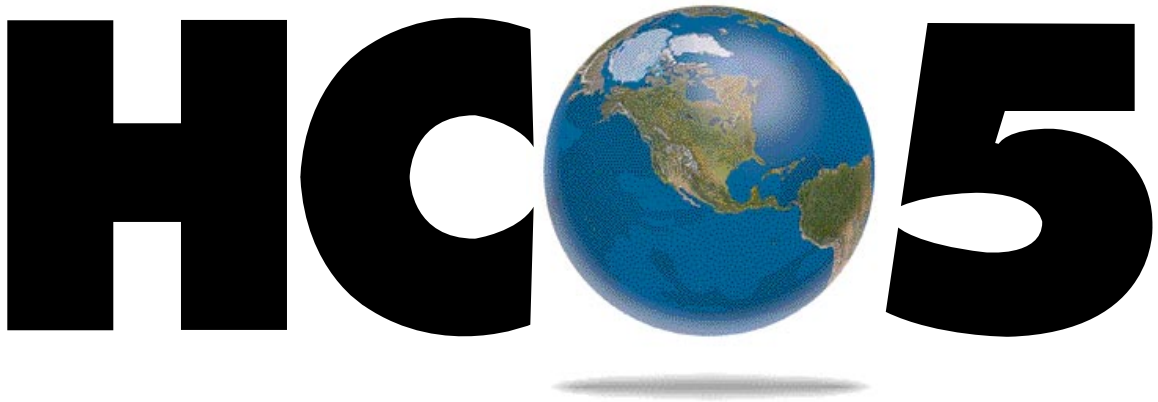
Order this document by
MC68HC05D9/D
Rev 3.0



MC68HC05D9
MC68HC05D24
MC68HC05D32
MC68HC705D32

HCMOS Microcontroller Unit

TECHNICAL DATA

List of Sections

List of Sections 3

Table of Contents 5

Introduction 9

Operating Modes and Pin Descriptions 13

Central Processing Unit (CPU) 23

Resets and Interrupts 43

Memory and Registers 53

Input/Output Ports 57

16-bit Programmable Timer 63

Serial Communications Interface (SCI) 79

Pulse Width Modulator (PWM) 99

Electrical Specifications 103

Mechanical Data and Ordering Information 111

Features Specific to the MC68HC05D24 117

Features Specific to the MC68HC05D32 121

List of Sections

Features Specific to the MC68HC705D32 125

Glossary 135

Index. 147

Literature Updates 153

Table of Contents

List of Sections

Table of Contents

Introduction	Contents	9
	General description	9
	Features	10
Operating Modes and Pin Descriptions	Contents	13
	Operating Modes	13
	Software-selectable options	16
	Pin Descriptions	17
Central Processing Unit (CPU)	Contents	23
	Introduction	24
	CPU Registers	24
	Arithmetic/Logic Unit (ALU)	27
	Instruction Set Overview	28
	Addressing Modes	28
	Instruction Types	31
	Instruction Set Summary	36
Resets and Interrupts	Contents	43
	Resets	43
	Interrupts	47
	Low power modes	50
Memory and Registers	Contents	53
	Introduction	53
	RAM	53
	ROM	54
	Registers	54

Table of Contents

Input/Output Ports
- Contents 57
- Introduction 57
- Input/output programming 57
- Ports A, B and C 59
- Port D 59
- Port registers 60
- Other port considerations 61

16-bit Programmable TImer
- Contents 63
- Introduction 63
- Counter 65
- Timer functions 67
- Timer during WAIT mode 72
- Timer during STOP mode 73
- Timer state diagrams 73

Serial Communications Interface (SCI)
- Contents 79
- Introduction 79
- Overview and features 80
- Functional description 81
- Data format 83
- Receiver wake-up operation 84
- Receive data (RDI) 85
- Start bit detection 86
- Transmit data (TDO) 89
- SCI registers 89

Pulse Width Modulator (PWM)
- Contents 99
- Introduction 99
- PWM counter 99
- PWM registers 100

Electrical Specifications
- Contents 103
- Introduction 103
- Maximum ratings 103
- Thermal characteristics and power considerations 103
- DC electrical characteristics 106
- Control timing 107

Mechanical Data and Ordering Information
- Contents . . . 111
- Mechanical Data . . . 112
- Ordering information . . . 115

Features Specific to the MC68HC05D24
- Contents . . . 117
- Introduction . . . 117
- Memory map and registers . . . 117
- Programming model . . . 117

Features Specific to the MC68HC05D32
- Contents . . . 121
- Introduction . . . 121
- Memory map and registers . . . 121
- Programming model . . . 121

Features Specific to the MC68HC705D32
- Contents . . . 125
- Introduction . . . 125
- Operating mode selection . . . 125
- Pin descriptions . . . 126
- Memory . . . 127
- MC68HC705D32 EPROM description . . . 130
- Bootloader mode . . . 132

Glossary

Index

Literature Updates
- Literature Distribution Centers . . . 153
- Customer Focus Center . . . 154
- Mfax . . . 154
- Freescale SPS World Marketing World Wide Web Server . . . 154
- Microcontroller Division’s Web Site . . . 154

Table of Contents

Introduction

Contents

General description 9
Features 10

General description

The MC68HC05D9, with 16K bytes of masked ROM, is a member of the Freescale M68HC05 family of advanced HCMOS 8-bit single chip microcomputers. Based around the industry standard M68HC05 CPU core and its familiar, efficient instruction set, this device includes five 6-bit pulse width modulation (PWM) channels, a serial communications interface (SCI), computer operating properly (COP) watchdog timer, high current output port capable of driving LEDs and a 16-bit timer with input capture and output compare.

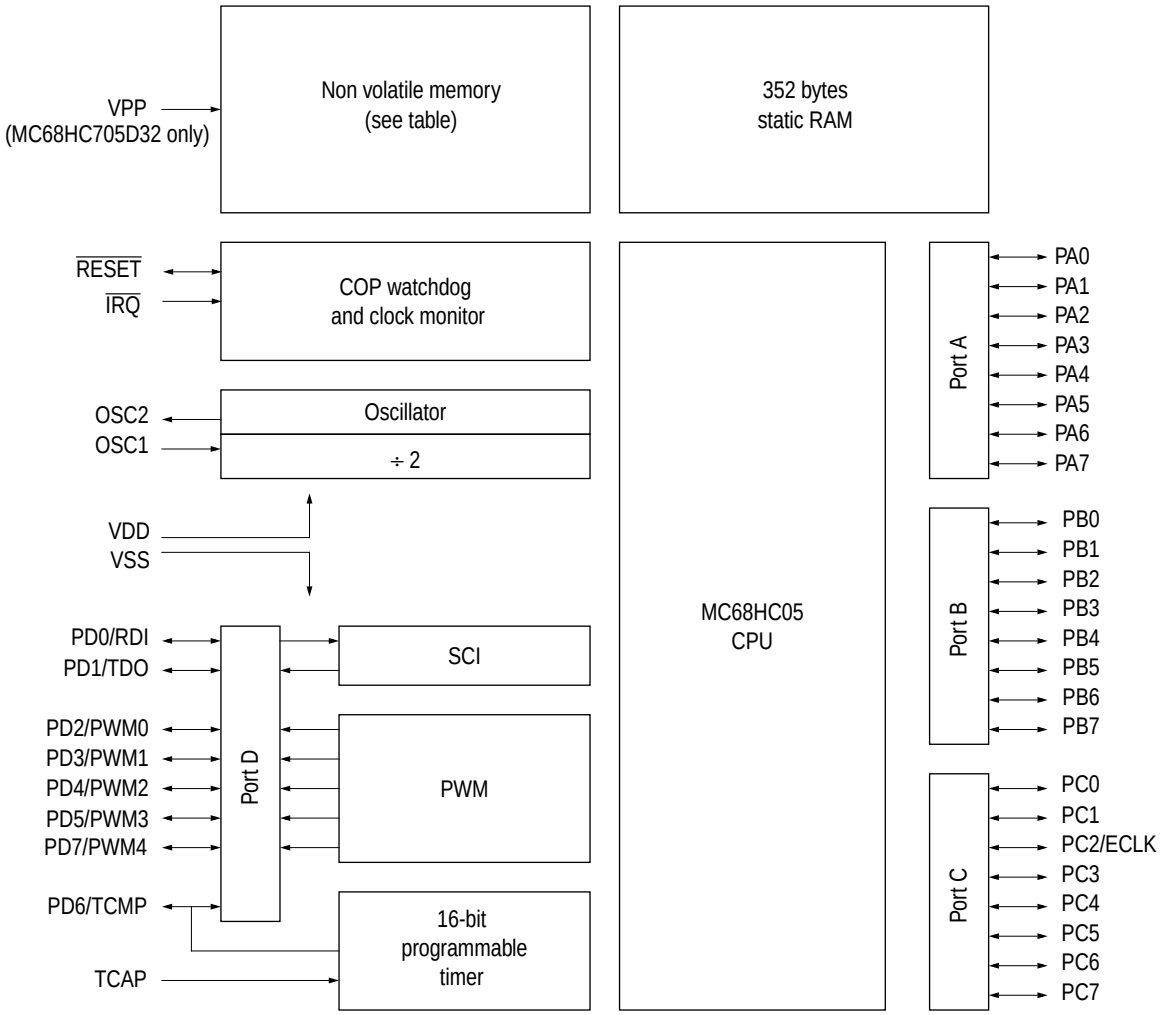
Devices similar to the MC68HC05D9 are descibed in a set of appendices at the back of this book.

Table 1. Data book appendices

Device	Appendix	Differences from MC68HC05D9
MC68HC05D24	A	24K bytes of ROM
MC68HC05D32	B	32K bytes of user
MC68HC705D32	C	32K bytes of EPROM

Features

- 15920 bytes of masked User ROM (plus 10 bytes for User vectors) on the MC68HC05D9
  - or 24112 bytes of masked User ROM (plus 10 bytes for vectors) on the MC68HC05D24
  - or 32304 bytes of masked User ROM (plus 10 bytes for vectors) on the MC68HC05D32
  - or 32304 bytes of User EPROM (plus 10 bytes for vectors) on the MC68HC705D32
- 239 bytes of self-check ROM (including self-check vectors) on the three mask-programmable ROM devices; 239 bytes of bootloader ROM (including bootloader vectors) on the MC68HC705D32
- 352 bytes of RAM
- Memory-mapped I/O
- 31 bidirectional I/O Lines
- Fully static operation
- On-chip oscillator with crystal/ceramic resonator
- 16-bit capture/compare timer sub-system
- Five 6-bit pulse width modulation channels operating at 30 kHz
- High current LED drive output port (port B)
- Asynchronous serial communications interface (SCI) system
- Power saving STOP, WAIT and data-retention modes
- Single 3.0 to 5.5 Volt supply (2 Volt data retention mode)
- Computer operating properly (COP) watchdog timer with clock monitor
- Software programmable external interrupt sensitivity



Device	Non volatile memory
MC68HC05D9	• 15920 bytes User ROM (plus 10 bytes for vectors) • 239 bytes self-check ROM (including vectors)
MC68HC05D24	• 24112 bytes User ROM (plus 10 bytes for vectors) • 239 bytes self-check ROM (including vectors)
MC68HC05D32	• 32304 bytes User ROM (plus 10 bytes for vectors) • 239 bytes self-check ROM (including vectors)
MC68HC705D32	• 32304 bytes User EROM (plus 10 bytes for vectors) • 239 bytes self-check ROM (including vectors)

Figure 1. Functional block diagram

Introduction

Operating Modes and Pin Descriptions

Contents

Operating Modes 13
- Single chip mode 14
- Self-check mode 14

Software-selectable options 16
- Option register 16

Pin Descriptions 17
- VDD and VSS 19
- VPP 19
- OSC1/OSC2 19
- RESET 21
- IRQ 21
- TCAP 21
- PA0–PA7 21
- PB0–PB7 21
- PC0–PC7 22
- PD0–PD7 22
- PD0/RDI 22
- PD1/TDO 22
- PD2–PD5, PD7/PWM0–PWM4 22
- PD6/TCMP 22

Operating Modes

The MCU has two modes of operation: single chip mode and self-check mode. The mode of operation is determined by the voltage level present on the IRQ pin when the device is brought out of reset (see **Table 2**).

Single chip mode

This is the normal operating mode of the MCU. In this mode the device functions as a self-contained microcomputer with all on-board peripherals available to the user, including the four 8-bit I/O ports.

CAUTION: *For the MC68HC705D32 all vectors are fetched from EPROM (locations $3FF6–$3FFF or $7FF6-$7FFF) in single chip mode; therefore, the EPROM must be programmed (via the bootloader mode) before the device is powered up in single chip mode.*

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

Table 2. Operating mode entry conditions

IRQ(1)	RESET	TCAP	Mode
V_{SS} to V_{DD}		Don't care	Single chip
2 V_{DD}		V_{DD}	Self-check

1. The voltage level on the IRQ pin should be maintained for at least 7x t_{CYC} after the rising edge of the reset signal to guarantee proper mode selection.

Self-check mode

The MCU contains, in masked ROM at locations $3F00 to $3FDE (or $7F00 to $7FDE), a program that checks the integrity of the device with a minimum of support hardware (see **Figure 2**). To enter self-check mode, a voltage equal to 2x V_{DD} must be present on the IRQ pin when the device is brought out of reset.

