MC68HC05F32/D

HC05

MC68HC05F32
MC68HC705F32

TECHNICAL
DATA

MOTOROLA

MC68HC05F32

TECHNICAL DATA

MOTOROLA

MC68HC05F32
MC68HC705F32

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

All Trade Marks recognized. This document contains information on new products. Specifications and information herein are subject to change without notice.

All products are sold on Motorola’s Terms & Conditions of Supply. In ordering a product covered by this document the Customer agrees to be bound by those Terms & Conditions and nothing contained in this document constitutes or forms part of a contract (with the exception of the contents of this Notice). A copy of Motorola’s Terms & Conditions of Supply is available on request.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

The Customer should ensure that it has the most up to date version of the document by contacting its local Motorola office. This document supersedes any earlier documentation relating to the products referred to herein. The information contained in this document is current at the date of publication. It may subsequently be updated, revised or withdrawn.

Conventions

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

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

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

– Cut along this line to remove –

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC05F32/D)

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

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

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

Comments: ______________________________

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

Selection of device for new application □ Other □ Please specify: ______________

System design □ ______________

Training purposes □ ______________

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

Completely □ □ □ □ Not at all Comments: ______________

____

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

Easy □ □ □ □ Difficult Comments: ______________

____

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

		Too little detail				Too much detail
SECTION 1	INTRODUCTION	□	□	□	□	□
SECTION 2	MODES OF OPERATION AND PIN DESCRIPTIONS	□	□	□	□	□
SECTION 3	MEMORY AND REGISTERS	□	□	□	□	□
SECTION 4	PARALLEL INPUT/OUTPUT PORTS	□	□	□	□	□
SECTION 5	CORE TIMER	□	□	□	□	□
SECTION 6	16-BIT PROGRAMMABLE TIMER	□	□	□	□	□
SECTION 7	DTMF/MELODY GENERATOR	□	□	□	□	□
SECTION 8	LIQUID CRYSTAL DISPLAY DRIVER MODULE	□	□	□	□	□
SECTION 9	A/D CONVERTER	□	□	□	□	□
SECTION 10	SERIAL PERIPHERAL INTERFACE	□	□	□	□	□
SECTION 11	SERIAL COMMUNICATIONS INTERFACE	□	□	□	□	□
SECTION 12	PULSE WIDTH MODULATOR	□	□	□	□	□
SECTION 13	32 KHZ CLOCK SYSTEM	□	□	□	□	□
SECTION 14	RESETS AND INTERRUPTS	□	□	□	□	□
SECTION 15	CPU CORE AND INSTRUCTION SET	□	□	□	□	□
SECTION 16	ELECTRICAL SPECIFICATIONS	□	□	□	□	□
SECTION 17	MECHANICAL DATA	□	□	□	□	□
SECTION 18	ORDERING INFORMATION	□	□	□	□	□
SECTION 19	APPENDICES	□	□	□	□	□

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

____

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

____

8. How could we improve this document? ________________________________

____

____

9. How would you rate Motorola’s documentation?

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

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

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

12. How many years have you worked with microprocessors?

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

– Second fold back along this line –

By air mail
Par avion

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

NE PAS AFFRANCHIR
NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

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

F.A.O. Technical Publications Manager
(re: MC68HC05F32/D)

MOTOROLA
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

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

____

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

Name: ________________ Phone No: ________________

Position: ________________ FAX No: ________________

Department: ________________

Company: ________________

Address: ________________

____

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

– Finally, tuck this edge into opposite flap –

TABLE OF CONTENTS

Paragraph Number | TITLE | Page Number

1
INTRODUCTION

2
MODES OF OPERATION AND PIN DESCRIPTIONS

3
MEMORY AND REGISTERS

4
PARALLEL INPUT/OUTPUT PORTS

5
CORE TIMER

6
16-BIT PROGRAMMABLE TIMER

7
DTMF/MELODY GENERATOR

8
LIQUID CRYSTAL DISPLAY DRIVER MODULE

9
A/D CONVERTER

10
SERIAL PERIPHERAL INTERFACE

11
SERIAL COMMUNICATIONS INTERFACE

12
PULSE WIDTH MODULATOR

13
32 KHZ CLOCK SYSTEM

14
RESETS AND INTERRUPTS

15
CPU CORE AND INSTRUCTION SET

16
ELECTRICAL SPECIFICATIONS

17
MECHANICAL DATA

18
ORDERING INFORMATION

A
MC68HC705F32

GLOSSARY

INDEX

THIS PAGE LEFT BLANK INTENTIONALLY

LIST OF FIGURES

Figure Number | TITLE | Page Number

LIST OF TABLES

Table Number | TITLE | Page Number

Table Number | TITLE | Page Number

1
INTRODUCTION

The MC68HC05F32 is a member of the M68HC05 family of HCMOS microcomputers. Its memory configuration comprises 32K bytes of ROM, 920 bytes of RAM and 256 bytes of EEPROM. The on-board features of this device make it particularly suitable for use in highly integrated telephone handsets; the timer and DTMF generator allow for both pulse and tone dialling and, in addition to telephone set-up parameters and features such as last number redial, the EEPROM can typically store up to 12 telephone numbers of 20 digits, even after power has been removed from the circuit. Other features of the device include the keyboard interrupt facility, which allows a direct interface to a telephone keypad, the LCD circuit, which can drive up to 160 segments of an LCD display, and the A/D converter which could be used, for example, as a volume control for a telephone in hands-free mode. A high level of integration has been achieved on the MC68HC05F32 and careful attention has been paid to its low-power and low-voltage performance, a major consideration in many telecommunications applications.

The MC68HC05F32 is very well suited to automotive applications; with its 8 analog inputs and many general I/O lines, it is especially useful in applications such as car dashboards. Also, the voltage levels of the LCD driver can be varied using external resistors, and the timer system is capable of driving two stepper motors (e.g. speedometer and odometer), as well as controlling a real time clock. The SCI subsystem is ideal for interfacing to diagnostic equipment, for example, and the on-board EEPROM can be used to store data such as mileage or calibration information.

This data sheet is structured such that devices similar to the MC68HC05F32 are described in a set of appendices.

Table 1-1 Data sheet appendices

Device	Appendix	Differences from MC68HC05F32
MC68HC705F32	A	32256 bytes EPROM; 496 bytes boot ROM

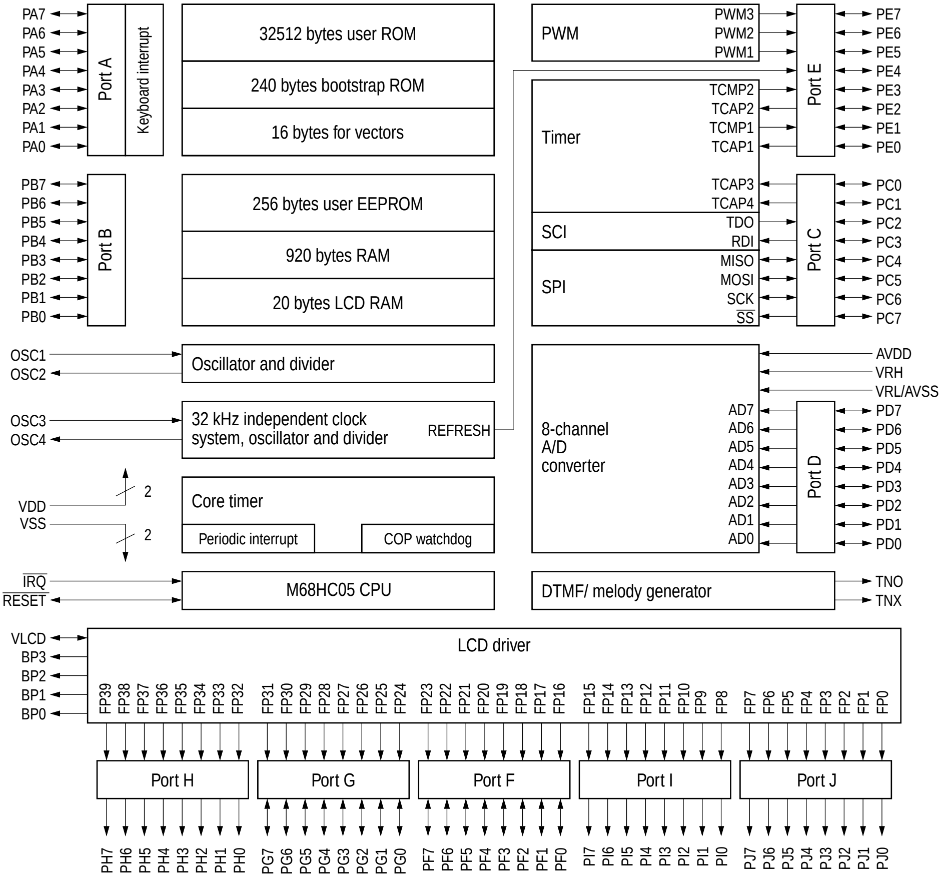
1.1 Features

- Fully static design featuring the industry-standard M68HC05 CPU core
- 32512 bytes of user ROM, plus 16 bytes for vectors
- 240 bytes of bootloader ROM
- 920 bytes of RAM plus 20 bytes of LCD RAM
- 256 bytes of user EEPROM
- DTMF/melody generator
- 16-bit programmable timer with four input captures and four output compares (the outputs of two of the output compares are used internally and do not have external connections)
- 15 stage multipurpose core timer with timer overflow, real time interrupt and COP watchdog
- LCD driver with 4 backplanes and 40 frontplanes
- 8-channel, 8-bit analog-to-digital (A/D) converter
- Power saving STOP and WAIT modes
- I/O lines
  - 100 QFP configuration – total of 80 I/O pins configured as:
    16 dedicated bidirectional I/O
    64 shared with peripherals
  - 80 QFP configuration – total of 69 I/O pins configured as:
    16 dedicated bidirectional I/O
    53 shared with peripherals
- Keyboard interrupt facility on eight of the I/O lines, with high or low voltage level interrupt triggers
- Hardware interrupt with edge or edge-and-level sensitive interrupt trigger
- SCI and SPI subsystems
- On-chip oscillators
- Three PWM channels
- Two selectable bus frequencies
- 32 kHz independent clock system
- Power-on and power-off resets; low voltage detection circuitry (EEPROM)
- Available in 100-pin QFP and 80-pin QFP

Note: The 80-pin version is only a bond option. Pins PE4, PD7–PD0, PC4, PC5 are shared with module functions which cannot work on the 80-pin package. These modules and their corresponding pin functions should not be enabled.

1.2 Mask options for the MC68HC05F32

There are three mask options available on the MC68HC05F32: STOP instruction (enable/disable), COP watchdog timer (enable/disable) and low voltage reset (LVR – enable/disable). These options are programmed during fabrication and must be specified by the customer at the time of ordering.



Note: When not being used to output the LCD frontplanes, port G and port F pins are input only, while port H, port I and port J pins are output only.

Figure 1-1 MC68HC05F32 block diagram

THIS PAGE LEFT BLANK INTENTIONALLY

2
MODES OF OPERATION AND PIN DESCRIPTIONS

The normal operating mode of the MC68HC05F32 is single chip mode. There is also a bootloader mode, primarily for factory test purposes. In addition to these modes, there are three low power modes which may be entered and exited at will from user mode: STOP, WAIT and data retention.

2.1 Single-chip mode

This is the normal user operating mode, in which the device functions as a self-contained microcomputer unit, with all on-board peripherals and I/O ports available to the user. All address and data activity occurs within the MCU.

2.2 Low power modes

2.2.1 STOP mode

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, 16-bit timers, SPI, SCI, PWM and A/D converter.

During STOP mode, the core timer interrupt flags (CTOF and RTIF) and interrupt enable bits (TOFE and RTIE) in the CTCSR as well as the 16-bit timer flags in register TSR and interrupt enable bits in register TCR are cleared by internal hardware. 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 interrupt ($\overline{\text{IRQ}}$, Keyboard, LVI or CPI from the 32 kHz clock system) if enabled or $\overline{\text{RESET}}$ (external reset or low voltage reset – LVR). See Figure 2-1.

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

2.2.2 WAIT mode

The WAIT instruction places the MCU in a low power consumption mode, though it consumes more power than in STOP mode. All CPU action is suspended, but the Core timer, the first 16-bit timer (TCAP1, 2 and TCMP1, 2), the DMG and the LCD remain active. If bit 7 (WTLCDO) of the LCD control register, $1E, is reset, the SPI, the SCI, the second 16-bit timer (TCAP3, 4 and TCMP3, 4) and the A/D converter, also remain active in WAIT mode. If, however, WTLCDO is set they are turned off.

An interrupt from the core timer, 16-bit timers, SPI, SCI, $\overline{\text{IRQ}}$, keyboard, LVI, OR CPI from the 32 kHz clock system, if enabled, will cause the MCU to exit the WAIT mode. An external reset, or LVR, causes the MCU to exit the 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. See Figure 2-1.

2.2.3 Data retention mode

The contents of the RAM and CPU registers are retained at supply voltages as low as 2.0Vdc. This is called the data retention mode, in which data is maintained but the device is not guaranteed to operate. If the voltage drops below V_{ROFF} the low voltage reset circuit generates a reset.

For lowest power consumption in data retention mode the device should be put into STOP mode before reducing the supply voltage, to ensure that all the clocks are stopped. If the device is not in STOP mode then it is recommended that $\overline{\text{RESET}}$ be held low whilst the power supply is outwith the normal operating range, to ensure that processing is suspended in an orderly manner.

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

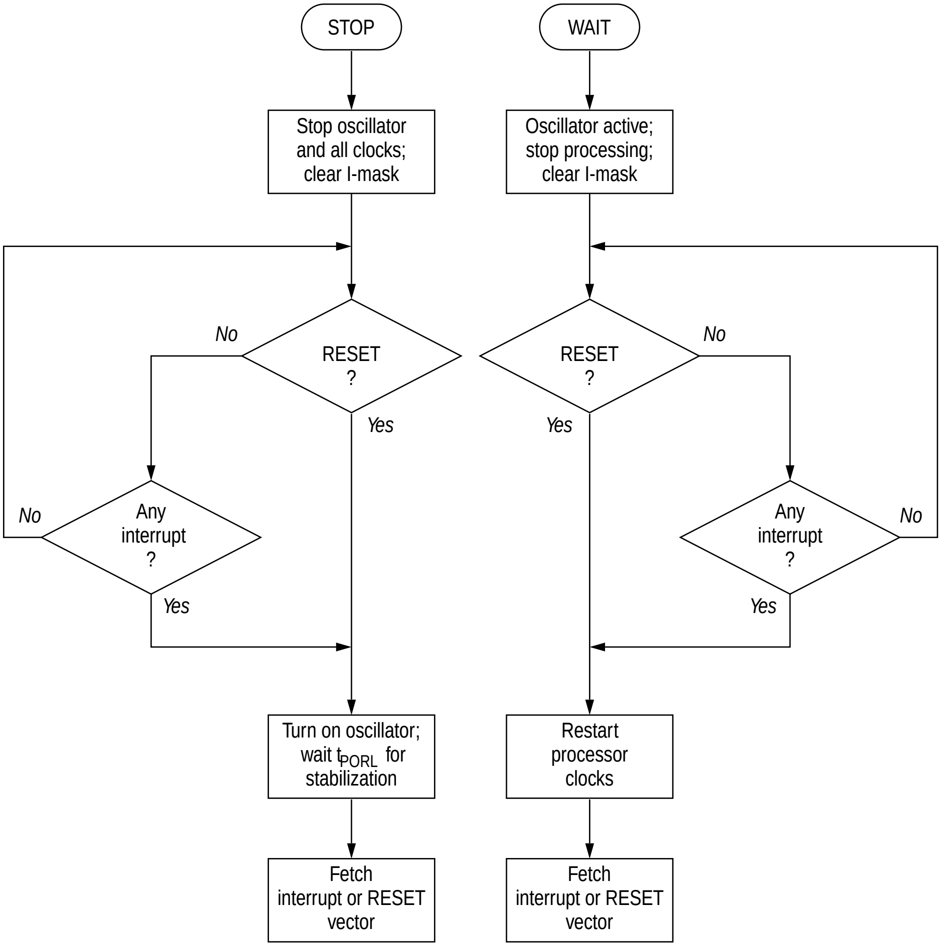

STOP
WAIT
Stop oscillator and all clocks; clear I-mask
Oscillator active; stop processing; clear I-mask
RESET ?
No
Yes
RESET ?
No
Yes
Any interrupt ?
No
Yes
Any interrupt ?
No
Yes
Turn on oscillator; wait t_{PORL} for stabilization
Restart processor clocks
Fetch interrupt or RESET vector
Fetch interrupt or RESET vector

Figure 2-1 STOP and WAIT flowcharts

2.3 System options register (SOR)

The MC68HC05F32 MCU contains a System Option Register which is located at address $4D. This register is used to control the LVI and the clock system.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System options register (SOR)	$004D	LVIF	LVIE	LVION	SC	IRQ	KEYMUX	KEYCLR	PUEN	0000 0000

LVIF, LVIE, LVION — Low voltage interrupt bits

The LVIF flag is set by the low voltage detection circuit, if the LVI is enabled and power supply VDD falls below V_{lvi}.

The low voltage interrupt must be enabled by first setting bit LVION Low Voltage Interrupt On and after that setting bit LVIE Low Voltage Interrupt Enable. After power on reset the LVI circuit is disabled.

SC — System clock option

After power on reset the internal bus frequency is f=3.58Mhz/2. If the bit SC System Clock is set the system speed is reduced to f=3.58Mhz/4, with the exception of the DTMF generator (Oscillator Frequency 3.58Mhz).

IRQ — Interrupt sensitivity

IRQ edge or level sensitivity

1 (set)	–	IRQ input edge and level sensitive
0 (clear)	–	IRQ input edge sensitive

KEYMUX — Multiplex bit for access of interrupt flag

The KEYMUX bit switches between the port A data register and the interrupt status register IRSTATE, that both have the address $0000. If KEYMUX is cleared normal read and write access to port A is possible. If KEYMUX is set, a read or write operation at address $0000 accesses the 8 interrupt status flags.

KEYCLR — Keyboard interrupt clear

The keyboard wake-up interrupt status flag (Bit 7, $1B) is cleared by writing a “1” to bit KEYCLR. A read access to this bit always returns “0”.

PUEN — PORTC pull-up enable

After power on reset the pull-up resistors in port C are disabled. If bit PUEN is set, the pull-up resistors in port C are enabled. Writing a “0” to PUEN disables the pull-up function.

2.4 Pin descriptions

2.4.1 VDD and VSS

Power is supplied to the microcomputer via these two pins. VDD is the positive supply pin and VSS is the ground pin.

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.

2.4.2 $\overline{\text{IRQ}}$

This is an input-only pin for external interrupt sources. Interrupt triggering is selected using the IRQ bit in the SOR register, to be one of two options: either edge and level sensitive or edge sensitive only.

The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity.

2.4.3 $\overline{\text{RESET}}$

This active low I/O pin is used to reset the MCU. Applying a logic zero to this pin forces the device to a known start-up state. An external RC-circuit can be connected to this pin to generate a power-on reset (POR) if required. In this case, the time constant must be great enough (at least 100ms) to allow the oscillator circuit to stabilise. This input has an internal Schmitt trigger to improve noise immunity. When a low voltage reset condition occurs internally, the $\overline{\text{RESET}}$ pin provides an active-low open drain output signal that may be used to reset external hardware. Other internal reset conditions are not visible at the $\overline{\text{RESET}}$ pin.

2.4.4 PA7–PA0/keyboard interrupt, PB7–PB0

These 16 I/O lines comprise the two 8-bit ports A and B. The state of any pin is software programmable, and on reset, the port pins are configured as inputs, with internal pull-up resistors. The eight I/O lines of port A are shared with the keyboard interrupt function.

2.4.5 PC7/$\overline{\text{SS}}$, PC6/SCK, PC5/MOSI, PC4/MISO, PC3/TDO, PC2/RDI, PC1/TCAP4, PC0/TCAP3

These eight I/O lines comprise the 8-bit port C, and are shared with other functions to give added flexibility. During reset, these lines are configured as inputs. Port pins PC0 and PC1 are shared with the input timer capture TCAP3 and TCAP4. Pins PC2 and PC3 are connected to the SCI system (RDI, TDO), if the SCI is enabled. The remaining four pins, PC7–PC4, are connected to the SPI system ($\overline{\text{SS}}$, SCK, MOSI, MISO), if the SPI is enabled. All eight lines have internal programmable pull-ups. If the PUEN bit in the system options register is cleared, the pull-ups are disabled after reset. Setting the PUEN bit enables all the pull-up resistors in port C.

2.4.6 PD7–PD0/AN7–AN0

The eight I/O lines of port D are configured as inputs during power-on or reset. As all port D output are open-drain, an external pull-up resistor is needed when a pin is being used as an output. These port lines, PD7–PD0, are shared with the A/D converter, and are connected to it when the corresponding port D control register bit is set to 1.

2.4.7 VRH

The VRH pin is the positive reference voltage for the A/D converter.

2.4.8 AVDD

AVDD is the positive supply voltage for the A/D converter.

2.4.9 AVSS

AVSS is the negative supply voltage and the negative reference voltage for the A/D converter.

2.4.10 PE7/PWM3, PE6/PWM2, PE5/PWM1, PE4/REFRESH, PE3/TCMP2, PE2/TCAP2, PE1/TCMP1, PE0/TCAP1

The pins PE7–PE0 comprise port E, providing eight I/O lines when the port E control bits are set to 0. As these pins are open-drain, an external pull-up resistor is needed when a pin is being used as an output. These pins also share functions. When the corresponding port E register control bit is set to 1, pins PE3–PE0 are connected to the timer system (TCMP2, TCAP2, TCMP1, TCAP1), pin PE4 becomes REFRESH, and pins PE7–PE5 are connected to the PWM (PWM3–PWM1).

2.4.11 BP3–BP0

The LCD driver subsystem has a maximum of four backplanes and forty frontplanes configured under software control. The pins BP3–BP0 provide the backplane drive signals and the forty output lines FP39–FP0 provide the frontplane drive signals for the LCD unit. The forty frontplane lines are shared with ports F, G, H, I and J.

2.4.12 VLCD

The analogue part of the LCD controller can be supplied with an external voltage, V_{LCD}, using the VLCD pin. The value of V_{LCD} may not exceed the positive power supply voltage $V_{DD.}$ When the INTVLCD bit in the LCD control register is set to 1, an internal voltage generator (approx. 3V, if V_{DD}>3V) is activated as the source of the analogue LCD supply voltage.

2.4.13 Ports F, G, H, I, J/FP39–FP0

These five ports are shared with the frontplanes FP39–FP0. The default setting of the register control bits is 0, setting all the pins in ports F and G input only, and all the pins in ports H, I and J output only. The port J outputs are all open-drain. When a register control bit is set to 1, the corresponding pin is connected to the LCD frontplane driver.

2.4.14 TNO and TNX

The TNO output provides dual tone DTMF or melody under program control. TNO is an open-drain output, and therefore requires an external pull-up resistor. The TNX output provides pacifier tones under program control.

2.4.15 OSC1 and OSC2

These pins provide control input for an on-chip oscillator circuit. A crystal or external clock signal connected to these pins supplies the oscillator clock. The oscillator frequency of 3.579 MHz provides the time base for the real-time clock and the DTMF/melody generator.

2.4.16 OSC3 and OSC4

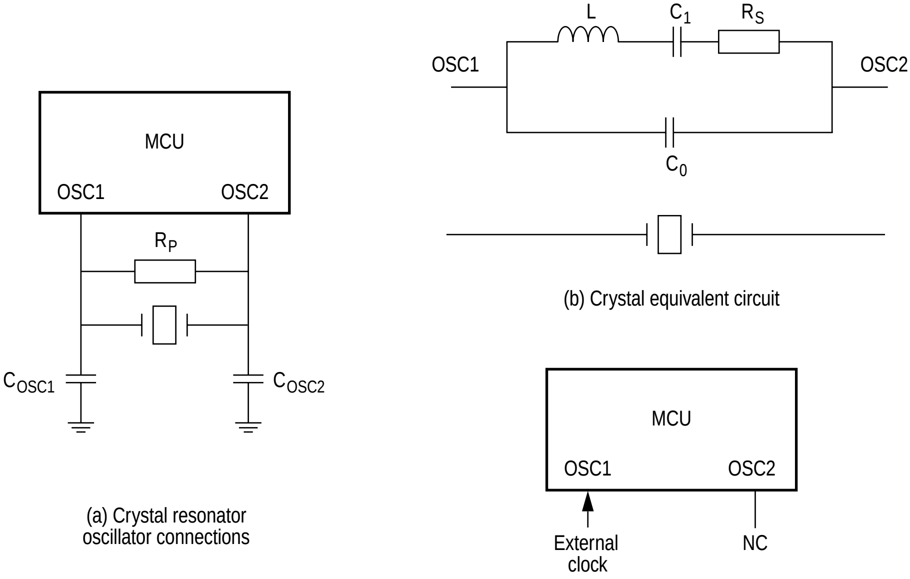
These pins provide control input for an independent on-chip oscillator circuit. A 32 kHz crystal connected across these pins, or an external clock signal connected to OSC3 provides the separate clock. The oscillator frequency (f_{OSC}=32 kHz) provides the time base for the divider, the real time custom periodic interrupt (CPI) and the clock system output (REFRESH).

2.4.16.1 Crystal

The circuit shown in Figure 2-2(a) is recommended when using either a crystal or a ceramic resonator. Figure 2-2(d) provides the recommended capacitance and feedback resistance values. The internal oscillator is designed to interface with an AT-cut parallel-resonant quartz crystal resonator in the frequency range specified for f_{OSC} (see Section 16.4). Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and associated components should be mounted as close as possible to the input pins to minimize output distortion and start-up stabilization time. The manufacturer of the particular crystal being considered should be consulted for specific information.

2.4.16.2 External clock

An external clock should be applied to the OSC1 input, with the OSC2 pin left unconnected, as shown in Figure 2-2(c). The t_{OXOV} specification (see Section 16.4) does not apply when using an external clock input. The equivalent specification of the external clock source should be used in lieu of t_{OXOV}.



(a) Crystal resonator oscillator connections

(b) Crystal equivalent circuit

(c) External clock source connections

Crystal

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

(d) Crystal resonator parameters

Figure 2-2 Oscillator connections

2.5 Alternative pin descriptions for the 80-pin QFP package

There is also an 80-pin version of the MC68HC05F32. As it has fewer pins and fewer modules, some of the pin descriptions vary. The reduction of the I/O count means that there is no longer a port H and that port C has only three pins available for use, one of which is shared with the timer (TCAP3). Port D's pins were shared with the A/D converter, but this can no longer be used.

2.5.1 PC5, PC4, PC0/TACP3

The three I/O lines of port C are configured as inputs during reset and each one has an internal pull-up resistor. Pin PC0 is shared with one of the timer's input captures (TCAP3).

2.5.2 PD7–PD0

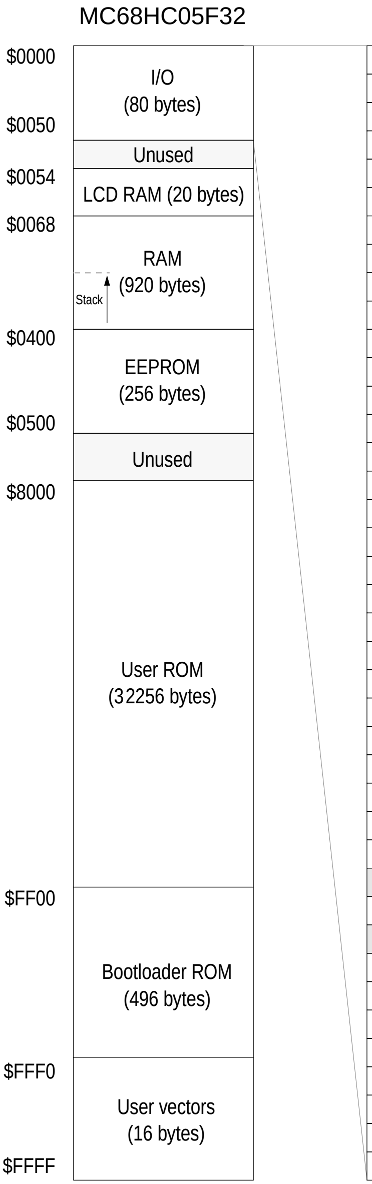
All eight port D lines are configured as inputs during reset. These pins are open drain outputs which means that each one requires an external pull-up resistor when it is used as an output.

3
MEMORY AND REGISTERS

The MC68HC05F32 has a 64K byte memory map consisting of registers (for I/O, control and status), user RAM, user ROM, EEPROM, bootloader ROM and reset and interrupt vectors as shown in Figure 3-1.

3.1 Registers

All the I/O, control and status registers of the MC68HC05F32 are contained within the first 80 byte block of the memory map, as detailed in Table 3-1.



$00 Port A data (PORTA)
$01 Port B data (PORTB)
$02 Port C data (PORTC)
$03 Port D data (PORTD)
$04 Port A DDR (DDRA)
$05 Port B DDR (DDRB)
$06 Port C DDR (DDRC)
$07 Port D DDR (DDRD)
$08 Ctimer control/status (CTCSR)
$09 Ctimer counter (CTCR)
$0A Port E data (PORTE)
$0B Port E DDR (DDRE)
$0C Port E control (PECR)
$0D Row freq. control (FCR)
$0E Column freq. control (FCC)
$0F Tone control (TNCR)
$10 Port F data (PORTF)
$11 Port F control (PFCR)
$12 Port G data (PORTG)
$13 Port G control (PGCR)
$14 Port H data (PORTH)[1]
$15 Port H control (PHCR)[1]
$16 Port I data (PORTI)
$17 Port I control (PICR)
$18 Port J data (PORTJ)
$19 Port J control (PJCR)
$1A Port D control (PDCR)
$1B Key control (KCR)
$1C EEPROM prog. (EEPROG)
▨
$1E LCD control (LCD)
▨
$20 Capture 1 high (ICR1H)
$21 Capture 1 low (ICR1L)
$22 Compare 1 high (OCR1H)
$23 Compare 1 low (OCR1L)
$24 Capture 2 high (ICR2H)
$25 Capture 2 low (ICR2L)
$26 Compare 2 high (OCR2H)
$27 Compare 2 low (OCR2L)

$28 Counter 1 high (CNTH/1)
$29 Counter 1 low (CNTL/1)
$2A Alt. counter high 1 (ACNTH/1)
$2B Alt. counter low 1 (ACNTL/1)
$2C Timer 1 control 1 (TCR1/1)
$2D Timer 1 control 2 (TCR2/1)
$2E Timer 1 status (TSR/1)
▨
$30 Capture 3 high (ICR3H)
$31 Capture 3 low (ICR3L)
$32 Compare 3 high (OCR3H)
$33 Compare 3 low (OCR3L)
$34 Capture 4 high (ICR4H)
$35 Capture 4 low (ICR4L)
$36 Compare 4 high (OCR4H)
$37 Compare 4 low (OCR4L)
$38 Counter 2 high (CNTH/2)
$39 Counter 2 low (CNTL/2)
$3A Alt. counter high 2 (ACNTH/2)
$3B Alt. counter low 2 (ACNTL/2)
$3C Timer 2 control 1 (TCR1/2)[1]
$3D Timer 2 control 2 (TCR2/2)[1]
$3E Timer 2 status (TSR/2)[1]
▨
$40 PWM control (PWMCR)
$41 PWM data 1 (PWMD1)
$42 PWM data 2 (PWMD2)
$43 PWM data 3 (PWMD3)
$44 SPI control (SPCR)[1]
$45 SPI status (SPSR)[1]
$46 SPI data I/O (SPDAT)[1]
$47 SCI data (SCDAT)[1]
$48 SCI control 1 (SCCR1)[1]
$49 SCI control 2 (SCCR2)[1]
$4A SCI status (SCSR)[1]
$4B SCI baud rate (BAUD)[1]
$4C CPI control/status (CPICSR)
$4D System options (SOR)
$4E A/D data (ADDATA)[1]
$4F A/D status/control (ADSCR)[1]

[1] Not applicable to 80-pin package.

