | | | H | I | N | Z | C | | | | |
| BHS <i>rel</i> | Branch if Higher or Same | $PC \leftarrow (PC) + 2 + rel ? C = 0$ | — | — | — | — | — | REL | 24 | rr | 3 |
| BIH <i>rel</i> | Branch if IRQ Pin High | $PC \leftarrow (PC) + 2 + rel ? IRQ = 1$ | — | — | — | — | — | REL | 2F | rr | 3 |
| BIL <i>rel</i> | Branch if IRQ Pin Low | $PC \leftarrow (PC) + 2 + rel ? IRQ = 0$ | — | — | — | — | — | REL | 2E | rr | 3 |
| BIT # <i>opr</i> BIT <i>opr</i> BIT <i>opr</i> BIT <i>opr</i> ,X BIT <i>opr</i> ,X BIT ,X | Bit Test Accumulator with Memory Byte | $(A) \wedge (M)$ | — | — | ↕ | ↕ | — | IMM DIR EXT IX2 IX1 IX | A5 B5 C5 D5 E5 F5 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| BLO <i>rel</i> | Branch if Lower (Same as BCS) | $PC \leftarrow (PC) + 2 + rel ? C = 1$ | — | — | — | — | — | REL | 25 | rr | 3 |
| BLS <i>rel</i> | Branch if Lower or Same | $PC \leftarrow (PC) + 2 + rel ? C \vee Z = 1$ | — | — | — | — | — | REL | 23 | rr | 3 |
| BMC <i>rel</i> | Branch if Interrupt Mask Clear | $PC \leftarrow (PC) + 2 + rel ? I = 0$ | — | — | — | — | — | REL | 2C | rr | 3 |
| BMI <i>rel</i> | Branch if Minus | $PC \leftarrow (PC) + 2 + rel ? N = 1$ | — | — | — | — | — | REL | 2B | rr | 3 |
| BMS <i>rel</i> | Branch if Interrupt Mask Set | $PC \leftarrow (PC) + 2 + rel ? I = 1$ | — | — | — | — | — | REL | 2D | rr | 3 |
| BNE <i>rel</i> | Branch if Not Equal | $PC \leftarrow (PC) + 2 + rel ? Z = 0$ | — | — | — | — | — | REL | 26 | rr | 3 |
| BPL <i>rel</i> | Branch if Plus | $PC \leftarrow (PC) + 2 + rel ? N = 0$ | — | — | — | — | — | REL | 2A | rr | 3 |
| BRA <i>rel</i> | Branch Always | $PC \leftarrow (PC) + 2 + rel ? 1 = 1$ | — | — | — | — | — | REL | 20 | rr | 3 |
| BRCLR <i>n opr rel</i> | Branch if Bit n Clear | $PC \leftarrow (PC) + 2 + rel ? Mn = 0$ | — | — | — | — | ↕ | DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7) | 01 03 05 07 09 0B 0D 0F | dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr | 5 5 5 5 5 5 5 5 |
| BRN <i>rel</i> | Branch Never | $PC \leftarrow (PC) + 2 + rel ? 1 = 0$ | — | — | — | — | — | REL | 21 | rr | 3 |
| BRSET <i>n opr rel</i> | Branch if Bit n Set | $PC \leftarrow (PC) + 2 + rel ? Mn = 1$ | — | — | — | — | ↕ | DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7) | 00 02 04 06 08 0A 0C 0E | dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr | 5 5 5 5 5 5 5 5 |
| BSET <i>n opr</i> | Set Bit n | $Mn \leftarrow 1$ | — | — | — | — | — | DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7) | 10 12 14 16 18 1A 1C 1E | dd dd dd dd dd dd dd dd | 5 5 5 5 5 5 5 5 |
| BSR <i>rel</i> | Branch to Subroutine | $PC \leftarrow (PC) + 2$; push (PCL) $SP \leftarrow (SP) - 1$; push (PCH) $SP \leftarrow (SP) - 1$ $PC \leftarrow (PC) + rel$ | — | — | — | — | — | REL | AD | rr | 6 |
| CLC | Clear Carry Bit | $C \leftarrow 0$ | — | — | — | — | 0 | INH | 98 | | 2 |

Central Processing Unit

Table 17 Instruction Set Summary (Continued)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|---|---|---------------|---|----------------------|----------------------|----------------------|---------------------------------------|----------------------------------|----------------------------------|----------------------------|
| | | | H | I | N | Z | C | | | | |
| CLI | Clear Interrupt Mask | $I \leftarrow 0$ | — | 0 | — | — | — | INH | 9A | | 2 |
| CLR <i>opr</i> CLRA CLR X CLR <i>opr</i> ,X CLR ,X | Clear Byte | $M \leftarrow \$00$ $A \leftarrow \$00$ $X \leftarrow \$00$ $M \leftarrow \$00$ $M \leftarrow \$00$ | — | — | 0 | 1 | — | DIR INH INH IX1 IX | 3F 4F 5F 6F 7F | dd ff | 5 3 3 6 5 |
| CMP # <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> ,X CMP <i>opr</i> ,X CMP ,X | Compare Accumulator with Memory Byte | $(A) - (M)$ | — | — | $\uparrow\downarrow$ | \uparrow | \uparrow | IMM DIR EXT IX2 IX1 IX | A1 B1 C1 D1 E1 F1 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| COM <i>opr</i> COMA COM X COM <i>opr</i> ,X COM ,X | Complement Byte (One's Complement) | $M \leftarrow (\bar{M}) = \$FF - (M)$ $A \leftarrow (\bar{A}) = \$FF - (A)$ $X \leftarrow (\bar{X}) = \$FF - (X)$ $M \leftarrow (\bar{M}) = \$FF - (M)$ $M \leftarrow (\bar{M}) = \$FF - (M)$ | — | — | $\uparrow\downarrow$ | $\uparrow\downarrow$ | 1 | DIR INH INH IX1 IX | 33 43 53 63 73 | dd ff | 5 3 3 6 5 |
| CPX # <i>opr</i> CPX <i>opr</i> CPX <i>opr</i> CPX <i>opr</i> ,X CPX <i>opr</i> ,X CPX ,X | Compare Index Register with Memory Byte | $(X) - (M)$ | — | — | $\uparrow\downarrow$ | $\uparrow\downarrow$ | $\uparrow\downarrow$ | IMM DIR EXT IX2 IX1 IX | A3 B3 C3 D3 E3 F3 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| DEC <i>opr</i> DECA DEC X DEC <i>opr</i> ,X DEC ,X | Decrement Byte | $M \leftarrow (M) - 1$ $A \leftarrow (A) - 1$ $X \leftarrow (X) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$ | — | — | $\uparrow\downarrow$ | $\uparrow\downarrow$ | — | DIR INH INH IX1 IX | 3A 4A 5A 6A 7A | dd ff | 5 3 3 6 5 |
| EOR # <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> ,X EOR <i>opr</i> ,X EOR ,X | EXCLUSIVE OR Accumulator with Memory Byte | $A \leftarrow (A) \oplus (M)$ | — | — | $\uparrow\downarrow$ | \uparrow | — | IMM DIR EXT IX2 IX1 IX | A8 B8 C8 D8 E8 F8 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| INC <i>opr</i> INCA INC X INC <i>opr</i> ,X INC ,X | Increment Byte | $M \leftarrow (M) + 1$ $A \leftarrow (A) + 1$ $X \leftarrow (X) + 1$ $M \leftarrow (M) + 1$ $M \leftarrow (M) + 1$ | — | — | $\uparrow\downarrow$ | $\uparrow\downarrow$ | — | DIR INH INH IX1 IX | 3C 4C 5C 6C 7C | dd ff | 5 3 3 6 5 |
| JMP <i>opr</i> JMP <i>opr</i> JMP <i>opr</i> ,X JMP <i>opr</i> ,X JMP ,X | Unconditional Jump | $PC \leftarrow \text{Jump Address}$ | — | — | — | — | — | DIR EXT IX2 IX1 IX | BC CC DC EC FC | dd hh ll ee ff ff | 2 3 4 3 2 |