NOTE: *The TCAP pin must be tied to V_{DD} to ensure correct selection of the self-check mode. Failure to do so could result in unpredictable operation.*

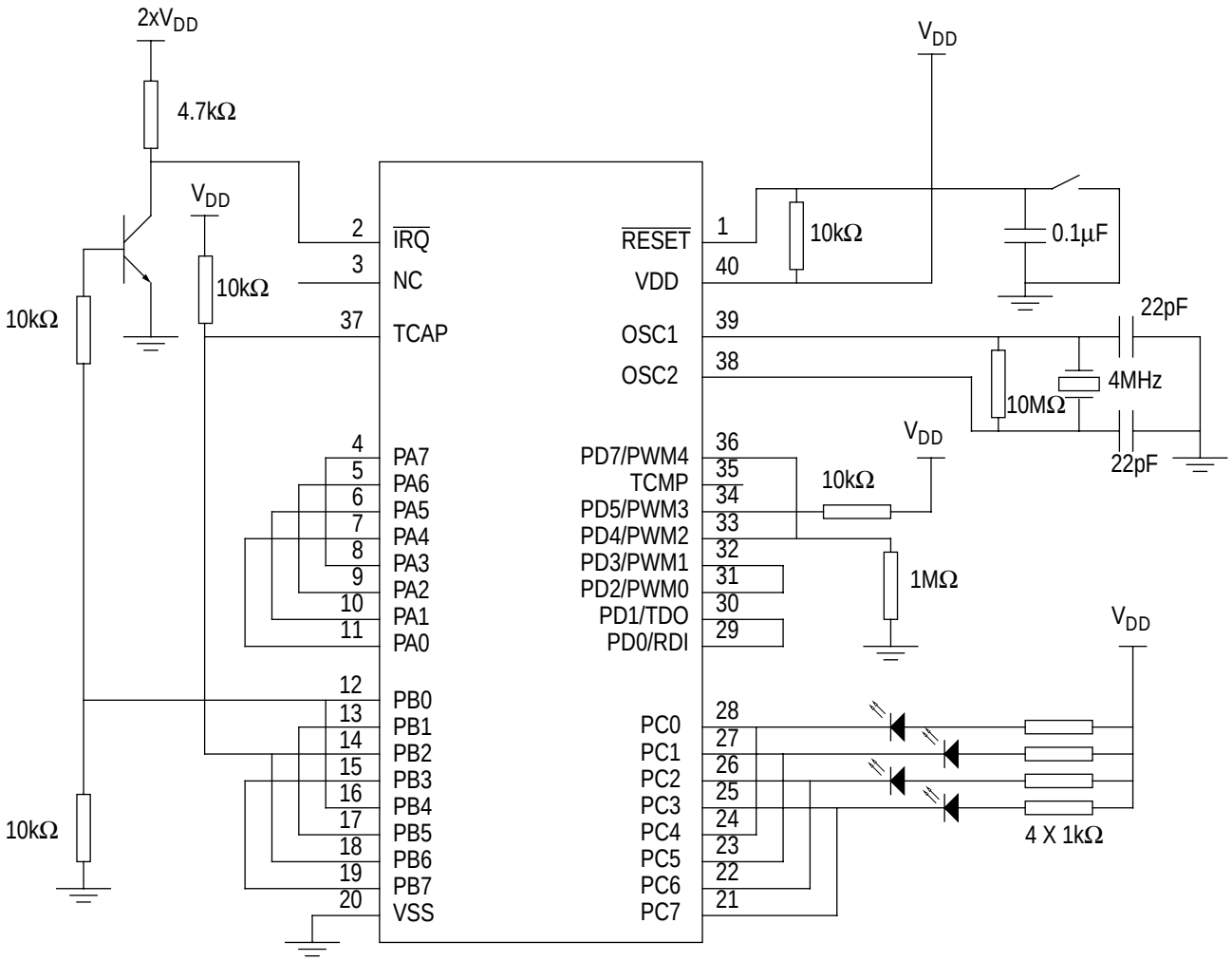


Figure 2. Self-check circuit diagram (for 40-pin DIL package)

The self-check program indicates the results of its tests by outputting a 4-bit code on the lower four bits of port C (PC0–3). The fault codes are detailed in **Table 3**.

Table 3. Self-check mode fault indication

PC3	PC2	PC1	PC0	Result
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad timer

Table 3. Self-check mode fault indication

1	1	0	0	Bad SCI
1	1	0	1	Bad ROM
1	1	1	0	Bad PWM
1	1	1	1	Bad interrupts or IRQ
Flashing				Good device
All other				Bad device

‘0’ denotes LED on; ‘1’ denotes LED off

Software-selectable options

Option register

The option register (OPTION) is a user-writable register located at $3FDF on the MC68HC05D9 ($7FDF on the MC68HC05D24 and MC68HC05D32). It allows the user to configure the memory map and the external interrupt IRQ.

Address: $3FDF

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RAM0	RAM1	0	0	0		IRQ	0
Reset:	0	0	0	0	0	u	1	0

Figure 3. Options register (OPTION)

RAM0

This bit maps 48 bytes of either RAM or ROM into the memory map from $0020 to $004F. This bit is readable and writable at all times, allowing the user software to switch back and forth between RAM and ROM when necessary. Reset clears this bit.

1 = Maps 48 bytes of RAM into the memory map starting at address $0020.

0 = Maps 48 bytes of ROM/EPROM into the memory map starting at address $0020.

RAM1

This bit maps 128 bytes of either RAM or ROM/EPROM into the memory map from $0100 to $017F. This bit is readable and writable at all times, allowing the user software to switch back and forth between RAM and ROM/EPROM when necessary. Reset clears this bit.

1 = Maps 128 bytes of RAM into the memory map starting at address $0100.
0 = Maps 128 bytes of ROM/EPROM into the memory map starting at address $0100.

IRQ

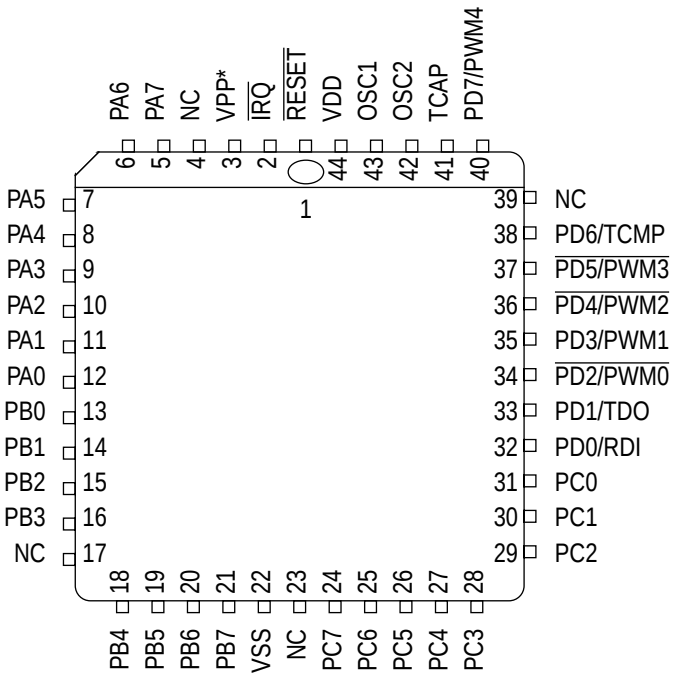
This bit selects the external interrupt IRQ sensitivity. This bit is not readable and can only be written once after reset. Reset sets this bit.

1 = Edge-and-level sensitive interrupt selected.
0 = Edge sensitive only option is selected.

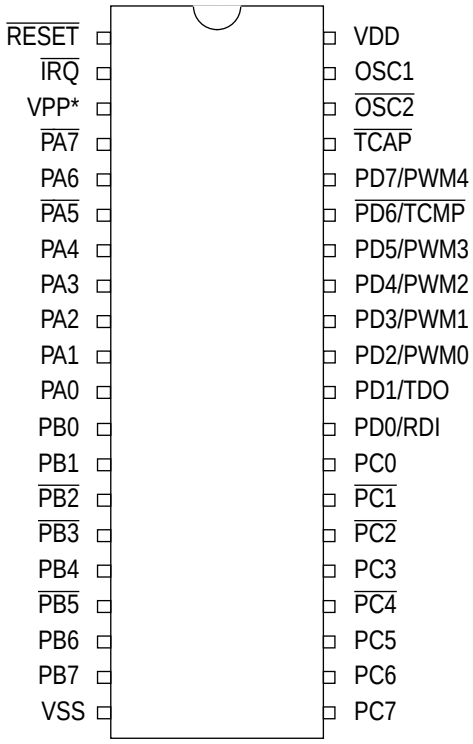
Pin Descriptions

Figure 4 shows the pin-out for this family of devices for the 40-pin plastic dual-in-line package (PDIP) and the 44-pin plastic leadless chip-carrier (PLCC).

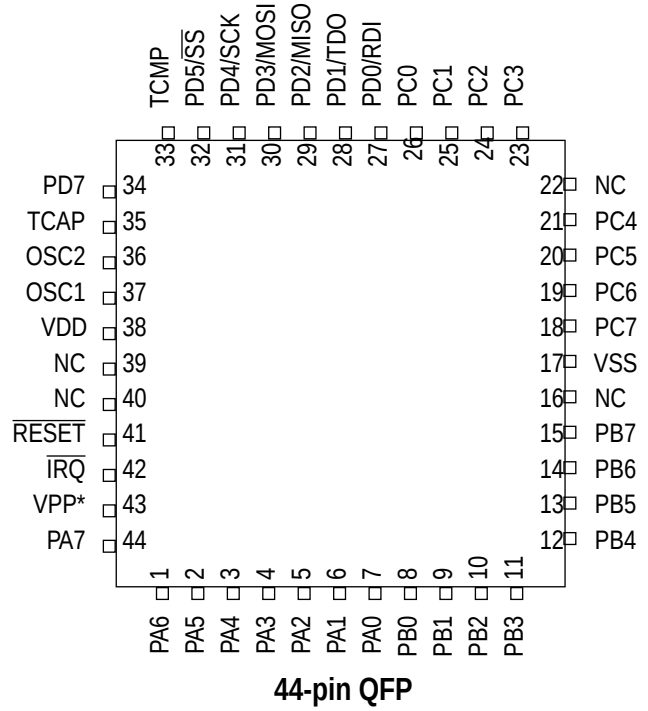
Operating Modes and Pin Descriptions



44-pin Plastic Leadless Chip Carrier



40-pin Plastic Dual-in-line Package



44-pin QFP

Note: * On MC68HC705D32 and MC68HC705D32 only. This pin should be connected to V_{DD} on the masked ROM devices in this family.

Figure 4. Pin assignments for MC68HC05D9 packages

VDD and VSS

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

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

VPP

This pin is used to supply programming power to the EPROM array on the MC68HC705D32. On the MC68HC05D9 (and the MC68HC05D24 and MC68HC05D32) this pin should be connected to V_{DD}. On the MC68HC705D32, the voltage on this pin should never be allowed to go below V_{DD}.

OSC1/OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, ceramic resonator or external clock signal connected to these pins provides the oscillator clock. The oscillator frequency is divided by 2 to provide the internal bus frequency.

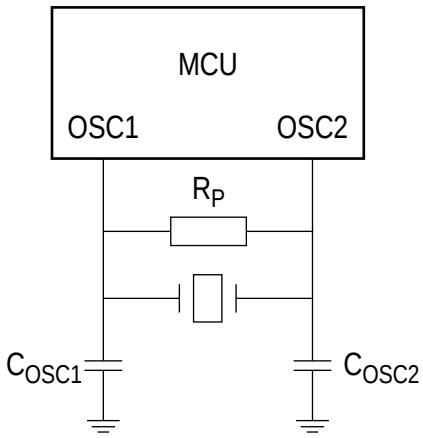
Crystal

The circuit shown in **Figure 5**(a) is recommended when using a crystal. 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 **Control timing**). Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimise output distortion and start-up stabilization time.

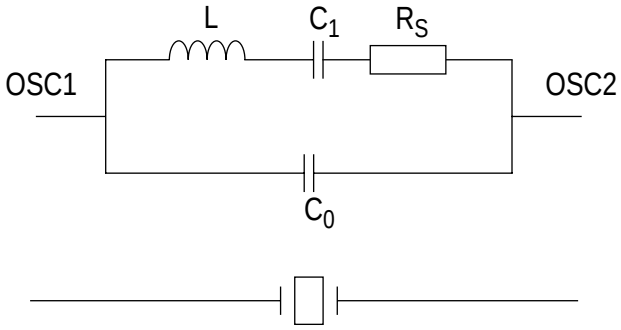
Ceramic resonator

A ceramic resonator may be used instead of a crystal in cost-sensitive applications. The circuit in **Figure 5**(a) is recommended when using a ceramic resonator. **Figure 5**(d) lists the recommended capacitance and feedback resistance values. The manufacturer of the particular ceramic

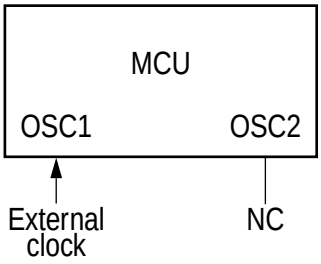
resonator being considered should be consulted for specific information. .



(a) Crystal/ceramic resonator oscillator connections



(b) Crystal equivalent circuit



(c) External clock source connections

Crystal

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

Ceramic resonator

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

(d) Crystal and ceramic resonator parameters

Figure 5. Oscillator connections

External clock

An external clock should be applied to the OSC1 input with the OSC2 pin not connected, as shown in **Figure 5**(c). The t_{OXOV} specification 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}.

RESET

This pin is used to apply an active low reset signal to 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). In this case, the time constant must be chosen high enough (minimum 100 ms) to allow the oscillator circuit to stabilise. This input has an internal Schmitt trigger to improve noise immunity. When a reset condition occurs internally, i.e. from the COP watchdog or clock monitor circuitry, this pin provides an active-low open drain output signal which may be used to reset external hardware.

IRQ

IRQ is an input pin for external interrupt sources. The interrupt type (edge or edge-and-level sensitive) can be selected via the option register.

TCAP

An external signal can be applied to this input pin to trigger an input capture event in the 16-bit timer. Input capture can be programmed to occur on either the rising edge or the falling edge of the signal applied to TCAP (see **16-bit Programmable TImer**).

PA0–PA7

Port A comprises eight bidirectional pins (PA0 to PA7). The direction and state of each pin is software programmable. All pins are configured as inputs during power-on or reset.

PB0–PB7

Port B comprises eight bidirectional pins (PB0 to PB7). The direction and state of each pin is software programmable, and each pin can drive one LED load. All pins are configured as inputs during power-on or reset.

PC0–PC7 Port C comprises eight bidirectional pins (PC0 to PC7). The direction and state of each pin is software programmable. All pins are configured as inputs during power-on or reset.

PD0–PD7 Port D comprises seven bidirectional pins (PD0 to PD5, PD7), with each pin supporting one of the on-chip hardware functions. PD6 is dedicated to the TCMP pin and is always an output, therefore a read of bit 6 of the port D data register will always return the value on the TCMP pin.

PD0/RDI When the SCI is enabled, this pin is configured as the high-impedance receive data input pin and the SCI becomes active. When the SCI is disabled, this pin is configured as a normal I/O port pin.

PD1/TDO When the SCI is enabled, this pin is configured as the transmit data output pin and the SCI becomes active. When the SCI is disabled, this pin is configured as a normal I/O port pin.

PD2–PD5, PD7/PWM0–PWM4 When the associated PWM enable bits in the PWM mode register are set, these pins are configured to output the PWM signals. When the PWM channels are disabled, these pins are configured as normal I/O pins.

PD6/TCMP As an output, this pin is always configured to support the output compare function in the 16-bit timer. Consequently, there is no corresponding data direction register bit associated with this pin. Reading the associated bit in the port D data register always returns the instantaneous value present on the pin, whether this is caused by the output compare function or by an external signal applied to the pin.

Central Processing Unit (CPU)

Contents

Introduction 24
CPU Registers 24
- Accumulator 24
- Index Register 25
- Stack Pointer 25
- Program Counter 26
- Condition Code Register 26

Arithmetic/Logic Unit (ALU) 27
Instruction Set Overview 28
Addressing Modes 28
- Inherent 28
- Immediate 29
- Direct 29
- Extended 29
- Indexed, No Offset 29
- Indexed, 8-Bit Offset 29
- Indexed,16-Bit Offset 30
- Relative 30

Instruction Types 31
- Register/Memory Instructions 31
- Read-Modify-Write Instructions 32
- Jump/Branch Instructions 33
- Bit Manipulation Instructions 35
- Control Instructions 35

Instruction Set Summary 36

Introduction

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

CPU Registers

Figure 1 shows the five CPU registers. CPU registers are not part of the memory map.