▨ – reserved

Figure 3-1 Memory map of the MC68HC05F32

Table 3-1 Register outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined
Key interrupt status (KISR)	$0000									0000 0000
Port B data (PORTB)	$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined
Port C data (PORTC)	$0002	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined
Port D data (PORTD)	$0003	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	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	TOF	RTIF	TOFE	RTIE	RTOF	RRTIF	RT1	RT0	0000 0011
Core timer counter (CTCR)	$0009									0000 0000
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined
Port E data direction (DDRE)	$000B									0000 0000
Port E control (PECR)	$000C						0		0	0000 0000
DTMF row freq. control (FCR)	$000D	0	0	0	FCR4	FCR3	FCR2	FCR1	FCR0	undefined
DTMF column freq. control (FCC)	$000E	0	0	0	FCC4	FCC3	FCC2	FCC1	FCC0	undefined
DTMF tone control (TNCR)	$000F	MS1	MS0	TGER	TGEC	TNOE	0	0	0	0000 0000
Port F data (PORTF)	$0010	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined
Port F control (PFCR)	$0011									0000 0000
Port G data (PORTG)	$0012	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	undefined
Port G control (PGCR)	$0013									0000 0000
Port H data (PORTH)	$0014	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	0000 0000
Port H control (PHCR)	$0015									0000 0000
Port I data (PORTI)	$0016	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0	0000 0000
Port I control (PICR)	$0017									0000 0000
Port J data (PORTJ)	$0018	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	0000 0000
Port J control (PJCR)	$0019									0000 0000
Port D control (PDCR)	$001A									0000 0000
Key control (KCR)	$001B	KF	KIE	EDG5	EDG4	EDG3	EDG2	EDG1	EDG0	0000 0000
EEPROM prog. (EEPROG)	$001C	0	CPEN	0	ER1	ER0	LATCH	EERC	EEPGM	0000 0000
LCD control (LCD)	$001E	WTLCDO	FSEL1	FSEL0 I	NTVLCD	FDISP	MUX4	MUX3	EXTVON	0000 0000
Capture 1 high (ICR1H)	$0020	(bit 15)							(bit 8)	undefined
Capture 1 low (ICR1L)	$0021									undefined

Table 3-1 Register outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Compare 1 high (OCR1H)	$0022	(bit 15)							(bit 8)	undefined
Compare 1 low (OCR1L)	$0023									undefined
Capture 2 high (ICR2H)	$0024	(bit 15)							(bit 8)	undefined
Capture 2 low (ICR2L)	$0025									undefined
Compare 2 high (OCR2H)	$0026	(bit 15)							(bit 8)	undefind
Compare 2 low (OCR2L)	$0027									undefined
Counter 1 high (CNTH/1)	$0028	(bit 15)							(bit 8)	1111 1111
Counter 1 low (CNTL/1)	$0029									1111 1100
Alternate counter 1 high (ACNTH/1)	$002A	(bit 15)							(bit 8)	1111 1111
Alternate counter 1 low (ACNTL/1)	$002B									1111 1100
Timer1 control 1 (TCR1/1)	$002C	ICI1E	ICI2E	OCI1E	TOIE	CO1E	IEDG1	IEDG2	OLVL1	0000 0uu0
Timer1 control 2 (TCR2/1)	$002D	0	0	OCI2E	0	CO2E	0	0	OLVL2	0000 0000
Timer1 status (TSR/1)	$002E	IC1F	IC2F	OC1F	TOF	TCAP1	TCAP2	OC2F	0	uuuu uuu0
Capture 3 high (ICR3H)	$0030	(bit 15)							(bit 8)	undefined
Capture 3 low (ICR3L)	$0031									undefined
Compare 3 high (OCR3H)	$0032	(bit 15)							(bit 8)	undefined
Compare 3 low (OCR3L)	$0033									undefined
Capture 4 high (ICR4H)	$0034	(bit 15)								undefined
Capture 4 low (ICR4L)	$0035									undefined
Compare 4 high (OCR4H)	$0036	(bit 15)								undefined
Compare 4 low (OCR4L)	$0037									undefined
Counter 1 high (CNTH/1)	$0038	(bit 15)							(bit 8)	1111 1111
Counter 1 low (CNTL/1)	$0039									1111 1100
Alternate counter 2 high (ACNTH/1)	$003A	(bit 15)							(bit 8)	1111 1111
Alternate counter 2 low (ACNTL/1)	$003B									1111 1100
Timer2 control 1 (TCR1/2)	$003C	ICI3E	ICI4E	OCI3E	TOIE	CO3E	IEDG3	IEDG4		0000 0uu0
Timer2 control 2 (TCR2/2)	$003D	0	0	OCI4E	0	CO4E	0	0		0000 0000
Timer2 status (TSR/2)	$003E	IC3F	IC4F	OC3F	TOF	TCAP3	TCAP4	OC4F	0	uuuu uuu0
PWM control (PWMCR)	$0040				POL3	POL2	POL1	RA1	RA0	0001 1100
PWM data 1 (PWMD1)	$0041									1000 0000
PWM data 2 (PWMD2)	$0042									1000 0000
PWM data 3 (PWMD3)	$0043									1000 0000
SPI control (SPCR)	$0044	SPIE	SPE	DOD	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu

Table 3-1 Register outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI status (SPSR)	$0045	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000
SPI data I/O (SPDAT)	$0046									undefined
SCI data (SCDAT)	$0047									undefined
SCI control 1 (SCCR1)	$0048	R8	T8	0	M	WAKE	0	0	0	uu00 0000
SCI control 2 (SCCR2)	$0049	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI status (SCSR)	$004A	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	1100 0000
SCI baud rate (BAUD)	$004B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	0000 0uuu
CPI control status (CPICSR)	$004C	0	CPIF	0	CPIE	0	0	RFQ1	RFQ0	0000 0000
System options (SOR)	$004D	LVIF	LVIE	LVION	SC	IRQ	KEYMUX	KEYCLR	PUEN	0000 0000
A/D data (ADDATA)	$004E									undefined
A/D status/control (ADSCR)	$004F	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0	0000 0000

u = undefined

3.2 RAM

The user RAM consists of 920 bytes of memory, from $0068 to $03FF. This is shared with a 64 byte stack area. The stack begins at $00FF, and may extend down to $00C0.

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

3.3 ROM

The user ROM occupies 32512 bytes of memory, from $8000 to $FEFF. In addition, there are 16 bytes of user vectors, from $FFF0 to $FFFF. The Bootloader ROM is located from $FF00 to $FFEF.

Note: For compatibility, unused bits (shaded) should always be cleared, when writing to them.

3.4 Bootloader ROM

The MC68HC05F32 has 224 bytes of bootloader ROM plus 16 bytes of bootloader vectors, from $FF00 to $FFEF. These are included primarily for factory test purposes.

3.5 EEPROM

256 bytes of user EEPROM reside at addresses $0400 to $04FF.

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

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

Caution: There is a restriction on the use of indexed addressing for EEPROM read operations. When the base address of an indexed read of an EEPROM location is within the EEPROM address range ($0400 to $04FF), the read may not be successful.

e.g. LDA (BASE ADDRESS), X – may not give the correct result when the base address is in the range $0400 to $04FF. However if the base address is outwith the EEPROM address range, the read operation will be successful. This restriction applies to all operations capable of using indexed addressing.

3.5.1 EEPROM programming register

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

CPEN — Charge pump enable

1 (set) – Charge pump enabled.

0 (clear) – Charge pump disabled.

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

ER1, ER0 — Erase select bits

ER1 and ER0 are used to select either single byte programming or one of three erase modes: byte, block, or bulk. Table 3-2 shows the mode selected for each bit configuration. These bits are readable and writable and are cleared by reset.

Table 3-2 Erase modes

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

- In byte erase mode, only the selected byte is erased.
- In block erase mode, a 32-byte block of EEPROM is erased. The EEPROM memory space is divided into four 64-byte blocks ($0400 – $043F, $0440 – $047F, $0480 – $04BF and $04C0 – $04FF) and performing a block erase on any address within a block will erase the entire block.
- In bulk erase mode, the entire 256 bytes of EEPROM are erased.

LATCH — EEPROM latch bit

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

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

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

EERC — EEPROM RC oscillator control

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

0 (clear) – Use CPU clock for EEPROM.

When this bit is set, the EEPROM memory array uses the internal RC oscillator instead of the CPU clock. After setting the EERC bit, the user should wait a time t_{RCON} to allow the RC oscillator to stabilize. This bit is readable and writable and should be set by the user when the internal bus frequency falls below 1.5 MHz. Reset clears this bit.

EEPGM — EEPROM programming power enable

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

0 (clear) – Programming power switched off.