Table 17 Instruction Set Summary (Continued)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|--------------------------------------|---|---------------|---|---|---|---|---------------------------------------|----------------------------------|----------------------------------|----------------------------|
| | | | H | I | N | Z | C | | | | |
| JSR <i>opr</i> JSR <i>opr</i> JSR <i>opr</i> ,X JSR <i>opr</i> ,X JSR ,X | Jump to Subroutine | PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address | — | — | — | — | — | DIR EXT IX2 IX1 IX | BD CD DD ED FD | dd hh ll ee ff ff | 5 6 7 6 5 |
| LDA # <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> ,X LDA <i>opr</i> ,X LDA ,X | Load Accumulator with Memory Byte | A ← (M) | — | — | ↕ | ↕ | — | IMM DIR EXT IX2 IX1 IX | A6 B6 C6 D6 E6 F6 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| LDX # <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> ,X LDX <i>opr</i> ,X LDX ,X | Load Index Register with Memory Byte | X ← (M) | — | — | ↕ | ↕ | — | IMM DIR EXT IX2 IX1 IX | AE BE CE DE EE FE | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| LSL <i>opr</i> LSLA LSLX LSL <i>opr</i> ,X LSL ,X | Logical Shift Left (Same as ASL) |  | — | — | ↕ | ↕ | ↕ | DIR INH INH IX1 IX | 38 48 58 68 78 | dd ff | 5 3 3 6 5 |
| LSR <i>opr</i> LSRA LSRX LSR <i>opr</i> ,X LSR ,X | Logical Shift Right |  | — | — | 0 | ↕ | ↕ | DIR INH INH IX1 IX | 34 44 54 64 74 | dd ff | 5 3 3 6 5 |
| MUL | Unsigned Multiply | X : A ← (X) × (A) | 0 | — | — | — | 0 | INH | 42 | | 11 |
| NEG <i>opr</i> NEGA NEGX NEG <i>opr</i> ,X NEG ,X | Negate Byte (Two's Complement) | M ← -(M) = \$00 - (M) A ← -(A) = \$00 - (A) X ← -(X) = \$00 - (X) M ← -(M) = \$00 - (M) M ← -(M) = \$00 - (M) | — | — | ↕ | ↕ | ↕ | DIR INH INH IX1 IX | 30 40 50 60 70 | dd ff | 5 3 3 6 5 |
| NOP | No Operation | | — | — | — | — | — | INH | 9D | | 2 |
| ORA # <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ,X ORA <i>opr</i> ,X ORA ,X | Logical OR Accumulator with Memory | A ← (A) ∨ (M) | — | — | ↕ | ↕ | — | IMM DIR EXT IX2 IX1 IX | AA BA CA DA EA FA | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| ROL <i>opr</i> ROLA ROLX ROL <i>opr</i> ,X ROL ,X | Rotate Byte Left through Carry Bit |  | — | — | ↕ | ↕ | ↕ | DIR INH INH IX1 IX | 39 49 59 69 79 | dd ff | 5 3 3 6 5 |

Central Processing Unit

Table 17 Instruction Set Summary (Continued)

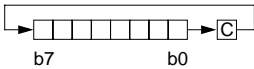
| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|---|--|---------------|---|---|---|---|---------------------------------------|----------------------------------|----------------------------------|----------------------------|
| | | | H | I | N | Z | C | | | | |
| ROR <i>opr</i> RORA RORX ROR <i>opr</i> ,X ROR ,X | Rotate Byte Right through Carry Bit |  | — | — | ↕ | ↕ | ↕ | DIR INH INH IX1 IX | 36 46 56 66 76 | dd ff | 5 3 3 6 5 |
| RSP | Reset Stack Pointer | SP ← \$00FF | — | — | — | — | — | INH | 9C | | 2 |
| RTI | Return from Interrupt | SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL) | ↕ | ↕ | ↕ | ↕ | ↕ | INH | 80 | | 9 |
| RTS | Return from Subroutine | SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL) | — | — | — | — | — | INH | 81 | | 6 |
| SBC # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> ,X SBC <i>opr</i> ,X SBC ,X | Subtract Memory Byte and Carry Bit from Accumulator | A ← (A) – (M) – (C) | — | — | ↕ | ↕ | ↕ | IMM DIR EXT IX2 IX1 IX | A2 B2 C2 D2 E2 F2 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| SEC | Set Carry Bit | C ← 1 | — | — | — | — | 1 | INH | 99 | | 2 |
| SEI | Set Interrupt Mask | I ← 1 | — | 1 | — | — | — | INH | 9B | | 2 |
| STA <i>opr</i> STA <i>opr</i> STA <i>opr</i> ,X STA <i>opr</i> ,X STA ,X | Store Accumulator in Memory | M ← (A) | — | — | ↕ | ↕ | — | DIR EXT IX2 IX1 IX | B7 C7 D7 E7 F7 | dd hh ll ee ff ff | 4 5 6 5 4 |
| STOP | Stop Oscillator and Enable IRQ Pin | | — | 0 | — | — | — | INH | 8E | | 2 |
| STX <i>opr</i> STX <i>opr</i> STX <i>opr</i> ,X STX <i>opr</i> ,X STX ,X | Store Index Register In Memory | M ← (X) | — | — | ↕ | ↕ | — | DIR EXT IX2 IX1 IX | BF CF DF EF FF | dd hh ll ee ff ff | 4 5 6 5 4 |
| SUB # <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> ,X SUB <i>opr</i> ,X SUB ,X | Subtract Memory Byte from Accumulator | A ← (A) – (M) | — | — | ↕ | ↕ | ↕ | IMM DIR EXT IX2 IX1 IX | A0 B0 C0 D0 E0 F0 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| SWI | Software Interrupt | PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte | — | 1 | — | — | — | INH | 83 | | 10 |
| TAX | Transfer Accumulator to Index Register | X ← (A) | — | — | — | — | — | INH | 97 | | 2 |

Table 17 Instruction Set Summary (Continued)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|---|--|-------------|---------------|--------|---|---|---|--------------------------------|----------------------------|--------------|-----------------------|
| | | | H | I | N | Z | C | | | | |
| TST <i>opr</i> TSTA TSTX TST <i>opr,X</i> TST <i>,X</i> | Test Memory Byte for Negative or Zero | (M) – \$00 | — | — | ↑ | ↑ | — | DIR INH INH IX1 IX | 3D 4D 5D 6D 7D | dd ff | 4 3 3 5 4 |
| TXA | Transfer Index Register to Accumulator | A ← (X) | — | — | — | — | — | INH | 9F | | 2 |
| WAIT | Stop CPU Clock and Enable Interrupts | | — | 0 ◇ | — | — | — | INH | 8F | | 2 |

A Accumulator
opr Operand
 C Carry/borrow flag
 PC Program counter
 CCR Condition code register
 PCH Program counter high byte
 dd Direct address of operand
 PCL Program counter low byte
 dd rr Direct address of operand and relative offset of branch instruction
 REL Relative addressing mode
 DIR Direct addressing mode
rel Relative program counter offset byte
 ee ff High and low bytes of offset in indexed, 16-bit offset addressing
 rr Relative program counter offset byte
 EXT Extended addressing mode
 SP Stack pointer
 ff Offset byte in indexed, 8-bit offset addressing
 X Index register
 H Half-carry flag
 Z Zero flag
 hh ll High and low bytes of operand address in extended addressing
 # Immediate value
 I Interrupt mask
 ^ Logical AND
 ii Immediate operand byte
 v Logical OR
 IMM Immediate addressing mode
 ⊕ Logical EXCLUSIVE OR
 INH Inherent addressing mode
 () Contents of
 IX Indexed, no offset addressing mode
 —() Negation (two's complement)
 IX1 Indexed, 8-bit offset addressing mode
 ← Loaded with
 IX2 Indexed, 16-bit offset addressing mode
 ? If
 M Memory location: Concatenated with
 N Negative flag
 ↑ Set or cleared
 n Any bit— Not affected

Operand (one or two bytes)
 Program counter
 Program counter high byte
 Program counter low byte
 Relative addressing mode
 Relative program counter offset byte
 Relative program counter offset byte
 Stack pointer
 Index register
 Zero flag
 Immediate value
 Logical AND
 Logical OR
 Logical EXCLUSIVE OR
 Contents of
 Negation (two's complement)
 Loaded with
 If
 Concatenated with
 Set or cleared
 Not affected