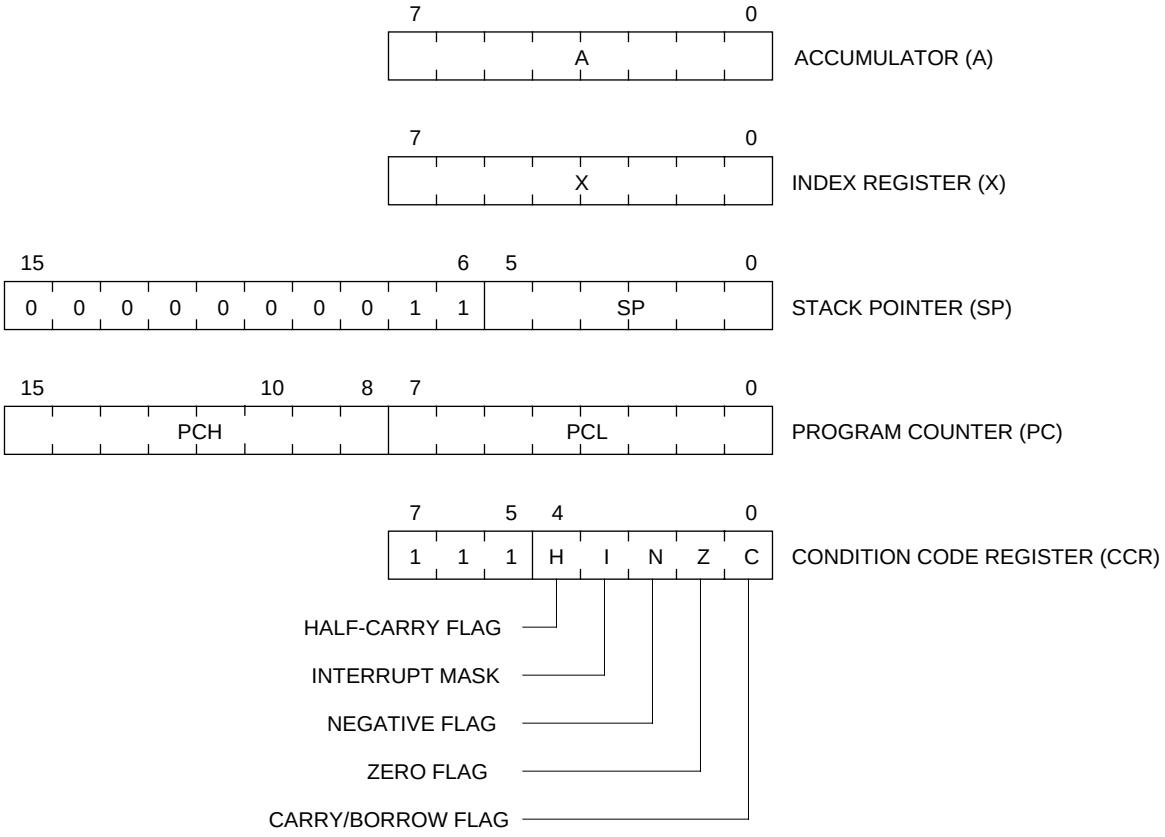


Figure 6. Programming Model

Accumulator

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

	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	Unaffected by reset							

Figure 7. Accumulator

Index Register

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

	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	Unaffected by reset							

Figure 8. Index Register

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

Stack Pointer

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

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	0	0	0	1	1						
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Figure 9. Stack Pointer

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

Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched. The two most significant bits of the program counter are ignored internally.

Normally, the address in the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

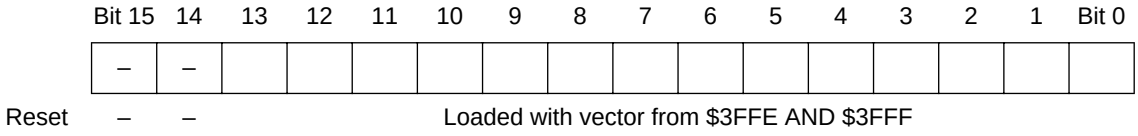


Figure 10. Program Counter

Condition Code Register

The condition code register is an 8-bit register whose three most significant bits are permanently fixed at 111. The condition code register contains the interrupt mask and four flags that indicate the results of the instruction just executed. The following paragraphs describe the functions of the condition code register.

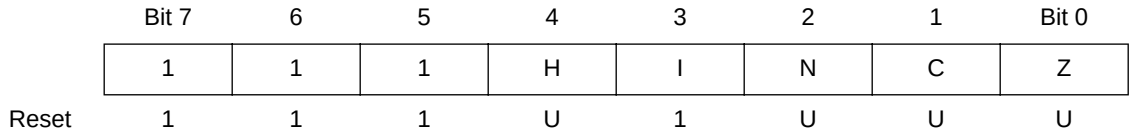


Figure 11. Condition Code Register

Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between bits 3 and 4 of the accumulator during an ADD or ADC operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations.

Interrupt Mask

Setting the interrupt mask disables interrupts. If an interrupt request occurs while the interrupt mask is logic zero, the CPU saves the CPU registers on the stack, sets the interrupt mask, and then fetches the

interrupt vector. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the CPU processes the latched interrupt as soon as the interrupt mask is cleared again.

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

Negative Flag

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

Zero Flag

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

Carry/Borrow Flag

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

Arithmetic/Logic Unit (ALU)

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

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

Instruction Set Overview

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

Addressing Modes

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

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

Inherent

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

Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

Direct

Direct instructions can access any of the first 256 memory locations with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address.

Extended

Extended instructions use three bytes and can access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Freescale assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the effective address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The

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

Indexed,16-Bit Offset

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

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

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

Relative

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

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

Instruction Types

The MCU instructions fall into the following five categories:

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

Register/Memory Instructions

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

Table 4. Register/Memory Instructions

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

Read-Modify-Write Instructions

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

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

Table 5. Read-Modify-Write Instructions

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

1. Unlike other read-modify-write instructions, BCLR and BSET use only direct addressing.
2. TST is an exception to the read-modify-write sequence because it does not write a replacement value.

Jump/Branch Instructions

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

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

Table 6. Jump and Branch Instructions

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

Bit Manipulation Instructions

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

Table 7. Bit Manipulation Instructions

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

Control Instructions

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

Table 8. Control Instructions

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

Instruction Set Summary

Table 9. Instruction Set Summary

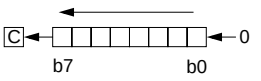
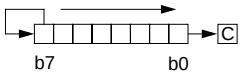
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr*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

Table 9. Instruction Set Summary (Continued)

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

Table 9. Instruction Set Summary (Continued)

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

Table 9. Instruction Set Summary (Continued)

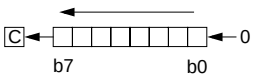
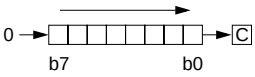
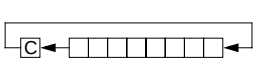
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2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

Table 9. Instruction Set Summary (Continued)

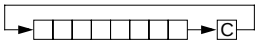
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕◊	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI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

Table 9. Instruction Set Summary (Continued)

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

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

Table 10. Opcode Map

MSB / LSB	Bit Manipulation		Branch	Read-Modify-Write						Control		Register/Memory						MSB / LSB
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
0	5 BRSET0 DIR 3	5 BSET0 DIR 2	3 BRA REL 2	5 NEG DIR 1	3 NEGA INH 1	3 NEGX INH 2	6 NEG IX1 1	5 NEG IX 1	9 RTI INH 1		2 SUB IMM 2	3 SUB DIR 3	4 SUB EXT 3	5 SUB IX2 2	4 SUB IX1 1	3 SUB IX 1	**0**
1	5 BRCLR0 DIR 3	5 BCLR0 DIR 2	3 BRN REL 2						6 RTS INH 1		2 CMP IMM 2	3 CMP DIR 3	4 CMP EXT 3	5 CMP IX2 2	4 CMP IX1 1	3 CMP IX 1	**1**
2	5 BRSET1 DIR 3	5 BSET1 DIR 2	3 BHI REL 2		11 MUL INH 1						2 SBC IMM 2	3 SBC DIR 3	4 SBC EXT 3	5 SBC IX2 2	4 SBC IX1 1	3 SBC IX 1	**2**
3	5 BRCLR1 DIR 3	5 BCLR1 DIR 2	3 BLS REL 2	5 COM DIR 1	3 COMA INH 1	3 COMX INH 2	6 COM IX1 1	5 COM IX 1	10 SWI INH		2 CPX IMM 2	3 CPX DIR 3	4 CPX EXT 3	5 CPX IX2 2	4 CPX IX1 1	3 CPX IX 1	**3**
4	5 BRSET2 DIR 3	5 BSET2 DIR 2	3 BCC REL 2	5 LSR DIR 1	3 LSRA INH 1	3 LSRX INH 2	6 LSR IX1 1	5 LSR IX 1			2 AND IMM 2	3 AND DIR 3	4 AND EXT 3	5 AND IX2 2	4 AND IX1 1	3 AND IX 1	**4**
5	5 BRCLR2 DIR 3	5 BCLR2 DIR 2	3 BCS/BLO REL 2								2 BIT IMM 2	3 BIT DIR 3	4 BIT EXT 3	5 BIT IX2 2	4 BIT IX1 1	3 BIT IX 1	**5**
6	5 BRSET3 DIR 3	5 BSET3 DIR 2	3 BNE REL 2	5 ROR DIR 1	3 RORA INH 1	3 RORX INH 2	6 ROR IX1 1	5 ROR IX 1			2 LDA IMM 2	3 LDA DIR 3	4 LDA EXT 3	5 LDA IX2 2	4 LDA IX1 1	3 LDA IX 1	**6**
7	5 BRCLR3 DIR 3	5 BCLR3 DIR 2	3 BEQ REL 2	5 ASR DIR 1	3 ASRA INH 1	3 ASRX INH 2	6 ASR IX1 1	5 ASR IX 1	2 TAX INH 1			3 STA DIR 3	4 STA EXT 3	5 STA IX2 2	4 STA IX1 1	3 STA IX 1	**7**
8	5 BRSET4 DIR 3	5 BSET4 DIR 2	3 BHCC REL 2	5 ASL/LSL DIR 1	3 ASLA/LSLA INH 1	3 ASLX/LSLX INH 2	6 ASL/LSL IX1 1	5 ASL/LSL IX 1	2 CLC INH 1		2 EOR IMM 2	3 EOR DIR 3	4 EOR EXT 3	5 EOR IX2 2	4 EOR IX1 1	3 EOR IX 1	**8**
9	5 BRCLR4 DIR 3	5 BCLR4 DIR 2	3 BHCS REL 2	5 ROL DIR 1	3 ROLA INH 1	3 ROLX INH 2	6 ROL IX1 1	5 ROL IX 1	2 SEC INH 1		2 ADC IMM 2	3 ADC DIR 3	4 ADC EXT 3	5 ADC IX2 2	4 ADC IX1 1	3 ADC IX 1	**9**
A	5 BRSET5 DIR 3	5 BSET5 DIR 2	3 BPL REL 2	5 DEC DIR 1	3 DECA INH 1	3 DECX INH 2	6 DEC IX1 1	5 DEC IX 1	2 CLI INH 1		2 ORA IMM 2	3 ORA DIR 3	4 ORA EXT 3	5 ORA IX2 2	4 ORA IX1 1	3 ORA IX 1	**A**
B	5 BRCLR5 DIR 3	5 BCLR5 DIR 2	3 BMI REL						2 SEI INH 1		2 ADD IMM 2	3 ADD DIR 3	4 ADD EXT 3	5 ADD IX2 2	4 ADD IX1 1	3 ADD IX 1	**B**
C	5 BRSET6 DIR 3	5 BSET6 DIR 2	3 BMC REL 2	5 INC DIR 1	3 INCA INH 1	3 INCX INH 2	6 INC IX1 1	5 INC IX 1	2 RSP INH 1		2 JMP DIR 2	3 JMP DIR 3	4 JMP EXT 3	5 JMP IX2 2	4 JMP IX1 1	3 JMP IX 1	**C**
D	5 BRCLR6 DIR 3	5 BCLR6 DIR 2	3 BMS REL 2	5 TST DIR 1	3 TSTA INH 1	3 TSTX INH 2	5 TST IX1 1	4 TST IX 1	2 NOP INH 1		6 BSR REL 2	5 JSR DIR 3	6 JSR EXT 3	7 JSR IX2 2	6 JSR IX1 1	5 JSR IX 1	**D**
E	5 BRSET7 DIR 3	5 BSET7 DIR 2	3 BIL REL								2 LDX IMM 2	3 LDX DIR 3	4 LDX EXT 3	5 LDX IX2 2	4 LDX IX1 1	3 LDX IX 1	**E**
F	5 BRCLR7 DIR 3	5 BCLR7 DIR 2	3 BIH REL 2	5 CLR DIR 1	3 CLRA INH 1	3 CLRX INH 2	6 CLR IX1 1	5 CLR IX 1	2 WAIT INH 1	2 TXA INH 1		4 STX DIR 3	5 STX EXT 3	6 STX IX2 2	5 STX IX1 1	4 STX IX 1	**F**

INH = InherentREL = Relative
IMM = ImmediateIX = Indexed, No Offset
DIR = DirectIX1 = Indexed, 8-Bit Offset
EXT = ExtendedIX2 = Indexed, 16-Bit Offset

LSB of Opcode in Hexadecimal

MSB of Opcode in Hexadecimal

MSB / LSB	0
0	5 BRSET0 DIR 3

Number of Cycles
Opcode Mnemonic
Number of Bytes/Addressing Mode

Resets and Interrupts

Contents

Resets 43
Power-on reset 43
RESET pin 43
Computer operating properly (COP) reset 44
Clock monitor reset 45
Interrupts 47
Non-maskable software interrupt (SWI) 48
Maskable hardware interrupts 48
Hardware controlled interrupt sequence 49
Low power modes 50
STOP 50
WAIT 50
Data retention mode 50

Resets

The MCU can be reset in four ways: by the initial power-on reset function, by an active low input to the RESET pin, by a COP watchdog timer reset and by an internal clock monitor reset. See **Figure 12**.

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 RESET pin is low at the end of this delay then the processor remains in the reset state until RESET goes high. The user must ensure that the voltage on VDD has risen to a point where the MCU can operate properly by the time t_{PORL} has elapsed.

If there is doubt, the external RESET pin should remain low until the voltage on VDD has reached the specified minimum operating voltage. This may be accomplished by connecting an external RC-circuit to this pin to generate a power-on reset (POR). In this case, the time constant must be great enough (at least 100 ms) to allow the oscillator circuit to stabilise.

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

RESET pin

When the oscillator is running in a stable state, the MCU is reset when a logic zero is applied to the RESET input for a minimum period of 8 machine cycles (t_{CYC}). This pin contains an internal Schmitt trigger as part of its input to improve noise immunity.