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

3.5.2 Programming and erasing procedures

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

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

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

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

3.5.3 Sample EEPROM programming sequence

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

```
TCSR   EQU    $0008         TIMER CONTROL AND STATUS REGISTER
TCNT   EQU    $0009         TIMER COUNTER REGISTER
TOF    EQU    7             TOF BIT OF TCSR
PROG   EQU    $001C         EEPROM PROGRAM REGISTER
CPEN   EQU    6             CHARGE PUMP ENABLE BIT
ER1    EQU    4             ERASE SELECT BIT 1
ER0    EQU    3             ERASE SELECT BIT 0
LATCH  EQU    2             LATCH BIT
EERC   EQU    1             RC/OSC SELECTOR BIT
EEPGM  EQU    0             EEPROM PROGRAM BIT
EESTARTEQU    $0400         START ADDRESS OF EEPROM
SUMPIN EQU    $FF           DUMMY DATA

ORG           $0680
START  EQU    *
       BSET   EERC, PROG    SELECT RC OSCILLATOR
       BSR    DELAY         RC OSCILLATOR STABILIZATION
       BSET   CPEN, PROG    TURN ON CHARGE PUMP
       BSET   LATCH, PROG   ENABLE LATCH BIT
       BCLR   ER1, PROG     SELECT PROGRAM (NOT ERASE)
       BCLR   ER0, PROG     SELECT PROGRAM (NOT ERASE)

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

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

OUT    RTS

OUT1   SEC                  FLAG AN ERROR
       RTS

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

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

THIS PAGE LEFT BLANK INTENTIONALLY

4
PARALLEL INPUT/OUTPUT PORTS

The MC68HC05F32 has a total of 80 I/O lines, arranged as ten 8-bit ports. The I/O lines are individually programmable as either input or output, under the software control of the data direction registers. Port A can also 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 in output mode.

4.1 Input/output programming

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

At power-on or reset, all DDRs are cleared, thus configuring all port pins as inputs. The data direction registers can be written to or read by the MCU. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin. The operation of the standard port hardware is shown schematically in Figure 4-2.

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

4.2 Port A

Port A is an 8-bit bidirectional port which is equipped with a keyboard interrupt. All eight lines have internal pull-up resistors, which are required when the port is in input mode. On reset, this port is configured as a standard I/O port comprising a data register and a data direction register.

Reset does not affect the state of the data register, but clears the data direction register, thereby returning all ports pins to input mode. Writing a 1 to any DDR bit sets the corresponding port pin to output mode. As every pin configured as an input contributes to the keyboard interrupt, it is possible to disable a single pin by configuring it as an output.

4.2.1 Keyboard interrupt

Provided that the interrupt mask bit of the condition code register is cleared, the keyboard interrupt facility is enabled by setting the keyboard interrupt bit (KIE) in the key control register.

On detection of a high-to-low transition, the interrupt inputs PA6 and PA7 are triggered. The trigger edges of the interrupt lines, PA0–PA5, can be programmed using the EDG0–EDG5 bits in the key control register. If one of these bits is cleared, after reset the corresponding interrupt is falling-edge sensitive. If, however, one of them is set, after reset the corresponding interrupt is rising-edge sensitive. The internal pull-up resistors of input lines, PA7–PA0, are disabled, if rising-edge sensitivity is selected.

When a correct transition is detected, on any of this port’s pins, a keyboard interrupt request is generated, and the corresponding interrupt status flag of the interrupt status register, IRSTATE, is set. The interrupt status register is an 8-bit register which has the same address as PORTA, $0000. This register can be read if the KEYMUX bit in the system option register is set. If KIE is set, a keyboard interrupt is generated and the keyboard status flag, KF, is set by generating the logical OR of the eight interrupt state register outputs.

The 8 interrupt state register flags can be reset in three ways:

1) Completely, if the chip is reset.
2) Completely, if a 1 is written to KEYCLR, in the system option register.
3) Individually, if a 1 is written to the corresponding bit position of the interrupt state register ($00 with KEYMUX = 1, in the system option register).

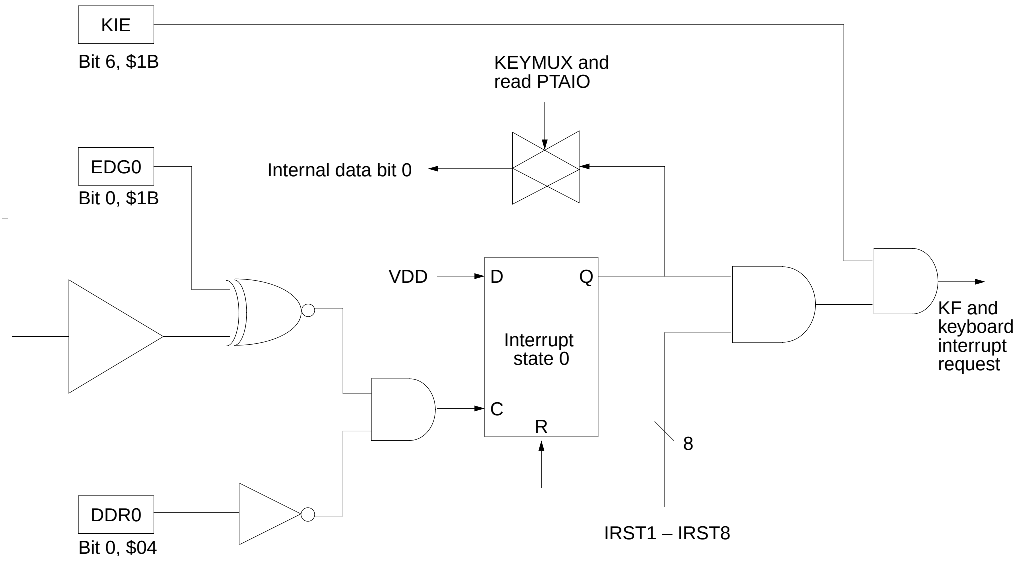


Figure 4-1 Structure of port with keyboard interrupt

4.2.1.1 Key control register (KCR)

This register contains eight bits, two of which are used to control the keyboard interrupt facility, the others determine the keyboard interrupt edges.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Key control register (KCR)	$001B	KF	KIE	EDG5	EDG4	EDG3	EDG2	EDG1	EDG0	0000 0000

KF — Keyboard interrupt status flag

1 (set) – A valid transition has occurred on one of the port pins.

0 (clear) – No valid transition has occurred on any of the port pins.

This bit is set when a valid transition is detected on any of the port A pins; a keyboard interrupt request will be generated, if keyboard interrupts are enabled (only if KIE is set). The KF flag is cleared by resetting the IRSTATE register, or by setting KEYCLR = 1 in the system option register.

KIE — keyboard interrupt enable

1 (set) – Keyboard interrupt enabled.

0 (clear) – Keyboard interrupt disabled.

An interrupt can only be generated if KIE and KF are both set and the I-bit in the CCR is clear.

EDG5–EDG0 — trigger edge control

1 (set) – Sets the corresponding interrupt line to rising-edge sensitive.

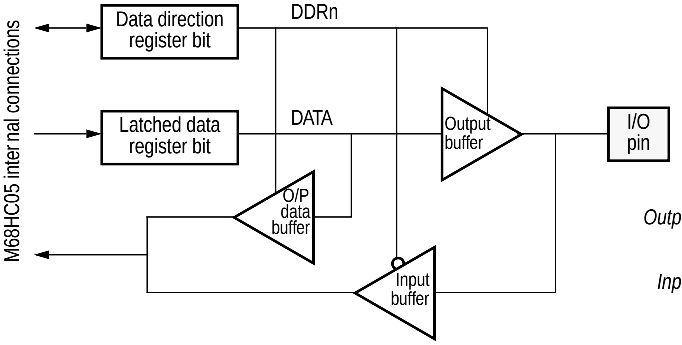
0 (clear) – Sets the corresponding interrupt line falling-edge sensitive.

The trigger edges of the interrupt lines PA5–PA0 are programmable with the EDG5–EDG0 bits in the key control register.

4.3 Port B

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

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


M68HC05 inter nal connections
Data direction register bit
DDRn
Latched data register bit
DATA
Output buffer
I/O pin
O/P data buffer
Input buffer

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

Figure 4-2 Standard I/O port structure

Table 4-1 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.

4.4 Port C

Port C is an 8-bit bidirectional port, which is shared with the SPI subsystem, the SCI subsystem and the timer system. If the SPI system is enabled, pins PC4–PC7 are connected to the functions MISO, MOSI, SCK and $\overline{\text{SS}}$, respectively. It the SCI system is enabled, pins PC2 and PC3 are connected to RDI and TDO. PC0 and PC1 are connected to TCAP3 and TCAP4 in the timer system. (These lines must be set to input, by resetting the DDR, to enable the correct TCAP function).

Reset does not affect the data register, but it clears the data direction register, returning the ports to inputs. Writing a 1 to a DDR bit, sets the corresponding port bit to output mode. All eight lines have internal pull-ups, which can be programmed using the PUEN bit in the system option register (SOR). The internal pull-ups are disabled after reset and when PUEN = 0, but are enabled by writing a 1 to PUEN.

4.5 Port D

Port D is an 8-bit bidirectional port, which is shared with the A/D converter. A pin becomes connected to the A/D converter, when its corresponding bit in the control register is set to 1.

Reset does not affect the data register, but it clears the data direction register and the control register. The default setting of the register control bits is 0, making the pins general purpose I/O lines. The direction of the pins is then determined by their corresponding bits in DDR (0 - input, 1 - output). Write access to DDR or the I/O register is blocked to reduce digital noise. Read access to DDR or the I/O register returns 0. Port D has open-drain outputs, it therefore requires external pull-up resistors for each pin when they are used as outputs.

Note: The maximum leakage current for I/O ports is 10µA. Thus, a high resistance from an analog source can limit the accuracy of the A/D converter. The analog source should therefore be less than 1 kΩ.

4.6 Port E

Port E is an 8-bit bidirectional port which is shared with the timer system, the independent 32 kHz clock system and the PWM. When the corresponding bit in the port E control register is set to 1, the pins PE1 and PE3 are connected to TCMP1 and TCMP2 of the timer system, PE4 is connected to the independent clock system, it becomes REFRESH, and PE5 – PE7 become PWM1 – PWM3 of the PWM system. Pins PE0 and PE2 are always connected to the timer system (TCAP1 and TCAP2). These two lines must be set to input by resetting the DDR to enable correct TCAP function.

Reset does not affect the data register, but it clears the data direction register and the control register. The default setting of the register control bits is 0, making the pins general purpose I/O lines. The direction of the pins is then determined by their corresponding bits in DDR (0 – input, 1 – output). Port E has open-drain outputs, it therefore requires external pull-up resistors for each pin when they are used as outputs.

Note: As the voltage at port D or port E is driven above V_{DD}, the protection device will begin to conduct and tend to clamp the input voltage to protect the input buffer. The voltage at which this condition will occur varies significantly, from lot to lot, and over the temperature range. At room temperature, the pin typically does not draw any current until approximately 18V.

4.7 Ports F, G, H, I and J

These five ports are shared with the frontplanes FP39 – FP0. The default setting of the port control bits, during reset, is 0, setting the pins in port F and port G to input only, and the pins in ports H, I and J, to output only. At power on or reset, the output only port data registers are cleared, so that these pins are driving logical 0. When the corresponding port control register bit is set to a 1, the pin is connected to the LCD frontplane driver. All port J outputs are open-drain.

4.8 Port registers

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

4.8.1 Port data registers (Ports A, B, C, D, E, F, G, H, I and J)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined
Port B data (PORTB)	$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined
Port C data (PORTC)	$0002	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined
Port D data (PORTD)	$0003	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	undefined
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined
Port F data (PORTF)	$0010	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined
Port G data (PORTG)	$0012	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	undefined
Port H data (PORTH)	$0014	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	0000 0000
Port I data (PORTI)	$0016	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0	0000 0000
Port J data (PORTJ)	$0018	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	0000 0000

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

Reset does not affect the state of the port A – port G data registers. However, port H, port I and port J data registers are reset to 0.

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

	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
Port E data direction (DDRE)	$000B									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 port pins as inputs.

4.8.3 Port control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset$
Port D control (PDCR)	$001A									0000 0000
Port E control (PECR)	$000C									0000 0000
Port F control (PFCR)	$0011									0000 0000
Port G control (PGCR)	$0013									0000 0000
Port H control (PHCR)	$0015									0000 0000
Port I control (PICR)	$0017									0000 0000
Port J control (PJCR)	$0019									0000 0000

Writing a 1 to any bit configures the corresponding port pin as a special function port (timer, A/D, LCD, PWM, refresh clock). However, clearing any bit to 0, configures the corresponding port pin in port D and port E as general purpose I/O, port F and port G as input, and port H, port I and port J as output.

5
CORE TIMER

The MC68HC05F32 has a 15-stage ripple counter called the core timer (CTIMER). Features of this timer are: timer overflow, power-on reset (POR), real time interrupt (RTI) with four selectable interrupt rates and a computer operating properly (COP) watchdog timer.

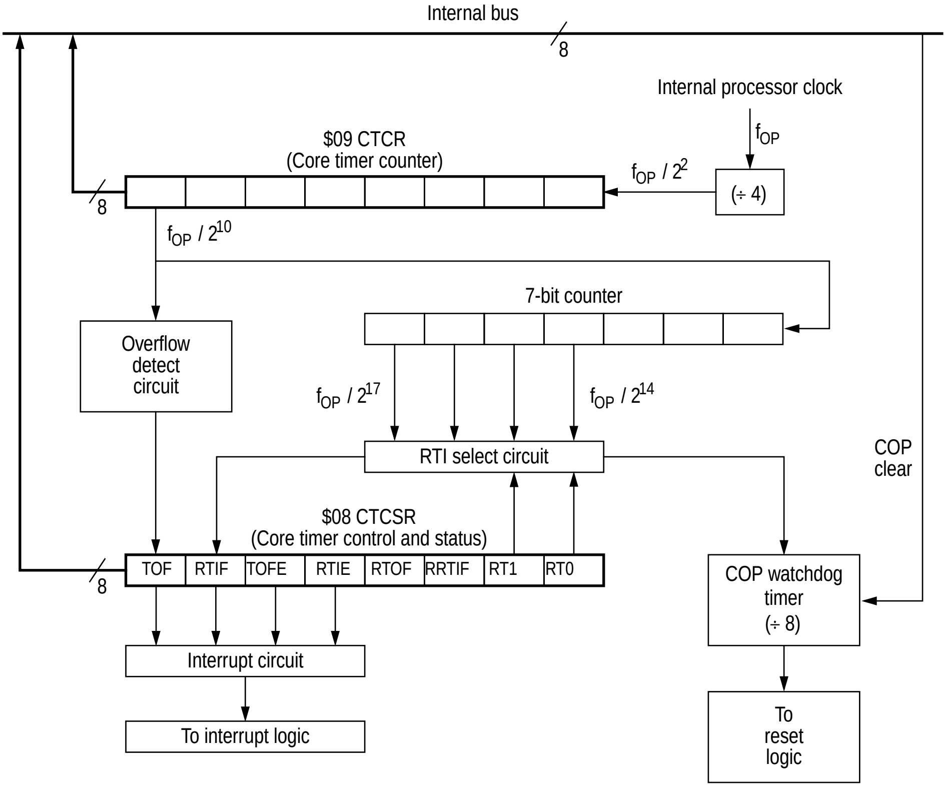


Figure 5-1 Core timer block diagram

As shown in Figure 5-1, the timer is driven by the internal bus clock divided by four with a fixed prescaler. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time, by accessing the CTIMER counter register (CTCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of $f_{OP}/1024$. (The POR signal (t_{PORL}) is also derived from this register, at $f_{OP}/4064$.) The counter register circuit is followed by four more stages, with the resulting clock ($f_{OP}/16384$) driving the real time interrupt circuit. The RTI circuit consists of three divider stages with a 1-of-4 selector. The output of the RTI circuit is further divided by 8 to drive the COP watchdog timer circuit. The RTI rate selector bits, and the RTI and CTIMER overflow enable bits and flags, are located in the CTIMER control and status register (CTCSR) at location $08.

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

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

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

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

5.1 Real time interrupts (RTI)

The real time interrupt circuit consists of a three stage divider and a 1-of-4 selector. The clock frequency that drives the RTI circuit is $f_{OP}/2^{14}$ (or $f_{OP}/16384$), with three additional divider stages, giving a maximum interrupt period of 4 seconds at a bus frequency (f_{OP}) of 32kHz. Register details are given in Section 5.2.

5.2 Core timer registers

5.2.1 Core timer control and status register (CTCSR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Core timer control/status (CTCSR)	$0008	CTOF	RTIF	CTOFE	RTIE	RTOF	RRTIF	RT1	RT0	0000 0011

CTOF — Core timer overflow

1 (set) – Core timer overflow has occurred.

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

CTOF is a read-only status bit and 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, CTOF may be cleared by writing a ‘1’ to RTOF.

RTIF — Real time interrupt flag

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

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

RTIF is a read-only status bit and 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 RRTIF. Reset also clears this bit.

CTOFE — Core timer overflow enable

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

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

Setting this bit enables the core timer overflow Interrupt. A CPU interrupt request will then be generated whenever the CTOF bit becomes set and the I-bit in the CCR is clear. Clearing this bit disables the core timer overflow interrupt capability.

RTIE — Real time interrupt enable

1 (set) – Real time interrupt is enabled.

0 (clear) – Real time interrupt is disabled.

Setting this bit enables the real time interrupt. A CPU interrupt request will then be generated whenever the RTIF bit becomes set and the I-bit in the CCR is clear. Clearing this bit disables the real time interrupt capability.

RT1, RT0 — Real time interrupt rate select

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

Table 5-1 Example RTI periods

			RTI Rates at f_{OP} Frequency Specified			
RT1	**RT0**	**Division ratio**	**16.384 kHz**	**447 kHz**	**895 kHz**	**1.789 MHz**
0	0	2^{14}	1 s	36.7 ms	18.35 ms	9.17 ms
0	1	2^{15}	2 s	73.4 ms	36.7 ms	18.35 ms
1	0	2^{16}	4 s	146.8 ms	73.4 ms	36.7 ms
1	1	2^{17}	8 s	293.6 ms	146.8 ms	73.4 ms

5.2.2 Core timer counter register (CTCR)

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

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

5.3 Computer operating properly (COP) watchdog timer

The COP watchdog timer function is implemented by taking the output of the RTI circuit and further dividing it by eight, as shown in Figure 5-1. Note that the minimum COP timeout period is seven times the RTI period. This is because the COP will be cleared asynchronously with respect to the value in the core timer counter register/RTI divider, hence the actual COP timeout period will vary between 7x and 8x the RTI period. The minimum COP reset rates are shown in Table 5-2.

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

If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. A COP timeout is prevented by writing a '0' to bit 0 of address $FFF0. When the COP is cleared, only the final divide-by-eight stage is cleared (see Figure 5-1).

Table 5-2 Minimum COP reset times

		Minimum COP reset at f_{OP} frequency specified				
RT1	**RT0**	**16.384 kHz**	**447 kHz**	**895 kHz**	**1.789 MHz**	f_{OP}
0	0	7 s	256.9 ms	128.45 ms	64.19 ms	7 x RTI rate
0	1	14 s	513.8 ms	256.9 ms	128.45 ms	7 x RTI rate
1	0	28 s	1.03 s	513.8 s	256.9 ms	7 x RTI rate
1	1	56 s	2.06 s	1.03 s	513.8 ms	7 x RTI rate

5.4 Core timer during WAIT

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

5.5 Core timer during STOP

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

THIS PAGE LEFT BLANK INTENTIONALLY

6
16-BIT PROGRAMMABLE TIMER

The MC68HC05F32 has two programmable 16-bit timers (TIMER1 and TIMER2), each with two channels. The output compare function in TIMER2 has no external output, and is therefore used for generating precision time intervals and interrupts only. The external connections are the only differences between the two timers. The internal operation is identical (each timer has its own set of registers), therefore only a complete description of TIMER1 is given.

The timer consists of a 16-bit read-only free-running counter, with a fixed divide-by-four prescaler, plus the input capture/output compare circuitry. The timer can be used for many purposes including measuring pulse length of two input signals and generating two output signals. Pulse lengths for both input and output signals can vary from several microseconds to many seconds. The timer is also capable of generating periodic interrupts or indicating passage of an arbitrary multiple of four CPU cycles. A block diagram is shown in Figure 6-1, and timing diagrams are shown in Figure 6-2, Figure 6-3, Figure 6-4 and Figure 6-5.

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

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

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

6.1 Counter

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

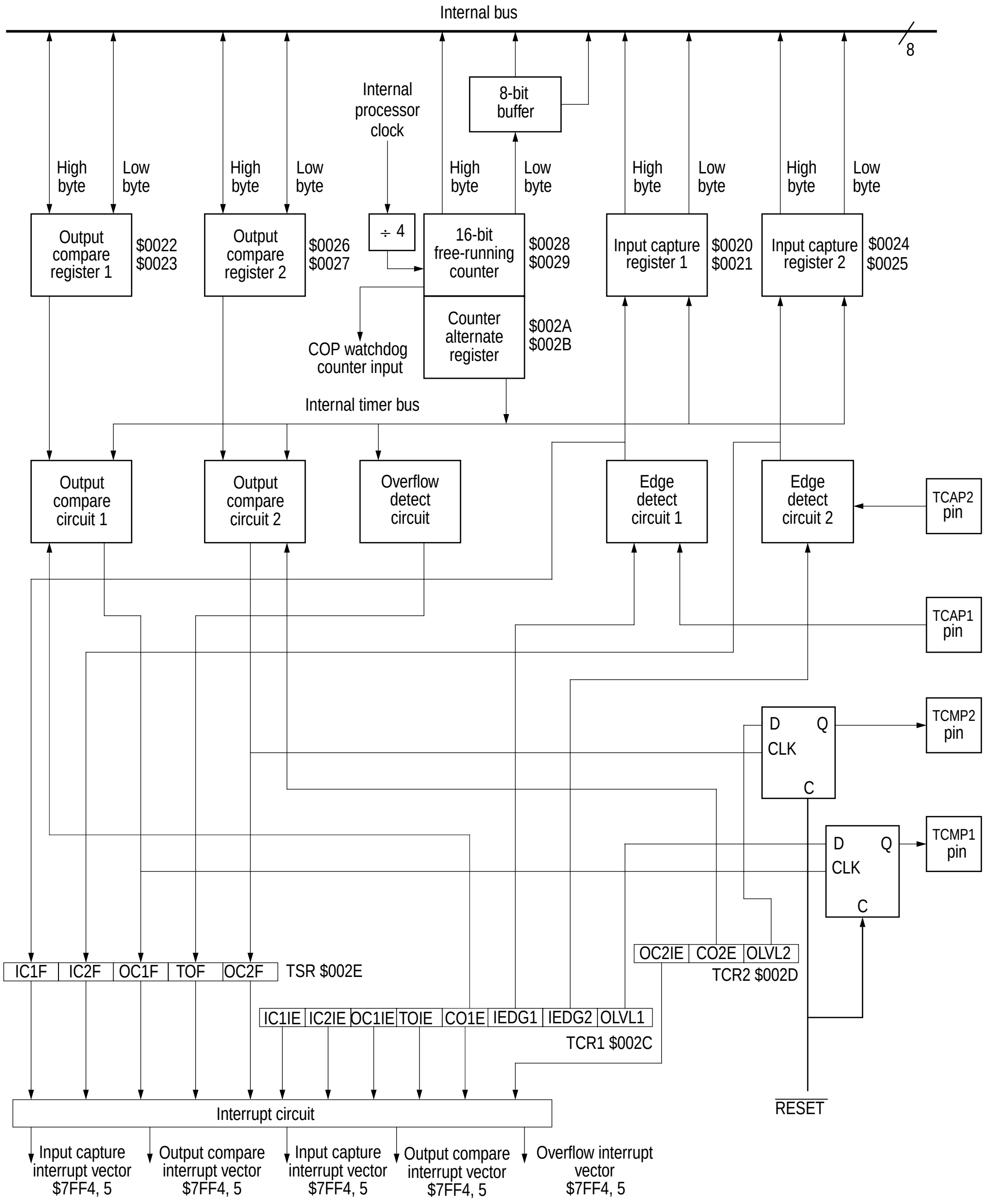


Figure 6-1 16-bit programmable timer block diagram

6.1.1 Counter register and alternate counter register

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

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

The double-byte, free-running counter can be read from either of two locations, $0028 – $0029 (counter register) or $002A – $002B (counter alternate register). A read from only the less significant byte (LSB) of the free-running counter ($0029 or $002B) receives the count value at the time of the read. If a read of the free-running counter or alternate counter register first addresses the more significant byte (MSB) ($0028 or $002A), the LSB is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or alternate counter register LSB and thus completes a read sequence of the total counter value. In reading either the free-running counter or alternate counter register, if the MSB is read, the LSB must also be read to complete the sequence. If the timer overflow flag (TOF) is set when the counter register LSB is read then a read of the timer status register (TSR) will clear the flag.

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

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

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

Caution: This operation may affect the function of the watchdog system (see Section 5.3).

6.2 Timer control and status

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

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

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

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 1 (TCR1)	$002C	IC1IE	IC2IE	OC1IE	TOIE	CO1E	IEDG1	IEDG2	OLVL1	0000 0uu0

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

IC1IE — Input capture 1 interrupt enable

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

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

IC2IE — Input capture 2 interrupt enable

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

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

OC1IE — Output compare 1 interrupt enable

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

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

TOIE — Timer overflow interrupt enable

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

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

CO1E — Timer compare 1 output enable

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

1 (set) – Output compare 1 enabled.

0 (clear) – Output compare 1 disabled.

IEDG1 — Input edge 1

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

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

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

IEDG2 — Input edge 2

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

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

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

OLVL1 — Output level 1

When OLV1 is set a high output level will be clocked into the output level register by the next successful output compare, and will appear on the TCMP1 pin. When clear, it will be a low level which will appear on the TCMP1 pin.

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

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

OC2IE — Output compare 2 interrupt enable

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

1 (set) – Interrupt enabled.

0 (clear) – Interrupt disabled.

CO2E — Timer compare 2 output enable

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

1 (set) – Output compare 2 enabled.

0 (clear) – Output compare 2 disabled.

OLVL2 — Output level 2

When OLV2 is set a high output level will be clocked into the output level register by the next successful output compare, and will appear on the TCMP2 pin. When clear, it will be a low level which will appear on the TCMP2 pin.

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

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

6.2.2 Timer status register (TSR)

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

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

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer status (TSR)	$002E	IC1F	IC2F	OC1F	TOF	TCAP1	TCAP2	OC2F	0	Undefined

IC1F — Input capture 1 flag

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

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

0 (clear) – No input capture has occurred.

IC2F — Input capture 2 flag

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

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

0 (clear) – No input capture has occurred.

OC1F — Output compare 1 flag

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

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

0 (clear) – No output compare has occurred.

TOF — Timer overflow status flag

This bit is set when the free-running counter overflows from $FFFF to $0000; a timer overflow interrupt will occur if TOIE is set. TOF is cleared by reading the TSR and the counter low register ($0029).

1 (set) – Timer overflow has occurred.

0 (clear) – No timer overflow has occurred.

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

1 The timer status register is read or written when TOF is set, and

2 The LSB of the free-running counter is read, but not for the purpose of servicing the flag.

Reading the alternate counter register instead of the counter register will avoid this potential problem.

TCAP1 — Timer capture 1 status flag

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

TCAP2 — Timer capture 2 status flag

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

OC2F — Output compare 2 flag

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

1 (set)	–	A valid output compare has occurred.
0 (clear)	–	No output compare has occurred.

6.3 Input capture

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

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

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

6.3.1 Input capture register 1 (ICR1)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Input capture 1 high (ICR1H)	$0020									Undefined
Input capture 1 low (ICR1L)	$0021									Undefined

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