Central Processing Unit

Table 18 Opcode Map

| Bit Manipulation | | | Branch | | Read-Modify-Write | | | | | Control | | | Register/Memory | | | | | | |
|------------------|----------------|-----------|----------------|----------------------|-----------------------|-----------------------|----------------------|--------------------|-------------------|---------|------------------|------------------|------------------|------------------|------------------|----------------|-----|-----|--|
| DIR | DIR | REL | DIR | REL | DIR | INH | INH | INH | IX1 | IX | INH | INH | IMM | DIR | EXT | IX2 | IX1 | IX | |
| MSB | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | MSB | LSB | |
| 0 | BRSET0 DIR2 | 5 DIR2 | BRA REL2 | 3 NEG DIR1 | 5 NEGA INH1 | 3 NEGX INH2 | 6 NEG IX11 | 5 NEG IX1 | 9 RTI INH | | 2 SUB IMM2 | 3 SUB DIR3 | 4 SUB EXT3 | 5 SUB IX22 | 4 SUB IX11 | 3 SUB IX | 0 | | |
| 1 | BRCLR0 DIR2 | 5 DIR2 | BRN REL | 3 BCLR0 DIR2 | | | | | 6 RTS INH1 | | 2 CMP IMM2 | 3 CMP DIR3 | 4 CMP EXT3 | 5 CMP IX22 | 4 CMP IX11 | 3 CMP IX | 1 | | |
| 2 | BRSET1 DIR2 | 5 DIR2 | BHI REL | 3 BSET1 DIR2 | 11 MUL INH1 | | | | | | 2 SBC IMM2 | 3 SBC DIR3 | 4 SBC EXT3 | 5 SBC IX22 | 4 SBC IX11 | 3 SBC IX | 2 | | |
| 3 | BRCLR1 DIR2 | 5 DIR2 | BLS REL2 | 3 COM DIR1 | 5 COMA INH1 | 3 COMX INH2 | 6 COM IX11 | 5 COM IX1 | 10 SWI INH | | 2 CPX IMM2 | 3 CPX DIR3 | 4 CPX EXT3 | 5 CPX IX22 | 4 CPX IX11 | 3 CPX IX | 3 | | |
| 4 | BRSET2 DIR2 | 5 DIR2 | BCC REL2 | 3 LSR DIR1 | 5 LSRA INH1 | 3 LSRX INH2 | 6 LSR IX11 | 5 LSR IX | | | 2 AND IMM2 | 3 AND DIR3 | 4 AND EXT3 | 5 AND IX22 | 4 AND IX11 | 3 AND IX | 4 | | |
| 5 | BRCLR2 DIR2 | 5 DIR2 | BCS/BLO REL | 3 ROR DIR1 | | | | | | | 2 BIT IMM2 | 3 BIT DIR3 | 4 BIT EXT3 | 5 BIT IX22 | 4 BIT IX11 | 3 BIT IX | 5 | | |
| 6 | BRSET3 DIR2 | 5 DIR2 | BNE REL2 | 3 ROR DIR1 | 5 RORA INH1 | 3 RORX INH2 | 6 ROR IX11 | 5 ROR IX | | | 2 LDA IMM2 | 3 LDA DIR3 | 4 LDA EXT3 | 5 LDA IX22 | 4 LDA IX11 | 3 LDA IX | 6 | | |
| 7 | BRCLR3 DIR2 | 5 DIR2 | BEQ REL2 | 3 ASR DIR1 | 5 ASRA INH1 | 3 ASRX INH2 | 6 ASR IX11 | 5 ASR IX | 2 TAX INH1 | | 2 STA IMM2 | 3 STA DIR3 | 4 STA EXT3 | 5 STA IX22 | 4 STA IX11 | 3 STA IX | 7 | | |
| 8 | BRSET4 DIR2 | 5 DIR2 | BHCC REL2 | 3 ASL/LSL DIR1 | 5 ASLA/SLA INH1 | 3 ASLX/SLX INH2 | 6 ASL/LSL IX11 | 5 ASL/LSL IX | | | 2 CLC IMM2 | 3 CLC DIR3 | 4 CLC EXT3 | 5 CLC IX22 | 4 CLC IX11 | 3 CLC IX | 8 | | |
| 9 | BRCLR4 DIR2 | 5 DIR2 | BHCS REL2 | 3 ROL DIR1 | 5 ROLA INH1 | 3 ROLX INH2 | 6 ROL IX11 | 5 ROL IX | | | 2 SEC IMM2 | 3 SEC DIR3 | 4 SEC EXT3 | 5 SEC IX22 | 4 SEC IX11 | 3 SEC IX | 9 | | |
| A | BRSET5 DIR2 | 5 DIR2 | BPL REL2 | 3 DEC DIR1 | 5 DECA INH1 | 3 DECX INH2 | 6 DEC IX11 | 5 DEC IX | | | 2 CLI IMM2 | 3 CLI DIR3 | 4 CLI EXT3 | 5 CLI IX22 | 4 CLI IX11 | 3 CLI IX | A | | |
| B | BRCLR5 DIR2 | 5 DIR2 | BMI REL | 3 ROR REL2 | | | | | | | 2 SEI IMM2 | 3 SEI DIR3 | 4 SEI EXT3 | 5 SEI IX22 | 4 SEI IX11 | 3 SEI IX | B | | |
| C | BRSET6 DIR2 | 5 DIR2 | BMC REL2 | 3 INC DIR1 | 5 INCA INH1 | 3 INCX INH2 | 6 INC IX11 | 5 INC IX | | | 2 RSP IMM2 | 3 RSP DIR3 | 4 RSP EXT3 | 5 RSP IX22 | 4 RSP IX11 | 3 RSP IX | C | | |
| D | BRCLR6 DIR2 | 5 DIR2 | BMS REL2 | 3 TST DIR1 | 4 TSTA INH1 | 3 TSTX INH2 | 5 TST IX11 | 4 TST IX | | | 2 NOP IMM2 | 3 NOP DIR3 | 4 NOP EXT3 | 5 NOP IX22 | 4 NOP IX11 | 3 NOP IX | D | | |
| E | BRSET7 DIR2 | 5 DIR2 | BIL REL | 3 ROR REL2 | | | | | 2 STOP INH1 | | 2 LDX IMM2 | 3 LDX DIR3 | 4 LDX EXT3 | 5 LDX IX22 | 4 LDX IX11 | 3 LDX IX | E | | |
| F | BRCLR7 DIR2 | 5 DIR2 | BIH REL2 | 3 CLR DIR1 | 5 CLRA INH1 | 3 CLR INH2 | 6 CLR IX11 | 5 CLR IX | 2 WAIT INH1 | | 2 TXA IMM2 | 3 TXA DIR3 | 4 TXA EXT3 | 5 TXA IX22 | 4 TXA IX11 | 3 TXA IX | F | | |

INH = Inherent/REL = Relative
 IMM = Immediate/IX = Indexed, No Offset
 DIR = Direct/IX1 = Indexed, 8-Bit Offset
 EXT = Extended/IX2 = Indexed, 16-Bit Offset

MSB of Opcode in Hexadecimal
 Number of Cycles
 Opcode Mnemonic
 Number of Bytes/Addressing Mode

Electrical Specifications

Contents

Introduction 105

Maximum ratings 106

Thermal characteristics and power considerations 107

DC electrical characteristics 108

AC electrical characteristics 111

A/D converter electrical characteristics 114

Introduction

This section contains the electrical specifications and associated timing information for the MC68HC05E6 and target data for the MC68HC705E6.

NOTE: *Information given cannot be guaranteed. All values are design targets only and may change before the MC68HC705E6 is qualified.*

Electrical Specifications

Maximum ratings

Table 19 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 (VPP – MC68HC705E6) | V_{IN} | $V_{SS} - 0.3$ to $2V_{DD} + 0.3$ | V |
| Operating temperature range ($V_{DD} = 5V \pm 10\%$) MC68HC05E6 / MC68HC705E6 MC68HC05EC / MC68HC705E6C MC68HC05EV / MC68HC705E6V MC68HC05EM / MC68HC705E6M | T_A | T_L to T_H 0 to 70 –40 to +85 –40 to +105 –40 to +125 | °C |
| Operating temperature range ($V_{DD} = 3.3V \pm 10\%$) MC68HC05E6 / MC68HC705E6 MC68HC05EC / MC68HC705E6C MC68HC05EV / MC68HC705E6V MC68HC05EM / MC68HC705E6M | T_A | T_L to T_H 0 to 70 –40 to +85 –40 to +105 –40 to +125 | °C |
| Storage temperature range | T_{STG} | – 65 to +150 | °C |
| Current drain per pin (excluding VDD and VSS) ⁽²⁾ | 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} .

Thermal characteristics and power considerations

Table 20 Package thermal characteristics

| Characteristics | Symbol | Value | Unit |
|-------------------------------|---------------|-------|------|
| Thermal resistance | | | |
| – Plastic 44 pin QFP package | θ_{JA} | 60 | °C/W |
| – Plastic 28 pin SOIC package | θ_{JA} | 60 | °C/W |

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

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

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 = $I_{DD} \cdot V_{DD}$ (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:

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

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 20](#).

| Pins | R1 | R2 | C |
|-----------------------------------|--------|--------|------|
| PA0–7, PB0–7, PC0–7, PD0–7, PG0–3 | 3.26kΩ | 2.38kΩ | 50pF |