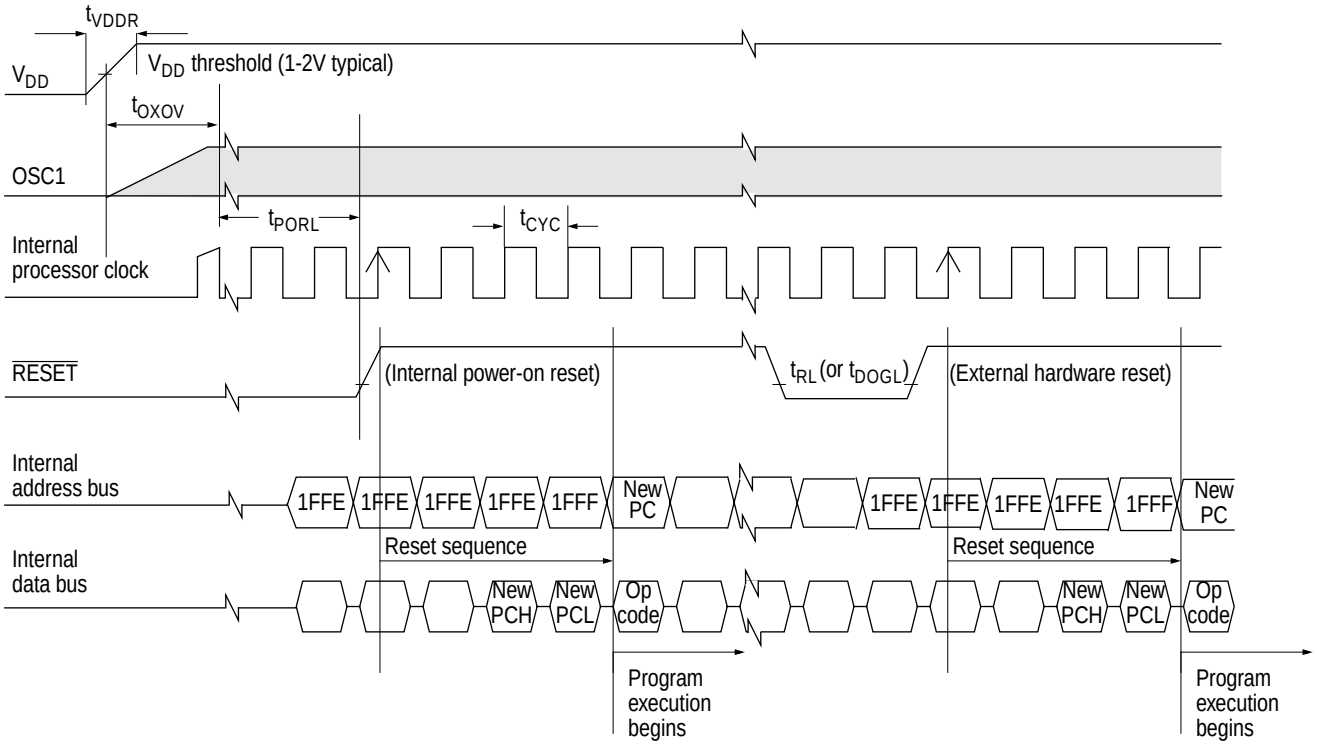


Figure 12. Reset timing diagram

Computer operating properly (COP) reset

The MCU contains a watchdog timer that automatically times out unless it is reset within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated which drives the RESET pin low to reset the MCU and the external system. The COPF flag in the COP control register (COPCR) is set whenever a COP timeout or clock monitor reset occurs; this allows COP and clock monitor resets to be distinguished from external and power-on resets.

The COP reset function can be enabled by programming the COPE control bit in the COP control register. This bit can be written once only after reset, but can be read at any time. Control bits CM0 and CM1 in COPCR allow the user to select one of four COP timeout periods. **Table 11** shows the relationship between CM0 and CM1 and the timeout period for various system clock frequencies.

The sequence for resetting the watchdog timer is as follows:

- Write $55 to the COP reset register (COPRST)
- Write $AA to the COP reset register

Both writes must occur in this sequence prior to the timeout, but any number of instructions may be executed between the two writes.

Clock monitor reset

The MCU contains a clock monitor circuit that measures the bus clock frequency (f_{OP}). The clock monitor function is enabled by the CME control bit in COPCR. Upon detection of a slow or absent clock, the clock monitor circuit (if enabled by CME = 1) will cause a system reset to be generated. If the bus clock input rate is above approximately 200 kHz, then the clock monitor does not generate a reset. If the bus clock is lost or its frequency falls below 10 kHz, then a reset is generated, and the RESET pin is driven low to reset the external system.

Special considerations are needed when using STOP and the clock monitor in the same system. Since the STOP function causes the clocks to be halted, the clock monitor function will generate a reset sequence if it is enabled prior to the STOP mode being entered. For systems that do not expect nor want a STOP function, this interaction can be useful to detect the unauthorized execution of a STOP instruction that could not be detected by the COP watchdog system. On the other hand, in

systems that use both the STOP and clock monitor functions, this interaction means that the CME bit must be written to zero just prior to executing a STOP instruction and should be written back to one as soon as the MCU resumes execution.

COP control register (COPCR)

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	0	COPF	CME	COPE	CM1	CM0
Reset:	0	0	0	0	0	0	0	0

Figure 13. COP Control Register (COPCR)

NOTE: *Bits 5, 6 and 7 are not used; these bits always read zero.*

COPF — Computer operating properly failure

The COPF flag in the COP control register is set whenever a COP timeout or clock monitor reset occurs; this allows COP and clock monitor resets to be distinguished from external and power-on resets. COPF is cleared by reading COPCR.

1 = COP or clock monitor reset has occurred
0 = COP or clock monitor reset has not occurred

CME — Clock monitor enable

This bit enables the clock monitor circuitry to generate a reset whenever the clock frequency falls below a specified level. CME can be read or written at any time.

1 = Clock monitor enabled
0 = Clock monitor disabled

COPE — Computer operating properly enable

This bit enables the COP circuitry to generate a reset whenever the COP timer times out. COPE can be read at any time, but can be written only once after reset.

1 = COP timeout enabled

0 = COP timeout disabled

CM1, CM0 — COP mode bits

The COP mode bits select one of four timeout durations for the COP timer. CM1 and CM0 are cleared by reset and can only be written once. **Table 11** shows examples of COP timeout periods for several different bus frequencies (f_{OP}).

Table 11. COP timeout periods (examples)

CM1	CM0	$f_{OP}/2^{15}$ divided by	XTAL = 4.0MHz f_{OP} = 2.0MHz	XTAL = 3.5795MHz f_{OP} = 1.7897MHz	XTAL = 2.0MHz f_{OP} = 1.0MHz	XTAL = 1.0MHz f_{OP} = 0.5MHz
0	0	1	16.38 ms	18.31 ms	32.77 ms	65.54 ms
0	1	4	65.54 ms	73.24 ms	131.07 ms	262.14 ms
1	0	16	262.14 ms	292.95 ms	524.29 ms	1.048 s
1	1	64	1.048 s	1.172 s	2.097 s	4.194 s

Interrupts

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

- External signal on the IRQ pin
- 16-bit programmable timer
- Serial communications interface
- Software interrupt instruction (SWI)

Interrupts cause the processor to save the register contents on the stack and to set the interrupt mask (I-bit) to prevent additional interrupts. The RTI instruction (return from interrupt) causes the register contents to be recovered from the stack and normal processing to resume.

Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete. The current instruction is the one already fetched and being operated on. When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not

masked (CCR I-bit clear) and the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

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

Table 12 shows the relative priority of all the possible interrupt sources. **Figure 14** shows the interrupt processing flow.

Table 12. Interrupt priorities

Source	Register	Flags	Vector address	Priority
Reset	—	—	$3FFE, $3FFF	highest
COP/Clock monitor reset	COPCR	COPF	$3FFE, $3FFF	
Software interrupt (SWI)	—	—	$3FFC, $3FFD	
External interrupt ($\overline{IRQ}$)	—	—	$3FFA, $3FFB	
16-bit programmable timer	TSR	ICF, OFC, TOF	$3FF8, $3FF9	
SCI	SCSR	TDRE, TC, RDRF, IDLE, OR	$3FF6, $3FF7	lowest

Non-maskable software interrupt (SWI)

The software interrupt (SWI) is an executable instruction and a non-maskable interrupt: it is executed regardless of the state of the I-bit in the CCR. If the I-bit is zero (interrupts enabled), SWI is executed after interrupts that were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The SWI interrupt service routine address is specified by the contents of memory locations $3FFC and $3FFD ($7FFC and $7FFD on the MC68HC05D24, MC68HC05D32 and MC68HC705D32).

Maskable hardware interrupts

If the interrupt mask bit (I-bit) of the CCR is set, all maskable interrupts (internal and external) are disabled. Clearing the I-bit enables interrupts.

NOTE: *The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I-bit is cleared.*

External interrupt (IRQ)

The interrupt request is latched immediately following the active edge or level on the IRQ pin. It is then synchronized internally and handled by the interrupt service routine, located at the address specified by the contents of $3FFA and $3FFB ($7FFA and $7FFB on the MC68HC05D24, MC68HC05D32 and MC68HC705D32). An IRQ interrupt will cause the MPU to exit from STOP mode.

16-bit programmable timer interrupt

There are three 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 register (TSR). All three interrupts will vector to the same service routine, located at the address specified by the contents of memory locations $3FF8 and $3FF9 ($7FF8 and $7FF9 on the MC68HC05D24, MC68HC05D32 and MC68HC705D32).

Serial communications interface interrupt

There are five different SCI interrupt flags that cause an SCI interrupt whenever they are set and enabled. The SCI interrupt enable bits are located in the serial communication control register 2 (SCCR2) and the SCI interrupt flags are located in the serial communication status register (SCSR). All three interrupts will vector to the same service routine, located at the address specified by the contents of memory locations $3FF6 and $3FF7 ($7FF6 and $7FF7 on the MC68HC05D24, MC68HC05D32 and MC68HC705D32).

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 15**.

RESET: *A reset condition causes the program to vector to its starting address, which is specified by the contents of memory locations $3FFE (MSB) and $3FFF (LSB). The I-bit in the condition code register is also set.*

STOP: *The STOP instruction causes the oscillator to be turned off and the processor to 'sleep' until an external interrupt (IRQ), if enabled, or a reset occurs.*

WAIT: *The WAIT instruction causes all processor clocks to stop, but leaves the timer clock running. This 'rest' state of the processor can be cleared by reset, an external interrupt (IRQ), if enabled, or a timer or SCI interrupt. There are no special wait vectors for these individual interrupts.*

Low power modes

STOP

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

During the STOP mode, all interrupt enable bits in TCR and SCCR2 are cleared by internal hardware to remove any pending timer or SCI interrupt requests and to disable any further interrupts from these sources. The timer prescaler is cleared. The I-bit in the CCR is cleared to enable external interrupts. All other bits, registers and memory contents remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt on (IRQ) or by a reset.

WAIT

The WAIT instruction places the MCU in a low-power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer remains active. Any enabled interrupt from the timer or the SCI can cause the MCU to exit the WAIT mode.

During the WAIT mode, the I-bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state.