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

Reset does not affect the contents of the input capture register 1, except when exiting STOP mode (see Section 6.5).

6.3.2 Input capture register 2 (ICR2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Input capture 2 high (ICR2H)	$0024									Undefined
Input capture 2 low (ICR2L)	$0025									Undefined

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

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

Reset does not affect the contents of the input capture register 2, except when exiting STOP mode (see Section 6.5).

6.4 Output compare

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

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

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

6.4.1 Output compare register 1 (OCR1)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 1 high (OCR1H)	$0022									Undefined
Output compare 1 low (OCR1L)	$0023									Undefined

The 16-bit output compare register 1 is made up of two 8-bit registers at locations $0022 (MSB) and $0023 (LSB). The contents of the output compare register 1 are compared with the contents of the free-running counter continually and, if a match is found, the corresponding output compare flag (OC1F) in the timer status register is set. If the timer compare output enable bit (CO1E) is set, the output level (OLVL1) is transferred to pin TCMP1. The output compare register 1 values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OC1IE) is set. (The free-running counter is updated every four internal bus clock cycles.)

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

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

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

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

6.4.2 Output compare register 2 (OCR2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 2 high (OCR2H)	$0026									Undefined
Output compare 2 low (OCR2L)	$0027									Undefined

The 16-bit output compare register 2 is made up of two 8-bit registers at locations $0026 (MSB) and $0027 (LSB). The contents of the output compare register 2 are compared with the contents of the free-running counter continually and, if a match is found, the corresponding output compare flag (OC2F) in the timer status register is set. If the timer compare 2 output enable bit (CO2E) is set, the output level (OLVL2) is transferred to pin TCMP2. The output compare register 2 values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OC2IE) is set. (The free-running counter is updated every four internal bus clock cycles.)

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

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

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

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

6.5 Timer during STOP mode

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

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

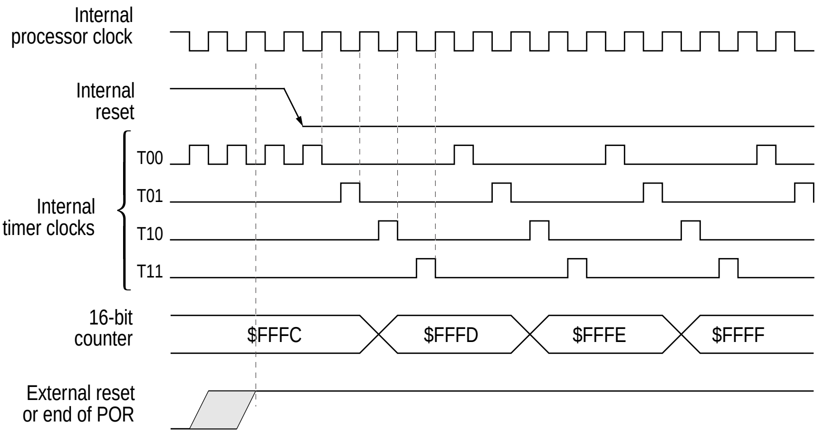
If STOP mode is exited by an external reset then no such input capture flag or data action takes place even if there was a valid input capture edge (at one of the TCAP pins) during STOP mode.

6.6 Timer during WAIT mode

During WAIT mode, the CPU clock halts but timer1 keeps running. Timer2 is disabled, if bit 7 (WTLCDO) of the LCD control register is set, however, if it is cleared, timer2 remains active. If interrupts are enabled, a timer interrupt will cause the processor to exit WAIT mode.

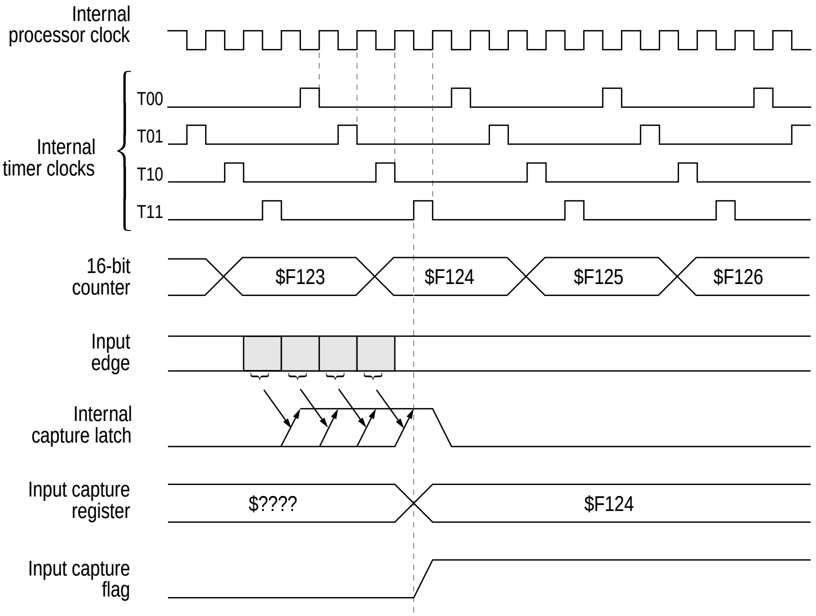
6.7 Timer state diagrams

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



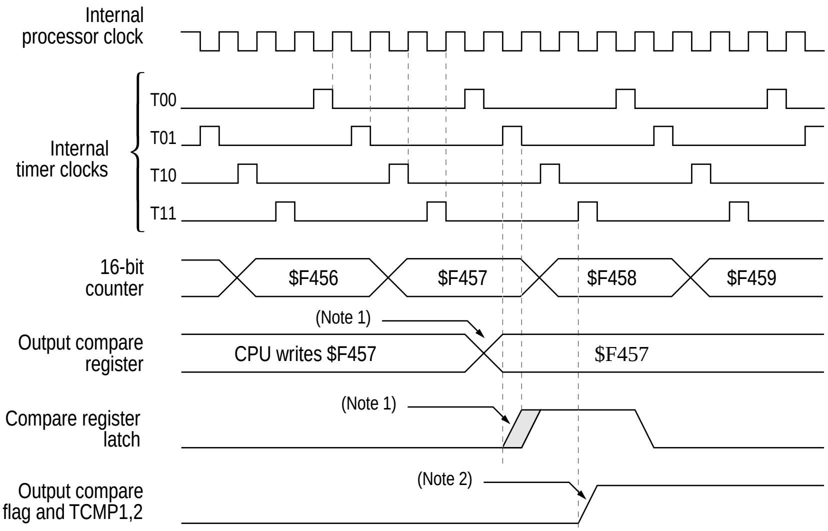
Note: The counter and timer control registers are the only ones affected by power-on or external reset.

Figure 6-2 Timer state timing diagram for reset



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

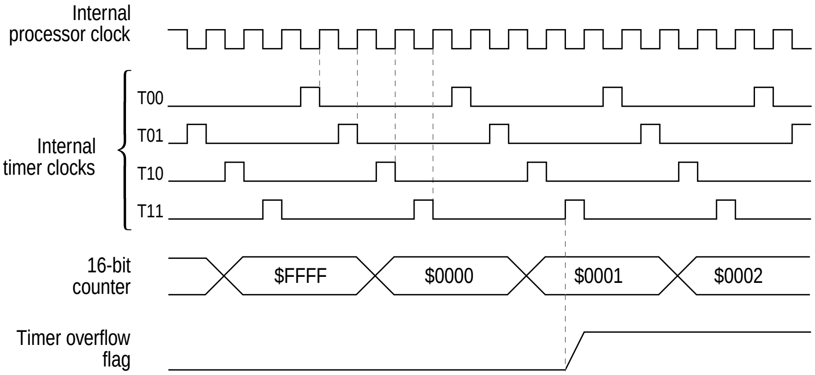
Figure 6-3 Timer state timing diagram for input capture



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

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

Figure 6-4 Timer state timing diagram for output compare



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

Figure 6-5 Timer state timing diagram for timer overflow

THIS PAGE LEFT BLANK INTENTIONALLY

7
DTMF/MELODY GENERATOR

7.1 Introduction

The DTMF/melody generator (DMG) is a multi-functional tone generator built into the MC68HC05F32 MCU which supports DTMF dialling, melody-on-hold and pacifier tone functions.

7.1.1 Features

- 4 row and 4 column frequencies for DTMF dialling
- 24 row and 24 column frequencies for dual tone melody
- 28 frequencies for pacifier tone to acknowledge button pressed for pulse dialling
- Power saving mechanism for output disable condition
- 3.579MHz/2 operation
- 6-bit D/A converter and 28 time steps for sine wave generation
- Sine wave or square wave selectable output for melody (or DTMF)
- Single or dual tone capability for melody (or DTMF)

7.2 Functional description

As shown in Figure 7-1, the DMG consists of 2 tone generation paths (the column and row paths). One path generates the row tone and the other the column tone, whose frequencies are determined by the values in the frequency control registers FCR and FCC respectively. The tones allowed at the TNO output are single/dual sine/square wave tones of DTMF and melody frequencies, whereas at the TNX output, only single square wave tones are allowed. The method of tone generation for the two paths is almost the same, and is described as follows.

To generate a sine wave tone with programmable frequency in a path, the internal clock (i.e. the 3.58MHz/2) is first divided by a frequency divider according to a number on the register (FCR or FCC). The output of the divider is a periodic pulse train whose frequency is the sampling rate of the desired ‘staircase sine wave’. This pulse train, in turn, clocks a divide-by-28 binary counter (PLA scanner) whose 28 decoded outputs scan sequentially 28 memory locations of a 28x6 sine wave generator (PLA) in 28 time steps (M). The six resulting digital sine wave bits are then fed separately to a 6-bit resistor ladder to produce a current signal.

The method for generating a square wave tone in a path is similar to that of a sine wave tone except that only the most significant bit of a sine wave PLA is fed to the 6-bit resistor ladder to produce a current signal (the other 5 least significant bits are masked by the sine/square wave select). Using this method, a square wave tone can be produced which has exactly the same frequency and phase as a sine wave tone, and uses the same frequency control register value.

After obtaining the current signals from the row and column paths, the row current signal is first attenuated by 2dB. It is then summed with the column current signal, and is finally fed to an active 7 KHz low pass filter to reduce harmonic distortion (note that square wave tones are also passed through this filter). The resulting DTMF or melody signal is output through the TNO pin which is normally connected to a speech circuit.

The generator provides not only DTMF and melody but also a square wave pacifier tone (ToneX). This signal is also extracted from the most significant bit of the sine wave PLA of the row path, but is not passed through the filter. The ToneX signal is output through the TNX pin which is normally connected to a loudspeaker.

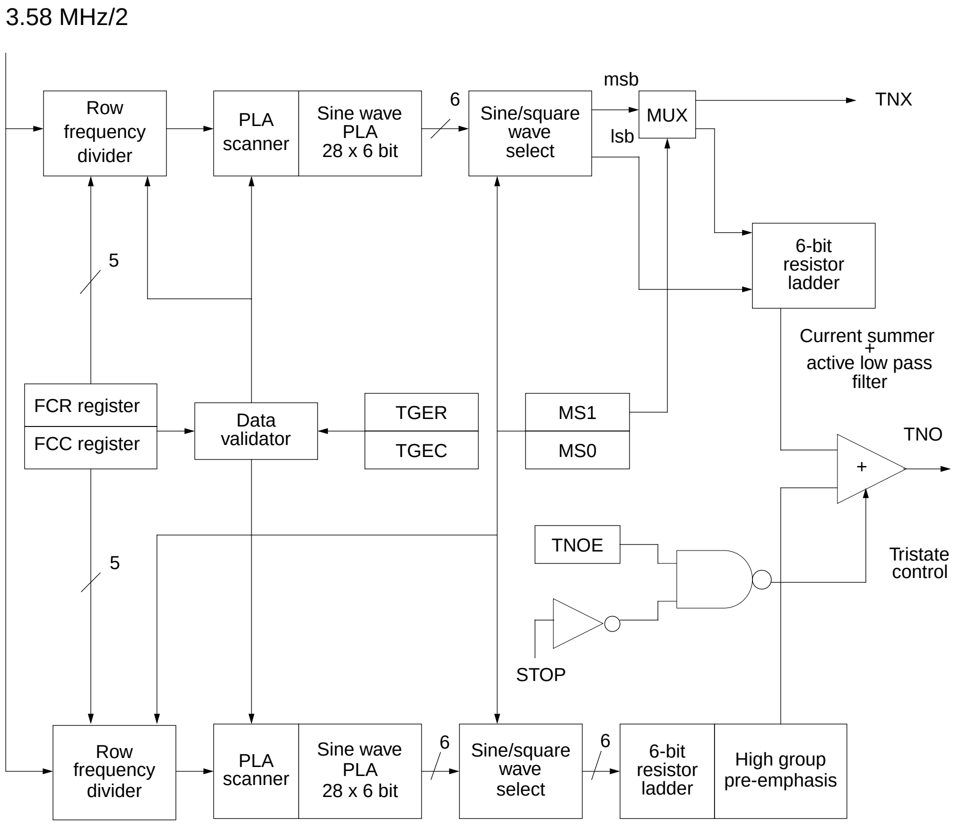


Figure 7-1 DTMF/melody generator (DMG) block diagram

7.3 DMG registers

The DMG has two registers (row frequency control register and column frequency control register) for row and column frequency selection respectively, and one register (tone control register) for tone output control and mode selection.

7.3.1 Row and column frequency control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Row frequency control register (FCR)	$000D	0	0	0	FCR4	FCR3	FCR2	FCR1	FCR0	undefined

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Column frequency control register (FCC)	$000E	0	0	0	FCC4	FCC3	FCC2	FCC1	FCC0	undefined

FCR4–FCR0 and FCC4–FCC0 control the frequency of the tone signals on the row and the column paths respectively. The row and column paths are not exactly identical owing to the presence of the high group pre-emphasis in the column path. In order to avoid the entry of the row DTMF tone values to the column, and vice versa, the above cases are treated as illegal. The data validator will disable all outputs when an illegal value is detected. The bit description for DTMF and melody tone generation are shown in Table 7-1 and Table 7-2 respectively. It is the user's responsibility to ensure good programming practice by initialising all registers to contain legal values for the desired function.

7.3.2 Tone control register (TNCR)

This register controls the internal configuration and tone output timing of the DTMF/melody generator.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Tone control register (TNCR)	$000F	MS1	MS0	TGER	TGEC	TNOE	0	0	0	0000 0000

MS1, MS0 — Melody select for operation

The MS0 and MS1 bits control the mode of operation of the DTMF/melody generator. There are sine wave, square wave 1, square wave 2 and square wave 3 modes. They are specified as shown in Table 7-3.

When square wave 2 or square wave 3 mode is selected, the TNX pin is activated. The idle state for TNX is a logic high. The final state of the TNX pin is still dependent on the values of TGER, TGEC (see Table 7-4), FCR and FCC bits (when illegal values are input).

The state of the TNO pin depends on the value of the TNOE bit. After a RESET, the TNOE is cleared and the TNO pin is tristate. When TNOE is set, the TNO output is activated. If the TGER and TGEC bits are held low and TNOE is set, the dc offset of $V_{DD}/2$ appears at TNO pin. In STOP mode, the TNX pin is high and the TNO pin is tristate.

When both MS1 and MS0 are set (square wave 3), the generator can generate both single tone melody at the column path, and ToneX at the row path simultaneously.

TGER — Tone generator enable row path

1 (set) – Row path on

0 (clear) – Row path off

TGEC — Tone generator enable column path

1 (set) – Column path on

0 (clear) – Column path off

TNOE — Tone output enable

1 (set) – TNO on

0 (clear) – TNO off

Table 7-1 Bit description for DTMF generation

FCR register	FCC register	Tone	Standard frequency (Hz)	Tone output frequency (Hz)	Frequency deviation
$00		f_{R1}	697.0	694.8	–0.32
$01		f_{R2}	770.0	770.1	0.02
$02		f_{R3}	852.0	854.2	0.03
$03		f_{R4}	941.0	940.0	–0.11
	$10	f_{C1}	1209.0	1206.0	–0.244
	$11	f_{C2}	1336.0	1331.7	–0.324
	$12	f_{C3}	1477.0	1486.5	0.645
	$13	f_{C4}	1633.0	1639.0	0.367

Note: The legal values in the FCR register column are illegal to the FCC register, and vice versa. An input of illegal values to these registers will produce a high at TNX output and $V_{DD}/2$ at TNO output (TNOE = 1)

.

Table 7-2 Bit description for melody generator

FCR/FCC register	Tone	Standard frequency (Hz)	Tone output frequency (Hz)	Frequency deviation (%)
$04	D#5	622.3	620.6	–0.28
$05	E5	659.3	659.0	–0.05
$06	F5	698.5	694.8	–0.53
$07	F#5	740.0	743.3	0.44
$08	G5	784.0	779.5	–0.57
$09	G#5	830.6	830.1	–0.06
$0A	A5	880.0	875.6	–0.50
$0B	A#5	932.0	926.4	–0.64
$0C	B5	987.8	983.4	–0.45
$0D	C6	1046.5	1047.9	0.13
$0E	C#6	1108.7	1102.1	–0.60
$0F	D6	1174.7	1183.7	0.77
$14	D#6	1244.5	1253.3	0.71
$15	E6	1318.5	1331.7	1.00
$16	F6	1396.9	1389.6	–0.52
$17	F#6	1480.0	1486.5	0.44
$18	G6	1568.0	1559.0	–0.57
$19	G#6	1661.2	1682.1	1.26
$1A	A6	1760.0	1775.6	0.89
$1B	A#6	1864.7	1880.0	0.82
$1C	B6	1975.5	1997.5	1.11
$1D	C7	2093.0	2062.0	–1.49
$1E	C#7	2217.5	2204.2	–0.60
$1F	D7	2349.3	2367.4	0.771

Table 7-3 Mode of operation for DMG

MS1	MS0	Mode	TNX output	TNO output
0	0	sine wave	high	sine wave row and column frequency
0	1	square wave 1	high	square wave row and column frequency
1	0	square wave 2	row frequency	square wave row and column frequency
1	1	square wave 3	row frequency	square wave column frequency

TGER, TGEC — Tone generation enable for row and column paths

When both bits are held low, the DMG is disabled by forcing the two frequency counters and the two PLA scanning counters to their reset states. The DMG should then consume zero dynamic power, if the TNOE bit is also cleared.

When a TGE bit for a path is held high (provided that the value in the frequency control register for that path is legal), the generator is enabled. All the counters associated with that path are then run from their reset states.

The reset state of a frequency counter defines the time=0 state of the time step, whereas at their reset state, the PLA scanning counters, scanning the memory location, contain the dc values of the staircase sine wave.

In DTMF dialling, the row and column tone values are first entered to the FCR and FCC registers. The TGER and TGEC bits are then set or reset simultaneously to achieve dual tone multiple frequency. Similarly, in melody generation, one path is chosen as the high part, and the other as the low part. The TGER and TGEC bits are then set and reset according to the rhythm required by the musical piece. One can exhibit only single tone melody by disabling either TGER or TGEC permanently. The DTMF column and row frequency tones can also be output separately for testing by enabling just the one path.

Table 7-4 Effect of tone generation on DMG

TGER	TGEC	RowPath	Column Path
0	0	off	off
0	1	off	active
1	0	active	off
1	1	active	active

7.4 Operation of the DMG

The DMG is recommended to be operated using the following procedures:

To operate melody generation, the choice of sine wave or square wave output mode is totally up to the user’s taste. The sine wave melody has a sound like a flute, whereas the square wave melody possesses much richer harmonics. The required tones are selected through the FCR and FCC registers. The selected tone is output when the corresponding TGER or TGEC bit and TNOE bit are set. The FCR register should contain the value representing the tone output frequency and the FCC register should contain a value of $03 or greater to ensure the output is not blocked by the data validator.

7.5 DMG during WAIT mode

The DMG is still active during the WAIT mode.

7.6 DMG during STOP mode

In STOP mode the oscillator is stopped causing the DMG to cease function.

8

LIQUID CRYSTAL DISPLAY DRIVER MODULE

The LCD driver module on the MC68HC05F32 supports 40 frontplanes and 4 backplanes, allowing a maximum of 160 LCD segments. Each segment is controlled by a corresponding bit in the LCD RAM. The mode of operation is determined by the values set in the LCD control register at $1E.

After reset and on leaving standby, the drivers are configured in the default duplex mode, 1/2 bias with 2 backplanes. At power-up or after reset, the ON/OFF control bits for the internal and external V_{LCD} voltage (INTVLCD and EXTVON) are cleared, disabling the LCD drivers. Figure 8-1 shows a block diagram of the LCD system. At power-up or after reset the LCD port's control bits are cleared, which disables the LCD frontplane drivers.

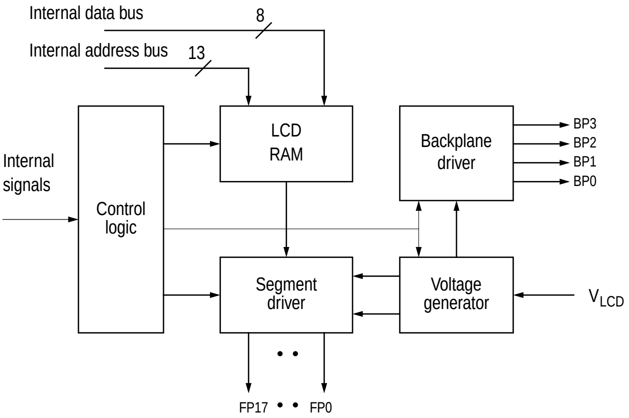


Figure 8-1 LCD system block diagram

8.1 LCD RAM

Data to be displayed on the LCD must be written into the LCD RAM. The LCD RAM is comprised of 20 bytes of RAM (in the MC68HC05F32’s memory map) at $0054 – $0067. The 160 bits in the LCD RAM correspond to the 160 segments that can be driven by the frontplane/backplane drivers. Table 8-1 shows how the LCD RAM is organized. Writing a ‘1’ to a given location will result in the corresponding display segment being activated when the EXTVON or INTVLCD bit is set. The LCD RAM is a dual port RAM that interfaces with the internal address and data buses of the MCU. It is possible to read from LCD RAM locations for scrolling purposes.

Table 8-1 LCD RAM organization

LCD RAM				Data				
Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$54	FP1-BP3	FP1-BP2	FP1-BP1	FP1-BP0	FP0-BP3	FP0-BP2	FP0-BP1	FP0-BP0
$55	FP3-BP3	FP3-BP2	FP3-BP1	FP3-BP0	FP2-BP3	FP2-BP2	FP2-BP1	FP2-BP0
$56	FP5-BP3	FP5-BP2	FP5-BP1	FP5-BP0	FP4-BP3	FP4-BP2	FP4-BP1	FP4-BP0
$57	FP7-BP3	FP7-BP2	FP7-BP1	FP7-BP0	FP6-BP3	FP6-BP2	FP6-BP1	FP6-BP0
$58	FP9-BP3	FP9-BP2	FP9-BP1	FP9-BP0	FP8-BP3	FP8-BP2	FP8-BP1	FP8-BP0
$59	FP11-BP3	FP11-BP2	FP11-BP1	FP11-BP0	FP10-BP3	FP10-BP2	FP10-BP1	FP10-BP0
$5A	FP13-BP3	FP13-BP2	FP13-BP1	FP13-BP0	FP12-BP3	FP12-BP2	FP12-BP1	FP12-BP0
$5B	FP15-BP3	FP15-BP2	FP15-BP1	FP15-BP0	FP14-BP3	FP14-BP2	FP14-BP1	FP14-BP0
$5C	FP17-BP3	FP17-BP2	FP17-BP1	FP17-BP0	FP16-BP3	FP16-BP2	FP16-BP1	FP16-BP0
$5D	FP19-BP3	FP19-BP2	FP19-BP1	FP19-BP0	FP18-BP3	FP18-BP2	FP18-BP1	FP18-BP0
$5E	FP21-BP3	FP21-BP2	FP21-BP1	FP21-BP0	FP20-BP3	FP20-BP2	FP20-BP1	FP20-BP0
$5F	FP23-BP3	FP23-BP2	FP23-BP1	FP23-BP0	FP22-BP3	FP22-BP2	FP22-BP1	FP22-BP0
$60	FP25-BP3	FP25-BP2	FP25-BP1	FP25-BP0	FP24-BP3	FP24-BP2	FP24-BP1	FP24-BP0
$61	FP27-BP3	FP27-BP2	FP27-BP1	FP27-BP0	FP26-BP3	FP26-BP2	FP26-BP1	FP26-BP0
$62	FP29-BP3	FP29-BP2	FP29-BP1	FP29-BP0	FP28-BP3	FP28-BP2	FP28-BP1	FP28-BP0
$63	FP31-BP3	FP31-BP2	FP31-BP1	FP31-BP0	FP30-BP3	FP30-BP2	FP30-BP1	FP30-BP0
$64	FP33-BP3	FP33-BP2	FP33-BP1	FP33-BP0	FP32-BP3	FP32-BP2	FP32-BP1	FP32-BP0
$65	FP35-BP3	FP35-BP2	FP35-BP1	FP35-BP0	FP34-BP3	FP34-BP2	FP34-BP1	FP34-BP0
$66	FP37-BP3	FP37-BP2	FP37-BP1	FP37-BP0	FP36-BP3	FP36-BP2	FP36-BP1	FP36-BP0
$67	FP39-BP3	FP39-BP2	FP39-BP1	FP39-BP0	FP38-BP3	FP38-BP2	FP38-BP1	FP38-BP0

8.2 LCD operation

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

- 1/2 bias, 2 backplanes
- 1/3 bias, 2 backplanes
- 1/3 bias, 3 backplanes
- 1/4 bias, 4 backplanes

The operating mode is selected at power on using the multiplex ratio bits (MUX3 and MUX4) in the LCD control register as shown in Table 8-4.

It is recommended that the EXTVON and INTVLCD bits in the LCD register are not set (display is disabled) until the multiplex rate is selected. The voltage levels required for the different multiplex rates are generated internally by a resistive divider chain between V_{LCD} and V_{SS}.

The 2-way multiplex with 1/3 bias and the three and four-way multiplex options require four voltage levels, whereas the two-way multiplex with 1/2 bias needs only three levels. Resistors R1, R2 and R3 are valued at 20kΩ ±40%. Figure 8-2 shows the resistive divider chain network that is used to produce the various LCD waveforms outlined in Section 8.3.

The LCD drivers can operate with an external V_{LCD} supply when EXTVON = 1, or with an internally generated LCD voltage when INTVLCD = 1. The EXTVON option is useful when a display with particular thresholds is being used. The LCD controller is enabled if the EXTVON bit or the INTVLCD bit is set. Table 8-2 shows the different modes of operation depending on the bits EXTVON and INTVLCD of the LCD control register.

Table 8-2 LCD controller operating modes

EXTVON	INTVLCD	LCD controller	Internal voltage generator	Resistor chain connected with
0	0	off	off	—
0	1	on	on	internal V_{LCD}
1	0	on	off	VLCD pin
1	1	on	on	both (for test)

Note: The external voltage V_{LCD} may not exceed the positive power supply voltage, V_{DD}.

Note: If both bits INTVLCD and EXTVON are set, an externally applied voltage source can cause damage to the LCD drivers.

.

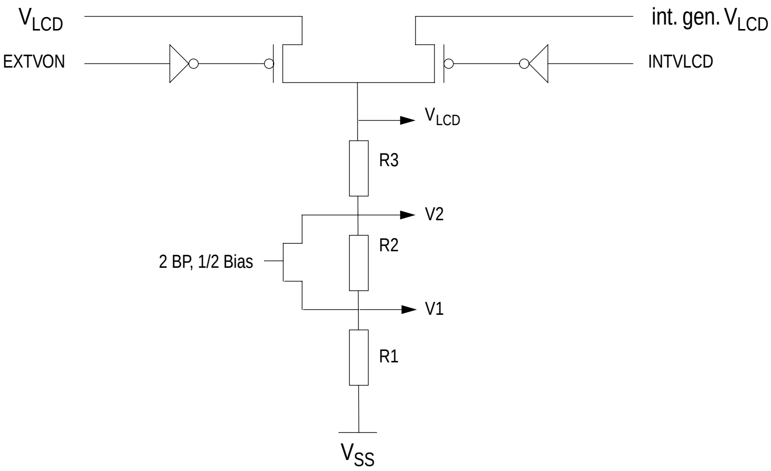


Figure 8-2 Voltage level selection

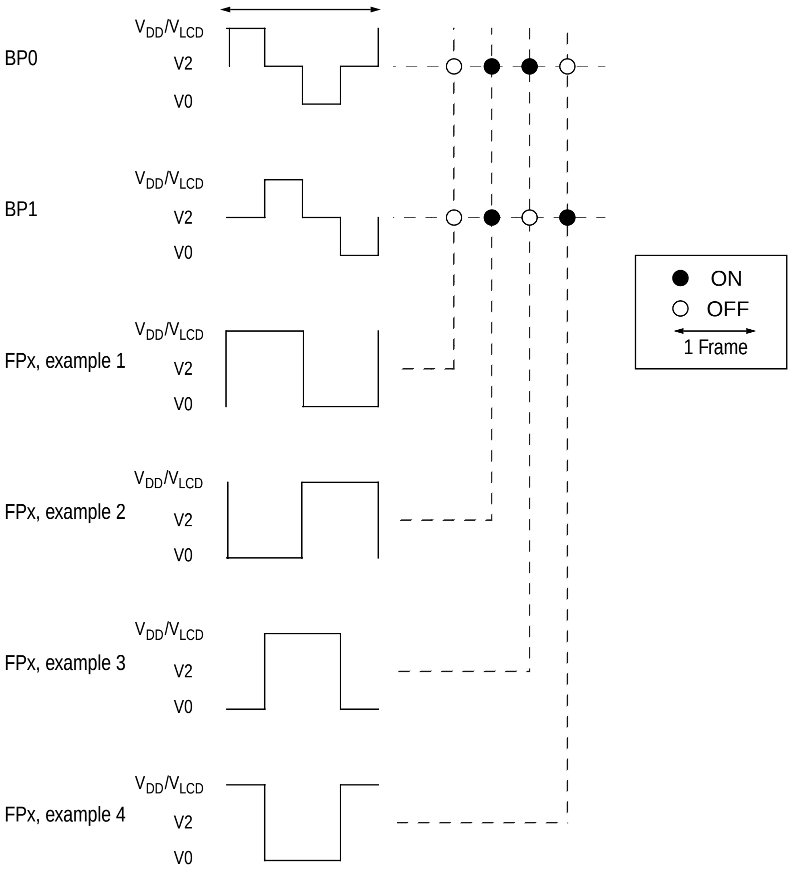
8.3 Timing signals and LCD voltage waveforms

The LCD timing signals are all derived from the main system clock. The frame rate will be $f_{OSC}/2^{16}$, therefore, if f_{OSC} = 3.579 MHz, the frame rate will be 54.6 Hz for two and four-way multiplexing and 72.8 Hz for three-way multiplexing (see Table 8-4). An extra divide-by-two stage can be included in the LCD clock generator by setting FDISP in the LCD register. This will result in the frame rate being halved. For example, when three-way multiplexing is used, a frame rate of 36.4 Hz instead of 72.8 Hz can be obtained. See Section 8.4.

Figure 8-3 to Figure 8-6 show the backplane waveforms and some examples of frontplane waveforms for each of the operating modes.

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

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



Note: In this mode V1=V2

Figure 8-3 LCD waveform with 2 backplanes, 1/2 bias

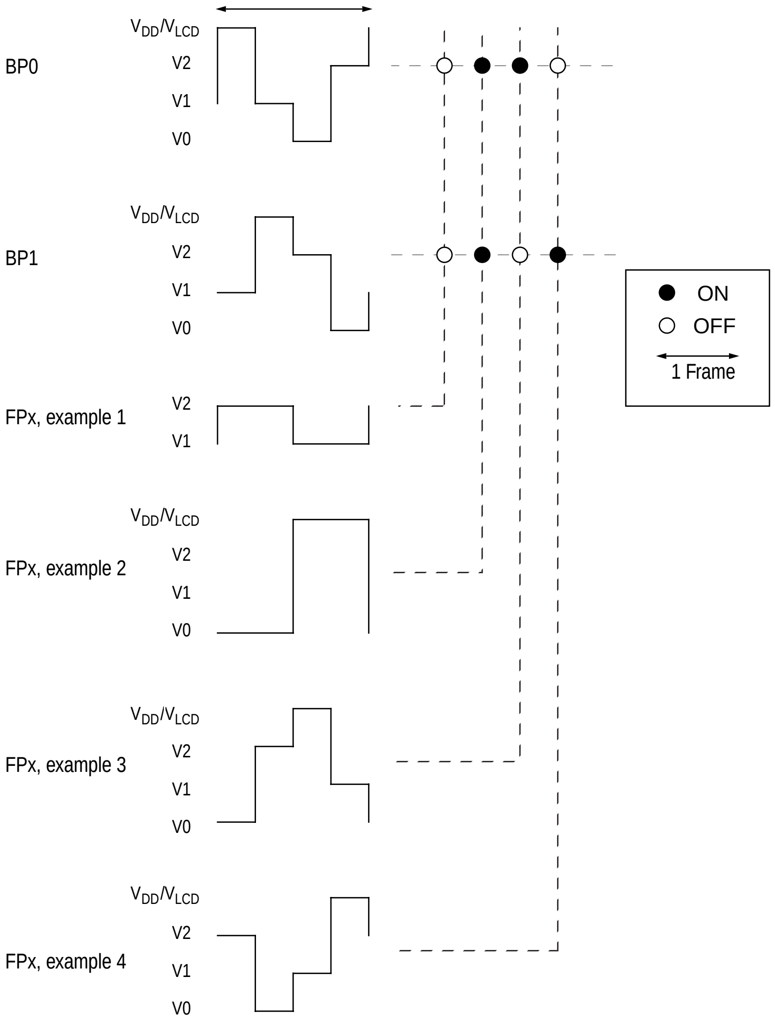


Figure 8-4 LCD waveform with 2 backplanes, 1/3 bias

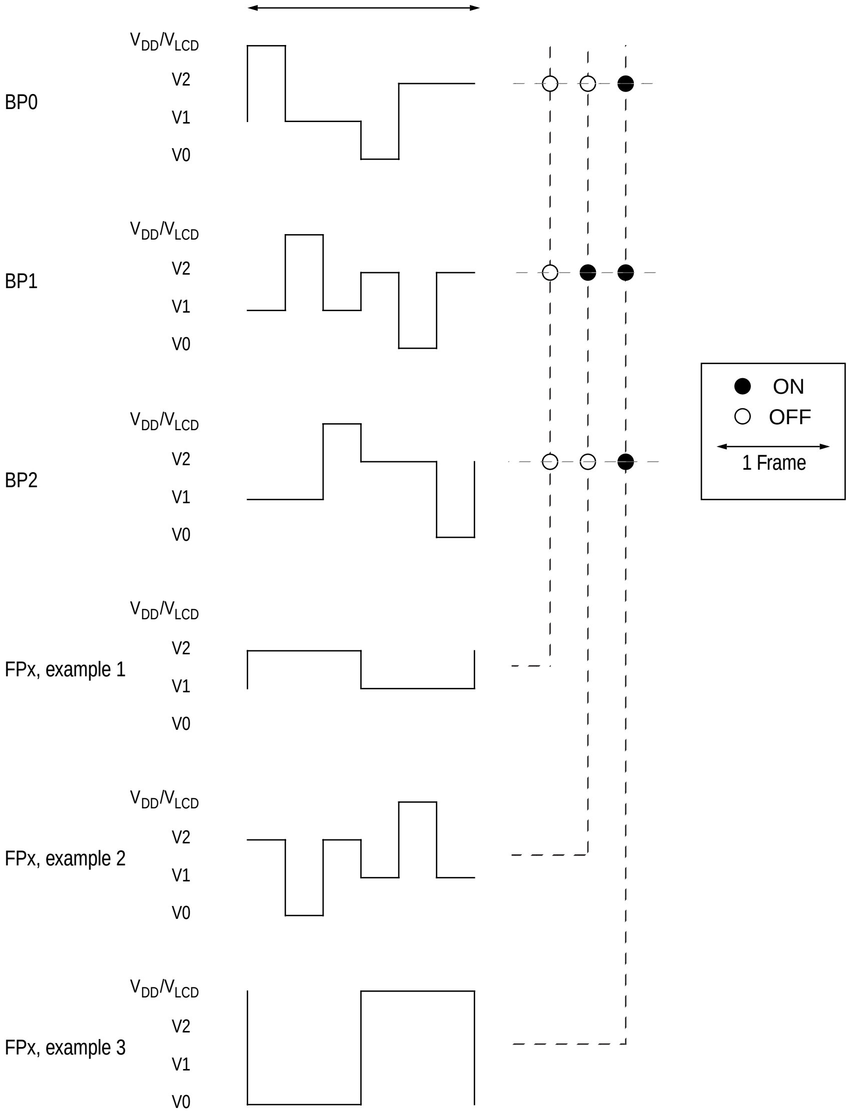


Figure 8-5 LCD waveform with 3 backplanes

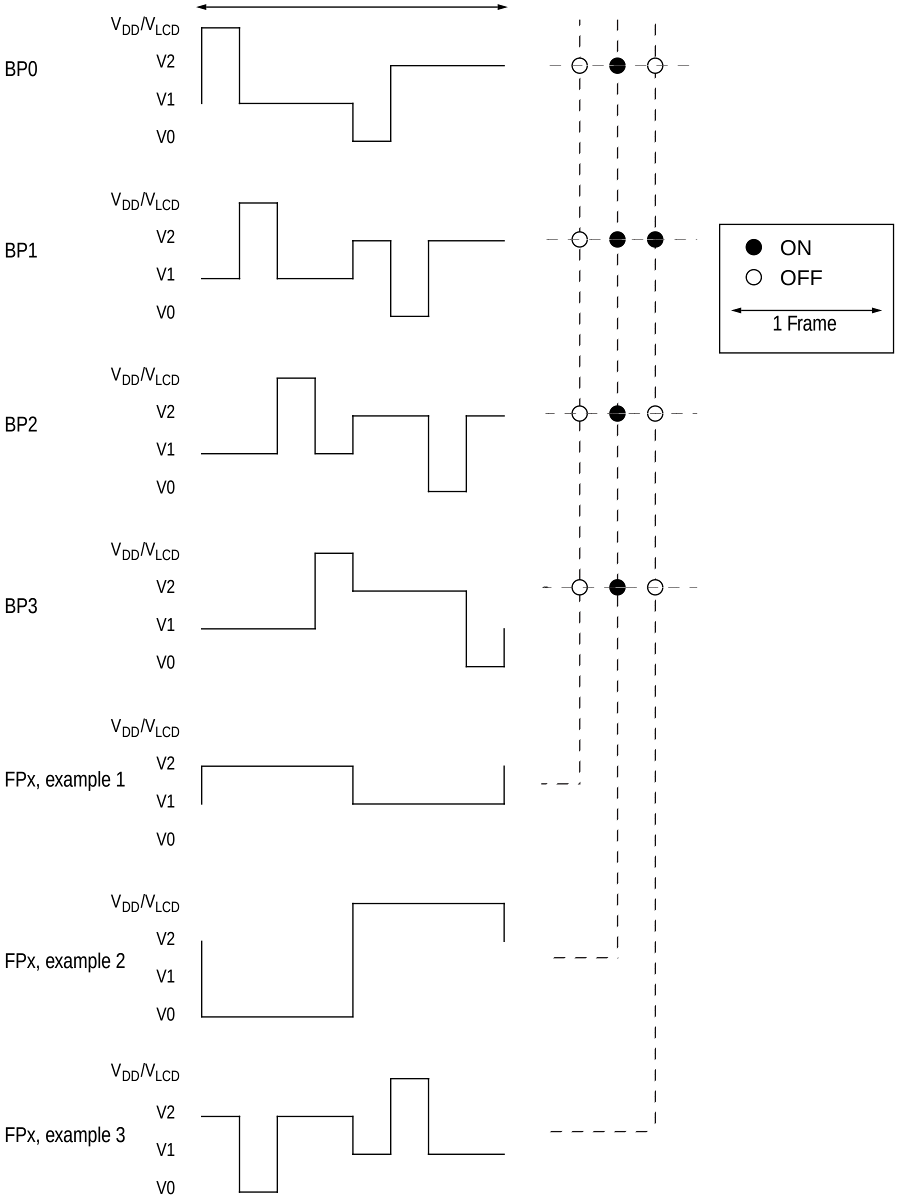


Figure 8-6 LCD waveform with 4 backplanes

8.4 LCD control register (LCD)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
LCD control register (LCD)	\$001E	WTLCDO	FSEL1	FSEL0	INTVLCD	FDISP	MUX4	MUX3	EXTVON	0000 0000

WTLCDO — WAIT mode LCD only

1 (set) – The SPI, the SCI, the second 16-bit timer and the A/D converter are turned off in WAIT mode.

0 (clear) – The SPI, the SCI, the second 16-bit timer and the A/D converter remain active during WAIT mode.

If this bit is set, the SPI, the SCI, the second 16-bit timer and the A/D converter are turned off in WAIT mode, reducing the power consumption, as only the core timer, the first 16-bit timer (timerA), the DMG and the LCD controller remain active.

FSEL1, FSEL0 — LCD operation frequency

These bits select the LCD operation frequency according to Table 8-3. The frequency shown in the right columns are calculated for an external frequency of 3.579 MHz.

Table 8-3 Frequency selection

FSEL1: FSEL0	Framefrequency (2, 4 backplanes)	Frame frequency (3 backplanes)	Frequency for 2,4 backplanes	Frequency for 3 backplanes
0 0	$F_{OSC}/2^{16}$	$4F_{OSC}/(3X2^{16})$	54.6 Hz	72.8 Hz
1 0	$F_{OSC}/2^{15}$	$4F_{OSC}/(3X2^{15})$	109.2 Hz	145.6 Hz
0 1	$F_{OSC}/2^{14}$	$4F_{OSC}/(3X2^{14})$	218.4 Hz	291.3 Hz
1 1	$F_{OSC}/2^{9}$	$4F_{OSC}/(3X2^{9})$	6990 Hz	9320 Hz

INTVLCD — Internal voltage generator ON/OFF

1 (set) – The display is on and an internal voltage generator is activated.

0 (clear) – The internal voltage generator is turned off.

When the INTVLCD bit is set, the display controller is on and an internal voltage generator is activated and connected to the resistor chain (V_{LCD} = 3V approx., if V_{DD} > 3V). See Table 8-2.

FDISP — Display frequency

1 (set) – Extra divide by two stage is included in the LCD clock generator when this bit is set, giving a reduced frame rate.

0 (clear) – Default frame rate is used.

For example, in the 3-way multiplexing mode, a frame rate of 36.8 Hz instead of 72.8 Hz can be achieved.

MUX4, MUX3 — Multiplex ratio

These two bits select the multiplex ratio to be 2, 3 or 4 backplanes. See Table 8-4.

Table 8-4 Multiplex ratio/backplane selection

MUX4	MUX3	Backplanes	Bias	Frequency
0	0	2	1/2	54.6 Hz
0	1	3	1/3	72.8 Hz
1	0	4	1/3	54.6 Hz
1	1	2	1/3	54.6 Hz

EXTVON — External LCD voltage ON/OFF

1 (set) – External LCD voltage is connected.

0 (clear) – External LCD voltage is disconnected.

Clearing this bit disconnects the voltage generator resistor chain from the external V_{LCD}. See Table 8-2.

8.5 LCD during WAIT mode

The LCD drivers function normally during WAIT mode and will keep the display active if the EXTVON bit or the INTVLCD bit is set.

8.6 LCD during STOP mode

During STOP mode the LCD controller is disabled. The driver outputs are discharged by the resistor chain.

9
A/D CONVERTER

The analog to digital converter system consists of a 12-channel, multiplexed input to a successive approximation A/D converter. Eight of the A/D input channels are connected to pins PD0–PD7 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 $4F. A further four channels are available internally for test purposes. In addition to the A/D status/control register (ADSCR) there is one 8-bit result data register at address $4E.

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 AVSS 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 AV_{SS}.

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 (ADRC) and can be used to give a sufficiently high clock rate to the A/D converter when the bus speed is too low to provide accurate results (see Section 9.2.1). When the A/D converter is not being used it can be disconnected using the ADON bit in the ADSCR register, in order to save power (see Section 9.2.1).

9.1 A/D converter operation

The A/D converter consists of an analog multiplexer, an 8-bit digital-to-analog capacitor array, a comparator and a successive approximation register (SAR). See Figure 9-1.

The A/D reference inputs 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 ($4E), and the conversion complete flag, COCO, is set in the A/D status/control register ($4F).

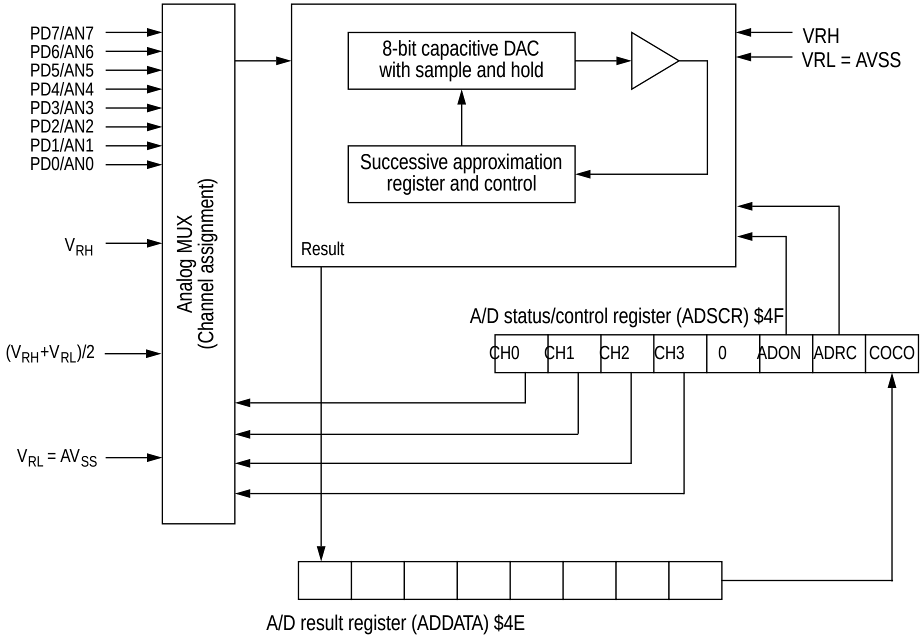


Figure 9-1 A/D converter block diagram

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

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

9.2 A/D registers

9.2.1 A/D status/control register (ADSCR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D status/control (ADSCR)	$004F	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0	0000 0000

COCO — Conversion complete flag

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

1 (set) – COCO flag is set each time a conversion is complete, allowing the new result to be read from the A/D result data register ($4E). The converter then starts a new conversion.