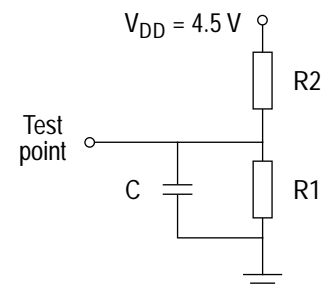


Figure 29 Equivalent test load

Electrical Specifications

DC electrical characteristics

Table 21 DC electrical characteristics for 5V operation

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

| Characteristic ⁽¹⁾ | Symbol | Min | Typ ⁽²⁾ | Max | Unit |
|--|---|-----------------------|---------------------------------|-------------------------|----------------------------|
| Output high voltage ⁽³⁾ (I _{LOAD} = − 10 μA) PA0–7, PB0–7, PC0–7, PD0–7 | V _{OH} | V _{DD} − 0.1 | — | — | V |
| Output low voltage ⁽³⁾ (I _{LOAD} = +10 μA) PA0–7, PB0–7, PC0–7, PD0–7 | V _{OL} | — | — | 0.1 | V |
| Output high voltage (I _{LOAD} = − 0.8 mA) PA0–7, PB0–7, PC0–7, PD0–7 | V _{OH} | V _{DD} − 0.8 | V _{DD} − 0.4 | — | V |
| Output low voltage (I _{LOAD} = +1.6 mA) PA0–7, PB0–7, PC0–7, PD0–7 | V _{OL} | — | 0.1 | 0.4 | V |
| Input high voltage PA0–7, PB0–5, PC0–7, PD0–7, PG0–3, OSC1, <u>IRQ</u> , <u>RESET</u> , <u>LVI</u> | V _{IH} | 0.7V _{DD} | — | V _{DD} | V |
| Input low voltage PA0–7, PB0–5, PC0–7, PD0–7, PG0–3, OSC1, <u>IRQ</u> , <u>RESET</u> , <u>LVI</u> | V _{IL} | V _{SS} | — | 0.2V _{DD} | V |
| Pull-up source current (V _{IN} = 0.2 V _{DD}) PA0–7, PC0–7, PD0–7 (if enabled), | I _{PU} | 30 | 75 | 180 | μA |
| Pull-down sink current (V _{IN} = 0.7 V _{DD}) PB0–7 | I _{PD} | 30 | 50 | 180 | μA |
| Supply current ⁽⁴⁾ RUN WAIT STOP (oscillators off) −40 to +85°C −40 to +105°C −40 to +125°C | I _{DD} | — — — — — | 3.5 1.0 1.0 1.0 1.0 | 6 2 8 20 45 | mA mA μA μA μA |
| I/O ports high-Z leakage current PC0–7, PD0–7, <u>IRQ</u> , <u>RESET</u> , <u>LVI</u> | I _{OZ} | — | 0.2 | 1.0 | μA |
| Capacitance ⁽³⁾ Ports (as input or output) | C _{OUT} | — | — | 12 | pF |
| MC68HC705E6 EPROM Programming voltage Programming current Programming time | V _{PP} I _{PP} t _{PROG} | 16 — 4 | 16.5 — — | 17 20 10 | V mA ms |

1. All I_{DD} measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs (see [VDD and VSS](#)).
2. Typical values are at mid point of voltage range and at 25°C only.
3. Characteristic guaranteed by design, but not tested.
4. RUN and WAIT I_{DD} : measured using an external square-wave clock source ($f_{OSC} = 4.2$ MHz); all inputs 0.2V from rail; no DC loads; maximum load on outputs 50pF (except OSC2 load 20pF).
WAIT I_{DD} : only the timer system active; current varies linearly with the OSC2 capacitance.
WAIT and STOP I_{DD} : all ports configured as inputs; $V_{IL} = 0.2V$ and $V_{IH} = V_{DD} - 0.2V$.
STOP I_{DD} : measured with $OSC1 = V_{DD}$.

Electrical Specifications

Table 22 DC electrical characteristics for 3.3V operation

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

| Characteristic ⁽¹⁾ | Symbol | Min | Typ ⁽²⁾ | Max | Unit |
|--|-----------|-----------------------|---------------------------------|-------------------------------|---|
| Output high voltage ⁽³⁾ ($I_{LOAD} = -10 \mu\text{A}$) PA0–7, PB0–7, PC0–7, PD0–7 | V_{OH} | $V_{DD} - 0.1$ | — | — | V |
| Output low voltage ⁽³⁾ ($I_{LOAD} = +10 \mu\text{A}$) PA0–7, PB0–7, PC0–7, PD0–7 | V_{OL} | — | — | 0.1 | V |
| Output high voltage ($I_{LOAD} = -0.4 \text{ mA}$) PA0–7, PB0–7, PC0–7, PD0–7 | V_{OH} | $V_{DD} - 0.8$ | — | — | V |
| Output low voltage ($I_{LOAD} = +0.8 \text{ mA}$) PA0–7, PB0–7, PC0–7, PD0–7 | V_{OL} | — | — | 0.4 | V |
| Input high voltage PA0–7, PB0–5, PC0–7, PD0–7, PG0–3, OSC1, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, $\overline{\text{LVI}}$ | V_{IH} | $0.7V_{DD}$ | — | V_{DD} | V |
| Input low voltage PA0–7, PB0–5, PC0–7, PD0–7, PG0–3, OSC1, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, $\overline{\text{LVI}}$ | V_{IL} | V_{SS} | — | $0.2V_{DD}$ | V |
| Pull-up source current ($V_{IN} = 0.2 V_{DD}$) PA0–7, PC0–7, PD0–7 (if enabled), | I_{PU} | 5 | 30 | 70 | μA |
| Pull-down sink current ($V_{IN} = 0.7 V_{DD}$) PB0–7 | I_{PD} | 5 | 15 | 70 | μA |
| Supply current ⁽⁴⁾ RUN (at 2.1 MHz bus frequency) WAIT (at 2.1 MHz bus frequency) STOP (oscillators off) –40 to +85°C –40 to +105°C –40 to +125°C | I_{DD} | — — — — — | 1.5 1.0 1.0 1.0 1.0 | 2.5 1.8 6.0 15 25 | mA mA μA μA μA |
| I/O ports high-Z leakage current PC0–7, PD0–7, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, $\overline{\text{LVI}}$ | I_{OZ} | — | 0.2 | 1.0 | μA |
| Capacitance ⁽³⁾ Ports (as input or output) | C_{OUT} | — | — | 12 | pF |

1. All I_{DD} measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs (see [VDD and VSS](#)).
2. Typical values are at mid point of voltage range and at 25°C only.
3. Characteristic guaranteed by design, but not tested.
4. RUN and WAIT I_{DD} : measured using an external square-wave clock source ($f_{OSC} = 2.1 \text{ MHz}$); all inputs 0.2V from rail; no DC loads; maximum load on outputs 50pF (except OSC2 load 20pF).
WAIT I_{DD} : only the timer system active; current varies linearly with the OSC2 capacitance.
WAIT and STOP I_{DD} : all ports configured as inputs; $V_{IL} = 0.2\text{V}$ and $V_{IH} = V_{DD} - 0.2\text{V}$.
STOP I_{DD} : measured with $\text{OSC1} = V_{DD}$.

AC electrical characteristics

Table 23 AC electrical characteristics for 5V operation

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

| Characteristic | Symbol | Min | Max | Unit |
|--|-----------------------------------|------|------|------------------|
| Frequency of operation | | | | |
| Crystal | f _{OSC} | — | 4.2 | MHz |
| External clock | f _{OSC} | dc | 4.2 | MHz |
| Internal operating frequency | | | | |
| Crystal (f _{OSC} /2) | f _{OP} | — | 2.1 | MHz |
| External clock (f _{OSC} /2) | f _{OP} | dc | 2.1 | MHz |
| Processor cycle time | t _{CYC} | 480 | — | ns |
| Ceramic resonator start-up time | t _{OCOV} | — | 10 | ms |
| Ceramic resonator STOP recovery start-up time | t _{ICCH} | — | 10 | ms |
| OSC1 pulse width | t _{OH} , t _{OL} | 90 | — | ns |
| RESET pulse width | t _{RL} | 1.5 | — | t _{CYC} |
| EEPROM byte erase time | t _{EBYT} | — | 10 | ms |
| EEPROM block erase time | t _{EBLOCK} | — | 100 | ms |
| EEPROM bulk erase time | t _{EBULK} | — | 200 | ms |
| EEPROM byte program time | t _{EEPGM} | — | 10 | ms |
| EEPROM programming voltage fall time | t _{FPV} | — | 10 | μs |
| RC oscillator stabilization (EEPROM, A/D) ⁽¹⁾ | t _{ADRC} | — | 5 | μs |
| A/D current stabilization time | t _{ADON} | — | 100 | μs |
| 16-bit timer | | | | |
| Resolution ⁽²⁾ | t _{RESL} | 4 | — | t _{CYC} |
| Input capture pulse width | t _{TLTH} | 250 | — | ns |
| Input capture pulse period | t _{TLTL} | (3) | — | t _{CYC} |
| Power-on reset delay | t _{PORL} | 4064 | 4064 | t _{CYC} |
| Interrupt pulse width low (edge-triggered) | t _{ILIH} | 250 | — | ns |
| Interrupt pulse period (see Figure 30) | t _{ILIL} | (4) | — | t _{CYC} |

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

2. Since the 2-bit prescaler in the timer must count four external cycles (t_{CYC}), this is the limiting factor in determining the timer resolution.3. The minimum period t_{TLTL} should not be less than the number of cycles it takes to execute the capture interrupt service routine plus 24 t_{CYC}.4. The minimum period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 21 t_{CYC}.