Data retention mode

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

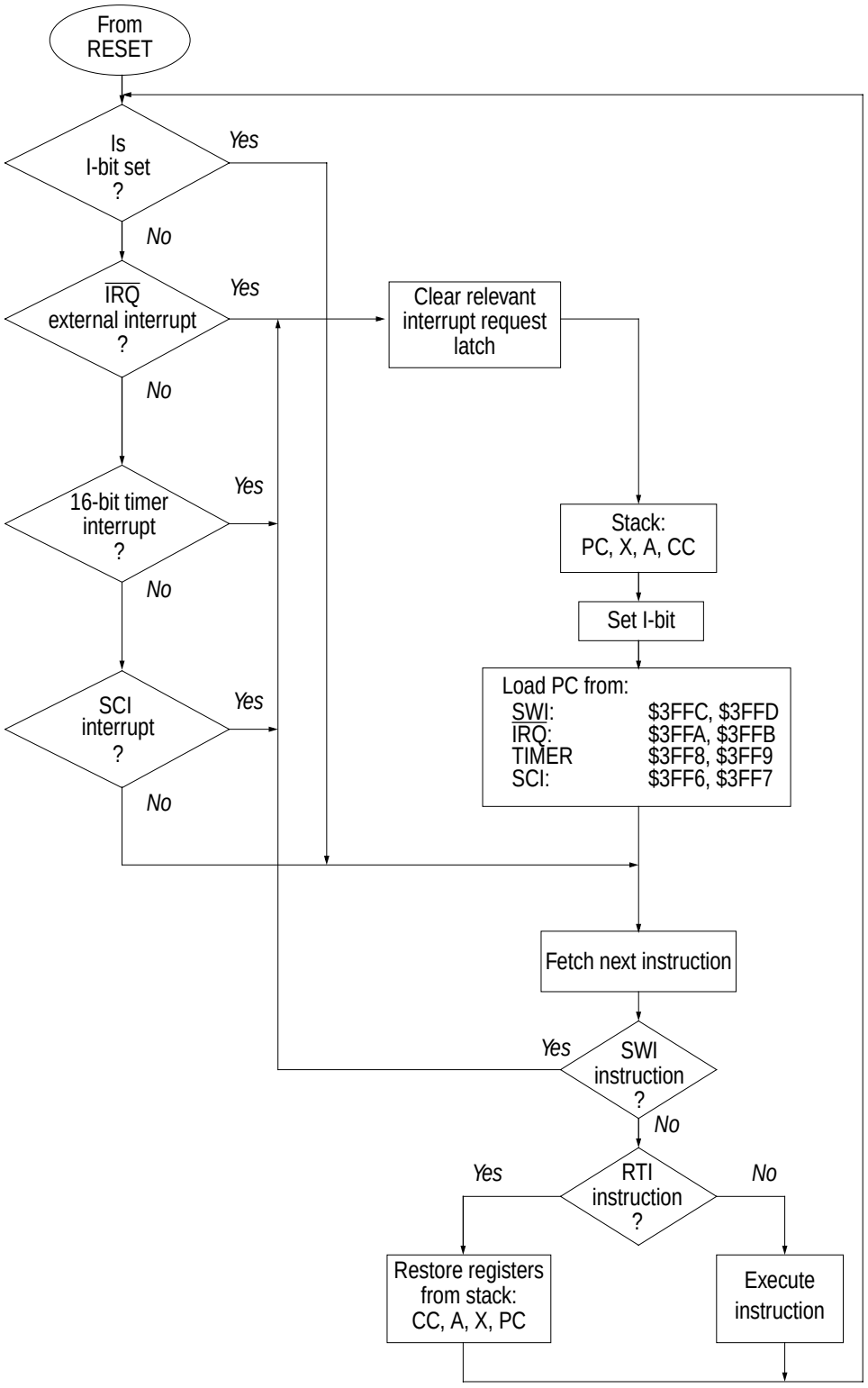


Figure 14. Interrupt flowchart

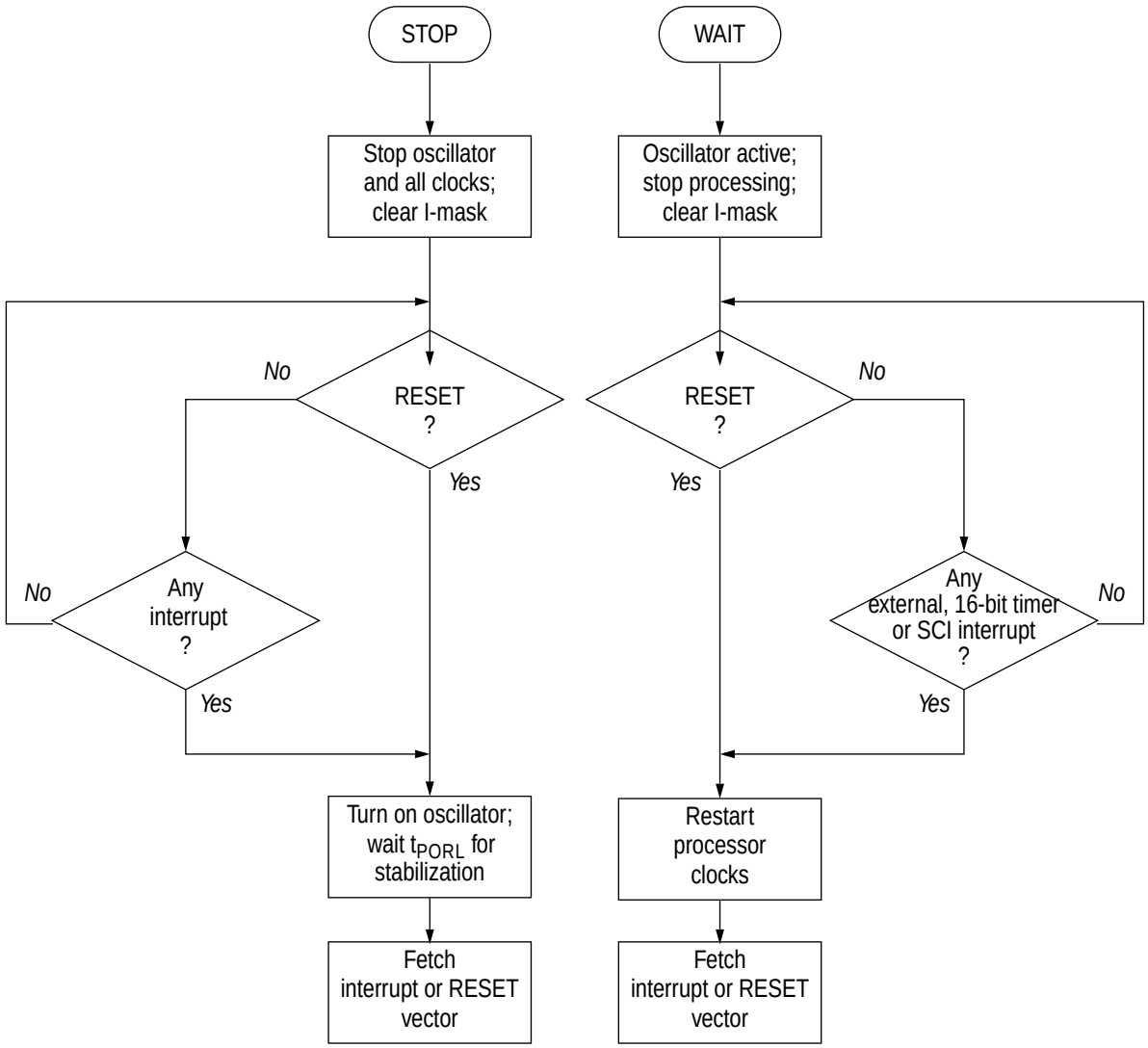


Figure 15. STOP and WAIT flowcharts

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 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 RESET line high.

Memory and Registers

Contents

Introduction 53
RAM 53
ROM 54
Registers 54

Introduction

The MCU has a 16K byte memory map consisting of registers (for I/O, control and status), User RAM, user ROM, self-check ROM and reset and interrupt vectors as shown in **Figure 16**.

Two control bits in the option register ($3FDF) allow the user to switch between RAM and ROM/EPROM, at any time, in two special areas of the memory map, $0020–$004F (48 bytes) and $0100–$017F (128 bytes).

RAM

The main user RAM consists of 176 bytes at $0050–$00FF. This RAM array is always present in the memory map and includes a 64 byte stack area. The stack pointer can access 64 bytes of RAM in the range $00FF to $00C0.

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

Two additional RAM arrays are available at $0020–$004F (48 bytes) and $0100–$017F (128 bytes). These may be accessed at any time by setting the RAM0 and RAM1 bits, respectively, in the option register (see **Software-selectable options**).

ROM

The main user ROM/EPROM occupies 15744 bytes in the memory space from $017F to $3FFF. This array includes 15744 bytes of main User ROM, 128 bytes of User ROM that are mapped into the memory space when the RAM0 bit in the option register is cleared and the User reset and interrupt vectors ($3FF6–$3FFF). It also includes the self-check ROM and vectors and the option register.

Two additional ROM arrays are mapped into the address space at $0020–$004F (48 bytes) and $0100–$017F (128 bytes) when RAM0 and RAM1, respectively, in the option register are cleared (see **Software-selectable options**).

Registers

Except for the option register, all I/O, control and status registers are contained within one 32-byte block in page zero of the address space ($0000–$001F). These registers are shown in **Table 13**.

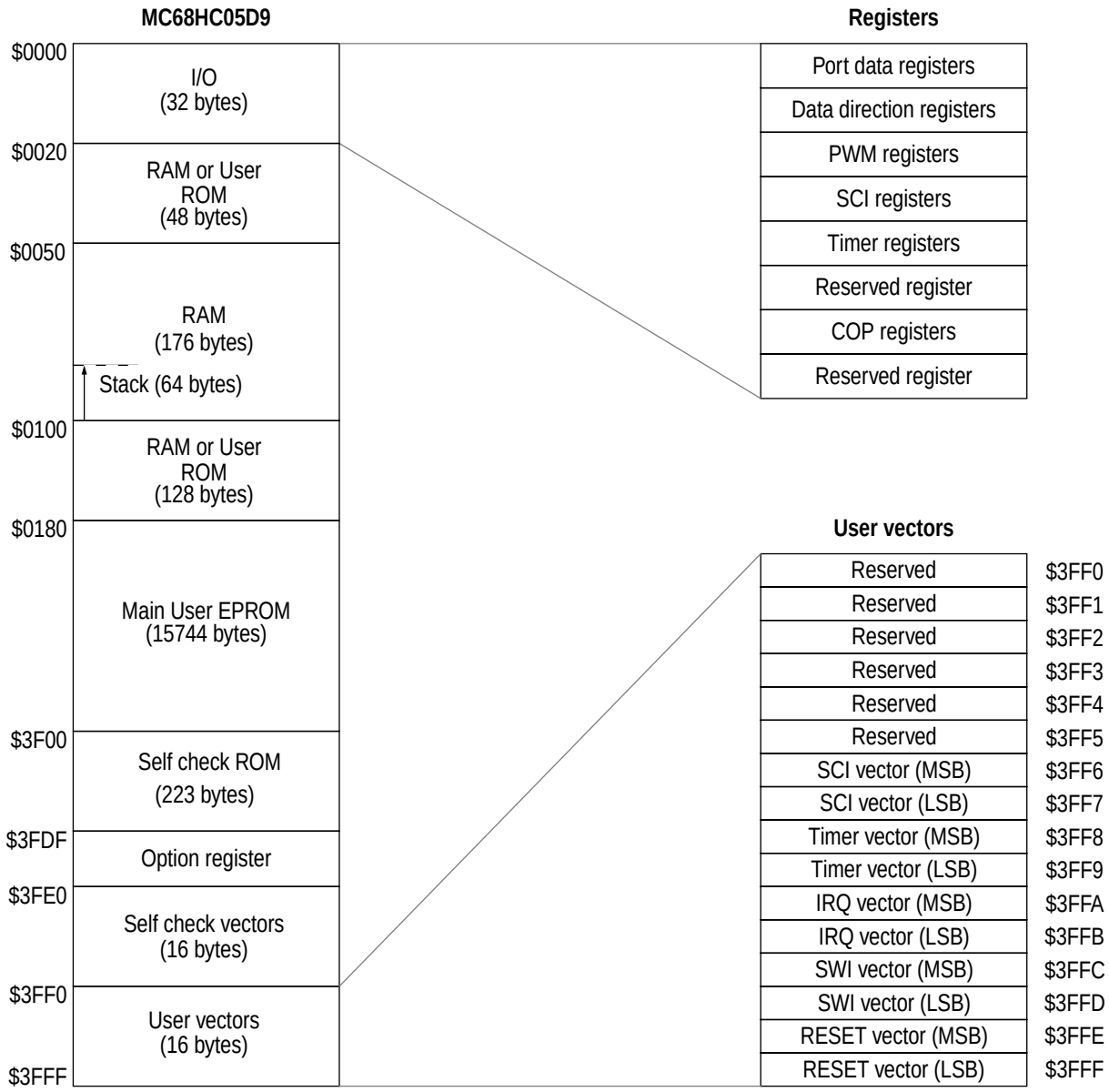


Figure 16. Memory map of the MC68HC05D9

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000									undefined
Port B data (PORTB)	$0001									undefined
Port C data (PORTC)	$0002									undefined
Port D data (PORTD)	$0003									undefined
Data direction A (DDRA)	$0004									0000 0000
Data direction B (DDRB)	$0005									0000 0000
Data direction C (DDRC)	$0006									0000 0000
Data direction D (DDRD)	$0007									0u00 0000
PWM mode (PWMM)	$0008	0	0	SCIB	PWM4	PWM3	PWM2	PWM1	PWM0	0010 0000
PWM channel 0 (PWM0)	$0009	0	0							0000 0000
PWM channel 1 (PWM1)	$000A	0	0							0000 0000
PWM channel 2 (PWM2)	$000B	0	0							0000 0000
PWM channel 3 (PWM3)	$000C	0	0							0000 0000
PWM channel 4 (PWM4)	$000D[1]	0	0							0000 0000
SCI baud rate (BAUD)				SCP1	SCP0		SCR2	SCR1	SCR0	uu00 uuuu
SCI control 1 (SCCR1)	$000E	R8	T8	0	M	WAKE	0	0	0	uu0u u000
SCI control 2 (SCCR2)	$000F	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI status (SCSR)	$0010	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	1100 0000
SCI data (SCDR)	$0011									undefined
Timer control (TCR)	$0012	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	0000 0000
Timer status (TSR)	$0013	ICF	OCF	TOF	0	0	0	0	0	uuu0 0000
Input capture (ICR)	$0014									undefined
	$0015									undefined
Output compare (OCR)	$0016									undefined
	$0017									undefined
Timer counter (TCNT)	$0018									1111 1111
	$0019									1111 1100
Alternate counter (ALTCNT)	$001A									1111 1111
	$001B									1111 1100
Reserved	$001C									undefined
COP reset (COPRST)	$001D	ICAF	ICBF	OCAF	TOF	0	0	0	0	0000 0000
COP control (COPCR)	$001E	0	0	0	COPF	CME	COPE	CM1	CM0	0000 0000
Reserved	$001F									undefined

OPTION	$3FDF	RAM0	RAM1	0	0	0		IRQ	0	0000 0u10

u = undefined

Table 13. MC68HC05D9 register assignment

1. If SCIB = 0, the PWM4 register is available at location $000D
   If SCIB = 1, the SCI baud rate generator register is available at location $000D

Input/Output Ports

Contents

Introduction 57
Input/output programming 57
Port D 59
Ports A, B and C 59
Port registers 60
Port data registers (PORTA, PORTB, PORTC and PORTD) 60
Data direction registers (DDRA, DDRB, DDRC and DDRD) 61
Other port considerations 61

Introduction

In single chip mode there are four 8-bit wide parallel ports comprising 31 bidirectional I/O lines and one output-only line. The bidirectional ports are programmable as either inputs or outputs under software control, via the data direction registers.

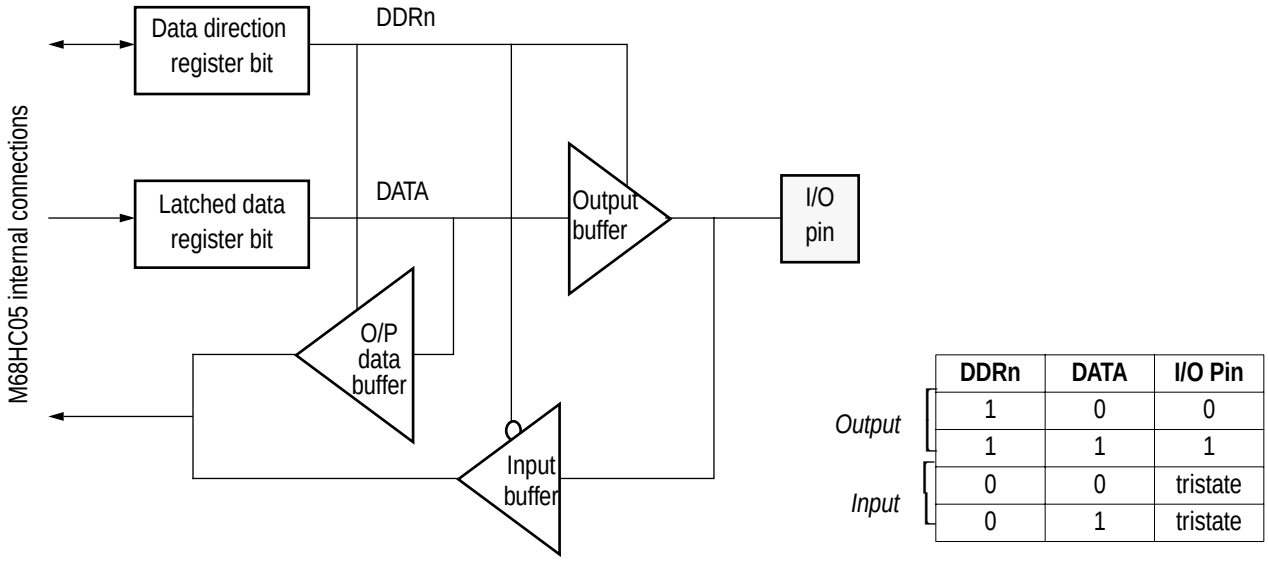
To prevent glitches on the output pins, data should be written to the I/O port data register before configuring the port as outputs, by setting the corresponding data direction register bits.

Input/output programming

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

corresponding DDR bit is set to a logic one (‘1’). A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero (‘0’).

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 processor. During the programmed output state, a read of the data register reads the value of the output data latch and not the I/O pin (see **Figure 17** and **Table 14**).



DDRn	DATA	I/O Pin
1	0	0
1	1	1
0	0	tristate
0	1	tristate

Figure 17. Standard I/O port structure

Table 14. Bidirectional I/O pin functions

R/$\overline{W}$(1)	DDR	I/O Pin function
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.