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

Reset clears the COCO flag.

ADRC — A/D RC oscillator control

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

1 (set) – The A/D RC oscillator is turned on and, if ADON is set, the A/D runs from the internal RC oscillator clock (see Table 9-1).

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

When the A/D RC oscillator is turned on, it takes a time t_{RCON} to stabilize (see Table 16-5). During this time A/D conversion results may be inaccurate.

Table 9-1 A/D clock selection

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

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

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

ADON — A/D converter on

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

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

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

When the A/D converter is switched on, it takes a time t_{ADON} for the current sources to stabilize (see Table 16-5). 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.

CH3 – CH0 — A/D channel selection

The CH3–CH0 bits allow the user to determine which channel of the A/D converter multiplexer is selected (see Table 9-2).

Table 9-2 A/D channel assignment

CH3	CH2	CH1	CH0	Channel	Signal
0	0	0	0	0	AD0/PD0
0	0	0	1	1	AD1/PD1
0	0	1	0	2	AD2/PD2
0	0	1	1	3	AD3/PD3
0	1	0	0	4	AD4/PD4
0	1	0	1	5	AD5/PD5
0	1	1	0	6	AD6/PD6
0	1	1	1	7	AD7/PD7
1	1	0	0	8	V_{REFH}
1	1	0	1	9	$(V_{REFH}+V_{REFL})/2$
1	1	1	0	10	V_{REFL}
1	1	1	1	11	Factory test

9.2.2 A/D result data register (ADDATA)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D data register	$004E									Undefined

The A/D data register 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 ADSCR register is set.

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

9.3 A/D converter during WAIT mode

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

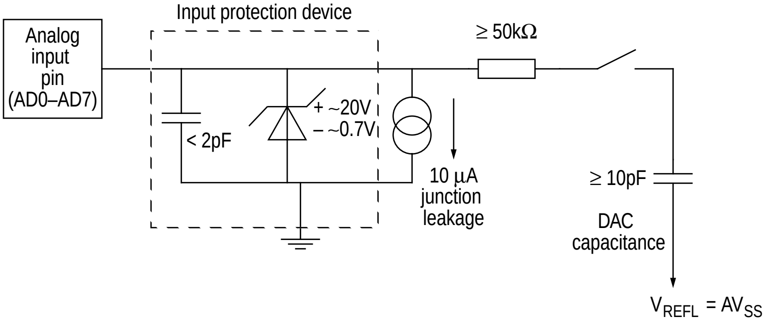
9.4 A/D converter during STOP mode

In STOP mode the comparator and charge pump are turned off and the A/D converter ceases to operate. Any pending conversion is aborted. When the clock begins oscillation upon leaving the STOP mode, a finite amount of time passes before the A/D circuits stabilize enough to provide conversions to the specified accuracy. Normally, the delays built into the MC68HC05F32 are sufficient for this purpose, therefore no explicit delays need to be built into the software.

9.5 A/D analog input

The external analog voltage value to be processed by the A/D converter is sampled on an internal capacitor through a resistive path, provided by input-selection switches and a sampling aperture time switch, as shown in Figure 9-2. Sampling time is limited to 12 bus clock cycles. After sampling, the analog value is stored on the capacitor and held until the end of conversion. During this hold time, the analog input is disconnected from the internal A/D system and the external voltage source sees a high impedance input.

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



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

Figure 9-2 Electrical model of an A/D input pin

10
SERIAL PERIPHERAL INTERFACE

10.1 Overview and features

The SPI is a synchronous interface which allows several SPI microcontrollers or SPI-type peripherals to be interconnected. In a serial peripheral interface, separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. The high-end SPI system may be configured either as a master or as a slave.

Features,

- Full-duplex, 3-wire synchronous transfers
- Master or slave operation
- Master bit frequency, $f_{OP}/2$
- Slave bit frequency, f_{OP}
- Four programmable master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection
- Easy interface to simple expansion parts (PLLs, D/As, latches, display drivers, etc.)

10.2 SPI signal descriptions

Four I/O pins located at port C (PC4 - PC7) are associated with the SPI data transfers. They are the serial clock (SCK), the master in/slave out data line (MISO), the master out / slave in data line (MOSI), and the active-low slave select ($\overline{SS}$). When the SPI system is not utilized (SPE bit cleared in the serial peripheral control register), the four pins (MISO, MOSI, SCK, and $\overline{SS}$) are configured as general-purpose I/O pins. The four SPI signals are discussed in the following paragraphs for both master mode and slave mode of operation.

10.2.1 Master in slave out (MISO)

The MISO line is configured as an input, in a master device, and as an output in a slave device. It is one of the two lines that transfer serial data in one direction. The MISO line of a slave device is placed in the high-impedance state if the slave is not selected.

10.2.2 Master out slave in (MOSI)

The MOSI line is configured as an output in a master device, and as an input in a slave device. It is one of the two lines that transfer serial data in one direction.

10.2.3 Serial clock (SCK)

The serial clock is used to synchronize data movement both in and out of the device through its MOSI and MISO lines. The master and slave devices are capable of exchanging a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in Figure 10-1, four different timing relationships may be selected by control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing. The master device always places data on the MOSI line a half cycle before the clock edge (SCK), in order for the slave device to latch the data.

Two bits (SPR0 and SPR1) in the SPI control register (SPCR) of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on the operation of the SPI.

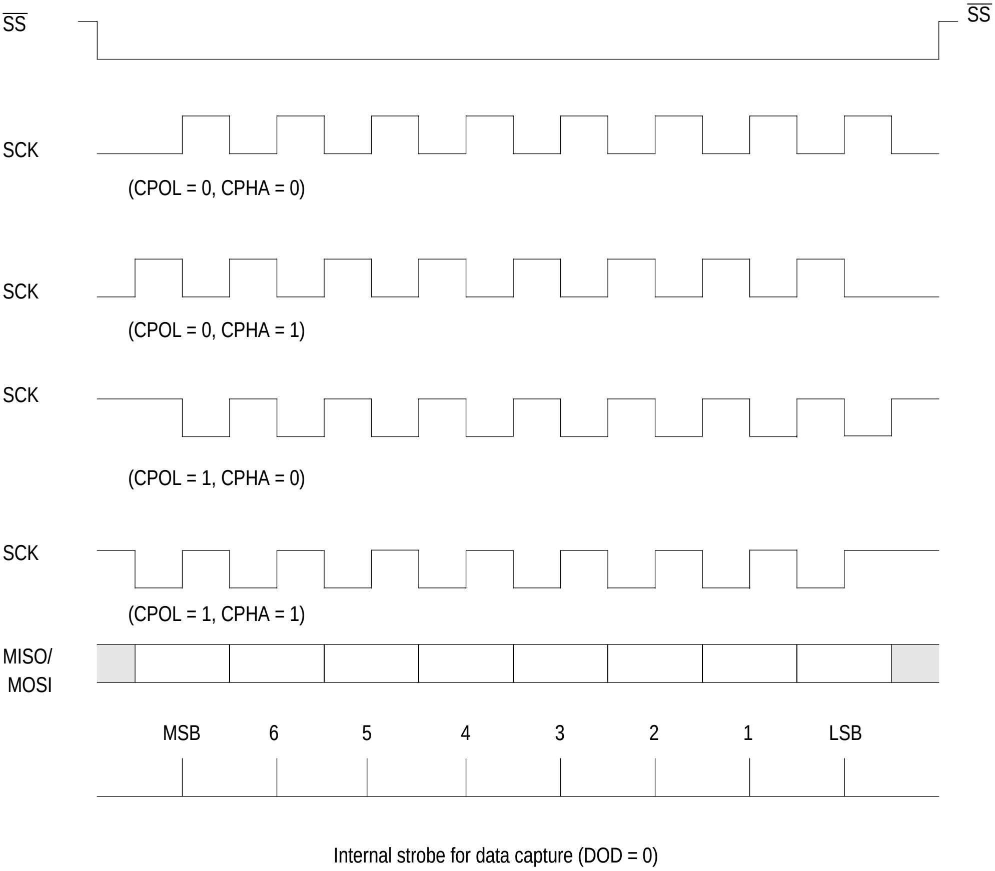


Figure 10-1 Data clock timing diagram

10.2.4 Slave select ($\overline{SS}$)

The slave select ($\overline{SS}$) input line is used to select a slave device. It must be in the active low state prior to data transactions and must stay low for the duration of the transaction. The $\overline{SS}$ line on the master must be tied high. If it goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR). When CPHA = 0, the shift clock is the logical OR of $\overline{SS}$ and SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ may be left low for several SPI characters. If there is only one SPI slave MCU, its $\overline{SS}$ line may be tied to $V_{SS,}$ provided CPHA = 1 clock modes are used.

10.3 Functional description

Figure 10-2 shows a block diagram of the serial peripheral interface circuitry. When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the master's MISO line. This implies full duplex transmission with both data out and data in synchronized to the same clock signal. Thus, the byte transmitted is replaced by the byte received, eliminating the need for separate transmitter-empty and receiver-full status bits. A single status bit (SPIF) is used to signify that the I/O operation has been completed.

10

The SPI is double buffered on read, but not on write. If a write is performed during data transfer, the transfer is not interrupted, and the write will be unsuccessful. This condition will cause the write collision status bit (WCOL) in the SPSR to be set. After a data byte is shifted, the SPIF flag in the SPSR is set.

In master mode, the SCK pin is an output. It idles high or low, depending on the CPOL bit in the SPCR, until data is written to the shift register. Eight clocks are then generated to shift the eight bits of data, after which SCK goes idle again.

In slave mode, the slave start logic receives a logic low on the $\overline{SS}$ pin and a clock input at the SCK pin, thus synchronizing the slave to the master. Data from the master is received serially via the slave MOSI line and is loaded into the 8-bit shift register. The data is then transferred, in parallel, from the 8-bit shift register to the read buffer. During a write cycle, data is written into the shift register, then the slave waits for a clock train from the master to shift the data out on the slave's MISO line.

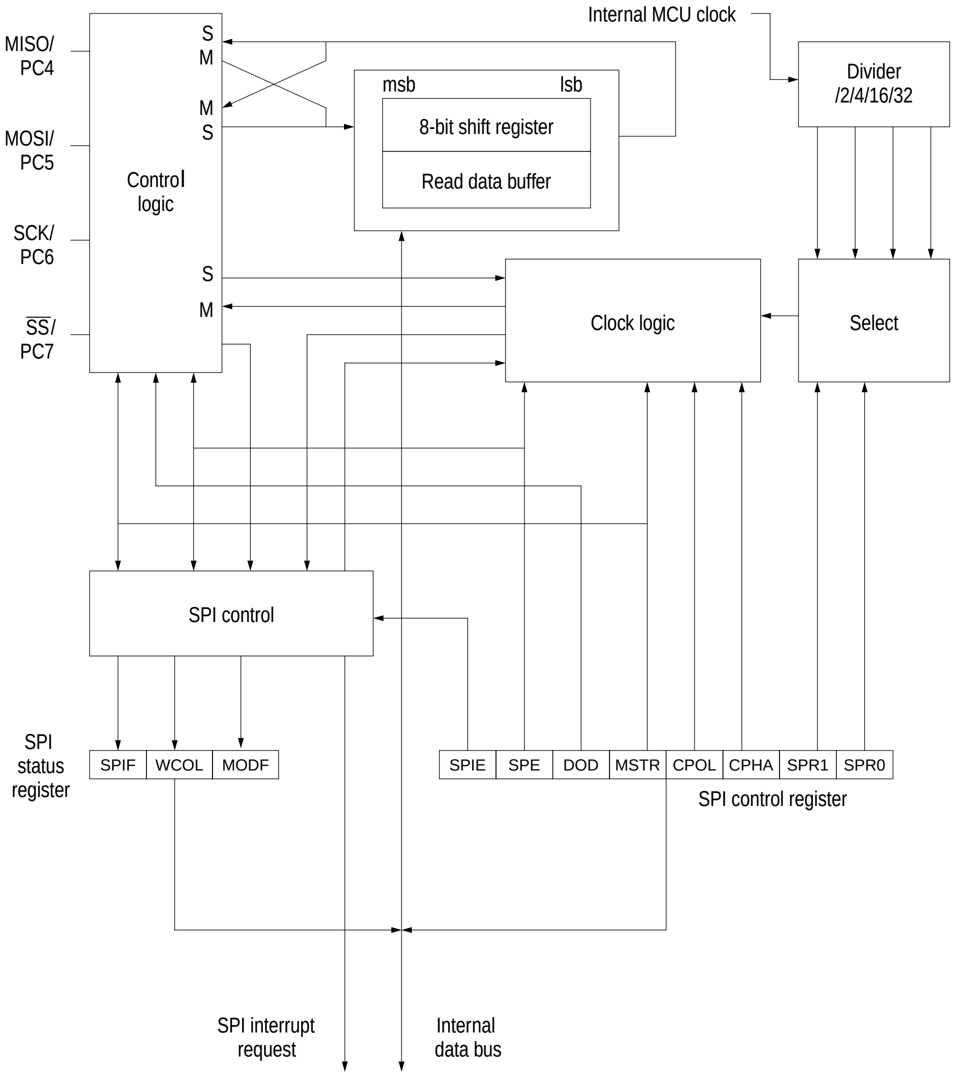


Figure 10-2 Serial peripheral interface block diagram

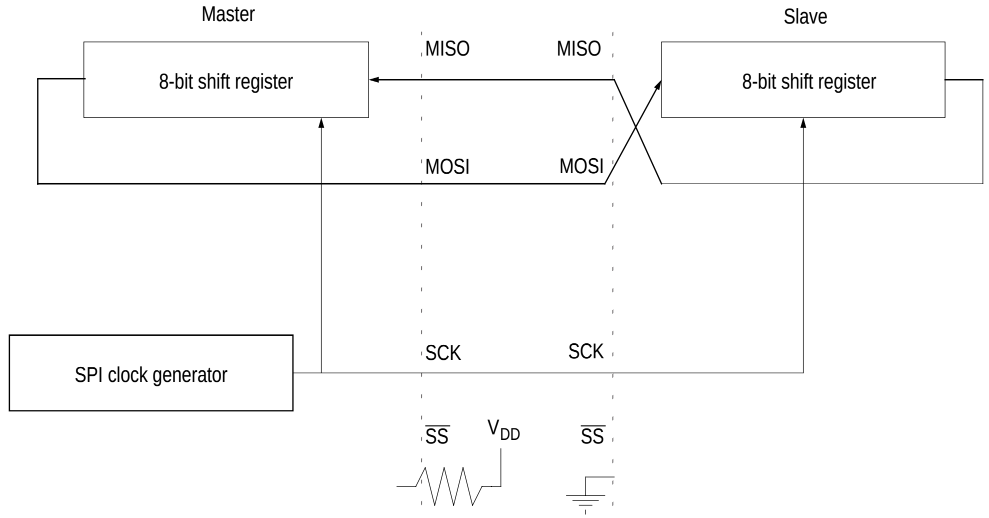


Figure 10-3 Serial peripheral interface master–slave interconnection

10.4 SPI registers

There are three registers in the serial peripheral interface which provide control, status and data storage functions. These registers are called: the serial peripheral control register (SPCR), the serial peripheral status register (SPSR) and the serial peripheral data I/O register (SPDAT).

10.4.1 Control register (SPCR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI control register (SPCR)	$0044	SPIE	SPE	DOD	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu

SPIE — SPI interrupt enable

1 (set) – SPI interrupts enabled.

0 (clear) – SPI interrupts disabled.

When this bit is set to one, a hardware interrupt sequence is requested each time the SPIF or MODF status flag is set. SPI interrupts are inhibited if this bit is clear or if the I bit in the CC register is set.

SPE — SPI system enable

1 (set) – SPI system on.

0 (clear) – SPI system off.

When the SPE bit is set, port C pins 4, 5, 6, and 7 are dedicated to the SPI function.

DOD — Direction of data

This bit determines the direction of the data flow in or out of the serial shift register.

1 (set) – data is transferred LSB first.

0 (clear) – data is transferred MSB first (default state).

MSTR — Master/slave mode select

1 (set) – master mode is selected.

0 (clear) – slave mode is selected.

CPOL — Clock polarity

When the clock polarity bit is cleared and data is not being transferred, a steady state low value is produced at the SCK pin of the master device. Conversely, if this bit is set, the SCK pin will idle high. This bit is also used in conjunction with the clock phase control bit to produce the desired clock-data relationship between master and slave. See Figure 10-1.

CPHA — Clock phase

The clock phase bit, in conjunction with the CPOL bit, controls the clock-data relationship between master and slave. The CPOL bit can be thought of simply as inserting an inverter in series with the SCK line. The CPHA bit selects one of two fundamentally different clocking protocols. When CPHA = 0, the shift clock is the logical OR of SCK and $\overline{\text{SS}}$. As soon as $\overline{\text{SS}}$ goes low, the transaction begins and the first edge on SCK invokes the first data sample. When CPHA = 1, the $\overline{\text{SS}}$ pin may be thought of as a simple output enable control. Refer to Figure 10-1.

SPR1, SPR0 — SPI clock (SCK) rate select bits

If the device is a master, the two serial peripheral rate bits select one of four division ratios of the E-clock to be used as SCK (See Table 10-1). These bits have no effect in slave mode.

Table 10-1 SPI rate selection

SPR1	SPR0	E clock divided by
0	0	2
0	1	4
1	0	16
1	1	32

10.4.2 Status register (SPSR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI status register (SPSR)	$0045	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000

SPIF — SPI interrupt request flag

The serial peripheral data transfer flag bit is set after the eighth SCK cycle in a data transfer and it is cleared by reading the SPSR register (with SPIF set) followed by reading from or writing to the SPI data register (SPDAT).

WCOL — Write collision

The write collision bit is used to indicate that a serial transfer was in progress when the MCU tried to write new data into the SPDAT data register. The MCU write is disabled to avoid writing over the data being transmitted. No interrupt is generated because the error status flag can be read upon completion of the transfer that was in progress at the time of the error. This flag is automatically cleared by a read of the SPSR (with WCOL set) followed by an access (read or write) to the SPDAT register.

MODF — SPI mode error interrupt status flag

This flag is set if the $\overline{SS}$ signal goes to its active-low level while the SPI is configured as a master (MSTR = 1). This condition is not permitted in normal operation. This flag is automatically cleared by a read of the SPSR (with MODF set) followed by a write to the SPCR register.

10.4.3 SPI data I/O register (SPDAT)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI data/IO register (SPDAT)	$0046									uuuu uuuu

The serial peripheral data I/O register is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte, and this will only occur in the master device. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices.

When the user reads the serial peripheral data I/O register, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist. In cases of overrun, the byte which causes the overrun is lost. A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

10.5 SPI during WAIT mode

When the MCU enters wait mode, the CPU clock is halted. All CPU action is suspended; however, the SPI system remains active. In fact an interrupt from the SPI causes the processor to exit the wait mode.

10.6 SPI during STOP mode

When the MCU enters the stop mode, the internal oscillator is turned off, and the baud rate generator which drives the SPI shuts down. This essentially stops all master mode SPI operation, thus transfer is halted until the MCU exits the stop mode. If the stop mode is exited by a reset, then the appropriate control/status bits are cleared and the SPI is disabled. If the device is in the slave mode when the stop instruction is executed, the slave SPI will still operate. It can still accept data and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in the stop mode, no flags are set until the MCU is “waked up” by an interrupt ($\overline{\text{IRQ}}$, keyboard, LVI or CPI). Caution should be observed when operating the SPI (as a slave) during the stop mode because none of the protection circuitry (write collision, mode fault, etc.) is active.

THIS PAGE LEFT BLANK INTENTIONALLY

11
SERIAL COMMUNICATIONS INTERFACE

A full-duplex asynchronous serial communications interface (SCI) is provided with a standard non-return-to-zero (NRZ) format and a variety of baud rates. The SCI transmitter and receiver are functionally independent and have their own baud rate generator; however they use the same baud rate and data format.

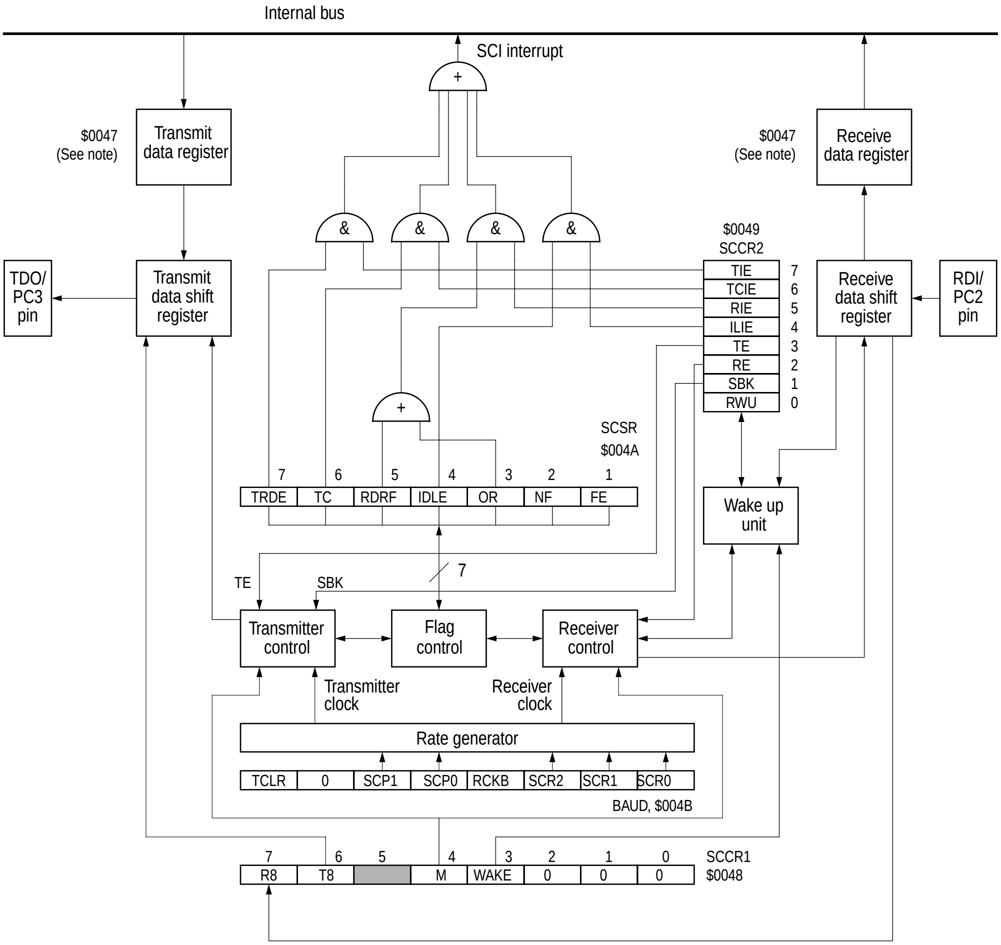
The serial data format is standard mark/space (NRZ) and provides one start bit, eight or nine data bits, and one stop bit.

Any SCI bidirectional communication requires a two-wire system: receive data in (RDI) and transmit data out (TDO).

‘Baud’ and ‘bit rate’ are used synonymously in the following description.

11.1 SCI two-wire system features

- Standard NRZ (mark/space) format
- Advanced error detection method with noise detection for noise duration of up to 1/16th bit time
- Full-duplex operation (simultaneous transmit and receive)
- 32 software selectable baud rates
- Software selectable word length (eight or nine bits)
- Separate transmitter and receiver enable bits
- Interrupt drive capability
- Four separate enable bits for interrupt control



Note: The serial communications data register (SCDAT) is controlled by the internal R/$\overline{\text{W}}$ signal. It is the transmit data register when written to and the receive data register when read.

Figure 11-1 Serial communications interface block diagram

11.2 SCI receiver features

- Receiver wake-up function (idle line or address bit)
- Idle line detection
- Framing error detection
- Noise detection
- Overrun detection
- Receiver data register full flag

11.3 SCI transmitter features

- Transmit data register empty flag
- Transmit complete flag
- Send break

11.4 External connections

The external operation of the SCI block is routed through bits 2 and 3 of port C. Bits PC2 and PC3 are the receive and transmit pins for the SCI (RDI, TDO). Refer to Section 4 for a full description of port C.

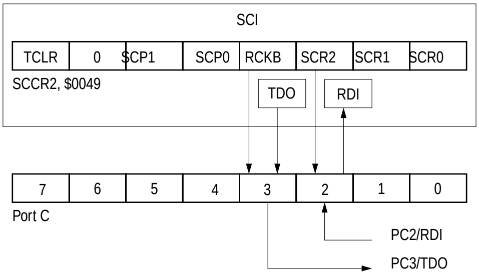


Figure 11-2 SCI and port C

11.5 Functional description

A block diagram of the SCI is shown in Figure 11-1. Option bits in serial control register1 (SCCR1) select the ‘wake-up’ method (WAKE bit) and data word length (M bit) of the SCI. SCCR2 provides control bits that individually enable the transmitter and receiver, enable system interrupts and provide the wake-up enable bit (RWU) and the send break code bit (SBK). Control bits in the baud rate register (BAUD) allow the user to select one of 32 different baud rates for the transmitter and receiver (see Section 11.11.5).

Data transmission is initiated by writing to the serial communications data register (SCDR). Provided the transmitter is enabled, data stored in the SCDR is transferred to the transmit data shift register. This transfer of data sets the transmit data register empty flag (TDRE) in the SCI status register (SCSR) and generates an interrupt (if transmitter interrupts are enabled). The transfer of data to the transmit data shift register is synchronized with the bit rate clock. All data is transmitted least significant bit first. Upon completion of data transmission, the transmission complete flag (TC) in the SCSR is set (provided no pending data, preamble or break is to be sent) and an interrupt is generated (if the transmit complete interrupt is enabled). If the transmitter is disabled, and the data, preamble or break (in the transmit data shift register) has been sent, the TC bit will also be set. This will also generate an interrupt if the transmission complete interrupt enable bit (TCIE) is set. If the transmitter is disabled during a transmission, the character being transmitted will be completed before the transmitter gives up control of the TDO pin.

When SCDR is read, it contains the last data byte received, provided that the receiver is enabled. The receive data register full flag bit (RDRF) in the SCSR is set to indicate that a data byte has been transferred from the input serial shift register to the SCDR; this will cause an interrupt if the receiver interrupt is enabled. The data transfer from the input serial shift register to the SCDR is synchronized by the receiver bit rate clock. The OR (overrun), NF (noise), or FE (framing) error flags in the SCSR may be set if data reception errors occurred.

An idle line interrupt is generated if the idle line interrupt is enabled and the IDLE bit (which detects idle line transmission) in SCSR is set. This allows a receiver that is not in the wake-up mode to detect the end of a message or the preamble of a new message, or to resynchronize with the transmitter. A valid character must be received before the idle line condition or the IDLE bit will not be set and idle line interrupt will not be generated.

11.6 Data format

Receive data or transmit data is the serial data that is transferred to the internal data bus from the receive data input pin (RDI) or from the internal bus to the transmit data output pin (TDO). The non-return-to-zero (NRZ) data format shown in Figure 11-3 is used and must meet the following criteria:

- The idle line is brought to a logic one state prior to transmission/reception of a character.
- A start bit (logic zero) is used to indicate the start of a frame.
- The data is transmitted and received least significant bit first.
- A stop bit (logic one) is used to indicate the end of a frame. A frame consists of a start bit, a character of eight or nine data bits, and a stop bit.
- A break is defined as the transmission or reception of a low (logic zero) for at least one complete frame time (10 zeros for 8-bit format, 11 zeros for 9-bit).

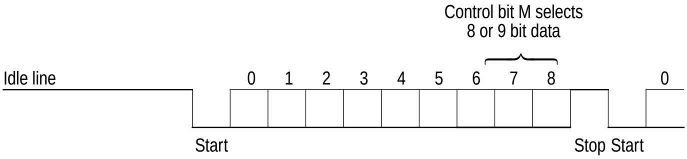


Figure 11-3 Data format

11.7 Receiver wake-up operation

The receiver logic hardware also supports a receiver wake-up function which is intended for systems having more than one receiver. With this function a transmitting device directs messages to an individual receiver or group of receivers by passing addressing information as the initial byte(s) of each message. The wake-up function allows receivers not addressed to remain in a dormant state for the remainder of the unwanted message. This eliminates any further software overhead to service the remaining characters of the unwanted message and thus improves system performance.

The receiver is placed in wake-up mode by setting the receiver wake-up bit (RWU) in the SCCR2 register. While RWU is set, all of the receiver related status flags (RDRF, IDLE, OR, NF, and FE) are inhibited (cannot become set). Note that the idle line detect function is inhibited while the RWU bit is set. Although RWU may be cleared by a software write to SCCR2, it would be unusual to do so. Normally RWU is set by software and is cleared automatically in hardware by one of the two methods described below.

11.7.1 Idle line wake-up

In idle line wake-up mode, a dormant receiver wakes up as soon as the RDI line becomes idle. Idle is defined as a continuous logic high level on the RDI line for ten (or eleven) full bit times. Systems using this type of wake-up must provide at least one character time of idle between messages to wake up sleeping receivers, but must not allow any idle time between characters within a message.

11.7.2 Address mark wake-up

In address mark wake-up, the most significant bit (MSB) in a character is used to indicate whether it is an address (1) or data (0) character. Sleeping receivers will wake up whenever an address character is received. Systems using this method for wake-up would set the MSB of the first character of each message and leave it clear for all other characters in the message. Idle periods may be present within messages and no idle time is required between messages for this wake-up method.

11.8 Receive data in (RDI)

Receive data is the serial data that is applied through the input line and the SCI to the internal bus. The receiver circuitry clocks the input at a rate equal to 16 times the baud rate. This time is referred to as the RT rate in Figure 11-5.

The receiver clock generator is controlled by the baud rate register, as shown in Figure 11-1; however, the SCI is synchronized by the start bit, independent of the transmitter.

Once a valid start bit is detected, the start bit, each data bit and the stop bit are sampled three times at RT intervals 8 RT, 9 RT and 10 RT (1 RT is the position where the bit is expected to start), as shown in Figure 11-4. The value of the bit is determined by voting logic which takes the value of the majority of the samples. A noise flag is set when all three samples on a valid start bit or data bit or the stop bit do not agree .

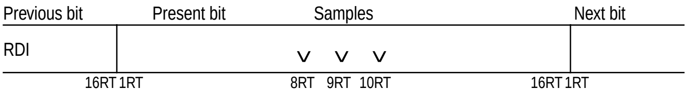


Figure 11-4 SCI sampling technique used on all bits

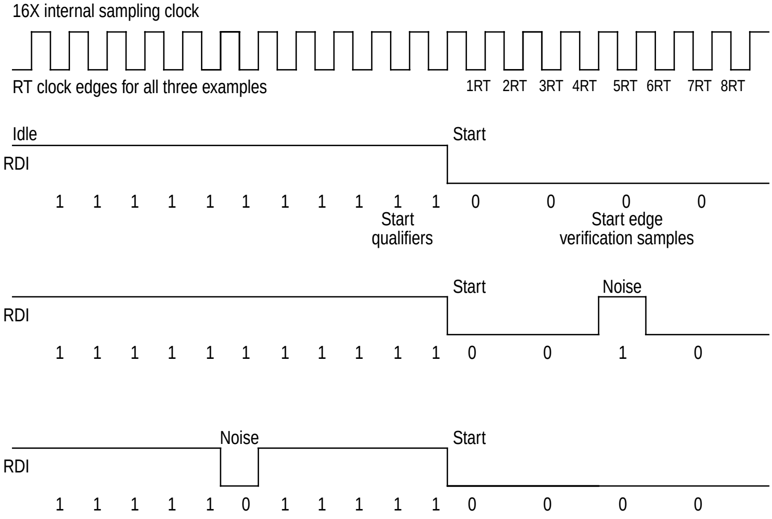


Figure 11-5 SCI examples of start bit sampling technique

11.9 Start bit detection

When the input (idle) line is detected low, it is tested for three more sample times (referred to as the start edge verification samples in Figure 11-5). If at least two of these three verification samples detect a logic zero, a valid start bit has been detected, otherwise the line is assumed to be idle. A noise flag is set if one of the three verification samples detect a logic one, thus a valid start bit could be assumed with a set noise flag present.

If there has been a framing error without detection of a break (10 zeros for 8 bit format or 11 zeros for 9 bit format), the circuit continues to operate as if there actually was a stop bit, and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic one start qualifiers (shown in Figure 11-5) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see Figure 11-6); therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $00) produced the framing error, the start bit will not be artificially induced and the receiver must actually detect a logic one before the start bit can be recognised (see Figure 11-7).

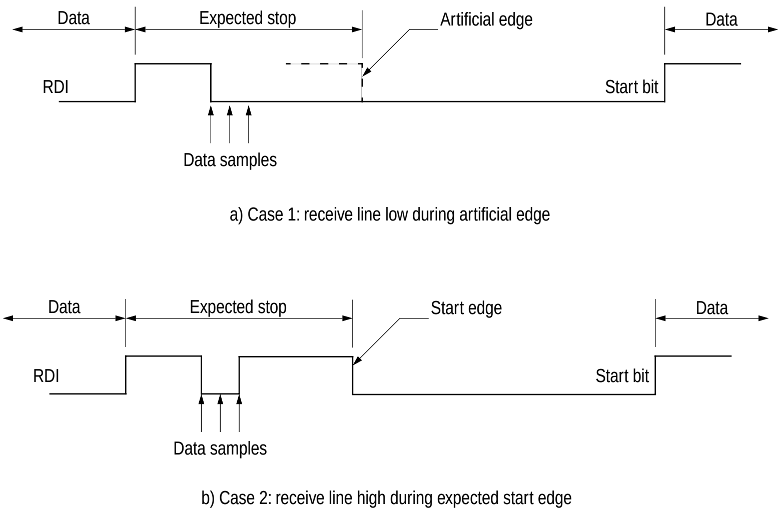


Figure 11-6 Artificial start following a framing error

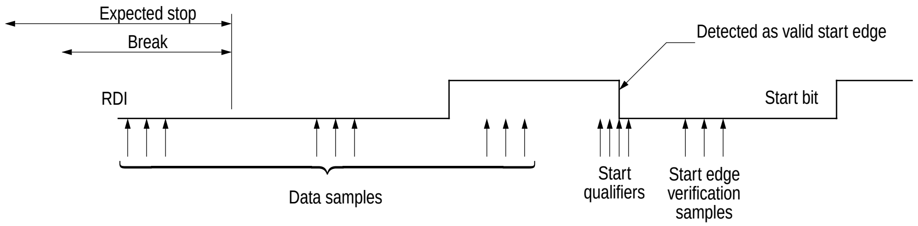


Figure 11-7 SCI start bit following a break

11.10 Transmit data out (TDO)

Transmit data is the serial data from the internal data bus that is applied through the SCI to the output line. Data format is as discussed in Section 11.6 and shown in Figure 11-3. The transmitter generates a bit time by using a derivative of the RT clock, thus producing a transmission rate equal to 1/16th that of the receiver sample clock.

11.11 SCI registers

The SCI system is configured and controlled by five registers: SCDAT, SCCR1, SCCR2, SCSR, and BAUD.

11.11.1 Serial communications data register (SCDAT)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI data (SCDAT)	$0047									undefined

The SCDAT is controlled by the internal R/$\overline{\text{W}}$ signal and performs two functions in the SCI. It acts as the receive data register (RDR) when it is read and as the transmit data register (TDR) when it is written. Figure 11-1 shows this register as two separate registers, RDR and TDR. The RDR provides the interface from the receive shift register to the internal data bus and the TDR provides the parallel interface from the internal data bus to the transmit shift register.