Electrical Specifications

Table 24 AC electrical characteristics for 3.3V operation

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

| Characteristic | Symbol | Min | Max | Unit |
|--|------------------|----------------|------|---------------|
| Frequency of operation | | | | |
| Crystal | f_{OSC} | — | 2.1 | MHz |
| External clock | f_{OSC} | dc | 2.1 | MHz |
| Internal operating frequency | | | | |
| Crystal ($f_{OSC} / 2$) | f_{OP} | — | 1.05 | MHz |
| External clock ($f_{OSC} / 2$) | f_{OP} | dc | 1.05 | MHz |
| Processor cycle time | t_{CYC} | 1000 | — | ns |
| Ceramic resonator start-up time | t_{OCOV} | — | 20 | ms |
| Ceramic resonator STOP recovery start-up time | t_{ICCH} | — | 20 | ms |
| OSC1 pulse width | t_{OH}, t_{OL} | 200 | — | ns |
| RESET pulse width | t_{RL} | 1.5 | — | t_{CYC} |
| EEPROM byte erase time | t_{EBYT} | — | 20 | ms |
| EEPROM block erase time | t_{EBLOCK} | — | 100 | ms |
| EEPROM bulk erase time | t_{EBULK} | — | 200 | ms |
| EEPROM byte program time | t_{EEPGM} | — | 20 | ms |
| EEPROM programming voltage fall time | t_{FPV} | — | 10 | μs |
| RC oscillator stabilization (EEPROM, A/D) ⁽¹⁾ | t_{ADRC} | — | 5 | μs |
| A/D current stabilization time | t_{ADON} | — | 100 | μs |
| 16-bit timer | | | | |
| Resolution ⁽²⁾ | t_{RESL} | 4 | — | t_{CYC} |
| Input capture pulse width | t_{TLTH} | 500 | — | ns |
| Input capture pulse period | t_{TLTL} | ⁽³⁾ | — | t_{CYC} |
| Power-on reset delay | t_{PORL} | 4064 | 4064 | t_{CYC} |
| Interrupt pulse width low (edge-triggered) | t_{ILIH} | 250 | — | ns |
| Interrupt pulse period (see Figure 30) | t_{ILIL} | ⁽⁴⁾ | — | t_{CYC} |

1. For bus frequencies less than 1MHz, the internal RC oscillator should be used when programming the EEPROM.
2. Since the 2-bit prescaler in the timer must count four external cycles (t_{CYC}), this is the limiting factor in determining the timer resolution.
3. The minimum period t_{TLTL} should not be less than the number of cycles it takes to execute the capture interrupt service routine plus 24 t_{CYC} .
4. The minimum period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 21 t_{CYC} .



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

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

Figure 30 External interrupt timing

Electrical Specifications

A/D converter electrical characteristics

Table 25 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 | — | ±1.5 | 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 |
| Conversion time | Total time to perform a single analog to digital conversion | | | |
| | Bus 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 | | | |
| | Bus clock Internal RC oscillator | — — | 12 12 | t _{cyc} μs |
| Sample/hold capacitance | Input capacitance during sample ADIN | — | 12 | pF |
| Input leakage ⁽²⁾ | Input leakage on A/D pins ADIN and V _{REFH} | — | 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 17).

Table 26 A/D converter electrical characteristics for 3.3V operation

($V_{DD} = 3.3 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ 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 $V_{RL} = 0\text{V}$) | — | ± 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 | — | ± 1.5 | 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 |
| Conversion time | Total time to perform a single analog to digital conversion | | | |
| | Bus 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 | | | |
| | Bus clock Internal RC oscillator | — — | 12 12 | t_{CYC} μs |
| Sample/hold capacitance | Input capacitance during sample ADIN | — | 12 | pF |
| Input leakage ⁽²⁾ | Input leakage on A/D pins ADIN and V_{RH} | — | 1 | μA |

1. Source impedances greater than 10 k Ω 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 17](#)).

Mechanical Data

Contents

| | |
|------------------------|-----|
| Introduction | 117 |
|------------------------|-----|

Introduction

Both the MC68HC05E6 and the MC68HC705E6 are available in a 28-pin SOIC package and a 44-pin QFP package. The pinout diagrams and associated mechanical drawings are illustrated in this chapter.

Mechanical Data

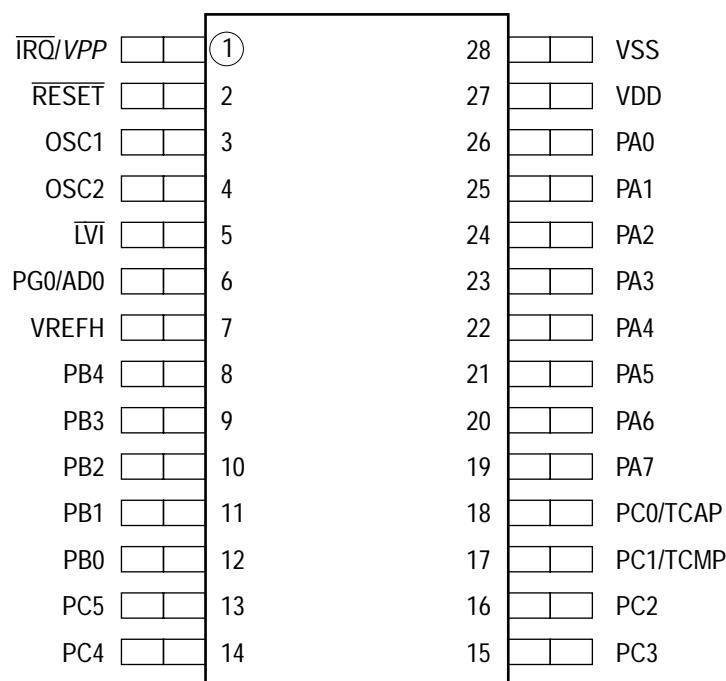


Figure 31 28-pin SOIC pinout

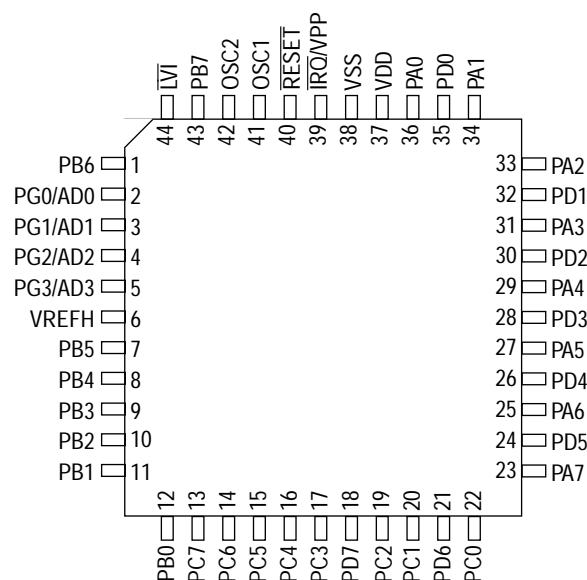
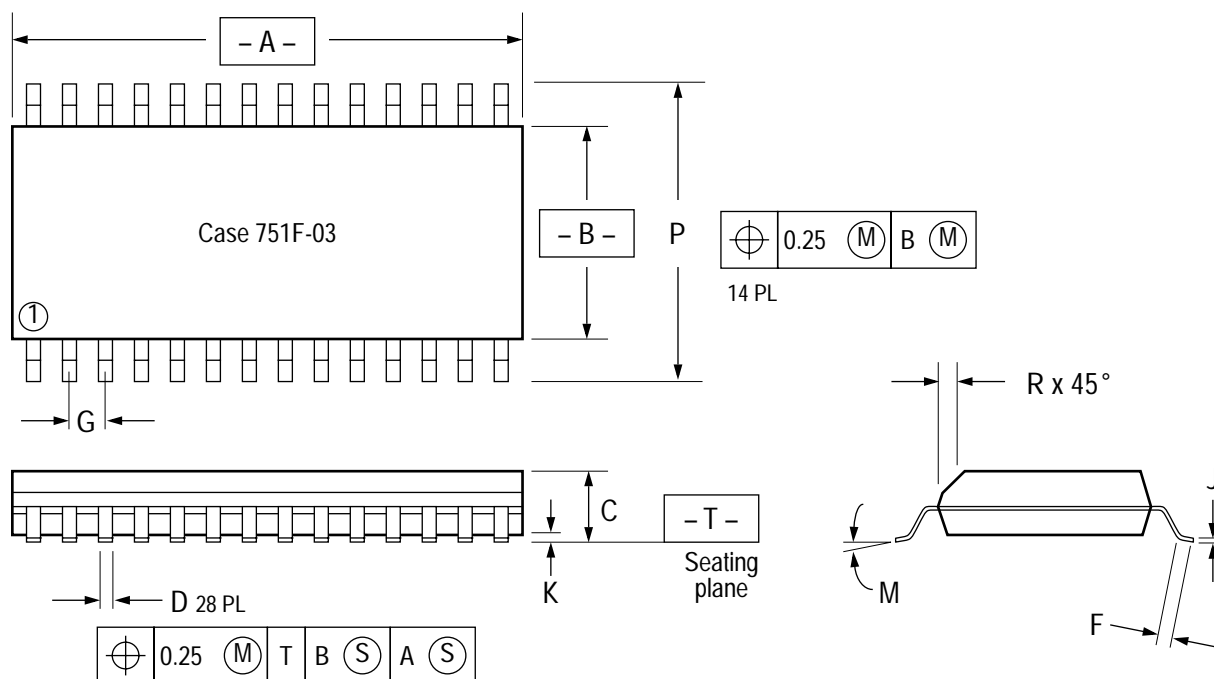