1. Note that R/$\overline{W}$ is an internal signal, not available to the user.

Ports A, B and C

Ports A, B and C are 8-bit bidirectional ports (see **Table 14**). Their data registers reside at addresses $0000–$0002 and their data direction registers (DDRn) at $0004–$0006, respectively. Reset does not affect the data register, but clears the data direction register, thereby returning the ports to inputs. Writing a ‘1’ to a data direction register bit sets the corresponding port bit to output mode.

Port D

Port D is an 8-bit wide port that shares pins with the on-board sub-systems (timer, SCI and PWM). When a sub-system function is enabled, the associated pin or pins are configured to support the input/output requirements of that function (see **PD0–PD7**). In particular, a bit whose pin is shared with the timer, SCI or PWM sub-system can be used to read the level on the pin, even when the associated sub-system is enabled. When a sub-system is enabled, certain pins associated with that sub-system are capable of becoming outputs. These pins then get their drive signals directly from the sub-system rather than from the port D data register. These pins, however, remain under the control of the data direction register and can be configured as input pins when driven directly from the sub-system.

The port D data register is located at address $0003 and the port D data direction register (DDR) at $0007. Bits 0–5 and 7 are bidirectional lines, their direction controlled by the corresponding bits in DDRD. Bit 6 is dedicated to the timer output compare function as an output; as an input, reading this bit returns the instantaneous level on the TCMP pin. Consequently, bit 6 does not have a corresponding data direction register bit in DDRD. Reset does not affect the data register, but clears the data direction register, thereby returning the ports to inputs. Writing a ‘1’ to a DDR bit sets the corresponding port bit to output mode. When a bit of DDRD is cleared, the corresponding bit in the data register monitors the level on the pin at all times.

NOTE: *The operation of the timer and PWM sub-systems can cause transitions on the port D pins that are asynchronous with respect to the read cycles on the port D data register. User software should be written in such a way as not to depend on the value of any bit on port D that was read while the operation of one of the sub-systems may have been causing transitions on the pin.*

Port registers

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

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

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

Each bit 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 data registers.

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

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction A (DDRA)	$0004									0000 0000
Data direction B (DDRB)	$0005									0000 0000
Data direction C (DDRC)	$0006									0000 0000
Data direction D (DDRD)	$0007									0u00 0000

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

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

Other port considerations

All input/output ports can emulate open-drain outputs. This is done by writing a ‘0’ to the relevant output port latch. By toggling the corresponding data direction bit, the port pin will be either output a ‘0’ or be tri-state (i.e. an input). (See **Figure 18**.)

When using a port pin as an open-drain output, certain precautions must be taken in the user software. If a read-modify-write instruction is used on a port where the open-drain is assigned and the pin at this time is programmed as an input, it will read it as a ‘1’. The read-modify-write instruction will then write this ‘1’ into the output data latch on the next cycle. This would cause the open-drain pin not to output a “0” when desired.

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

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

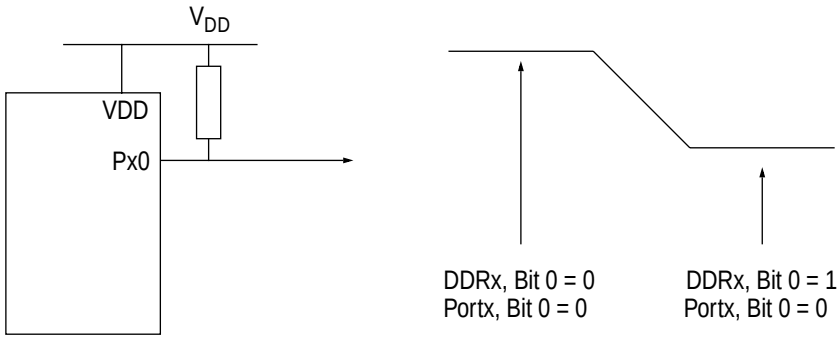


Figure 18. Port logic levels

16-bit Programmable TImer

Contents

Introduction 63
Counter 65
Counter registers 65
Timer functions 67
Timer control register (TCR) 67
Timer status register (TSR) 68
Input capture register 70
Output compare register 71
Timer during WAIT mode 72
Timer during STOP mode 73
Timer state diagrams 73

Introduction

The programmable input capture/output compare timer on the MC68HC05D9 consists of a 16-bit read-only free-running counter, preceded by a fixed divide-by-four prescaler. Selected input edges cause the current counter value to be latched into a 16-bit input capture register so that software can later read this value to determine when the edge occurred. When the free running counter value matches the value in the output compare registers, the programmed pin action takes place. Refer to **Figure 19** for a block diagram of the timer.

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

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

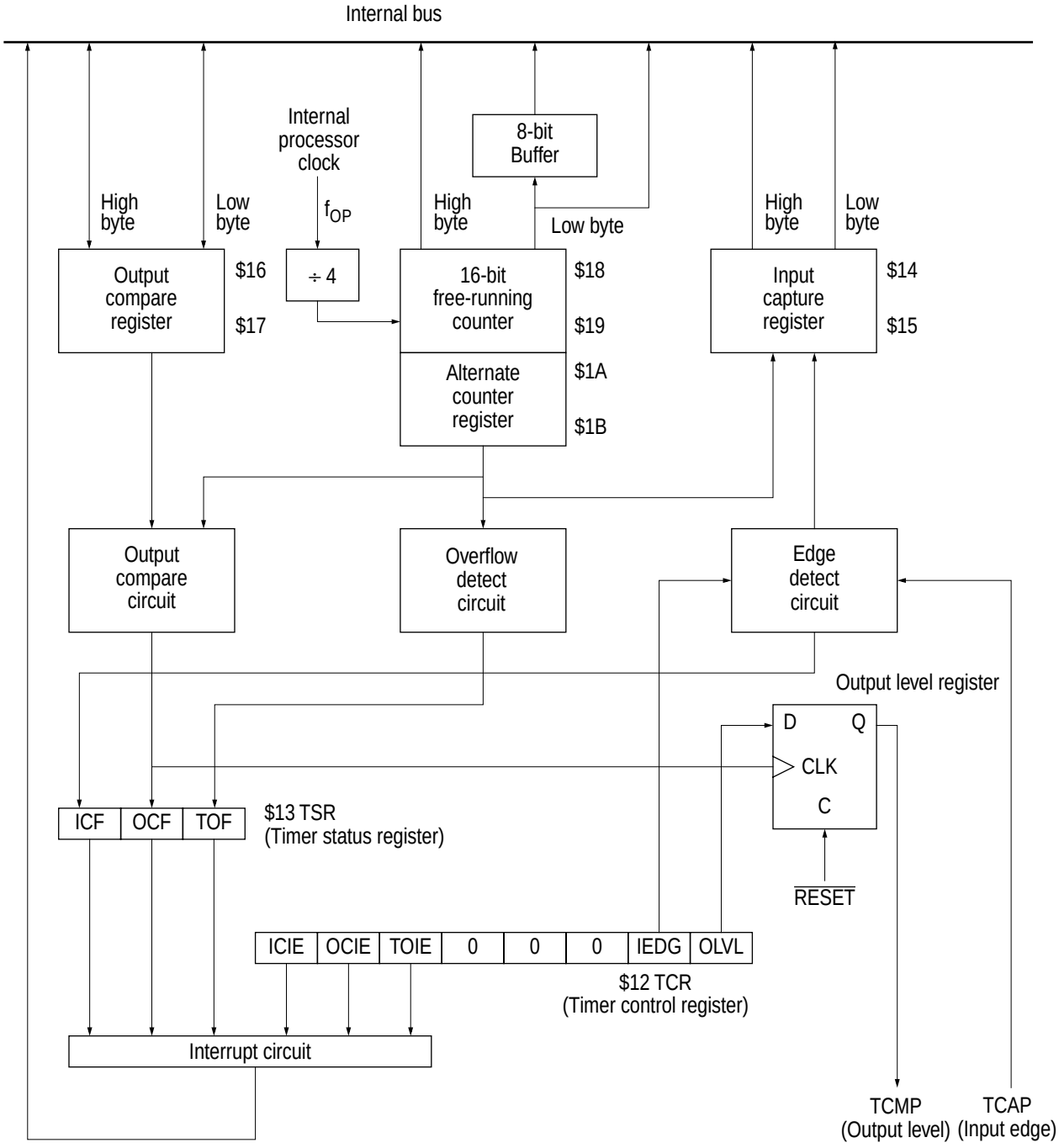


Figure 19. 16-bit programmable timer block diagram

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.

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

Counter registers

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

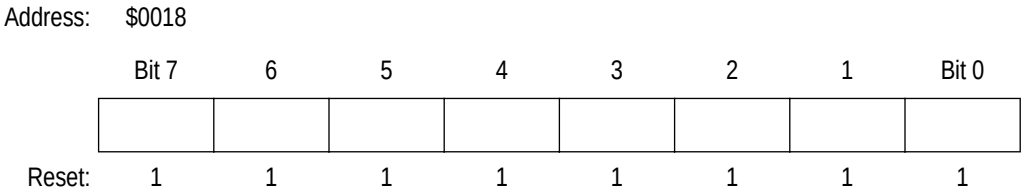


Figure 20. Timer Counter Register (TCNT) MSB

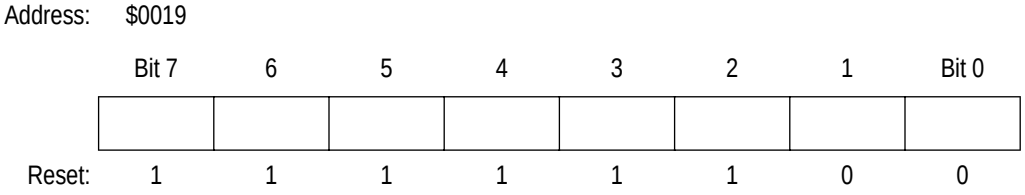


Figure 21. Timer Counter Register (TCNT) LSB

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

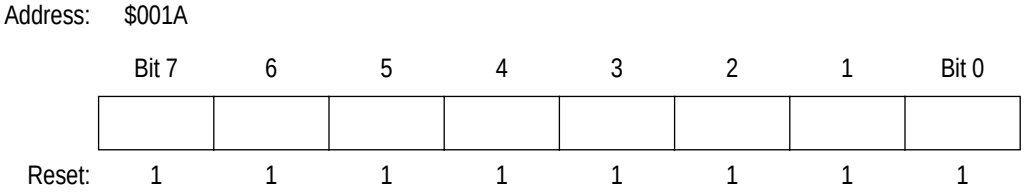


Figure 22. Alternate Counter Register (ALTCNT) MSB

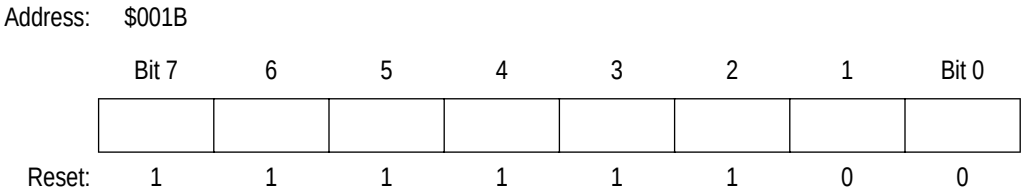


Figure 23. Alternate Counter Register (ALTCNT) LSB

Timer functions

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

Timer control register (TCR)

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

Address: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
Reset:	0	0	0	0	0	0	u	0

Figure 24. Timer Control Register (TCR)

ICIE — Input capture interrupt enable

ICIE is a read/write control bit that is used to enable or disable interrupts coming from the input capture circuitry. Reset clears this bit.

1 = Input capture interrupt enabled.
0 = Input capture interrupt disabled.

OCIE — Output compare interrupt enable

OCIE is a read/write control bit that is used to enable or disable interrupts coming from the output compare circuitry. Reset clears this bit.

1 = Output compare interrupt enabled.
0 = Output compare interrupt disabled.

TOIE — Timer overflow interrupt enable

TOIE is a read/write control bit that is used to enable or disable interrupts coming from the timer overflow detection circuitry. Reset clears this bit.

1 = Timer overflow interrupt enabled.
0 = Timer overflow interrupt disabled.

IEDG — Input edge

IEDG is a read/write control bit that is used to specify the input capture trigger mechanism. Reset clears this bit.

1 = A positive going edge on the TCAP pin will trigger free-running counter transfer to the input capture register.

0 = A negative going edge on the TCAP pin will trigger free-running counter transfer to the input capture register.

OLVL — Output level

OLVL is a read/write control bit that is used to specify whether a logic high or logic low level will appear on the TCMP pin following the next successful output compare. Reset clears this bit.

1 = A high output value will be clocked into the output level register by the next successful output compare and will appear on the TCMP pin.

0 = A low output value will be clocked into the output level register by the next successful output compare and will appear on the TCMP pin.

Timer status register (TSR)

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

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

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
	ICF	OCF	TOF	0	0	0	0	0
Reset:	u	u	u	0	0	0	0	0

Figure 25. Timer Status Register (TSR)

ICF — Input capture flag

ICF is a read-only bit that is set whenever an input capture occurs. If input capture interrupts are enabled (via ICIE in TCR), the setting of this bit will interrupt the processor. It is cleared by the sequence of reading TSR followed by the input capture low register ($15).

1 = Input capture has occurred.
0 = No interrupt capture has occurred.

OCF — Output compare flag

OCF is a read-only bit that is set whenever an output compare occurs. If output compare interrupts are enabled (via OCIE in TCR), the setting of this bit will interrupt the processor. It is cleared by the sequence of reading TSR followed by the output compare low register ($17).

1 = Output compare has occurred.
0 = No output compare has occurred.

TOF — Timer overflow flag

TOF is a read-only bit that is set whenever a timer overflow occurs. If timer overflow interrupts are enabled (via TOIE in TCR), the setting of this bit will interrupt the processor. It is cleared by the sequence of reading TSR followed by the counter low register ($19).

1 = Timer overflow has occurred.
0 = No timer overflow has occurred.

NOTE: *When using the timer overflow function and reading the free-running counter at random times to measure an elapsed time, a problem could occur where the timer overflow flag may be cleared unintentionally if the timer status register is read or written when TOF is set and the LSB of the free-running counter is read, but not for the purpose of servicing the flag.*

Input capture register

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

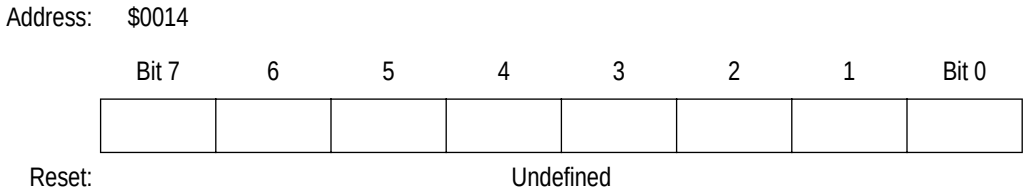


Figure 26. Input Capture Register (ICR) High Byte

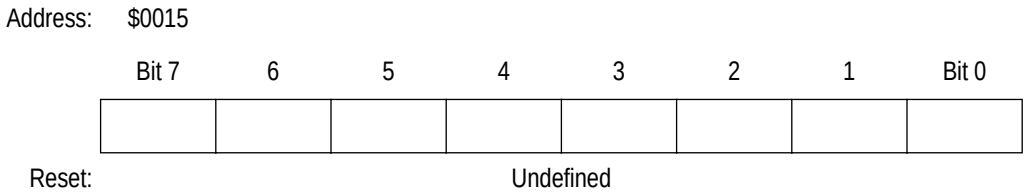


Figure 27. Input Capture Register (ICR) Low Byte

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

The result obtained from an input capture will be one greater than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal 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 on each valid signal transition whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture. After a read of the input capture register MSB ($14), the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input

capture software routine and its interaction with the main program to determine the minimum pulse period. A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since the two actions occur on opposite edges of the internal bus clock.