The receive data register is a read-only register containing the last byte of data received from the shift register for the internal data bus. The RDR full bit (RDRF) in the serial communications status register is set to indicate that a byte has been transferred from the input serial shift register to the SCDAT. The transfer is synchronized with the receiver bit rate clock (from the receiver control) as shown in Figure 11-1. All data is received with the least significant bit first.

The transmit data register (TDR) is a write-only register containing the next byte of data to be applied to the transmit shift register from the internal data bus. As long as the transmitter is enabled, data stored in the SCDAT is transferred to the transmit shift register (after the current byte in the shift register has been transmitted).

The transfer is synchronized with the transmitter bit rate clock (from the transmitter control) as shown in Figure 11-1. All data is received with the least significant bit first.

11.11.2 Serial communications control register 1 (SCCR1)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI control 1 (SCCR1)	$0048	R8	T8	0	M	WAKE	0	0	0	0000 0000

The SCI control register 1 (SCCR1) contains control bits related to the nine data bit character format and the receiver wake-up feature.

R8 — Receive data bit 8

This read-only bit is the ninth serial data bit received when the SCI system is configured for nine data bit operation (M = 1). The most significant bit (bit 8) of the received character is transferred into this bit at the same time as the remaining eight bits (bits 7–0) are transferred from the serial receive shift register to the SCI receive data register.

T8 — Transmit data bit 8

This read/write bit is the ninth data bit to be transmitted when the SCI system is configured for nine data bit operation (M = 1). When the eight low order bits (bits 7–0) of a transmit character are transferred from the SCI data register to the serial transmit shift register, this bit (bit 8) is transferred to the ninth bit position of the shift register.

M — Mode (select character format)

The read/write M-bit controls the character length for both the transmitter and receiver at the same time. The 9th data bit is most commonly used as an extra stop bit or it can also be used as a parity bit (see Table 11-1).

1 (set) – Start bit, 9 data bits, 1 stop bit.

0 (clear) – Start bit, 8 data bits, 1 stop bit.

Table 11-1 Method of receiver wake-up

WAKE	M	Method of receiver wake-up
0	x	Detection of an idle line allows the next data type received to cause the receive data register to fill and produce an RDRF flag.
1	0	Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags.
1	1	Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags.

x = Don’t care

WAKE — Wake-up mode select

This bit allows the user to select the method for receiver wake-up. The WAKE bit can be read or written to any time. See Table 11-1.

1 (set) – Wake-up on address mark; if RWU is set, SCI will wake-up if the 8th (if M = 0) or the 9th (if M = 1) bit received on the Rx line is set.

0 (clear) – Wake-up on idle line; if RWU is set, SCI will wake-up after 11 (if M = 0) or 12 (if M = 1) consecutive ‘1’s on the Rx line.

11.11.3 Serial communications control register 2 (SCCR2)

The SCI control register 2 (SCCR2) provides the control bits that enable/disable individual SCI functions.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI control (SCCR2)	$0049	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000

TIE — Transmit interrupt enable

1 (set) – TDRE interrupts enabled.

0 (clear) – TDRE interrupts disabled.

TCIE — Transmit complete interrupt enable

1 (set) – TC interrupts enabled.

0 (clear) – TC interrupts disabled.

RIE — Receiver interrupt enable

1 (set) – RDRF and OR interrupts enabled.

0 (clear) – RDRF and OR interrupts disabled.

ILIE — Idle line interrupt enable

1 (set) – IDLE interrupts enabled.

0 (clear) – IDLE interrupts disabled.

TE — Transmitter enable

When the transmit enable bit is set, the transmit shift register output is applied to the TDO line. Depending on the state of control bit M (SCCR1), a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted when software sets the TE bit from a cleared state.

After loading the last byte in the serial communications data register and receiving the TDRE flag, the user should clear TE. Transmission of the last byte will then be completed before the transmitter gives up control of the TDO pin. While the transmitter is active, PC3 is forced to be an output.

1 (set) – Transmitter enabled.

0 (clear) – Transmitter disabled.

RE — Receiver enable

1 (set)	–	Receiver enabled.
0 (clear)	–	Receiver disabled.

When RE is clear (receiver disabled) all the status bits associated with the receiver (RDRF, IDLE, OR, NF and FE) are inhibited. While the receiver is enabled, PC2 is forced to be an input.

RWU — Receiver wake-up

When the receiver wake-up bit is set by the user software, it puts the receiver to sleep and enables the wake-up function. The type of wake-up mode for the receiver is determined by the WAKE bit discussed above (in SCCR1). When the RWU bit is set, no status flags will be set. Flags which were set previously will not be cleared when RWU is set.

If the WAKE bit is cleared, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M =1) consecutive ones. Under these conditions, RWU cannot be set if the line is idle. If the WAKE bit is set, RWU is cleared after receiving an address bit. The RDRF flag will then be set and the address byte stored in the receiver data register.

SBK — Send break

If the send break bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle sending data. If SBK remains set, the transmitter will continually send whole blocks of zeros (sets of 10 or 11) until cleared. At the completion of the break code, the transmitter sends at least one high bit to guarantee recognition of a valid start bit. If the transmitter is currently empty and idle, setting and clearing SBK is likely to queue two character times of break because the first break transfers almost immediately to the shift register and the second is then queued into the parallel transmit buffer.

11.11.4 Serial communications status register (SCSR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI status (SCSR)	$004A	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	1100 0000

The serial communications status register (SCSR) provides inputs to the interrupt logic circuits for generation of the SCI system interrupt. In addition, a noise flag bit and a framing error bit are also contained in the SCSR.

TDRE — Transmit data register empty flag

This bit is set when the contents of the transmit data register are transferred to the serial shift register. New data will not be transmitted unless the SCSR register is read before writing to the transmit data register to clear the TDRE flag.

If the TDRE bit is clear, this indicates that the transfer has not yet occurred and a write to the serial communications data register will overwrite the previous value. The TDRE bit is cleared by accessing the serial communications status register (with TDRE set) followed by writing to the serial communications data register.

TC — Transmit complete flag

This bit is set to indicate that the SCI transmitter has no meaningful information to transmit (no data in shift register, no preamble, no break). When TC is set the serial line will go idle (continuous MARK). The TC bit is cleared by accessing the serial communications data register (with TC set) followed by writing to the serial communications data register. It does not inhibit the transmitter function in any way.

RDRF — Receive data register full flag

This bit is set when the contents of the receiver serial shift register are transferred to the receiver data register.

If multiple errors are detected in any one received word, the NF and RDRF bits will be affected as appropriate during the same clock cycle. The RDRF bit is cleared when the serial communications status register is accessed (with RDRF set) followed by a read of the serial communications data register.

IDLE — Idle line detected flag

This bit is set when a receiver idle line is detected (the receipt of a minimum of ten/eleven consecutive ‘1’s). This bit will not be set by the idle line condition when the RWU bit is set. This allows a receiver that is not in the wake-up mode to detect the end of a message, detect the preamble of a new message or resynchronize with the transmitter. The IDLE bit is cleared by accessing the serial communications status register (with IDLE set) followed by a read of the serial communications data register. Once cleared, IDLE will not be set again until after RDRF has been set, (i.e. until after the line has been active and becomes idle again).

OR — Overrun error flag

This bit is set when a new byte is ready to be transferred from the receiver shift register to the receiver data register and the receive data register is already full (RDRF bit is set). Data transfer is inhibited until the RDRF bit is cleared. Data in the serial communications data register is valid in this case, but additional data received during an overrun condition (including the byte causing the overrun) will be lost.

The OR bit is cleared when the serial communications status register is accessed (with OR set) followed by a read of the serial communications data register.

NF — Noise error flag

This bit is set if there is noise on a ‘valid’ start bit, any of the data bits or on the stop bit. The NF bit is not set by noise on the idle line nor by invalid start bits. If there is noise, the NF bit is not set until the RDRF flag is set. Each data bit is sampled three times as described in Section 11.8.

The NF bit represents the status of the byte in the serial communications data register. For the byte being received (shifted in) there will be also a ‘working’ noise flag, the value of which will be transferred to the NF bit when the serial data is loaded into the serial communications data register. The NF bit does not generate an interrupt because the RDRF bit gets set with NF and can be used to generate the interrupt.

The NF bit is cleared when the serial communications status register is accessed (with NF set) followed by a read of the serial communications data register.

FE — Framing error flag

This bit is set when the word boundaries in the bit stream are not synchronized with the receiver bit counter (generated by the reception of a logic zero bit where a stop bit was expected). The FE bit reflects the status of the byte in the receive data register and the transfer from the receive shift register to the receive data register is inhibited by an overrun. The FE bit is set during the same cycle as the RDRF bit but does not get set in the case of an overrun (OR). The framing error flag inhibits further transfer of data into the receive data register until it is cleared.

The FE bit is cleared when the serial communications status register is accessed (with FE set) followed by a read of the serial communications data register.

11.11.5 Baud rate register (BAUD)

The baud rate register (BAUD) is used to set the bit rate for the SCI system. Normally this register is written once, during initialization, to set the baud rate for SCI communications. Both the receiver and the transmitter use the same baud rate which is derived from the MCU bus rate clock. A two stage divider is used to develop custom baud rates from normal MCU crystal frequencies, therefore it is not necessary to use special baud rate crystal frequencies.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI baud rate (BAUD)	$004B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	0000 0uuu

TCLR — Clear baud rate counters (test purposes only)

This bit is disabled and remains low in any mode other than test or bootstrap. Reset clears this bit. While in test or bootstrap mode, setting this bit causes the baud rate counter chains to be reset. The logic one state of this bit is transitory, reads always a return a logic zero. This control bit is only intended for factory testing of the MCU

SCP1, SCP0 — Serial prescaler select bits

These read/write bits determine the prescale factor by which the internal processor clock is divided before it is applied to the transmitter and receiver rate control dividers. This common prescaled output is used as the input to a divider that is controlled by the SCR0–SCR2 bits for the SCI receiver and transmitter.

Table 11-2 First prescaler stage

SCP1	SCP0	Prescaler division ratio (PRS1)
0	0	1
0	1	3
1	0	4
1	1	13

SCR2, SCR1, SCR0 — SCI rate select bits

These three read/write bits select the baud rates for the transmitter and the receiver. The prescaler output is divided by the factors shown in Table 11-3.

Table 11-3 Second prescaler stage

SCR2	SCR1	SCR0	Prescaler division ratio (PRS2)
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

RCKB — SCI receive baud rate clock test

This bit is disabled and remains low in any mode other than test or bootstrap. Reset clears this bit. While in test or bootstrap mode, this bit may be written but not read (reads always return a logic zero). Setting this bit enables a baud rate counter test mode, where the exclusive-or of the receiver clock (16 times the baud rate) is driven out of the PC3/TDO pin. This control bit is intended only for factory testing of the MCU.

11.12 Baud rate selection

The flexibility of the baud rate generator allows many different baud rates to be selected, depending on the CPU clock frequency. A particular baud rate may be generated by manipulating the various prescaler and division ratio bits.

The SCI baud rate can be calculated from the internal bus clock and the two prescaler factors, PRS1 and PRS2. The first prescaler factor, PRS1, is selected with SCP0 and SCP1, as shown in Table 11-2. The second prescaler factor, PRS2, is selected with SCR0, SCR1 and SCR2, as shown in Table 11-3. The SCI baud rate B equals the internal bus clock E, divided by 16, divided by PRS1, divided by PRS2 (B = E/16/PRS1/PRS2).

Note: For the receiver, the internal clock frequency is 16 times higher than the selected baud rate.

11.13 SCI during STOP mode

When the MCU enters STOP mode, the baud rate generator driving the receiver and transmitter is shut down. This stops all SCI activity. Both the receiver and the transmitter are unable to operate.

If the STOP instruction is executed during a transmitter transfer, that transfer is halted. When STOP mode is exited as a result of an external interrupt, that particular transmission resumes.

If the receiver is receiving data when the STOP instruction is executed, received data sampling is stopped (baud generator stops) and the rest of the data is lost.

Warning: For the above reasons, all SCI transactions should be in the idle state when the STOP instruction is executed.

11.14 SCI during WAIT mode

The SCI system is not affected by WAIT mode and continues normal operation. Any valid SCI interrupt will wake-up the system. If required, the SCI system can be disabled prior to entering WAIT mode by writing a zero to the transmitter and receiver enable bits in the serial communication control register 2 at $0049. This action will result in a reduction of power consumption during WAIT mode.

12
PULSE WIDTH MODULATOR

12.1 PWM introduction

The pulse width modulator (PWM) system has three 8-bit channels (PWM1, PWM2, and PWM3). The PWM has a programmable period of 256xT, where T can be E/2, E/4, and E/8 for an output frequency of 4 KHz, 2KHz, and 1 KHz respectively with E = 2MHz. E is the internal bus frequency fixed to half of the external oscillator frequency.

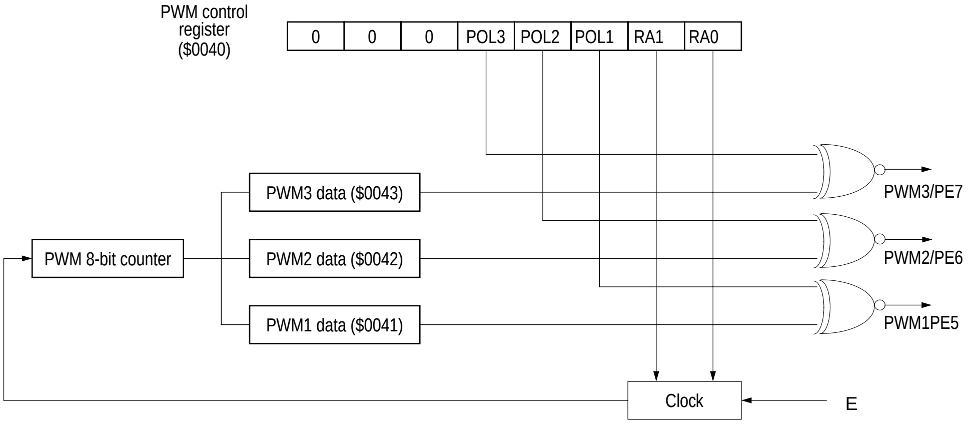


Figure 12-1 PWM block diagram

12.2 Functional description

The PWM is capable of generating signals from 0% to 100% duty cycle. A $00 in the PWM data register yields an ‘OFF’ output (0%), but an $FF yields a duty of 255/256. To achieve the 100% duty (‘ON’ output), the polarity control bit is set to active low (POL = 0) for that channel (i.e. PWM0 and PWM1) while the data register has $00 in it.

When not in use, the PWM system can be shut off to save power by clearing the clock rate select bits RA0 and RA1 in the PWM control register (PWCR). The PWM starts conversion immediately after programming bits RA0 and RA1 in the PWM control register. The PWM outputs are connected to port E if the corresponding bit in the port E control register is set.

The PWM output can have an active high or an active low pulse under software control.

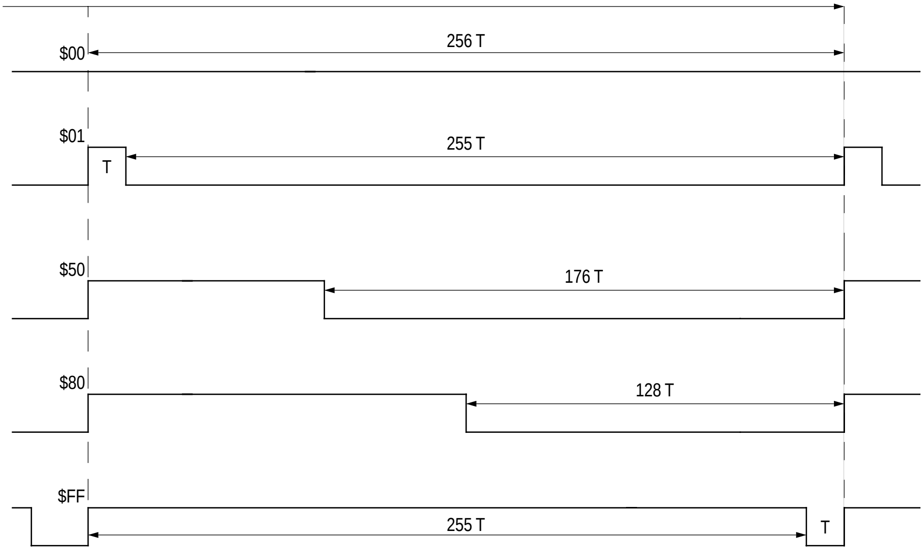


Figure 12-2 PWM output waveforms (POL = 1, active high)

12.3 Registers

There are three PWM data registers and a control register associated with the PWM system. These registers can be written to and read at any time.

After reset the user should write to the data registers and to the polarity select bits prior to enabling the PWM system (i.e. prior to setting RA1 and/or RA0 for PWM input clock rate). This will avoid an erroneous duty cycle being driven.

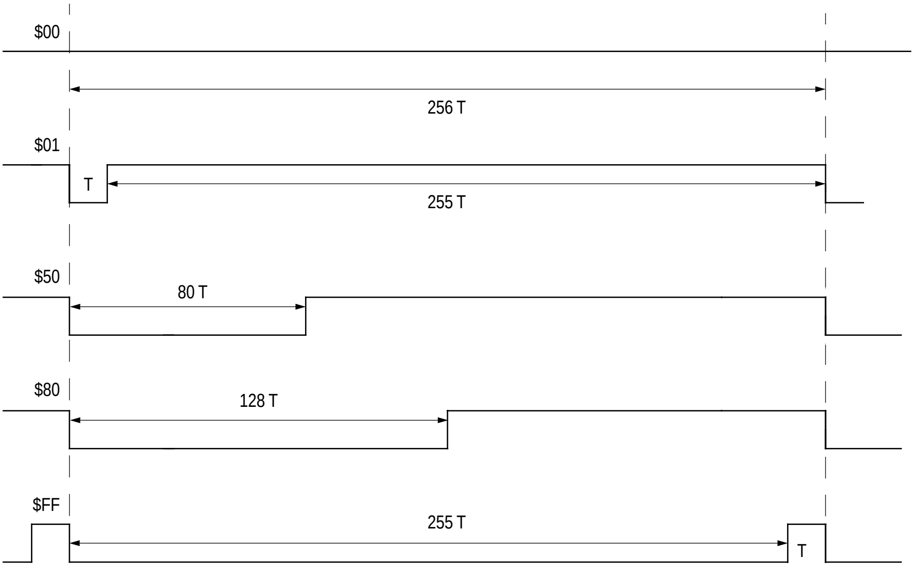


Figure 12-3 PWM waveforms (POL = 0, active low)

12.3.1 PWM control (PWMCR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PWM control (PWMCR)	$0040	0	0	0	POL3	POL2	POL1	RA1	RA0	0001 1100

POL1 — PWM1 polarity

1 (set) – makes the PWM1 pulse active high

0 (clear) – makes the PWM1 pulse active low

POL2 — PWM2 polarity

1 (set) – makes the PWM2 pulse active high

0 (clear) – makes the PWM2 pulse active low

POL3 — PWM3 polarity

1 (set) – makes the PWM3 pulse active high

0 (clear) – makes the PWM3 pulse active low

RA1, RA0 — PWM clock rate bits

These bits select the input clock rate and determine the period.

Note: The polarity bits and the PWM clock rate bits are not latched until the end of conversion. They affect the PWM output immediately. For proper operation these control bits must not be changed during conversion.

Table 12-1 PWM clock rate

RA1:RA0	PWM input clock
0 0	OFF
0 1	E/2
1 0	E/4
1 1	E/8

12.3.2 PWM data registers (PWMD)

The PWM system has three 8-bit data registers which hold the duty cycle for each PWM output. PWM data1, PWM data2, and PWM data3 are the data registers located at $41-$43 respectively.

Note: These registers are affected by RESET

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PWM data1 (PWMD1)	$0041									1000 0000
PWM data2 (PWMD2)	$0042									1000 0000
PWM data3 (PWMD3)	$0043									1000 0000

12.4 PWM during WAIT mode

The PWM continues normal operation during WAIT mode. To decrease power consumption during WAIT, it is recommended that the rate select bits in the PWM control register are cleared if the PWM D/A converter is not used.

12.5 PWM during STOP mode

In STOP mode the oscillator is stopped causing the PWM to cease operation. Any signal in process is aborted in whatever phase the signal happens to be in.

12.6 PWM during reset

Upon RESET the RA0 and RA1 bits in the PWM control register are cleared, the port E control register is cleared, the data registers are written with $80 and the polarity bits are set. This in effect disables the PWM system. The user should write to the data registers prior to enabling the PWM system (i.e. prior to setting RA1 or RA0). This will avoid an erroneous duty cycle being driven.

THIS PAGE LEFT BLANK INTENTIONALLY

13
32 KHZ CLOCK SYSTEM

13.1 32 kHz clock system

The 32 kHz clock system is mostly independent from the rest of the MCU. WAIT mode and STOP mode do not affect the work of the 32 kHz clock system. For the reason of power saving the oscillator and the divider can be stopped if the oscillator input pin OSC3 is held on fixed potential. The 32 kHz clock system is provided to generate a refresh signal at port E pin 4 and an custom periodic interrupt (CPI) with a period of 0.5s. The refresh frequency and the periodic interrupt are under the control of the custom periodic interrupt control/status register located at $4C.

13.1.1 Custom periodic interrupt control/status register (CPICSR)

The CPICSR contains the interrupt flag CPIF, the interrupt enable bit CPIE and refresh frequency select bits RFQ1, RFQ0.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
CPI control/status (CPICSR)	$004C	0	CPIF	0	CPIE	0	0	RFQ1	RFQ0	0000 0000

CPIF — Custom periodic interrupt flag

CPIF is a clearable, read-only status bit and is set when the 14-bit counter changes from $3FFF to $0000. A CPU interrupt request will be generated if CPIE is set. Clearing the CPIF is done by writing a ‘0’ to it. Writing a ‘1’ to CPIF has no effect on the bit’s value. Reset clears CPIF.

CPIE — Custom periodic interrupt enable

When this bit is cleared, the CPI interrupts are disabled. When this bit is set, a CPU interrupt request is generated when the CPIF bit is set. Reset clears this bit.

RFQ1–RFQ0 — Refresh frequency select

These two read/write bits select one of four taps from the 14-stage counter to provide a refresh clock with a frequency according to Table 13-1. Reset clears these bits, selecting the highest frequency.

Table 13-1 Refresh clock (32.768 kHz crystal)

RFQ1	RFQ0	Refresh clock frequency
0	0	8.192 kHz (reset condition)
0	1	4.096 kHz
1	0	2.048 kHz
1	1	1.024 kHz

13.1.1.1 Refresh clock

If bit 4 in the control register of port E is set, the output of the 32 kHz clock system is connected to the pin PE4/REFRESH. The refresh clock rate is under software control and is specified in Table 13-1.

13.2 Operation during STOP mode

Stop mode does not affect the work of the 32 kHz clock system. If the CPI interrupt is enabled, a custom periodic interrupt will cause the processor to wake up from the STOP mode.

13.3 Operation during WAIT mode

The CPU clock halts during the WAIT mode, but the 32 kHz clock system remains active. If the CPI interrupt is enabled, a custom periodic interrupt will cause the processor to exit the WAIT mode.

14
RESETS AND INTERRUPTS

14.1 Resets

The MC68HC05F32 can be reset in five ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by an on-chip low voltage reset, by an opcode fetch from an illegal address, and by a COP watchdog timer reset. Any of these resets will cause the program to return to its starting address, specified by the contents of memory locations $FFFE and $FFFF, and cause the interrupt mask of the CCR to be set.

14.1.1 Power-on reset

A power-on reset occurs when a positive transition is detected on VDD. The power-on reset function is strictly for power turn-on conditions and should not be used to detect drops in the power supply voltage. The power-on circuitry provides a stabilization delay (t_{PORL}) from when the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of this delay then the processor remains in the reset state until $\overline{\text{RESET}}$ goes high.

14.1.2 $\overline{\text{RESET}}$ pin

When the oscillator is running in a stable state, the MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a minimum period of 1.5 machine cycles (t_{CYC}). This pin contains an internal Schmitt trigger as part of its input to improve noise immunity. When the reset pin goes high, the MCU will resume operation on the following cycle. The $\overline{\text{RESET}}$ pin is also an output device for the internal low voltage reset.

14.1.3 Illegal address reset

When an opcode fetch occurs from an address which is not part of the RAM ($0068 – $03FF) or of the ROM ($8000 – $FFFF) or EEPROM ($0400 – $04FF), the device is automatically reset.

Note: No RTS or RTI instruction should be placed at the end of a memory block since this could result in an illegal address reset.

14.1.4 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.

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 function is a mask option, enabled or disabled during device manufacture. See Section 1.2.

Refer to Section 5.3 for more information on the COP watchdog timer.

14.1.5 Low voltage reset

The MCU contains a low voltage detection circuit which drives the external reset.

For a positive transition of supply voltage v_{DD}, the low voltage reset occurs as long as V_{DD} is below the V_{RON} level. In this case the external reset pin is pulled down. If the supply voltage drops off above the V_{RON} level, the reset is released. If the supply voltage falls off below the V_{ROFF} level, the $\overline{\text{RESET}}$ pin is pulled down.

14.2 Interrupts

The MCU can be interrupted by nine different sources, eight maskable hardware interrupts and one nonmaskable software interrupt:

- External signal on the $\overline{\text{IRQ}}$ pin; $\overline{\text{IRQ}}$ is mask selectable as edge or edge-and-level sensitive
- Keyboard interrupt
- Core timer interrupt
- 16-bit programmable timer interrupt
- Low voltage interrupt (LVI) – EEPROM
- Serial peripheral interface (SPI) interrupt
- Serial communications interface (SCI) interrupt
- 32 kHz clock system interrupt
- Software interrupt instruction (SWI)

Interrupts cause the processor to save the register contents on the stack and to set the interrupt mask (I-bit) to prevent additional interrupts. The RTI instruction (return from interrupt) causes the register contents to be recovered from the stack and normal processing to resume. While executing the RTI instruction, the interrupt mask bit (I-bit) will be cleared providing the corresponding enable bit stored on the stack is zero, i.e. the interrupt is disabled.

Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete. The current instruction is the one already fetched and being operated on. When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not masked (CCR I-bit clear) and the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Figure 14-1 shows the interrupt processing flow.

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

14.2.1 Interrupt priorities

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

14.2.2 Non-maskable 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 $FFFC and $FFFD.

14.2.3 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. $\overline{\text{IRQ}}$ is software selectable as either edge or edge-and-level sensitive (bit 3 of the system option register).

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.

14.2.3.1 Real time and core timer (CTIMER) interrupts

There are two different core timer interrupt flags that cause a CTIMER interrupt whenever an interrupt is enabled and its flag becomes set, namely RTIF and CTOF. The interrupt flags and enable bits are located in the CTIMER control and status register (CTCSR). These interrupts will vector to the same interrupt service routine, whose start address is contained in memory locations $FFF8 and $FFF9 (see Section 5.2.1 and Figure 5-1).

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

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

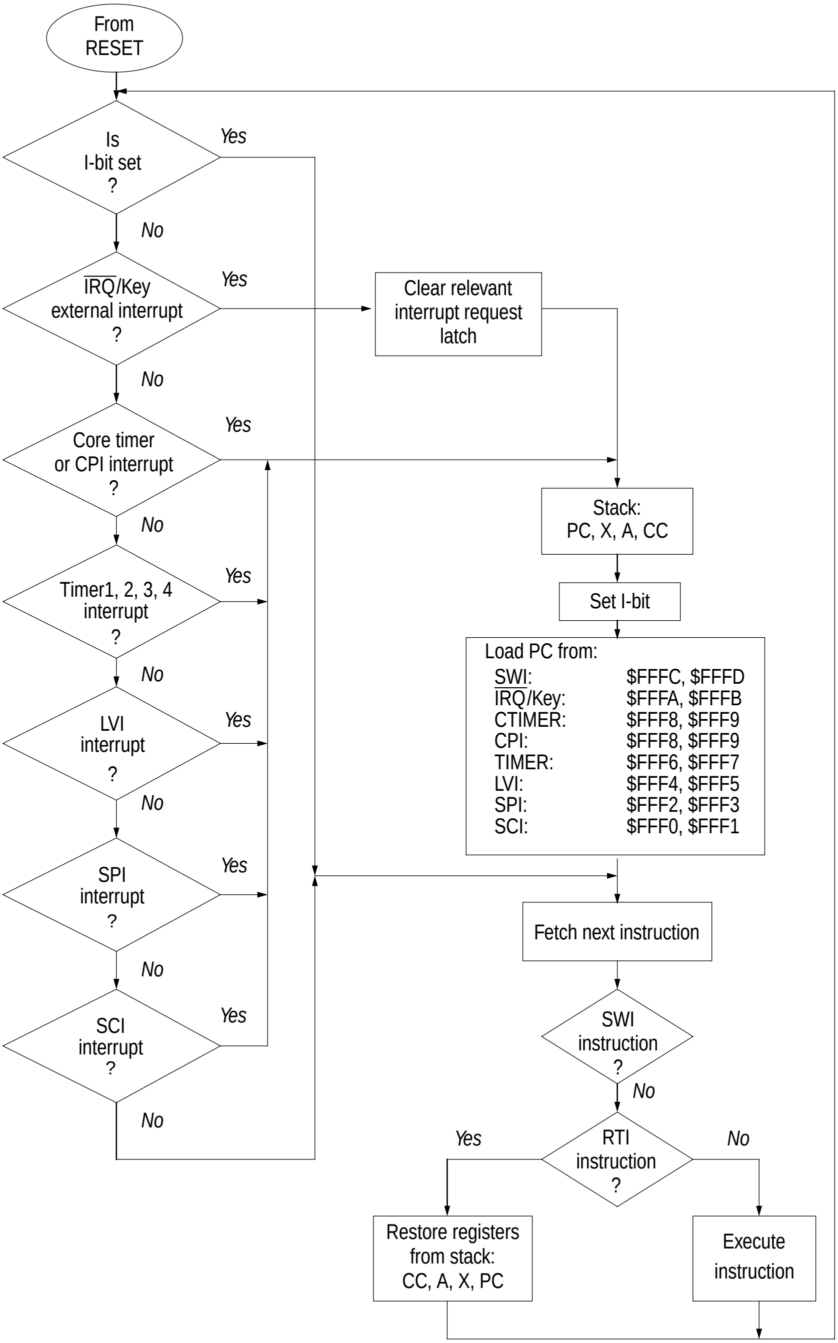


Figure 14-1 Interrupt flowchart

Table 14-1 Vector address for interrupts and reset

Register	Flagname	Interrupts	CPU interrupt	Vector address
—	—	Reset	$\overline{\text{RESET}}$	$FFFE–$FFFF
—	—	Software interrupt	SWI	$FFFC–$FFFD
—	—	External interrupt	$\overline{\text{IRQ}}$	$FFFA–$FFFB
CTCSR	CTOF	Core timer overflow	CTIMER	$FFF8–$FFF9
CTCSR	RTIF	Real time interrupt	CTIMER	$FFF8–$FFF9
CPICSR	CPIF	Custom periodic interrupt	CPI	$FFF8–$FFF9
TSR	ICF1	Timer input capture1	TIMER	$FFF6–$FFF7
TSR	OCF1	Timer output compare1	TIMER	$FFF6–$FFF7
TSR	ICF2	Timer input capture2	TIMER	$FFF6–$FFF7
TSR	OCF2	Timer output compare2	TIMER	$FFF6–$FFF7
TSR2	ICF3	Timer input capture3	TIMER	$FFF6–$FFF7
TSR2	OCF3	Timer output compare3	TIMER	$FFF6–$FFF7
TSR2	ICF4	Timer input capture4	TIMER	$FFF6–$FFF7
TSR2	OCF4	Timer output compare4	TIMER	$FFF6–$FFF7
TSR	TOF	Timer1 overflow	TIMER	$FFF6–$FFF7
TSR2	TOF	Timer2 overflow	TIMER	$FFF6–$FFF7
KEY	KSF	Keyboard interrupt	KEYF	$FFFA–$FFFB
SOR	LVI	Low voltage interrupt	LVI	$FFF4–$FFF5
SPSR	SPIF	SPI request interrupt	SPI	$FFF2–$FFF3
SPSR	MODF	SPI mode error	SPI	$FFF2–$FFF3
SCSR	TDRE	SCI transmit interrupt	SCI	$FFF0–$FFF1
SCSR	TC	SCI transmit complete	SCI	$FFF0–$FFF1
SCSR	RDRF	SCI receive interrupt	SCI	$FFF0–$FFF1
SCSR	IDLE	SCI idle line interrupt	SCI	$FFF0–$FFF1
SCSR	OR	SCI overrun error	SCI	$FFF0–$FFF1

14.2.3.2 Programmable 16-bit timer interrupt

There are ten different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The timer interrupt enable bits are located in the timer control register (TCR) and the timer interrupt flags are located in the timer status registers (TSR1, TSR2). All three interrupts will vector to the same service routine, whose start address is contained in memory locations $FFF6 and $FFF7.

14.2.3.3 Keyboard interrupt

When configured as input pins, all eight port A lines provide a wired-OR keyboard interrupt facility and will generate an interrupt, provided that the keyboard interrupt enable bit (KIE) in the keyboard/timer register (KEY/TIM) is set. The address of the interrupt service routine is specified by the contents of memory locations $0FFA and $0FFB. Since this interrupt vector is shared with the IRQ external interrupt function the interrupt service routine should check KSF to determine the interrupt source. KSF should be cleared by software in the interrupt service routine. Care must be taken to allow adequate time for switch debounce before clearing the flag.

14.2.3.4 Low voltage interrupt

There is a low voltage interrupt flag that causes an interrupt whenever it is set and enabled. The low voltage interrupt enable bit and the interrupt flag are located in the system option register (SOR). This interrupt will vector to the service routine, located at the address specified by the contents of memory locations $FFF4 and $FFF5.

14.2.3.5 Serial peripheral interface (SPI) interrupt

An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag bits in the SPI status register SPSR is set, provided the I-bit in the condition code register is clear and the enable bit SPIE in the SPI control register is enabled. The SPI interrupt causes the program to vector to memory location $FFF2 and $FFF3 which contains the starting address of the interrupt service routine. Software in the SPI service routine must determine the priority and cause of the SPI interrupt by examined the interrupt flag bits located in the SPI status register.

14.2.3.6 Serial communications interface (SCI) interrupt

There are five different interrupt flags (TDRE, TC, OR, RDRE, IDLE) that will cause an SCI interrupt whenever they are set and enabled. These five interrupt flags are found in the five most significant bits of the SCI status register SCSR. The actual processor interrupt is generated only if the I-bit in the condition code register is clear and the enable bit in the serial communication control register 2 (SCCR2) is enabled. The SCI interrupt causes the program counter to vector to the address pointed to by memory locations $FFF0–$FFF1 which contain the start address of the interrupt service routine. Software in the SCI interrupt service routine must determine the priority and cause of the SCI interrupt by examining the interrupt flags and the status bits in the serial communications status register SCSR.

14.2.3.7 Custom periodic interrupt (CPI)

There is a timer interrupt flag that causes a CPI interrupt from the 32 kHz clock system whenever set and enabled. The interrupt flag and enable bits are located in the CPI control and status register (CPICSR). An interrupt will vector to the same interrupt service routine as the core timer interrupts, located at the address specified by the contents of memory location $FFF8 and $FFF9.

14.2.4 Hardware controlled interrupt sequence

The following three functions (RESET, STOP, and WAIT) are not in the strictest sense interrupts. However, they are acted upon in a similar manner. Flowcharts for STOP and WAIT are shown in Figure 2-1.

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

STOP: The STOP instruction causes the oscillator to be turned off and the processor to ‘sleep’ until an external interrupt ($\overline{\text{IRQ}}$), a low voltage interrupt (LVI), a custom periodic interrupt (CPI), or a keyboard interrupt occurs, or the device is reset.

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

15
CPU CORE AND INSTRUCTION SET

This section provides a description of the CPU core registers, the instruction set and the addressing modes of the MC68HC05F32.

15.1 Registers

The MCU contains five registers, as shown in the programming model of Figure 15-1. The interrupt stacking order is shown in Figure 15-2.

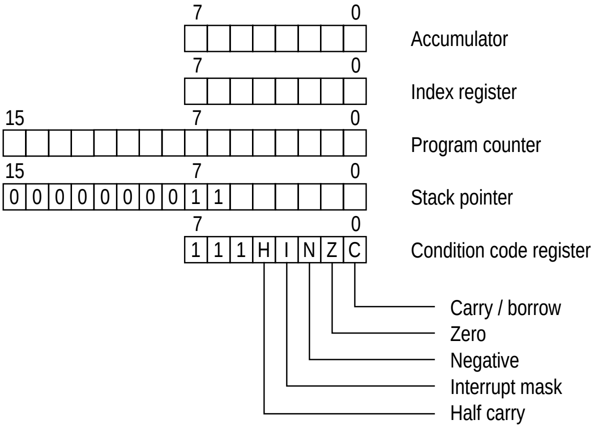


Figure 15-1 Programming model

15.1.1 Accumulator (A)

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

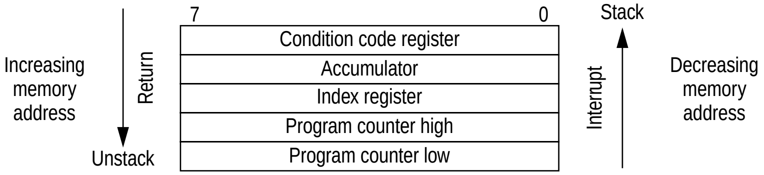


Figure 15-2 Stacking order

15.1.2 Index register (X)

The index register is an 8-bit register, which can contain the indexed addressing value used to create an effective address. The index register may also be used as a temporary storage area.

15.1.3 Program counter (PC)

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

15.1.4 Stack pointer (SP)

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

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

15.1.5 Condition code register (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed, and the fifth bit indicates whether interrupts are masked. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

Half carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, all maskable interrupts are masked. If an interrupt occurs while this bit is set, the interrupt is latched and remains pending until the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

15.2 Instruction set

The MCU has a set of 62 basic instructions. They can be grouped into five different types as follows:

- Register/memory
- Read/modify/write
- Branch
- Bit manipulation
- Control

The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

This MCU uses all the instructions available in the M146805 CMOS family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown in Table 15-1.

15.2.1 Register/memory Instructions

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 15-2 for a complete list of register/memory instructions.

15.2.2 Branch instructions

These instructions cause the program to branch if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to Table 15-3.

15.2.3 Bit manipulation instructions

The MCU can set or clear any writable bit that resides in the first 256 bytes of the memory space (page 0). All port data and data direction registers, timer and serial interface registers, control/status registers and a portion of the on-chip RAM reside in page 0. An additional feature allows the software to test and branch on the state of any bit within these locations. The bit set, bit clear, bit test and branch functions are all implemented with single instructions. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 15-4.

15.2.4 Read/modify/write instructions

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to this sequence of reading, modifying and writing, since it does not modify the value. Refer to Table 15-5 for a complete list of read/modify/write instructions.

15.2.5 Control instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 15-6 for a complete list of control instructions.

15.2.6 Tables

Tables for all the instruction types listed above follow. In addition there is a complete alphabetical listing of all the instructions (see Table 15-7), and an opcode map for the instruction set of the M68HC05 MCU family (see Table 15-8).

Table 15-1 MUL instruction

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

15.3 Addressing modes

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

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

Table 15-2 Register/memory instructions

		Addressing modes																	
		Immediate			Direct			Extended			Indexed (no offset)			Indexed (8-bit offset)			Indexed (16-bit offset)		
Function	Mnemonic	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles
Load A from memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in memory	STA				B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in memory	STX				BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add memory and carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract memory from A with borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND memory with A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic compare A with memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic compare X with memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit test memory with A (logical compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump unconditional	JMP				BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to subroutine	JSR				BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

15.3.1 Inherent

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

15.3.2 Immediate

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

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

Table 15-3 Branch instructions

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

Table 15-4 Bit manipulation instructions

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

15.3.3 Direct

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

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

Table 15-5 Read/modify/write instructions

		Addressing modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (no offset)			Indexed (8-bit offset)		
Function	Mnemonic	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (two's complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate left through carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate right through carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical shift left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical shift right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic shift right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for negative or zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5
Multiply	MUL	42	1	11												

Table 15-6 Control instructions

		Inherent addressing mode		
Function	Mnemonic	Opcode	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set carry bit	SEC	99	1	2
Clear carry bit	CLC	98	1	2
Set interrupt mask bit	SEI	9B	1	2
Clear interrupt mask bit	CLI	9A	1	2
Software interrupt	SWI	83	1	10
Return from subroutine	RTS	81	1	6
Return from interrupt	RTI	80	1	9
Reset stack pointer	RSP	9C	1	2
No-operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

Table 15-7 Instruction set

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

Address mode abbreviations

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

▨ Not implemented

Condition code symbols

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

Table 15-7 Instruction set (Continued)

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

Address mode abbreviations

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

▨ Not implemented

Condition code symbols

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

Table 15-8 M68HC05 opcode map

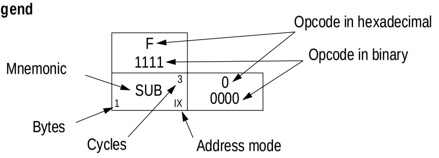
	Bit manipulation		Branch	Read/modify/write					Control		Register/memor y						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
High Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	High Low
0 0000	5 BRSET0 3 BTB	5 BSET0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

Abbre viations for address modes and register s

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

▨ Not implemented

Legend



15.3.4 Extended

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the short form of the instruction.

$$EA = (PC+1):(PC+2);\ PC \leftarrow PC+3$$
$$\text{Address bus high} \leftarrow (PC+1);\ \text{Address bus low} \leftarrow (PC+2)$$

15.3.5 Indexed, no offset

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

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

15.3.6 Indexed, 8-bit offset

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. Therefore the operand can be located anywhere within the lowest 511 memory locations. This addressing mode is useful for selecting the mth element in an n element table.

$$EA = X+(PC+1);\ PC \leftarrow PC+2$$
$$\text{Address bus high} \leftarrow K;\ \text{Address bus low} \leftarrow X+(PC+1)$$
$$\text{where } K = \text{the carry from the addition of } X \text{ and } (PC+1)$$

15.3.7 Indexed, 16-bit offset

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

$$EA = X+[(PC+1):(PC+2)];\ PC \leftarrow PC+3$$
$$\text{Address bus high} \leftarrow (PC+1)+K;\ \text{Address bus low} \leftarrow X+(PC+2)$$
$$\text{where } K = \text{the carry from the addition of } X \text{ and } (PC+2)$$

15.3.8 Relative

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

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

15.3.9 Bit set/clear

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

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

15.3.10 Bit test and branch

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

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

THIS PAGE LEFT BLANK INTENTIONALLY

16
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the MC68HC05F32.

16.1 Maximum ratings

Table 16-1 Maximum ratings

Rating[1]	Symbol	Value	Unit
Supply voltage	V_{DD}	– 0.3 to + 0.7	V
Input voltage	V_{IN}	V_{SS} – 0.3 to V_{SS} + 0.3	V
Bootloader mode ($\overline{IRQ}$ pin only)	V_{IN}	V_{SS} – 0.3 to 2 x V_{DD} + 0.3	V
Current drain per pin[2] — excluding VDD and VSS	I	25	mA
Operating temperature range — standard — extended	T_A	T_L to T_H 0 to + 70 –40 to + 85	°C
Storage temperature range	T_{STG}	– 65 to + 150	°C

[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}.

16.2 Thermal characteristics and power considerations

Table 16-2 Package thermal characteristics

Characteristics	Symbol	Value	Unit
Thermal resistance — 100-pin QFP package — 80-pin QFP package	θ_{JA}	55	°C/W

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

$$T_J = T_A + (P_D \bullet \theta_{JA})$$

where:

T_A = Ambient Temperature (°C)

θ_{JA} = Package Thermal Resistance,

Junction-to-ambient (°C/W)

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

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

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

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

$$P_D = \frac{K}{T_J + 273}$$

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

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

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 16-2.

16.3 DC electrical characteristics

Table 16-3 DC electrical characteristics (V_{DD} = 5.0 V)

(V_{DD} = 5.0V_{DC} ± 10%, V_{SS} = 0 V_{DC}, T_A = –40°C to +85°C, unless otherwise stated)

Characteristic	Symbol	Min.	Typ. [1]	Max.	Unit
Output voltage					
I_{LOAD} = –10 µA	V_{OH}	V_{DD} – 0.1	—	—	V
I_{LOAD} = +10 µA	V_{OL}	—	—	0.1	V
Output high voltage (I_{LOAD} = –0.8 mA) Ports (PA0–7, PB0–7, PC0–7, PH0–7, PI7–0)	V_{OH}	V_{DD} – 0.8	—	—	V
Output low voltage (I_{LOAD} = +1.6 mA) Ports(PA0–7, PB0–7, PC4–7, PD4–7, PE4–7, PH0–7, PI0–7, PJ0–7)	V_{OL}	—	—	0.4	V
Input high voltage Ports (PD0–7, PE0–7)	V_{IH}	0.7V_{DD}	—	15.0	V
Input high voltage Ports (PA0–7, PB0–7, PC0–7, PF0–7, PG0–7) $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$,OSC1, OSC3	V_{IH}	0.7V_{DD}	—	V_{DD}	V
Input low voltage Ports (PA0–7, PB0–7, PC0–7, PF0–7, PG0–7) $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, OSC3	V_{IL}	—	—	0.2V_{DD}	V
Supply Current[2]	I_{DD}				
RUN		—	5	10	mA
WAIT		—	0.6	1.2	mA
STOP		—	—	80	µA
I/O ports hi-Z leakage current Ports (PA0–7, PB0–7, PC0–7, PD0–7, PE0–7)	I_{OZ}	—	—	10	µA
Input current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, OSC1	I_{IN}	—	—	1	µA
Capacitance					
Ports (as input or output)	C_{OUT}	—	—	12	pF
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$	C_{IN}	—	—	8	pF
Input current low Ports (PA0–7, PB0–7, PC0–7), $\overline{\text{RESET}}$	I_{IL}	– 30	– 90	– 170	µA
LCD step down resistor	R_{LCDSD}	—	20	—	k Ω

[1] Typical values are at midpoint of voltage range and at 25°C only.

[2] All I_{DD} measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs. RUN and WAIT I_{DD}: measured using an external square-wave clock source (f_{OSC} = 3.58 MHz); all inputs0.2V from rail; no DC loads; maximum load on outputs 50pF (20pF on OSC2). WAIT I_{DD}: only the timer system active; current varies linearly with the OSC2 capacitance. STOP and WAIT I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V. STOP I_{DD}: measured with OSC1 = V_{SS}.

Table 16-4 DC electrical characteristics (V_{DD} = 2.7 V)

(V_{DD} = 2.7V_{DC} **min**, V_{SS} = 0 V_{DC}, T_A = –40°C to +85°C, unless otherwise stated**)**

Characteristic	Symbol	Min.	Typ. [(1)]	Max.	Unit
Output voltage I_{LOAD} = –10 µA I_{LOAD} = +10 µA	 V_{OH} V_{OL}	 V_{DD} – 0.1 —	 — —	 — 0.1	 V V
Output high voltage (I_{LOAD} = –0.8 mA) Ports (PA0–7, PB0–7, PC0–7, PH0–7, PI7–0)	V_{OH}	V_{DD} – 0.3	—	—	V
Output low voltage (I_{LOAD} = +1.6 mA) Ports(PA0–7, PB0–7, PC4–7, PD4–7, PE4–7, PH0–7, PI0–7, PJ0–7)	V_{OL}	—	—	0.3	V
Input high voltage Ports (PD0–7, PE0–7)	V_{IH}	0.7V_{DD}	—	15.0	V
Input high voltage Ports (PA0–7, PB0–7, PC0–7, PF0–7, PG0–7) $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$,OSC1, OSC3	V_{IH}	0.7V_{DD}	—	V_{DD}	V
Input low voltage Ports (PA0–7, PB0–7, PC0–7, PF0–7, PG0–7) $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, OSC3	V_{IL}	—	—	0.2V_{DD}	V
Supply Current[(2)] RUN WAIT STOP	I_{DD}	 — — —	 1.8 0.2 —	 3.0 1.0 40	 mA mA µA
I/O ports hi-Z leakage current Ports (PA0–7, PB0–7, PC0–7, PD0–7, PE0–7)	I_{OZ}	—	—	10	µA
Input current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, OSC1	I_{IN}	—	—	1	µA
Capacitance Ports (as input or output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
Input current low Ports (PA0–7, PB0–7, PC0–7), $\overline{\text{RESET}}$	I_{IL}	– 5	– 15	40	µA
LCD step down resistor	R_{LCDSD}	—	20	—	k Ω

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

(2) All I_{DD} measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs. RUN and WAIT I_{DD}: measured using an external square-wave clock source (f_{OSC} = 3.58 MHz); all inputs0.2V from rail; no DC loads; maximum load on outputs 50pF (20pF on OSC2). WAIT I_{DD}: only the timer system active; current varies linearly with the OSC2 capacitance. STOP and WAIT I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V. STOP I_{DD}: measured with OSC1 = V_{SS}.

16.4 Control timing

Table 16-5 Control timing (V_{DD} = 5V)

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

Characteristic	Symbol	Min.	Max.	Unit
Frequency of operation: Crystal External clock	f_{OSC}	 — DC	 3.579 3.579	MHz
Internal operating frequency: Crystal External clock	f_{OP}	 — DC	 1.789 1.789	MHz
Processor cycle time	t_{CYC}	550.0	—	ns
Stop recovery start-up time	t_{ILCH}	—	20.0	ms
Crystal oscillator start-up time	t_{OXOV}	—	20.0	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{LIH}	250.0	—	ns
Interrupt pulse period	t_{ILIL}	(1)	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	100.0	—	ns
RC oscillator stabilization time	t_{RCON}	—	5.0	µs
A/D on current stabilization time	t_{ADON}	—	100.0	µs
EEPROM byte programming time	t_{EPGM}	—	15.0	ms
EEPROM byte erase time	t_{EBYTE}	—	15.0	ms
EEPROM block erase time	t_{EBLOCK}	—	100.0	ms
EEPROM bulk erase time	t_{EBULK}	—	300.0	ms
EEPROM programming voltage fall time	t_{FPV}	—	10.0	µs
EEPROM minimum programming voltage	V_{CCMIN}	2.7		V

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

Table 16-6 Control timing (V_{DD} = 2.7V)

(V_{DD} = 2.7 V_{DC} min, V_{SS} = 0 V_{DC}, T_A = T_L to T_H)

Characteristic	Symbol	Min.	Max.	Unit
Frequency of operation: Crystal External clock	f_{OSC}	 — DC	 3.579 3.579	MHz
Internal operating frequency: Crystal External clock	f_{OP}	 — DC	 1.789 1.789	MHz
Processor cycle time	t_{CYC}	550.0	—	ns
Stop recovery start-up time	t_{ILCH}	—	20.0	ms
Crystal oscillator start-up time	t_{OXOV}	—	20.0	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{LIH}	250.0	—	ns
Interrupt pulse period	t_{ILIL}	(1)	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	100.0	—	ns
RC oscillator stabilization time	t_{RCON}	—	10.0	µs
A/D on current stabilization time	t_{ADON}	—	200.0	µs
EEPROM byte programming time	t_{EPGM}	—	15.0	ms
EEPROM byte erase time	t_{EBYTE}	—	15.0	ms
EEPROM block erase time	t_{EBLOCK}	—	100.0	ms
EEPROM bulk erase time	t_{EBULK}	—	300.0	ms
EEPROM programming voltage fall time	t_{FPV}	—	10.0	µs
EEPROM minimum programming voltage	V_{CCMIN}	2.7		V

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

16.5 DC levels for low voltage $\overline{\text{RESET}}$ and LVI

Table 16-7 DC levels for low voltage reset and LVI

(TA = –40°C to +85°C, unless otherwise stated)

Characteristic	Symbol	Min.	Typ.	Max.	Unit
Power-on reset voltage	V_{RON}	2.55	2.8	3.05	V
Power-off reset voltage	V_{ROFF}	2.45	2.7	2.95	V
Low voltage interrupt	V_{LVI}	2.75	3.0	3.25	V

16.6 Electrical specifications for DTMF/melody generator

Table 16-8 Sine wave tones at TNO

Characteristic	Min.	Typ.	Max.	Unit
Operating voltage	2.7	—	5.5	V
Tone output level: Low group – row High group – column	 0.120 0.160	 0.160 0.205	 0.210 0.280	V_{rms} V_{rms}
Frequency deviation (DTMF)	– 0.65		+ 0.65	%
Frequency deviation (Melody)	– 1.5		+ 1.5	%
Tone output DC level	0.45	0.50	0.55	Vdd
High group pre-emphasis	1	2.15	3	dB
Total harmonic distortion	—	–25	—	dB

Table 16-9 Square wave tones at TNO

Characteristic	Min.	Typ.	Max.	Unit
Operating voltage	2.7	—	5.5	V
Tone output level: Low group – row High group – column	 — —	 0.270 0.360		V_{p-p} V_{p-p}
Frequency deviation (Melody)	– 1.5		+ 1.5	%
Tone output DC level (+ 1/2 V_{p-p} value)	0.45	0.50	0.55	Vdd

Table 16-10 TONEX at TNX output

Characteristic	Min.	Typ.	Max.	Unit
Operating voltage	2.7	—	5.5	V
Tone output level (square wave)		V_{DD}		V_{p-p}
Frequency deviation	– 1.5		+ 1.5	%

16.7 EEPROM additional information

Table 16-11 EEPROM additional information

Temperature	Read/write cycles	Remarks
0 °C – 85 °C	10 000	The value is regularly tested and monitored
50 °C	35 000	This value is predicted from the tested ones
25 °C	100 000	This value is predicted from the tested ones

16.8 PWM timing

Table 16-12 PWM timing

Characteristic	Symbol	Min.	Max.	Unit
PWM rise time	t_{PWMR}	15.0	35.0	ns
PWM fall time	t_{PWMF}	15.0	35.0	ns

16.9 A/D converter characteristics

Table 16-13 A/D converter characteristics

(V_{DD} = 5.0 V_{DC} ± 10%, V_{SS} = 0 V_{DC}, T_A = –40°C to +85°C, unless otherwise stated)

Characteristic	Parameter	Min.	Max.	Unit