Figure 32 44-pin QFP pinout

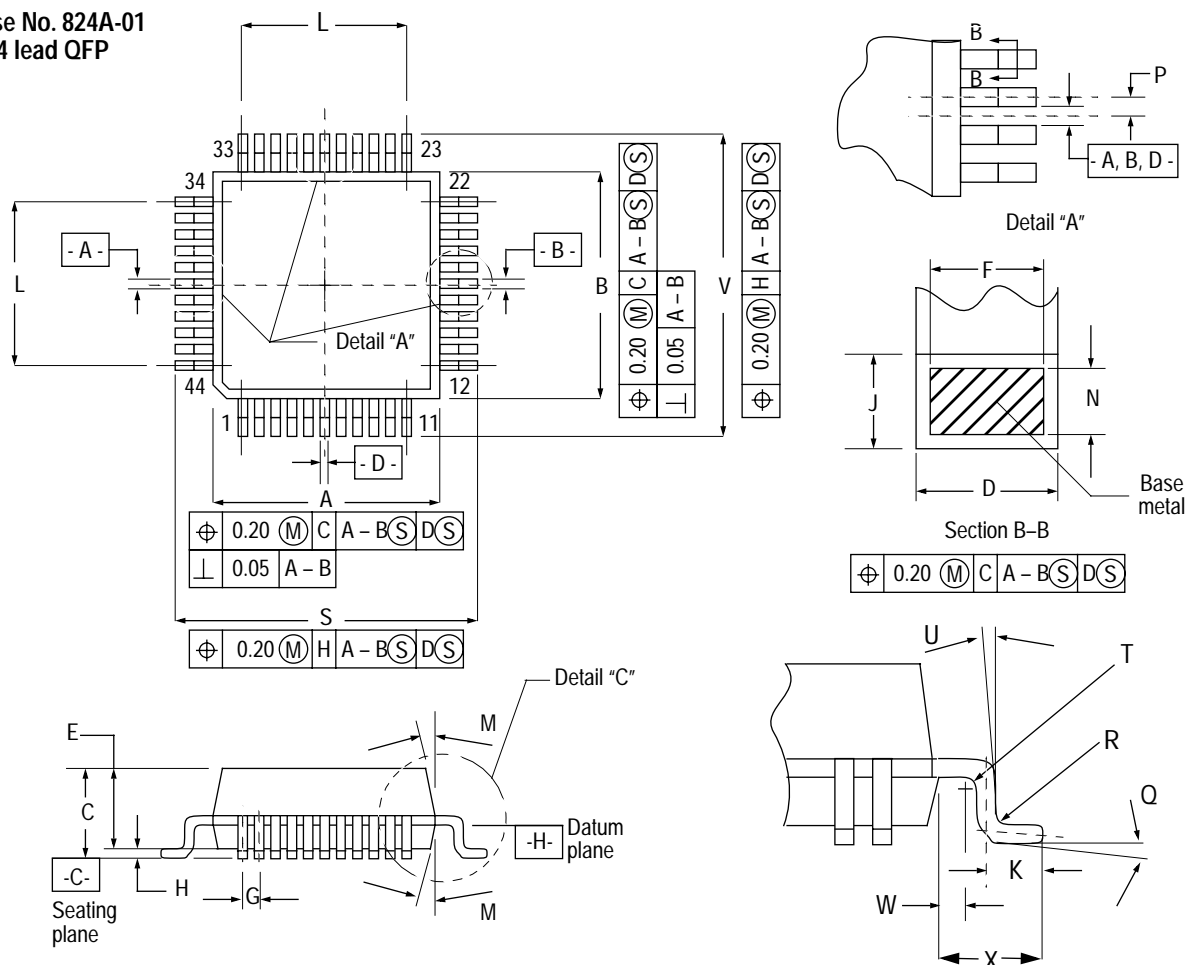


| Dim. | Min. | Max. | Notes | Dim. | Min. | Max. |
|------|----------|-------|--|------|-------|-------|
| A | 17.80 | 18.05 | 1. Dimensions 'A' and 'B' are datums and 'T' is a datum surface. 2. Dimensioning and tolerancing per ANSI Y14.5M, 1982. 3. All dimensions in mm. 4. Dimensions 'A' and 'B' do not include mould protrusion. 5. Maximum mould protrusion is 0.15 mm per side. | J | 0.229 | 0.317 |
| B | 7.40 | 7.60 | | K | 0.127 | 0.292 |
| C | 2.35 | 2.65 | | M | 0° | 8° |
| D | 0.35 | 0.49 | | P | 10.05 | 10.55 |
| F | 0.41 | 0.90 | | R | 0.25 | 0.75 |
| G | 1.27 BSC | | | — | — | — |

Figure 33 28-pin SOIC mechanical dimensions

Mechanical Data

Case No. 824A-01
44 lead QFP



| Dim. | Min. | Max. | Notes | Dim. | Min. | Max. |
|------|-------|-------|--|------|-------|-------|
| A | 9.90 | 10.10 | 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. | M | 5° | 10° |
| B | 9.90 | 10.10 | | N | 0.130 | 0.170 |
| C | 2.10 | 2.45 | | Q | 0° | 7° |
| D | 0.30 | 0.45 | | R | 0.13 | 0.30 |
| E | 2.00 | 2.10 | | S | 12.95 | 13.45 |
| F | 0.30 | 0.40 | | T | 0.13 | — |
| G | 0.80 | BSC | | U | 0° | — |
| H | — | 0.250 | | V | 12.95 | 13.45 |
| J | 0.130 | 0.230 | | W | 0.40 | — |
| K | 0.65 | 0.95 | | X | 1.6 | REF |
| L | 8.00 | REF | | | | |

Figure 34 44-pin QFP mechanical dimensions

Ordering Information

Contents

Introduction 121

EPROMS. 123

Verification media 123

Introduction

This section describes the information needed to order the MC68HC05E6 or MC68HC705E6.

To initiate a ROM pattern for the MCU, it is necessary to contact your local field service office, local sales person or Motorola representative. Please note that you will need to supply details such as mask option selections, temperature range, oscillator frequency, package type, electrical test requirements and device marking details so that an order can be processed, and a customer specific part number allocated. Refer to [Table 27](#) for appropriate part numbers.

NOTE: *When making a decision about packaging for the MC68HC05E6 or MC68HC705E6, it is important to remember that not all pins are bonded out in the 28-pin package and therefore some of the I/O ports of the device are not available to the user. [Modes of Operation and Pin Descriptions](#) shows the pin-out diagrams for each package option.*

Table 27 MC order numbers

| Device title | Package type | Temperature | Part number |
|--------------|---------------------|---------------|----------------|
| MC68HC05E6 | 28-pin plastic SOIC | 0 to +70°C | MC68HC05E6DW |
| | 44-pin QFP | 0 to +70°C | MC68HC05E6FB |
| | 28-pin plastic SOIC | −40 to +85°C | MC68HC05E6CDW |
| | 44-pin QFP | −40 to +85°C | MC68HC05E6CFB |
| | 28-pin plastic SOIC | −40 to +105°C | MC68HC05E6VDW |
| | 44-pin QFP | −40 to +105°C | MC68HC05E6VFB |
| | 28-pin plastic SOIC | −40 to +125°C | MC68HC05E6MDW |
| | 44-pin QFP | −40 to +125°C | MC68HC05E6MFB |
| MC68HC705E6 | 28-pin plastic SOIC | 0 to +70°C | MC68HC705E6DW |
| | 44-pin QFP | 0 to +70°C | MC68HC705E6FB |
| | 28-pin plastic SOIC | −40 to +85°C | MC68HC705E6CDW |
| | 44-pin QFP | −40 to +85°C | MC68HC705E6CFB |
| | 28-pin plastic SOIC | −40 to +105°C | MC68HC705E6VDW |
| | 44-pin QFP | −40 to +105°C | MC68HC705E6VFB |
| | 28-pin plastic SOIC | −40 to +125°C | MC68HC705E6MDW |
| | 44-pin QFP | −40 to +125°C | MC68HC705E6MFB |

EPROMS

An 8K 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.

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.

\$xxxx — The digits following the '\$' are in hexadecimal format.

%xxxx — The digits following the '%' are in binary format.

A — See “accumulator (A).”

accumulator (A) — An 8-bit general-purpose register in the CPU08. The CPU08 uses the accumulator to hold operands and results of arithmetic and logic operations.

acquisition mode — A mode of PLL operation during startup before the PLL locks on a frequency. Also see “tracking mode.”