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

Output compare register

'Output compare' is a technique that may be used, for example, to generate an output waveform, or to signal when a specified time has elapsed, by presetting the output compare register to the appropriate value. Note that the output pin used is the shared PD6/TCMP pin.

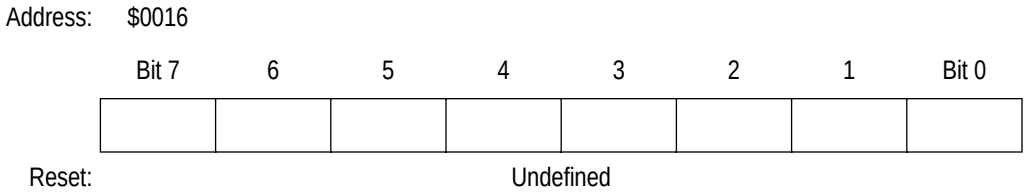


Figure 28. Output Compare Register (OCR) High Byte

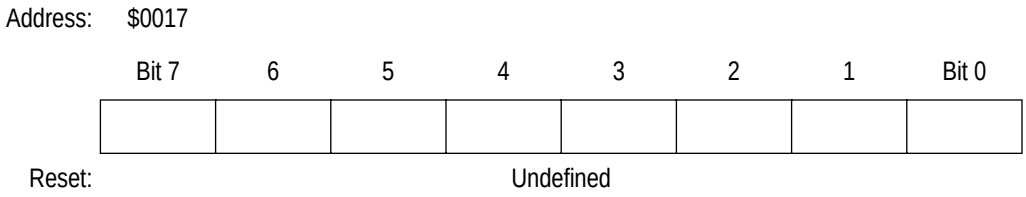


Figure 29. Output Compare Register (OCR) Low Byte

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

interrupt enable bit (OCIE) is set. (The free-running counter is updated every four internal bus clock cycles.)

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

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

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

Timer during WAIT mode

Only the CPU clock halts during WAIT mode, hence the timer continues to run. A timer interrupt may be used to bring the CPU out of WAIT mode. If $\overline{\text{RESET}}$ is used to exit WAIT mode, the counters are forced to $FFFC.

Timer during STOP mode

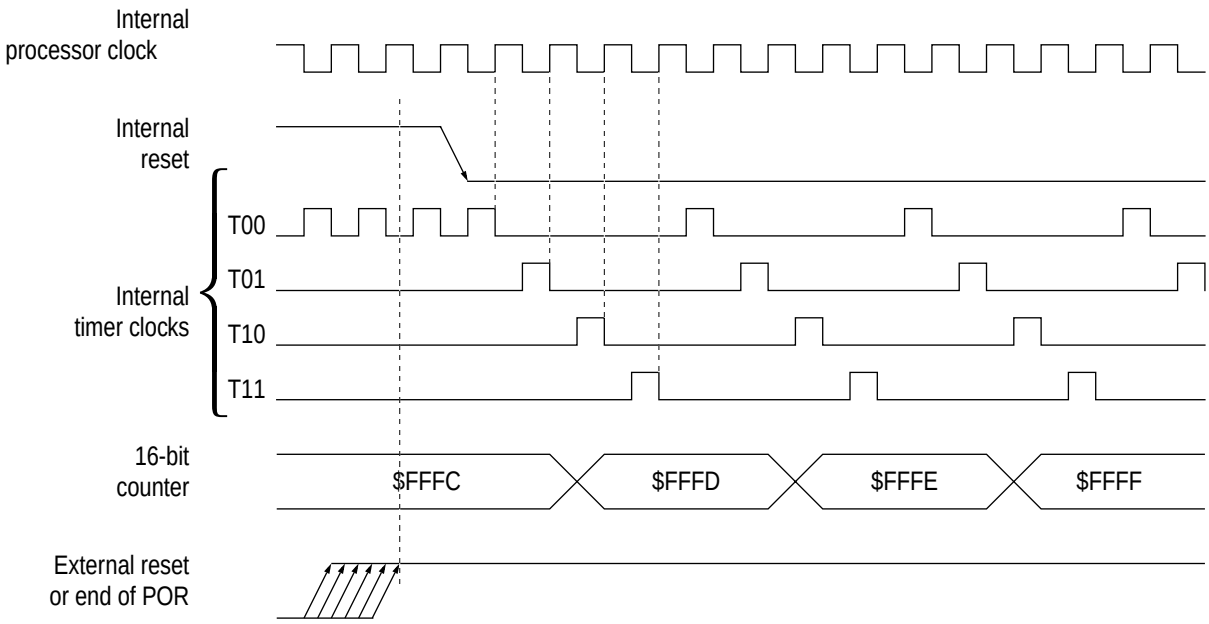
In the STOP mode all MCU clocks are stopped, hence the timer stops counting. If STOP is exited by an interrupt the counter retains the last count value. If the device is reset then the counter is forced to $FFFC.

During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detection circuitry is armed. This does not set any timer flags nor wake up the MCU. When the MCU does wake up, however, there is an active input capture flag and data from the first valid edge that occurred during the STOP period. If the device is reset to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

Timer state diagrams

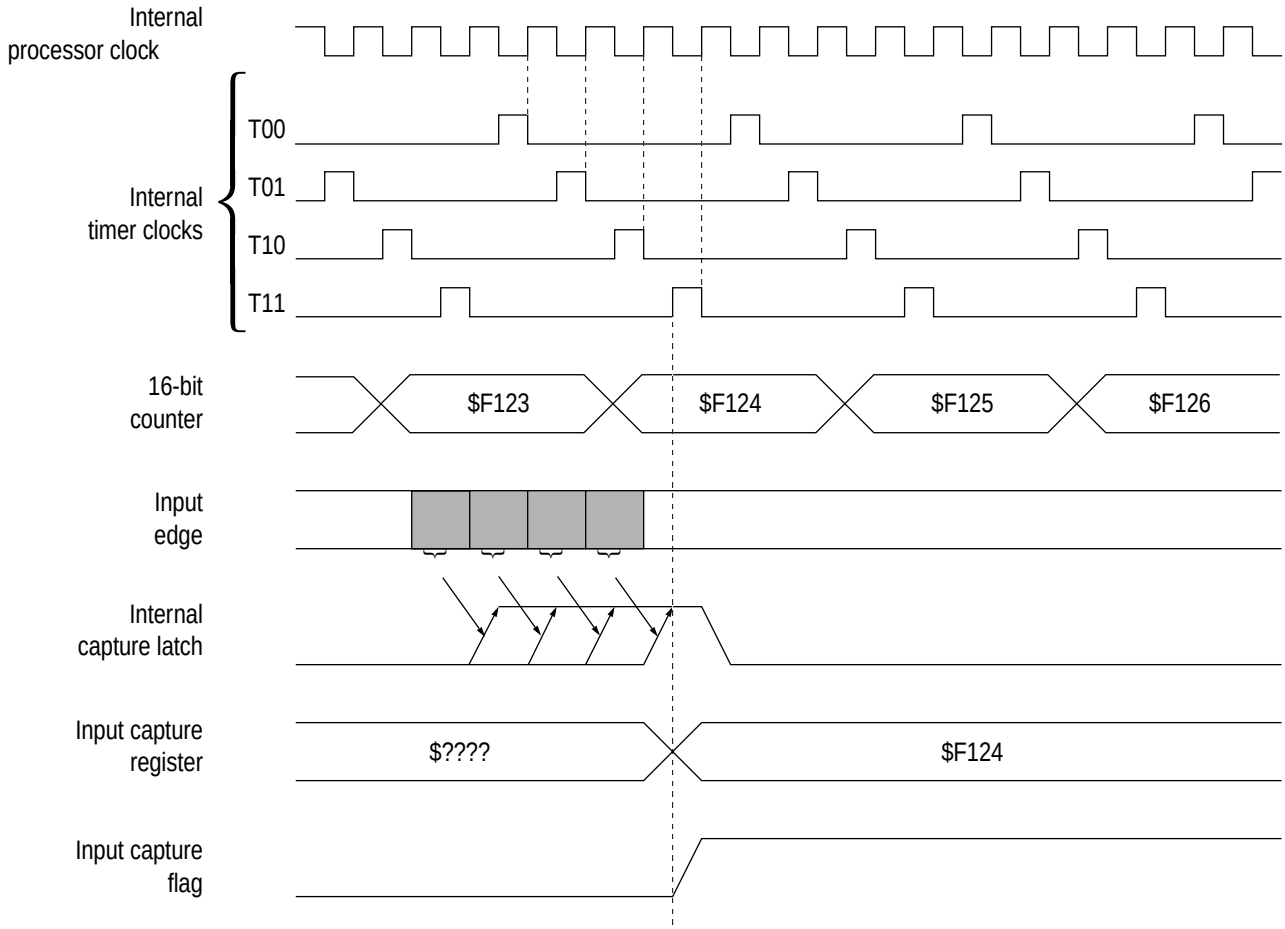
The relationships between the internal clock signals, the counter contents and the status of the flag bits are shown in the following figures.

The signals labelled ‘internal’ (processor clock, timer clocks and reset) are not available to the user.



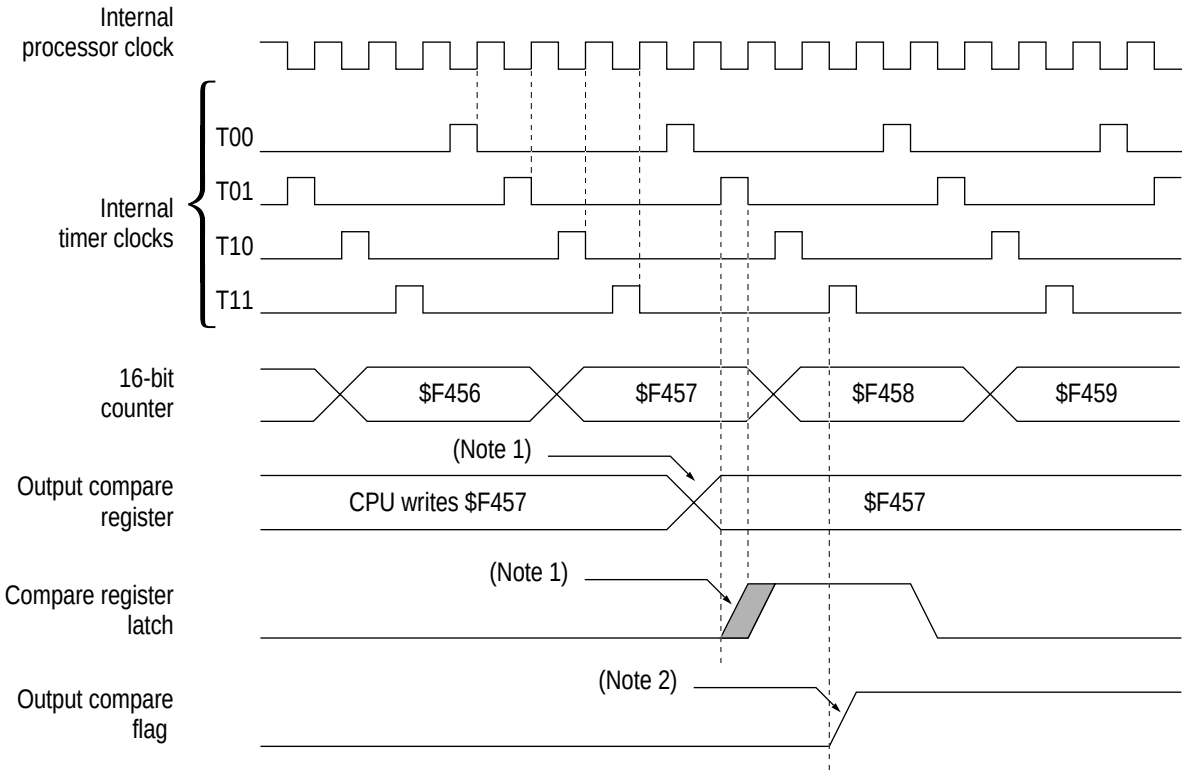
Note: The counter and timer control registers are the only ones affected by power-on or external reset.

Figure 30. 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..

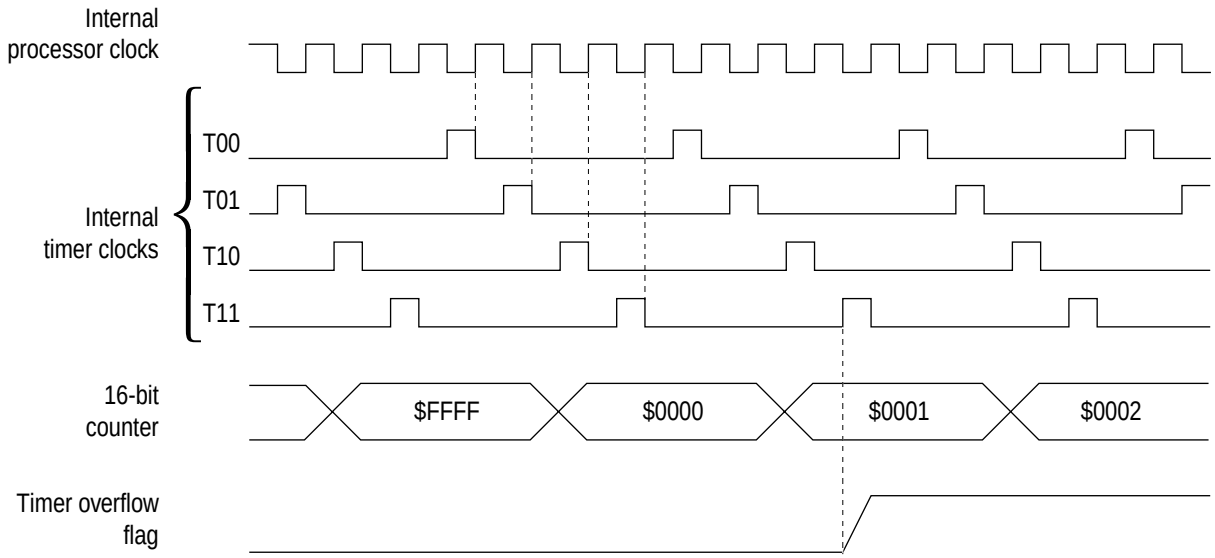
Figure 31. Timer state timing diagram for input capture



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 32. 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 33. Timer state timing diagram for timer overflow

Serial Communications Interface (SCI)

Contents

Introduction 79
Overview and features 80
SCI two-wire system features 80
SCI receiver features 80
SCI transmitter features 80
Functional description 81
Data format 83
Receiver wake-up operation 84
Idle line wake-up 84
Address mark wake-up 85
Receive data (RDI) 85
Start bit detection 86
Transmit data (TDO) 89
SCI registers 89
PWM mode register (PWMM) 89
SCIB — Serial communications interface baud 89
Serial communications data register (SCDR) 90
Serial communications control register 1 (SCCR1) 90
Serial communications control register 2 (SCCR2) 91
Serial communications status register (SCSR) 94
Baud rate register (BAUD) 96

Introduction

This section describes the UART type serial communications interface system (SCI). The SCI can be used, for example, to connect a CRT terminal or personal computer to the MCU or to form a serial communication network connecting several widely distributed MCUs.