Resolution	Number of bits resolved by the A/D	8	—	bit
Non-linearity	Maximum deviation from the best straight line through A/D transfer characteristics (AV_{SS} =V_{RH} = V_{DD}, AV_{SS} = V_{SS})	—	±1/2	LSB
Quantization error	Uncertainty due to converter resolution	—	±1/2	LSB
Absolute accuracy	Difference between the actual input voltage and the full scale equivalent of the binary output code for all errors	—	±1	LSB
Conversion range	Analog input voltage range	AV_{SS}§	V_{RH}	V
V_{RH}	Maximum analog reference voltage	AV_{SS}	V_{DD} + 0.1	V
AV_{SS}	Analog supply voltage	V_{SS} – 0.1	—	V
Conversion time	Total time to perform a single analog to digital conversion a. External clock b. internal RC oscillator	— —	32 32	t_{CYC} µs
Monotinicity	Conversion result never decreases with an increase in input voltage and has no missing codes	GUARANTEED		
Zero input reading	Conversion result when V_{IN} = AV_{SS}	00		Hex
Full scale reading	Conversion result when V_{IN} = V_{RH}	—	FF	Hex
Sample acquisition time(1)	Analog input acquisition sampling a. External clock b. Internal RC oscillator	— —	12 12	t_{CYC} µs
Sample/hold capacitance	Input capacitance on AN0–AN7		12	pF
Input leakage	Input leakage on AN0–AN7 Input leakage on V_{RH}	— —	10 1	µA µA
AV_{DD}	Analog supply voltage	—	1.125V_{RH}	V

(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.

(3) A/D accuracy may decrease as V_{RH} is reduced below 4V.

THIS PAGE LEFT BLANK INTENTIONALLY

17
MECHANICAL DATA

17.1 100-pin QFP pinout for the MC68HC05F32

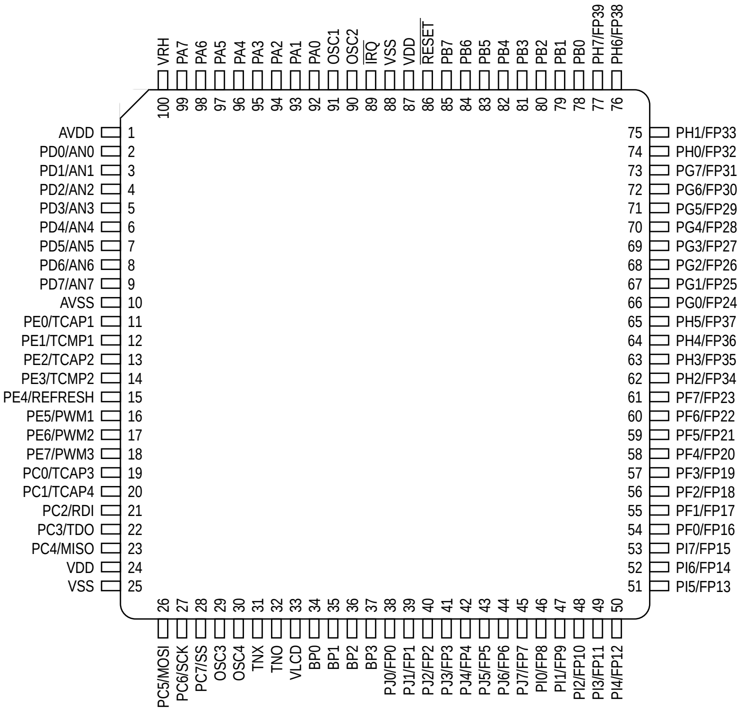
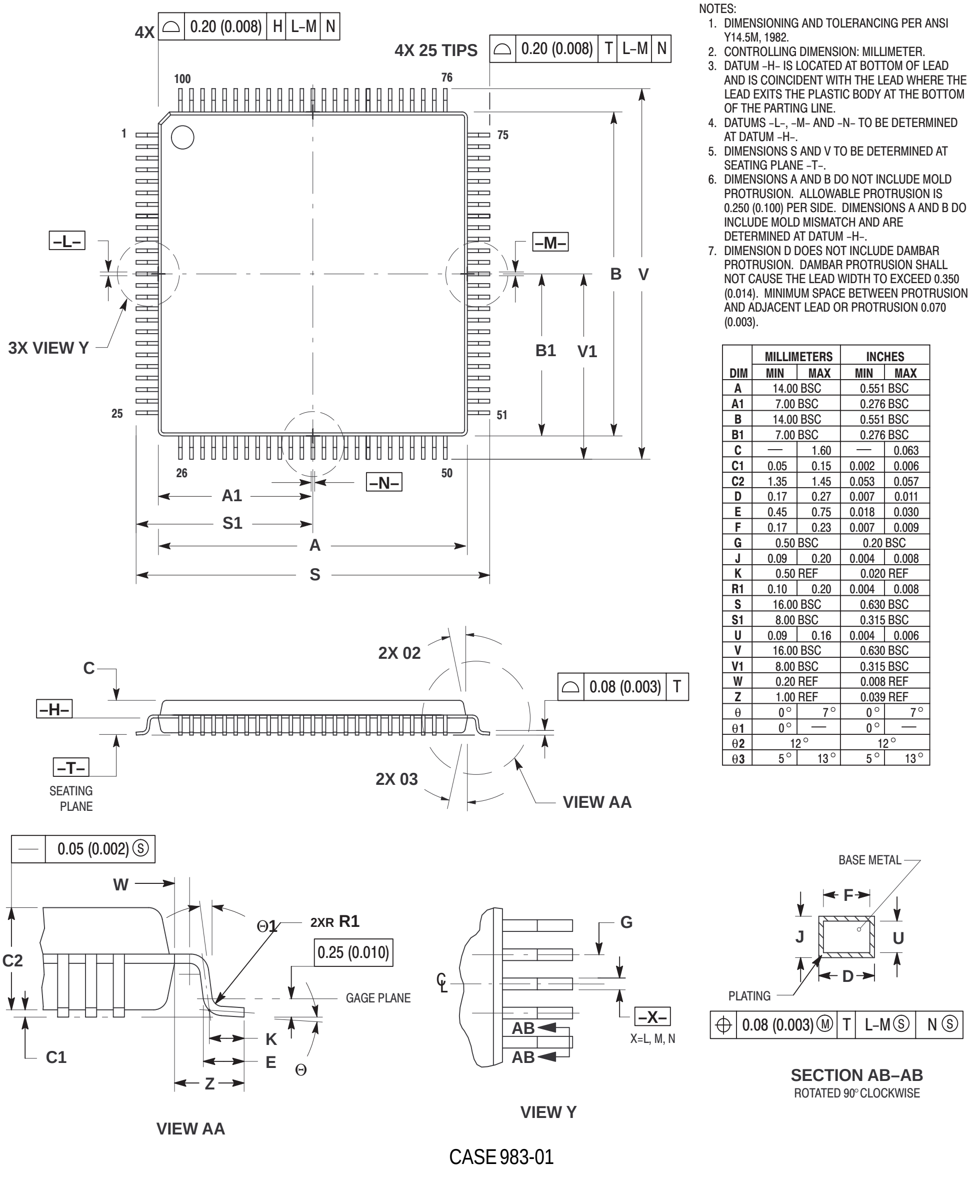


Figure 17-1 100-pin QFP pinout for the MC68HC05F32

17.2 100-pin QFP mechanical dimensions



NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM –H– IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS –L–, –M– AND –N– TO BE DETERMINED AT DATUM –H–.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE –T–.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 (0.100) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM –H–.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.350 (0.014). MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD OR PROTRUSION 0.070 (0.003).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	14.00 BSC		0.551 BSC	
A1	7.00 BSC		0.276 BSC	
B	14.00 BSC		0.551 BSC	
B1	7.00 BSC		0.276 BSC	
C	—	1.60	—	0.063
C1	0.05	0.15	0.002	0.006
C2	1.35	1.45	0.053	0.057
D	0.17	0.27	0.007	0.011
E	0.45	0.75	0.018	0.030
F	0.17	0.23	0.007	0.009
G	0.50 BSC		0.20 BSC	
J	0.09	0.20	0.004	0.008
K	0.50 REF		0.020 REF	
R1	0.10	0.20	0.004	0.008
S	16.00 BSC		0.630 BSC	
S1	8.00 BSC		0.315 BSC	
U	0.09	0.16	0.004	0.006
V	16.00 BSC		0.630 BSC	
V1	8.00 BSC		0.315 BSC	
W	0.20 REF		0.008 REF	
Z	1.00 REF		0.039 REF	
θ	0°	7°	0°	7°
θ1	0°	—	0°	—
θ2	12°		12°	
θ3	5°	13°	5°	13°

Figure 17-2 100-pin QFP mechanical dimensions

17.3 80-pin QFP pinout for the MC68HC05F32

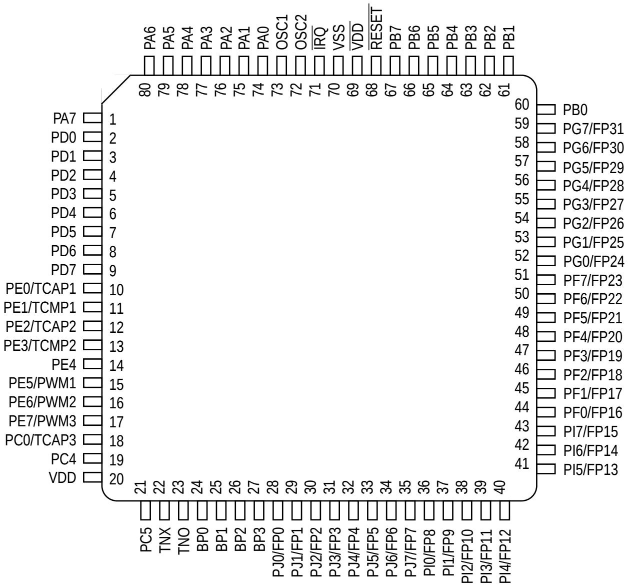
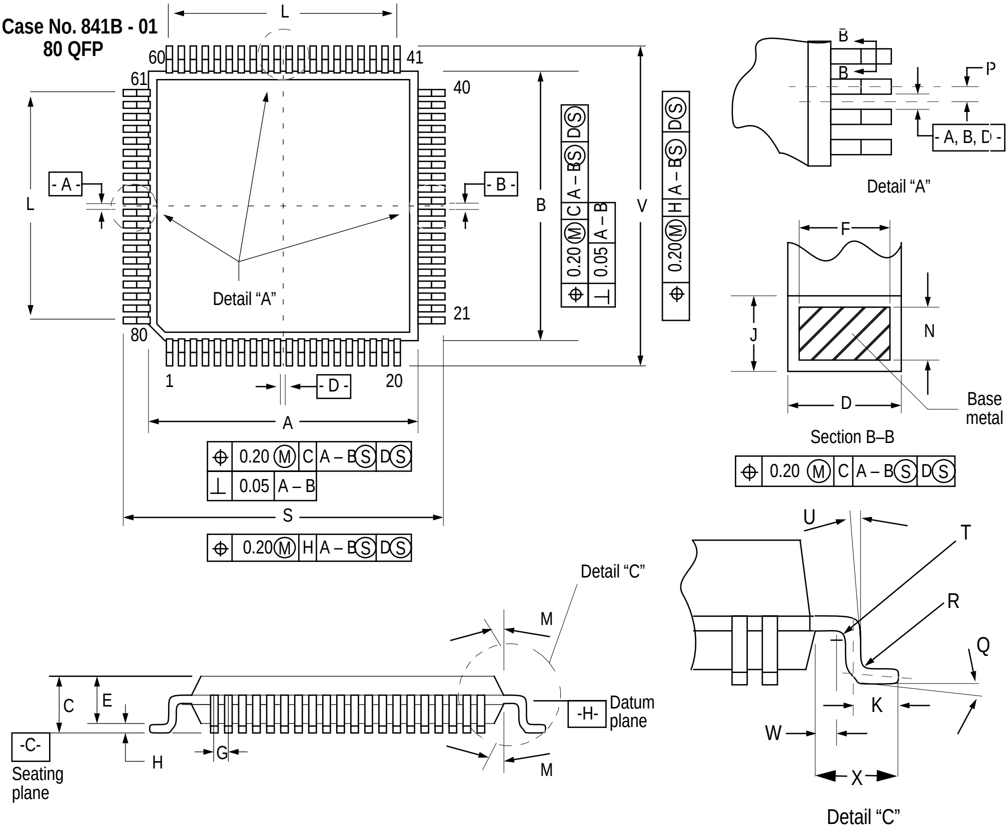


Figure 17-3 80-pin QFP pinout for the MC68HC05F32

Note: The 80-pin version is only a bond option. Pins PE4, PD7–PD0, PC4, PC5 are shared with module functions which cannot work on the 80-pin package. These modules and their corresponding pin functions should not be enabled.

17.4 80-pin QFP mechanical dimensions



Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	13.90	14.10	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 0.25 mm 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	13.90	14.10		N	0.130	0.170
C	2.15	2.45		Q	0°	7°
D	0.22	0.38		R	0.13	0.30
E	2.00	2.40		S	16.95	17.45
F	0.22	0.33		T	0.13	—
G	0.65 BSC			U	0°	—
H	—	0.250		V	16.95	17.45
J	0.130	0.230		W	0.35	0.45
K	0.65	0.95		X	1.6 REF	
L	12.35 REF					

Figure 17-4 80-pin QFP mechanical dimensions

18
ORDERING INFORMATION

This section describes the information needed to order the MC68HC05F32.

To initiate a ROM pattern for the MCU, it is necessary to first contact your local field service office, local sales person or Motorola representative. Please note that you will need to supply details such as: mask option selections; temperature range; oscillator frequency; package type; electrical test requirements; and device marking details so that an order can be processed, and a customer specific part number allocated. Refer to Table 18-1 for appropriate part numbers.

Table 18-1 MC order numbers

Device title	Package type	Temperature	Part number
MC68HC05F32	100-pin QFP	0 to 70 °C	MC68HC05F32PU
	80-pin QFP		MC68HC05F32FU
MC68HC705F32	100-pin QFP	0 to 70 °C	MC68HC705F32PU
	80-pin QFP		MC68HC705F32FU
MC68HC05F32	100-pin QFP	–40 to 85 °C	MC68HC05F32CPU
	80-pin QFP		MC68HC05F32CFU
MC68HC705F32	100-pin QFP	–40 to 85 °C	MC68HC705F32CPU
	80-pin QFP		MC68HC705F32CFU

18.1 EPROMs

For the MC68HC05F32, a 64K 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.

18.2 Verification media

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

18.3 ROM verification units(RVU)

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

A
MC68HC705F32

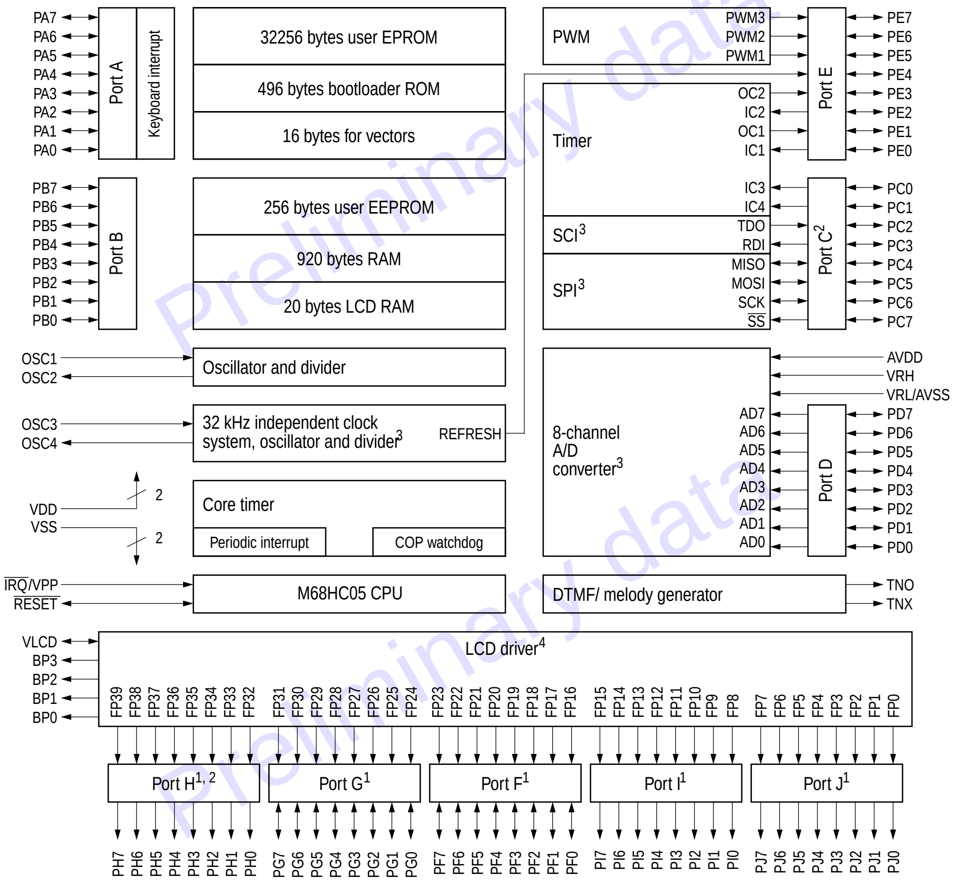
The MC68HC705F32 is a device very similar to the MC68HC05F32 but has 32256 bytes of user EPROM with 496 bytes of bootloader ROM. It does have the same amount of RAM, LCD RAM, EEPROM, I/O, and user vectors. It also has the same on-board peripherals as the MC68HC05F32.

There is also an 80-pin version of the MC68HC705F32, this has a reduced I/O count and reduced functionality. It has no 32 kHz clock system, SPI, SCI or A/D converter. The timer has three input captures (no TCAP4) and the LCD driver only has 32 frontplanes.

Note: The 80-pin version is only a bond option. Pins PE4, PD7–PD0, PC4, PC5 are shared with module functions which cannot work on the 80-pin package. These modules and their corresponding pin functions should not be enabled.

A.1 Features

- 32256 bytes of user EPROM plus 16 bytes of user vectors
- 496 bytes of bootloader ROM



1. When not being used to output the LCD frontplanes, port G and port F are input only, while port H, port I and port J are output only.
2. In the 80-pin package there is no port H and only pins PC0, PC4 and PC5 are available on port C.
3. These modules are not available in the 80-pin package.
4. In the 80-pin package there are only 32 frontplanes.

Figure A-1 MC68HC705F32 block diagram

A.2 Pin descriptions

A.2.1 $\overline{\text{IRQ}}$/VPP

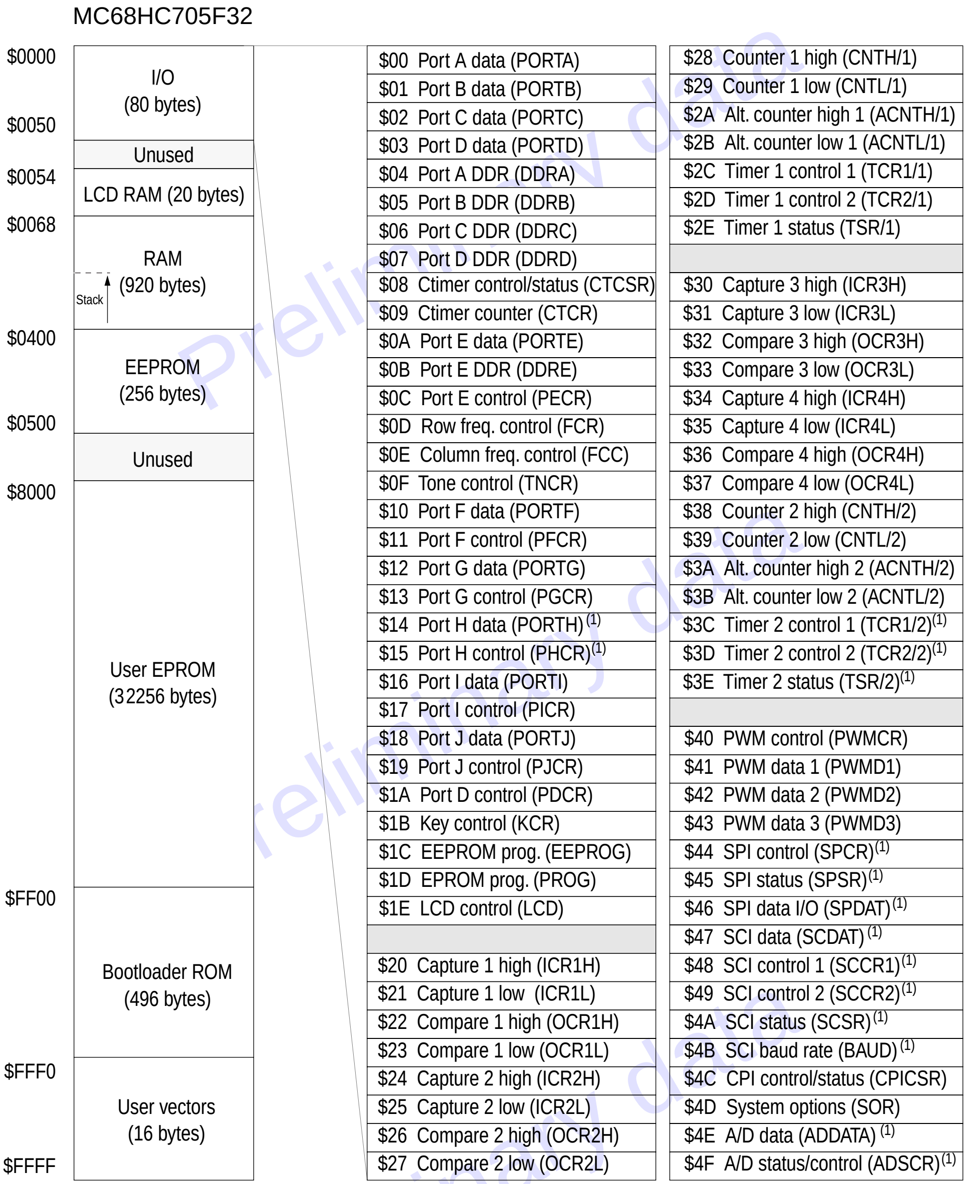
As for the MC68HC05F32, this is an input-only pin for external interrupt sources. It also serves as the EPROM programming voltage input pin (VPP) on the MC68HC705F32.

A.3 Memory and registers

The MC68HC705F32 has a 64K byte memory map consisting of registers (for I/O, control and status), user RAM, user ROM, EEPROM, bootloader ROM and reset and interrupt vectors as shown in Figure A-2.

A.3.1 Registers

All the I/O, control and status registers of the MC68HC705F32 are contained within the first 80 byte block of the memory map, as detailed in Table A-1.



$00 Port A data (PORTA)	$28 Counter 1 high (CNTH/1)
$01 Port B data (PORTB)	$29 Counter 1 low (CNTL/1)
$02 Port C data (PORTC)	$2A Alt. counter high 1 (ACNTH/1)
$03 Port D data (PORTD)	$2B Alt. counter low 1 (ACNTL/1)
$04 Port A DDR (DDRA)	$2C Timer 1 control 1 (TCR1/1)
$05 Port B DDR (DDRB)	$2D Timer 1 control 2 (TCR2/1)
$06 Port C DDR (DDRC)	$2E Timer 1 status (TSR/1)
$07 Port D DDR (DDRD)	
$08 Ctimer control/status (CTCSR)	$30 Capture 3 high (ICR3H)
$09 Ctimer counter (CTCR)	$31 Capture 3 low (ICR3L)
$0A Port E data (PORTE)	$32 Compare 3 high (OCR3H)
$0B Port E DDR (DDRE)	$33 Compare 3 low (OCR3L)
$0C Port E control (PECR)	$34 Capture 4 high (ICR4H)
$0D Row freq. control (FCR)	$35 Capture 4 low (ICR4L)
$0E Column freq. control (FCC)	$36 Compare 4 high (OCR4H)
$0F Tone control (TNCR)	$37 Compare 4 low (OCR4L)
$10 Port F data (PORTF)	$38 Counter 2 high (CNTH/2)
$11 Port F control (PFCR)	$39 Counter 2 low (CNTL/2)
$12 Port G data (PORTG)	$3A Alt. counter high 2 (ACNTH/2)
$13 Port G control (PGCR)	$3B Alt. counter low 2 (ACNTL/2)
$14 Port H data (PORTH) (1)	$3C Timer 2 control 1 (TCR1/2)(1)
$15 Port H control (PHCR)(1)	$3D Timer 2 control 2 (TCR2/2)(1)
$16 Port I data (PORTI)	$3E Timer 2 status (TSR/2)(1)
$17 Port I control (PICR)	
$18 Port J data (PORTJ)	$40 PWM control (PWMCR)
$19 Port J control (PJCR)	$41 PWM data 1 (PWMD1)
$1A Port D control (PDCR)	$42 PWM data 2 (PWMD2)
$1B Key control (KCR)	$43 PWM data 3 (PWMD3)
$1C EEPROM prog. (EEPROG)	$44 SPI control (SPCR)(1)
$1D EPROM prog. (PROG)	$45 SPI status (SPSR)(1)
$1E LCD control (LCD)	$46 SPI data I/O (SPDAT)(1)
	$47 SCI data (SCDAT) (1)
$20 Capture 1 high (ICR1H)	$48 SCI control 1 (SCCR1)(1)
$21 Capture 1 low (ICR1L)	$49 SCI control 2 (SCCR2)(1)
$22 Compare 1 high (OCR1H)	$4A SCI status (SCSR) (1)
$23 Compare 1 low (OCR1L)	$4B SCI baud rate (BAUD) (1)
$24 Capture 2 high (ICR2H)	$4C CPI control/status (CPICSR)
$25 Capture 2 low (ICR2L)	$4D System options (SOR)
$26 Compare 2 high (OCR2H)	$4E A/D data (ADDATA) (1)
$27 Compare 2 low (OCR2L)	$4F A/D status/control (ADSCR)(1)

(1) Not applicable to 80-pin package.

Figure A-2 Memory map of the MC68HC705F32

Table A-1 Register outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined
Key interrupt status (KISR)	$0000									0000 0000
Port B data (PORTB)	$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined
Port C data (PORTC)	$0002	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined
Port D data (PORTD)	$0003	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	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	TOF	RTIF	TOFE	RTIE	RTOF	RRTIF	RT1	RT0	0000 0011
Core timer counter (CTCR)	$0009									0000 0000
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined
Port E data direction (DDRE)	$000B									0000 0000
Port E control (PECR)	$000C						0		0	0000 0000
DTMF row freq. control (FCR)	$000D	0	0	0	FCR4	FCR3	FCR2	FCR1	FCR0	undefined
DTMF column freq. control (FCC)	$000E	0	0	0	FCC4	FCC3	FCC2	FCC1	FCC0	undefined
DTMF tone control (TNCR)	$000F	MS1	MS0	TGER	TGEC	TNOE	0	0	0	0000 0000
Port F data (PORTF)	$0010	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined
Port F control (PFCR)	$0011									0000 0000
Port G data (PORTG)	$0012	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	undefined
Port G control (PGCR)	$0013									0000 0000
Port H data (PORTH)	$0014	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	0000 0000
Port H control (PHCR)	$0015									0000 0000
Port I data (PORTI)	$0016	PI7	PI6	PI5	PI4	PI3	PI2	PI1	PI0	0000 0000
Port I control (PICR)	$0017									0000 0000
Port J data (PORTJ)	$0018	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	0000 0000
Port J control (PJCR)	$0019									0000 0000
Port D control (PDCR)	$001A									0000 0000
Key control (KCR)	$001B	KF	KIE	EDG5	EDG4	EDG3	EDG2	EDG1	EDG0	0000 0000
EEPROM prog. (EEPROG)	$001C	0	CPEN	0	ER1	ER0	LATCH	EERC	EEPGM	0000 0000
EPROM prog. (PROG)	$001D	0	0	0	TS1	TS0	ELATCH	0	EPGM	0000 0000
LCD control (LCD)	$001E	WTLCDO	FSEL1	FSEL0 I	NTVLCD	FDISP	MUX4	MUX3	EXTVON	0000 0000
Capture 1 high (ICR1H)	$0020	(bit 15)							(bit 8)	undefined
Capture 1 low (ICR1L)	$0021									undefined

Table A-1 Register outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Compare 1 high (OCR1H)	$0022	(bit 15)							(bit 8)	undefined
Compare 1 low (OCR1L)	$0023									undefined
Capture 2 high (ICR2H)	$0024	(bit 15)							(bit 8)	undefined
Capture 2 low (ICR2L)	$0025									undefined
Compare 2 high (OCR2H)	$0026	(bit 15)							(bit 8)	undefind
Compare 2 low (OCR2L)	$0027									undefined
Counter 1 high (CNTH/1)	$0028	(bit 15)							(bit 8)	1111 1111
Counter 1 low (CNTL/1)	$0029									1111 1100
Alternate counter 1 high (ACNTH/1)	$002A	(bit 15)							(bit 8)	1111 1111
Alternate counter 1 low (ACNTL/1)	$002B									1111 1100
Timer1 control 1 (TCR1/1)	$002C	ICI1E	ICI2E	OCI1E	TOIE	CO1E	IEDG1	IEDG2	OLVL1	0000 0uu0
Timer1 control 2 (TCR2/1)	$002D	0	0	OCI2E	0	CO2E	0	0	OLVL2	0000 0000
Timer1 status (TSR/1)	$002E	IC1F	IC2F	OC1F	TOF	TCAP1	TCAP2	OC2F	0	uuuu uuu0
Capture 3 high (ICR3H)	$0030	(bit 15)							(bit 8)	undefined
Capture 3 low (ICR3L)	$0031									undefined
Compare 3 high (OCR3H)	$0032	(bit 15)							(bit 8)	undefined
Compare 3 low (OCR3L)	$0033									undefined
Capture 4 high (ICR4H)	$0034	(bit 15)								undefined
Capture 4 low (ICR4L)	$0035									undefined
Compare 4 high (OCR4H)	$0036	(bit 15)								undefined
Compare 4 low (OCR4L)	$0037									undefined
Counter 1 high (CNTH/1)	$0038	(bit 15)							(bit 8)	1111 1111
Counter 1 low (CNTL/1)	$0039									1111 1100
Alternate counter 2 high (ACNTH/1)	$003A	(bit 15)							(bit 8)	1111 1111
Alternate counter 2 low (ACNTL/1)	$003B									1111 1100
Timer2 control 1 (TCR1/2)	$003C	ICI3E	ICI4E	OCI3E	TOIE	CO3E	IEDG3	IEDG4		0000 0uu0
Timer2 control 2 (TCR2/2)	$003D	0	0	OCI4E	0	CO4E	0	0		0000 0000
Timer2 status (TSR/2)	$003E	IC3F	IC4F	OC3F	TOF	TCAP3	TCAP4	OC4F	0	uuuu uuu0
PWM control (PWMCR)	$0040				POL3	POL2	POL1	RA1	RA0	0001 1100
PWM data 1 (PWMD1)	$0041									1000 0000
PWM data 2 (PWMD2)	$0042									1000 0000
PWM data 3 (PWMD3)	$0043									1000 0000
SPI control (SPCR)	$0044	SPIE	SPE	DOD	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu

Table A-1 Register outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI status (SPSR)	$0045	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000
SPI data I/O (SPDAT)	$0046									undefined
SCI data (SCDAT)	$0047									undefined
SCI control 1 (SCCR1)	$0048	R8	T8	0	M	WAKE	0	0	0	uu00 0000
SCI control 2 (SCCR2)	$0049	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI status (SCSR)	$004A	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	1100 0000
SCI baud rate (BAUD)	$004B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	0000 0uuu
CPI control status (CPICSR)	$004C	0	CPIF	0	CPIE	0	0	RFQ1	RFQ0	0000 0000
System options (SOR)	$004D	LVIF	LVIE	LVION	SC	IRQ	KEYMUX	KEYCLR	PUEN	0000 0000
A/D data (ADDATA)	$004E									undefined
A/D status/control (ADSCR)	$004F	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0	0000 0000

u = undefined

A.3.2 EPROM

The MC68HC705F32 has 32256 bytes of EPROM located from $8000 to $FDFF, plus 16 bytes of user vectors from $FFF0 to $FFFF. Up to 16 bytes of EPROM can be programmed simultaneously by correctly manipulating the bits in the EPROM programming register.

A.3.2.1 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

EPGM — EPROM program control

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

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

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.

ELATCH — EPROM latch control

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

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

ELATCH causes address and data buses to be latched when a write to EPROM is carried out. The EPROM cannot be read if ELATCH = 1. This bit should not be set unless a programming voltage is applied to the VPP pin.

A.3.2.2 EPROM programming operation

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

1) Apply the programming voltage V_{PP} to the $\overline{\text{IRQ}}$ 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–A4 do not change, i.e. all bytes are located within the same 16 byte address block, then multibyte programming is permitted. The multibyte programming facility allows up to 16 bytes of data to be written to the desired addresses after the ELATCH bit has been set.

A.4 Electrical specifications

This section gives the electrical specifications for the MC68HC705F32, the EPROM version of the MC68HC05F32. Contained in this section is the information specific to the MC68HC705F32 which differs from that detailed in Section 16.

A.4.1 EPROM characteristics

Table A-2 EPROM characteristics

Characteristic	Symbol	Value	Unit
EPROM programming voltage rate	V_{PP}	V_{SS} - 0.3 to +17 + 0.5	V
EPROM programming voltage	V_{PP}	typ. 17.0	V
EPROM programming time	t_{EPGM}	min. 4.0	ms

A.4.2 DC levels for low voltage reset and LVI

Table A-3 DC levels for low voltage reset and LVI

(TA = 0C to 60°C, unless otherwise stated)

Characteristic	Symbol	Min.	Typ.	Max.	Unit
Power-on reset voltage	V_{RON}	2.55	2.8	3.05	V
Power-off reset voltage	V_{ROFF}	2.45	2.7	2.95	V
Low voltage interrupt	V_{LVI}	2.75	3.0	3.25	V

A.5 Mechanical data

A.5.1 100-pin QFP pinout for the MC68HC705F32

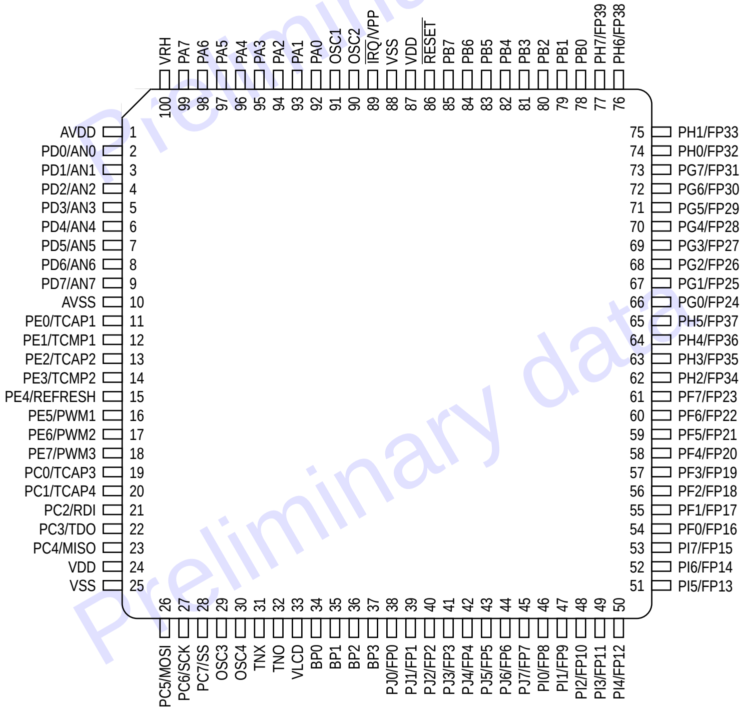


Figure 18-1 100-pin QFP pinout for the MC68HC705F32

For package dimensions, refer to Section 17.2.

A.5.2 80-pin QFP pinout for the MC68HC705F32

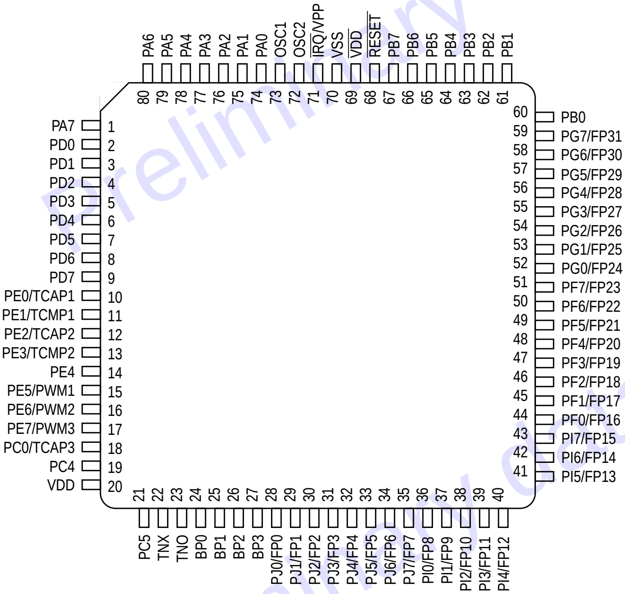


Figure 18-2 80-pin QFP pinout for the MC68HC705F32

Note: The 80-pin version is only a bond option. Pins PE4, PD7–PD0, PC4, PC5 are shared with module functions which cannot work on the 80-pin package. These modules and their corresponding pin functions should not be enabled.

For package dimensions, refer to Section 17.4.

THIS PAGE LEFT BLANK INTENTIONALLY

GLOSSARY

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

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

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

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

Bootstrap mode In this mode the device automatically loads its internal memory from an external source on reset and then allows this program to be executed.

Byte Eight bits.

CCR Condition codes register; an integral part of the CPU.

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

Clear '0' — the logic zero state; the opposite of 'set'.

CMOS Complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.

COP Computer operating properly. *aka* 'watchdog'. This circuit is used to detect device runaway and provide a means for restoring correct operation.

CPU Central processing unit.

D/A, DAC Digital-to-analog (converter).

EEPROM Electrically erasable programmable read only memory. *aka* 'EEROM'.

EPROM Erasable programmable read only memory. This type of memory requires exposure to ultra-violet wavelengths in order to erase previous data. *aka* 'PROM'.

ESD Electrostatic discharge.

Expanded mode In this mode the internal address and data bus lines are connected to external pins. This enables the device to be used in much more complex systems, where there is a need for external memory for example.

EVS Evaluation system. One of the range of platforms provided by Motorola for evaluation and emulation of their devices.

HCMOS High-density complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.

I/O Input/output; used to describe a bidirectional pin or function.

Input capture (IC) This is a function provided by the timing system, whereby an external event is ‘captured’ by storing the value of a counter at the instant the event is detected.

Interrupt This refers to an asynchronous external event and the handling of it by the MCU. The external event is detected by the MCU and causes a predetermined action to occur.

$\overline{\mathbf{IRQ}}$ Interrupt request. The overline indicates that this is an active-low signal format.

K byte A kilo-byte (of memory); 1024 bytes.

LCD Liquid crystal display.

LSB Least significant byte.

M68HC05 Motorola’s family of 8-bit MCUs.

MCU Microcontroller unit.

MI BUS Motorola interconnect bus. A single wire, medium speed serial communications protocol.

MSB Most significant byte.

Nibble Half a byte; four bits.

NRZ Non-return to zero.

Opcode The opcode is a byte which identifies the particular instruction and operating mode to the CPU. See also: prebyte, operand.

Operand The operand is a byte containing information the CPU needs to execute a particular instruction. There may be from 0 to 3 operands associated with an opcode. See also: opcode, prebyte.

Output compare (OC) This is a function provided by the timing system, whereby an external event is generated when an internal counter value matches a predefined value.

PLCC Plastic leaded chip carrier package.

PLL Phase-locked loop circuit. This provides a method of frequency multiplication, to enable the use of a low frequency crystal in a high frequency circuit.

Prebyte This byte is sometimes required to qualify an opcode, in order to fully specify a particular instruction. See also: opcode, operand.

Pull-down, pull-up These terms refer to resistors, sometimes internal to the device, which are permanently connected to either ground or V_{DD}.

PWM Pulse width modulation. This term is used to describe a technique where the width of the high and low periods of a waveform is varied, usually to enable a representation of an analog value.

QFP Quad flat pack package.

RAM Random access memory. Fast read and write, but contents are lost when the power is removed.

RFI Radio frequency interference.

RTI Real-time interrupt.

ROM Read-only memory. This type of memory is programmed during device manufacture and cannot subsequently be altered.

RS-232C A standard serial communications protocol.

SAR Successive approximation register.

SCI Serial communications interface.

Set ‘1’ — the logic one state; the opposite of ‘clear’.

Silicon glen An area in the central belt of Scotland, so called because of the concentration of semiconductor manufacturers and users found there.

Single chip mode In this mode the device functions as a self contained unit, requiring only I/O devices to complete a system.

SPI Serial peripheral interface.

Test mode This mode is intended for factory testing.

TTL Transistor-transistor logic.

UART Universal asynchronous receiver transmitter.

VCO Voltage controlled oscillator.

Watchdog *see* ‘COP’.

Wired-OR A means of connecting outputs together such that the resulting composite output state is the logical OR of the state of the individual outputs.

Word Two bytes; 16 bits.

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

THIS PAGE LEFT BLANK INTENTIONALLY

INDEX

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

– Cut along this line to remove –

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC05F32/D)

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

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

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

Comments: ______________________________

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

Selection of device for new application	□	Other □	Please specify: ______________
System design	□		______________________________
Training purposes	□		______________________________

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

Completely			Not at all	Comments: ______________
□	□	□	□	______________________________

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

Easy			Difficult	Comments: ______________
□	□	□	□	______________________________

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

		Too little detail				Too much detail
SECTION 1	INTRODUCTION	□	□	□	□	□
SECTION 2	MODES OF OPERATION AND PIN DESCRIPTIONS	□	□	□	□	□
SECTION 3	MEMORY AND REGISTERS	□	□	□	□	□
SECTION 4	PARALLEL INPUT/OUTPUT PORTS	□	□	□	□	□
SECTION 5	CORE TIMER	□	□	□	□	□
SECTION 6	16-BIT PROGRAMMABLE TIMER	□	□	□	□	□
SECTION 7	DTMF/MELODY GENERATOR	□	□	□	□	□
SECTION 8	LIQUID CRYSTAL DISPLAY DRIVER MODULE	□	□	□	□	□
SECTION 9	A/D CONVERTER	□	□	□	□	□
SECTION 10	SERIAL PERIPHERAL INTERFACE	□	□	□	□	□
SECTION 11	SERIAL COMMUNICATIONS INTERFACE	□	□	□	□	□
SECTION 12	PULSE WIDTH MODULATOR	□	□	□	□	□
SECTION 13	32 KHZ CLOCK SYSTEM	□	□	□	□	□
SECTION 14	RESETS AND INTERRUPTS	□	□	□	□	□
SECTION 15	CPU CORE AND INSTRUCTION SET	□	□	□	□	□
SECTION 16	ELECTRICAL SPECIFICATIONS	□	□	□	□	□
SECTION 17	MECHANICAL DATA	□	□	□	□	□
SECTION 18	ORDERING INFORMATION	□	□	□	□	□
SECTION 19	APPENDICES	□	□	□	□	□

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

____

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

____

8. How could we improve this document? ______________________________

____

____

9. How would you rate Motorola’s documentation?

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

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

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

12. How many years have you worked with microprocessors?

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

– Second fold back along this line –

By air mail
Par avion

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

NE PAS AFFRANCHIR

NO STAMP REQUIRED

REPONSE PAYEE GRANDE-BRETAGNE

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

F.A.O. Technical Publications Manager
(re: MC68HC05F32/D)

MOTOROLA
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

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

____

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

Name: ______________ Phone No: ______________

Position: ______________ FAX No: ______________

Department: ______________

Company: ______________

Address: ______________

____

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

– Finally, tuck this edge into opposite flap –

How to reach us:

Mfax™: RMFAX0@email.sps.mot.com – TOUCHTONE (602) 244-6609
INTERNET: http://www.mot.com/SPS/
USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405; Denver, Colorado 80217. 303-675-2140
JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki, 6F Seibu-Butsuryu-Center,
3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 81-3-3521-8315
HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road,
Tai Po, N.T., Hong Kong. 852-26629298

MOTOROLA