A/D, ADC — Analog-to-digital (converter).

address bus — The set of wires that the CPU or DMA uses to read and write memory locations.

addressing mode — The way that the CPU determines the operand address for an instruction. The M68HC08 CPU has 16 addressing modes.

ALU — See “arithmetic logic unit (ALU).”

arithmetic logic unit (ALU) — The portion of the CPU that contains the logic circuitry to perform arithmetic, logic, and manipulation operations on operands.

asynchronous — Refers to logic circuits and operations that are not synchronized by a common reference signal.

baud rate — The total number of bits transmitted per unit of time.

BCD — See “binary-coded decimal (BCD).”

binary — Relating to the base 2 number system.

binary number system — The base 2 number system, having two digits, 0 and 1. Binary arithmetic is convenient in digital circuit design because digital circuits have two permissible voltage levels, low and high. The binary digits 0 and 1 can be interpreted to correspond to the two digital voltage levels.

Glossary

binary-coded decimal (BCD) — A notation that uses 4-bit binary numbers to represent the 10 decimal digits and that retains the same positional structure of a decimal number. For example,

234 (decimal) = 0010 0011 0100 (BCD)

bit — A binary digit. A bit has a value of either logic 0 or logic 1.

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.

branch instruction — An instruction that causes the CPU to continue processing at a memory location other than the next sequential address.

break module — A module in the M68HC08 Family. The break module allows software to halt program execution at a programmable point in order to enter a background routine.

breakpoint — A number written into the break address registers of the break module. When a number appears on the internal address bus that is the same as the number in the break address registers, the CPU executes the software interrupt instruction (SWI).

break interrupt — A software interrupt caused by the appearance on the internal address bus of the same value that is written in the break address registers.

bus — A set of wires that transfers logic signals.

bus clock — The bus clock is derived from the CGMOUT output from the CGM. The bus clock frequency, f_{op} , is equal to the frequency of the oscillator output, CGMXCLK, divided by four.

byte — A set of eight bits.

C — The carry/borrow bit in the condition code register. The CPU08 sets the carry/borrow bit when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some logical operations and data manipulation instructions also clear or set the carry/borrow bit (as in bit test and branch instructions and shifts and rotates).

CCR — See “condition code register.”

central processor unit (CPU) — The primary functioning unit of any computer system. The CPU controls the execution of instructions.

CERQUAD — A ceramic package type, principally used for EPROM and high temperature devices.

CGM — See “clock generator module (CGM).”

clear — To change a bit from logic 1 to logic 0; the opposite of set.

clock — A square wave signal used to synchronize events in a computer.

clock generator module (CGM) — A module in the M68HC08 Family. The CGM generates a base clock signal from which the system clocks are derived. The CGM may include a crystal oscillator circuit and or phase-locked loop (PLL) circuit.

comparator — A device that compares the magnitude of two inputs. A digital comparator defines the equality or relative differences between two binary numbers.

computer operating properly module (COP) — A counter module in the M68HC08 Family that resets the MCU if allowed to overflow.

condition code register (CCR) — An 8-bit register in the CPU08 that contains the interrupt mask bit and five bits that indicate the results of the instruction just executed.

control bit — One bit of a register manipulated by software to control the operation of the module.

control unit — One of two major units of the CPU. The control unit contains logic functions that synchronize the machine and direct various operations. The control unit decodes instructions and generates the internal control signals that perform the requested operations. The outputs of the control unit drive the execution unit, which contains the arithmetic logic unit (ALU), CPU registers, and bus interface.

COP — See "computer operating properly module (COP)."

counter clock — The input clock to the TIM counter. This clock is the output of the TIM prescaler.

CPU — See "central processor unit (CPU)."

CPU08 — The central processor unit of the M68HC08 Family.

CPU clock — The CPU clock is derived from the CGMOUT output from the CGM. The CPU clock frequency is equal to the frequency of the oscillator output, CGMXCLK, divided by four.

CPU cycles — A CPU cycle is one period of the internal bus clock, normally derived by dividing a crystal oscillator source by two or more so the high and low times will be equal. The length of time required to execute an instruction is measured in CPU clock cycles.

Glossary

CPU registers — Memory locations that are wired directly into the CPU logic instead of being part of the addressable memory map. The CPU always has direct access to the information in these registers. The CPU registers in an M68HC08 are:

- A (8-bit accumulator)
- H:X (16-bit index register)
- SP (16-bit stack pointer)
- PC (16-bit program counter)
- CCR (condition code register containing the V, H, I, N, Z, and C bits)

CSIC — customer-specified integrated circuit

cycle time — The period of the operating frequency: $t_{CYC} = 1/f_{OP}$.

decimal number system — Base 10 numbering system that uses the digits zero through nine.

direct memory access module (DMA) — A M68HC08 Family module that can perform data transfers between any two CPU-addressable locations without CPU intervention. For transmitting or receiving blocks of data to or from peripherals, DMA transfers are faster and more code-efficient than CPU interrupts.

DMA — See "direct memory access module (DMA)."

DMA service request — A signal from a peripheral to the DMA module that enables the DMA module to transfer data.

duty cycle — A ratio of the amount of time the signal is on versus the time it is off. Duty cycle is usually represented by a percentage.

EEPROM — Electrically erasable, programmable, read-only memory. A nonvolatile type of memory that can be electrically reprogrammed.

EPROM — Erasable, programmable, read-only memory. A nonvolatile type of memory that can be erased by exposure to an ultraviolet light source and then reprogrammed.

exception — An event such as an interrupt or a reset that stops the sequential execution of the instructions in the main program.

external interrupt module (IRQ) — A module in the M68HC08 Family with both dedicated external interrupt pins and port pins that can be enabled as interrupt pins.

fetch — To copy data from a memory location into the accumulator.

firmware — Instructions and data programmed into nonvolatile memory.

free-running counter — A device that counts from zero to a predetermined number, then rolls over to zero and begins counting again.

full-duplex transmission — Communication on a channel in which data can be sent and received simultaneously.

H — The upper byte of the 16-bit index register (H:X) in the CPU08.

H — The half-carry bit in the condition code register of the CPU08. This bit indicates a carry from the low-order four bits of the accumulator value to the high-order four bits. The half-carry bit is required for binary-coded decimal arithmetic operations. The decimal adjust accumulator (DAA) instruction uses the state of the H and C bits to determine the appropriate correction factor.

hexadecimal — Base 16 numbering system that uses the digits 0 through 9 and the letters A through F.

high byte — The most significant eight bits of a word.

illegal address — An address not within the memory map

illegal opcode — A nonexistent opcode.

I — The interrupt mask bit in the condition code register of the CPU08. When I is set, all interrupts are disabled.

index register (H:X) — A 16-bit register in the CPU08. The upper byte of H:X is called H. The lower byte is called X. In the indexed addressing modes, the CPU uses the contents of H:X to determine the effective address of the operand. H:X can also serve as a temporary data storage location.

input/output (I/O) — Input/output interfaces between a computer system and the external world. A CPU reads an input to sense the level of an external signal and writes to an output to change the level on an external signal.

instructions — Operations that a CPU can perform. Instructions are expressed by programmers as assembly language mnemonics. A CPU interprets an opcode and its associated operand(s) and instruction.

interrupt — A temporary break in the sequential execution of a program to respond to signals from peripheral devices by executing a subroutine.

interrupt request — A signal from a peripheral to the CPU intended to cause the CPU to execute a subroutine.

I/O — See "input/output (I/O)."

IRQ — See "external interrupt module (IRQ)."

Glossary

jitter — Short-term signal instability.

latch — A circuit that retains the voltage level (logic 1 or logic 0) written to it for as long as power is applied to the circuit.

latency — The time lag between instruction completion and data movement.

least significant bit (LSB) — The rightmost digit of a binary number.

logic 1 — A voltage level approximately equal to the input power voltage (V_{DD}).

logic 0 — A voltage level approximately equal to the ground voltage (V_{SS}).

low byte — The least significant eight bits of a word.

low voltage inhibit module (LVI) — A module in the M68HC08 Family that monitors power supply voltage.

LVI — See "low voltage inhibit module (LVI)."

M68HC08 — A Motorola family of 8-bit MCUs.

mark/space — The logic 1/logic 0 convention used in formatting data in serial communication.

mask — 1. A logic circuit that forces a bit or group of bits to a desired state. 2. A photomask used in integrated circuit fabrication to transfer an image onto silicon.

mask option — A optional microcontroller feature that the customer chooses to enable or disable.

mask option register (MOR) — An EPROM location containing bits that enable or disable certain MCU features.

MCU — Microcontroller unit. See "microcontroller."

memory location — Each M68HC08 memory location holds one byte of data and has a unique address. To store information in a memory location, the CPU places the address of the location on the address bus, the data information on the data bus, and asserts the write signal. To read information from a memory location, the CPU places the address of the location on the address bus and asserts the read signal. In response to the read signal, the selected memory location places its data onto the data bus.

memory map — A pictorial representation of all memory locations in a computer system.

microcontroller — Microcontroller unit (MCU). A complete computer system, including a CPU, memory, a clock oscillator, and input/output (I/O) on a single integrated circuit.

modulo counter — A counter that can be programmed to count to any number from zero to its maximum possible modulus.

monitor ROM — A section of ROM that can execute commands from a host computer for testing purposes.

MOR — See "mask option register (MOR)."

most significant bit (MSB) — The leftmost digit of a binary number.

multiplexer — A device that can select one of a number of inputs and pass the logic level of that input on to the output.

N — The negative bit in the condition code register of the CPU08. The CPU sets the negative bit when an arithmetic operation, logical operation, or data manipulation produces a negative result.

nibble — A set of four bits (half of a byte).

object code — The output from an assembler or compiler that is itself executable machine code, or is suitable for processing to produce executable machine code.

opcode — A binary code that instructs the CPU to perform an operation.

open-drain — An output that has no pullup transistor. An external pullup device can be connected to the power supply to provide the logic 1 output voltage.

operand — Data on which an operation is performed. Usually a statement consists of an operator and an operand. For example, the operator may be an add instruction, and the operand may be the quantity to be added.

oscillator — A circuit that produces a constant frequency square wave that is used by the computer as a timing and sequencing reference.

OTPROM — One-time programmable read-only memory. A nonvolatile type of memory that cannot be reprogrammed.

overflow — A quantity that is too large to be contained in one byte or one word.

page zero — The first 256 bytes of memory (addresses \$0000–\$00FF).

parity — An error-checking scheme that counts the number of logic 1s in each byte transmitted. In a system that uses odd parity, every byte is expected to have an odd number of logic 1s. In an even parity system, every byte should have an even number of logic 1s. In the transmitter, a parity generator appends an extra bit to each byte to make the number of logic 1s odd for odd parity or even for even parity. A parity checker in the receiver counts the number of logic 1s in each byte. The parity checker generates an error signal if it finds a byte with an incorrect number of logic 1s.

Glossary

PC — See “program counter (PC).”

peripheral — A circuit not under direct CPU control.

phase-locked loop (PLL) — A oscillator circuit in which the frequency of the oscillator is synchronized to a reference signal.

PLL — See "phase-locked loop (PLL)."

pointer — Pointer register. An index register is sometimes called a pointer register because its contents are used in the calculation of the address of an operand, and therefore points to the operand.

polarity — The two opposite logic levels, logic 1 and logic 0, which correspond to two different voltage levels, V_{DD} and V_{SS} .

polling — Periodically reading a status bit to monitor the condition of a peripheral device.

port — A set of wires for communicating with off-chip devices.

prescaler — A circuit that generates an output signal related to the input signal by a fractional scale factor such as 1/2, 1/8, 1/10 etc.

program — A set of computer instructions that cause a computer to perform a desired operation or operations.

program counter (PC) — A 16-bit register in the CPU08. The PC register holds the address of the next instruction or operand that the CPU will use.

pull — An instruction that copies into the accumulator the contents of a stack RAM location. The stack RAM address is in the stack pointer.

pullup — A transistor in the output of a logic gate that connects the output to the logic 1 voltage of the power supply.

pulse-width — The amount of time a signal is on as opposed to being in its off state.

pulse-width modulation (PWM) — Controlled variation (modulation) of the pulse width of a signal with a constant frequency.

push — An instruction that copies the contents of the accumulator to the stack RAM. The stack RAM address is in the stack pointer.

PWM period — The time required for one complete cycle of a PWM waveform.

RAM — Random access memory. All RAM locations can be read or written by the CPU. The contents of a RAM memory location remain valid until the CPU writes a different value or until power is turned off.

RC circuit — A circuit consisting of capacitors and resistors having a defined time constant.

read — To copy the contents of a memory location to the accumulator.

register — A circuit that stores a group of bits.

reserved memory location — A memory location that is used only in special factory test modes. Writing to a reserved location has no effect. Reading a reserved location returns an unpredictable value.

reset — To force a device to a known condition.

ROM — Read-only memory. A type of memory that can be read but cannot be changed (written). The contents of ROM must be specified before manufacturing the MCU.

SCI — See "serial communication interface module (SCI)."

serial — Pertaining to sequential transmission over a single line.

serial communications interface module (SCI) — A module in the M68HC08 Family that supports asynchronous communication.

serial peripheral interface module (SPI) — A module in the M68HC08 Family that supports synchronous communication.

set — To change a bit from logic 0 to logic 1; opposite of clear.

shift register — A chain of circuits that can retain the logic levels (logic 1 or logic 0) written to them and that can shift the logic levels to the right or left through adjacent circuits in the chain.

signed — A binary number notation that accommodates both positive and negative numbers. The most significant bit is used to indicate whether the number is positive or negative, normally logic 0 for positive and logic 1 for negative. The other seven bits indicate the magnitude of the number.

software — Instructions and data that control the operation of a microcontroller.

software interrupt (SWI) — An instruction that causes an interrupt and its associated vector fetch.

SPI — See "serial peripheral interface module (SPI)."

stack — A portion of RAM reserved for storage of CPU register contents and subroutine return addresses.

stack pointer (SP) — A 16-bit register in the CPU08 containing the address of the next available storage location on the stack.

Glossary

start bit — A bit that signals the beginning of an asynchronous serial transmission.

status bit — A register bit that indicates the condition of a device.

stop bit — A bit that signals the end of an asynchronous serial transmission.

subroutine — A sequence of instructions to be used more than once in the course of a program. The last instruction in a subroutine is a return from subroutine (RTS) instruction. At each place in the main program where the subroutine instructions are needed, a jump or branch to subroutine (JSR or BSR) instruction is used to call the subroutine. The CPU leaves the flow of the main program to execute the instructions in the subroutine. When the RTS instruction is executed, the CPU returns to the main program where it left off.

synchronous — Refers to logic circuits and operations that are synchronized by a common reference signal.

TIM — See "timer interface module (TIM)."

timer interface module (TIM) — A module used to relate events in a system to a point in time.

timer — A module used to relate events in a system to a point in time.

toggle — To change the state of an output from a logic 0 to a logic 1 or from a logic 1 to a logic 0.

tracking mode — Mode of low-jitter PLL operation during which the PLL is locked on a frequency. Also see "acquisition mode."

two's complement — A means of performing binary subtraction using addition techniques. The most significant bit of a two's complement number indicates the sign of the number (1 indicates negative). The two's complement negative of a number is obtained by inverting each bit in the number and then adding 1 to the result.

unbuffered — Utilizes only one register for data; new data overwrites current data.

unimplemented memory location — A memory location that is not used. Writing to an unimplemented location has no effect. Reading an unimplemented location returns an unpredictable value. Executing an opcode at an unimplemented location causes an illegal address reset.

V — The overflow bit in the condition code register of the CPU08. The CPU08 sets the V bit when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow bit.

variable — A value that changes during the course of program execution.

VCO — See "voltage-controlled oscillator."

vector — A memory location that contains the address of the beginning of a subroutine written to service an interrupt or reset.

voltage-controlled oscillator (VCO) — A circuit that produces an oscillating output signal of a frequency that is controlled by a dc voltage applied to a control input.

waveform — A graphical representation in which the amplitude of a wave is plotted against time.

wired-OR — Connection of circuit outputs so that if any output is high, the connection point is high.

word — A set of two bytes (16 bits).

write — The transfer of a byte of data from the CPU to a memory location.

X — The lower byte of the index register (H:X) in the CPU08.

Z — The zero bit in the condition code register of the CPU08. The CPU08 sets the zero bit when an arithmetic operation, logical operation, or data manipulation produces a result of \$00.

Literature Updates

This document contains the latest data available at publication time. For updates, contact one of the centers listed below:

Literature Distribution Centers

Order literature by mail or phone.

USA/Europe Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado, 80217
Phone 1-303-675-2140

US & Canada only <http://sps.motorola.com/mfax>

Japan Nippon Motorola Ltd.
Tatsumi-SPD-JLDC
Toshikatsu Otsuki
6F Seibu-Butsuryu Center
3-14-2 Tatsumi Koto-Ku
Tokyo 135, Japan
Phone 03-3521-8315

Literature Updates

Hong Kong

Motorola Semiconductors H.K. Ltd.
8B Tai Ping Industrial Park
51 Ting Kok Road
Tai Po, N.T., Hong Kong
Phone 852-26629298

Customer Focus Center

1-800-521-6274

Mfax

To access this worldwide faxing service call or contact by electronic mail or the internet:

RMFAX0@email.sps.mot.com
TOUCH-TONE 1-602-244-6609
<http://sps.motorola.com/mfax>

Motorola SPS World Marketing World Wide Web Server

Use the Internet to access Motorola's World Wide Web server. Use the following URL:

<http://design-net.com>

Microcontroller Division's Web Site

Directly access the Microcontroller Division's web site with the following URL:

http://design-net.com/csic/CSIC_home.html

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

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

Freescale Semiconductor, Inc.

Home Page:

www.freescale.com

email:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130

support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

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

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

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



MC68HC05E6/D

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