Overview and features

The SCI on the MCU is a full duplex UART type asynchronous system. The SCI uses standard non-return-to-zero (NRZ) format (one start bit, eight or nine data bits, and one stop bit). An on-chip baud rate generator derives standard baud rate frequencies from the MCU oscillator. Both the transmitter and the receiver are double buffered, so back-to-back characters can be handled easily even if the CPU is delayed in responding to the completion of an individual character. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate.

SCI two-wire system features

- Standard NRZ (mark/space) format
- Advanced error detection method includes noise detection for noise duration of up to 1/16th bit time
- Full-duplex operation
- Software programmable for one of 32 different baud rates
- Software selectable word length (eight- or nine-bit words)
- Separate transmitter and receiver enable bits
- Can be interrupt driven
- Four separate enable bits available for interrupt control

SCI receiver features

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

SCI transmitter features

- Transmit data register empty flag

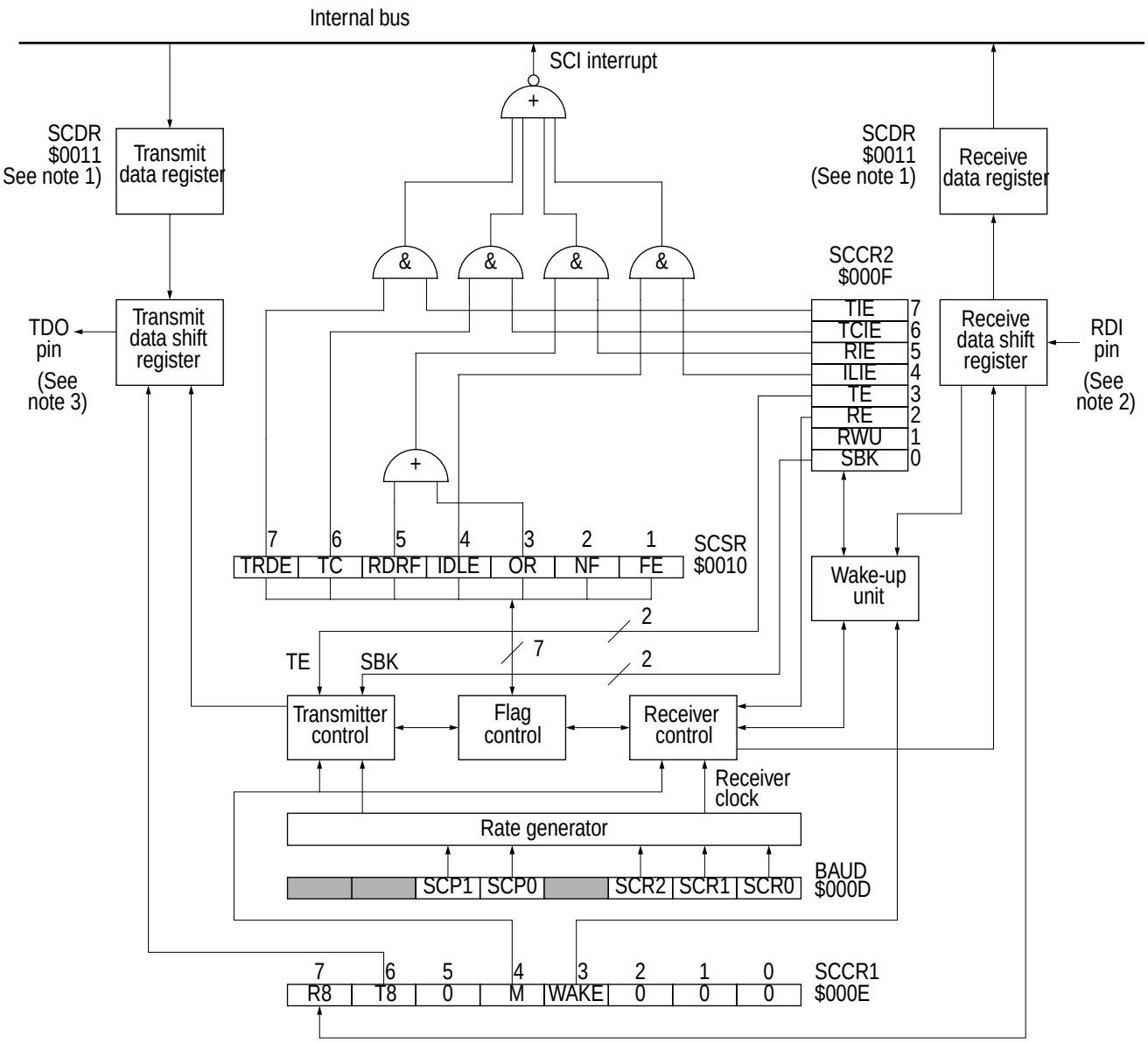
- Transmit complete flag
- Send break

Functional description

A block diagram of the SCI is shown in Figure 8-1. The user has option bits in serial communications control register 1 (SCCR1) to select the “wake-up” method (WAKE bit) and data word length (M bit) of the SCI. Serial communications control register 2 (SCCR2) provides control bits which individually enable/disable the transmitter or receiver (TE and RE, respectively), enable system interrupts (TIE, TCIE, ILIE) 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.

Data transmission is initiated by a write 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 may generate an interrupt if the transmit interrupt is enabled. The transfer of data to the transmit data shift register is synchronized with the bit rate clock (**Figure 35**). 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 may be 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 in the middle of a transmission, that character 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, which can cause an interrupt



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

Receive data in (RDI) is served by pin 0 of port B (PD0)

Transmit data out (TDO) is served by pin 1 of port B (PD1)

Figure 34. Serial communications interface block diagram

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 an idle line interrupt will not be generated.

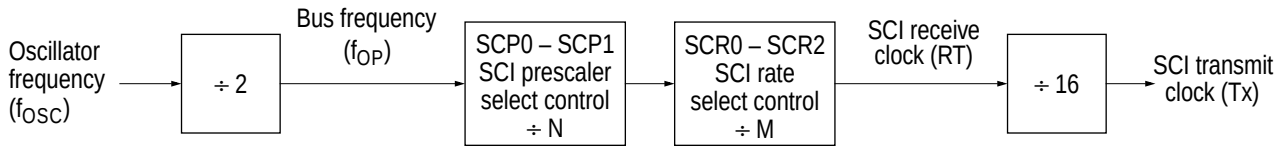


Figure 35. Rate generator division

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 8-3 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.

.

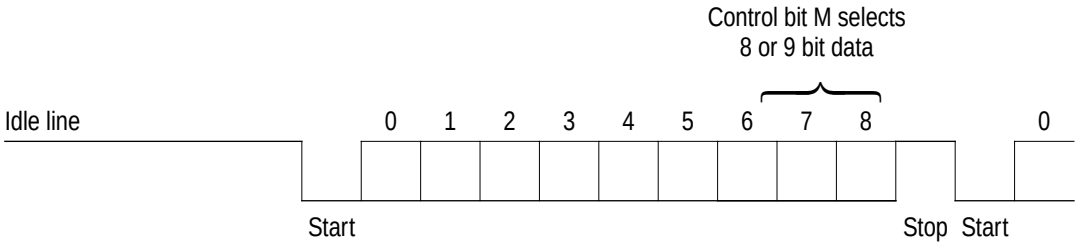


Figure 36. Data format

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 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 gets cleared automatically with hardware by one of the two methods described below.

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.

Address mark wake-up

In address mark wake-up, the most significant bit (MSB) in a character is used to indicate that the character is an address (1) or a 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.

Receive data (RDI)

Receive data is the serial data that is applied through the input line and the serial communications interface to the internal bus. The receiver circuitry clocks the input at a rate equal to 16 times the baud rate and this time is referred to as the RT clock.

Once a valid start bit is detected the start bit, each data bit and the stop bit are sampled three times at positions 8 RT, 9 RT and 10 RT (1 RT is the position where the bit is expected to start), as shown in **Figure 37**. The value of the bit is determined by voting logic which takes the value of the majority of the samples.

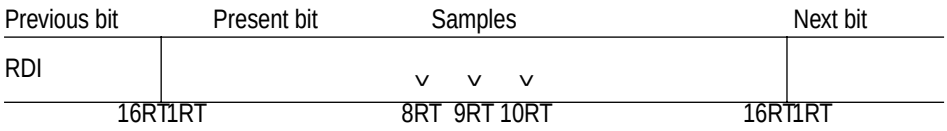


Figure 37. SCI sampling technique used on all bits

Start bit detection

When the RDI input is detected low, it is tested for three more sample times (referred to as the start edge verification samples in **Figure 38**). 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 all three verification samples do not detect a logic zero. 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 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 38**) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see **Figure 39**); therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break produced the framing error, the start bit will not be artificially induced and the receiver must detect a logic one before the start bit can be recognised (see **Figure 40**).

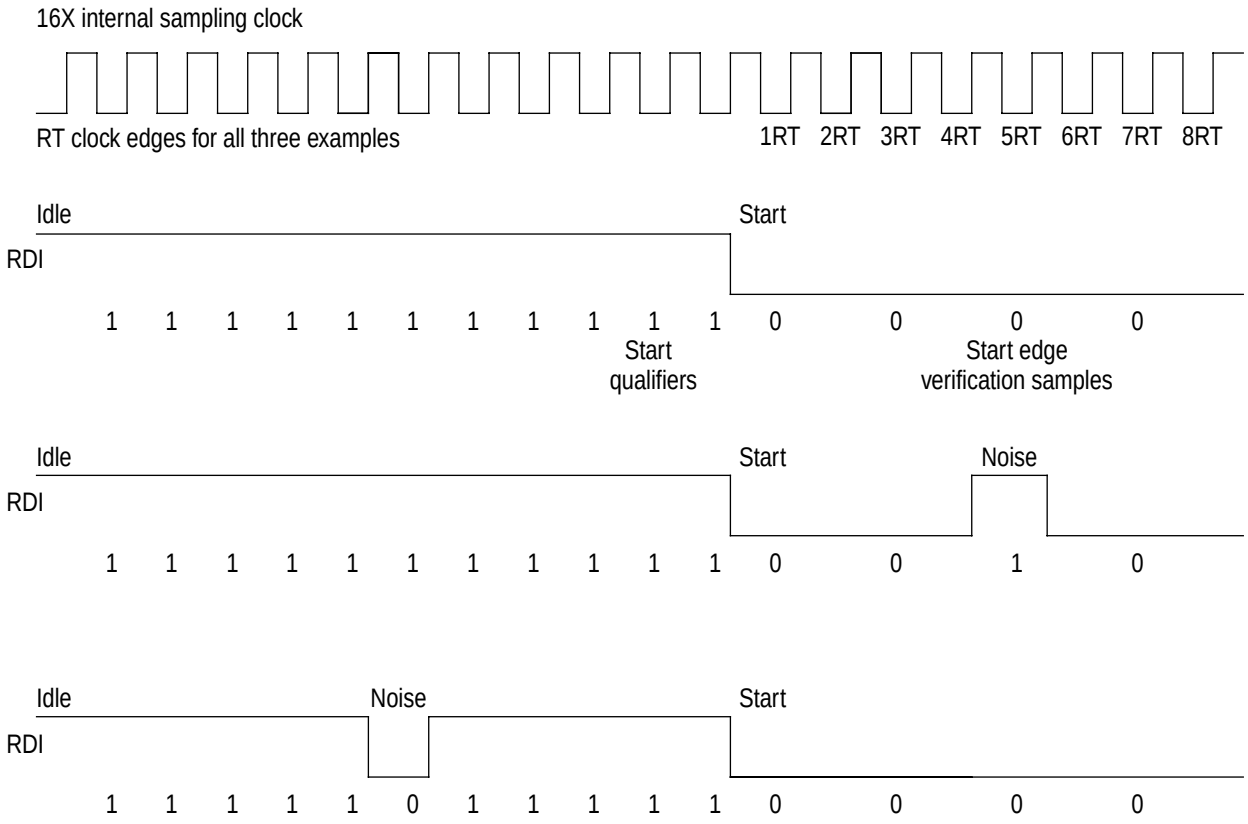
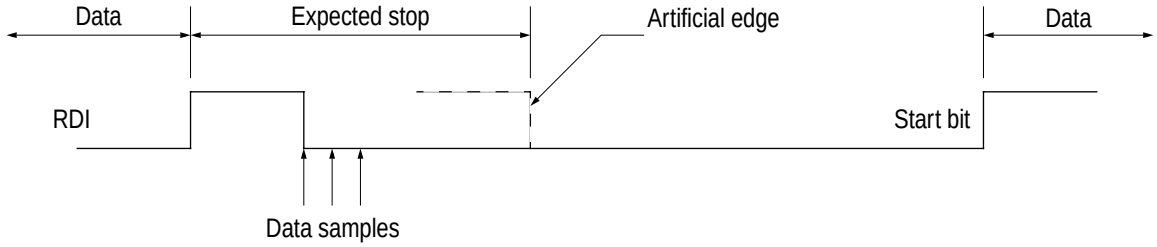
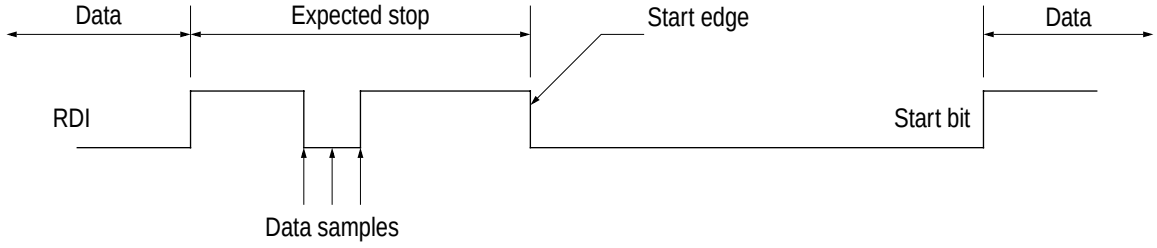


Figure 38. SCI examples of start bit sampling technique



a) Case 1, receive line low during artificial edge



b) Case 2, receive line high during expected start edge

Figure 39. Artificial start following a framing error

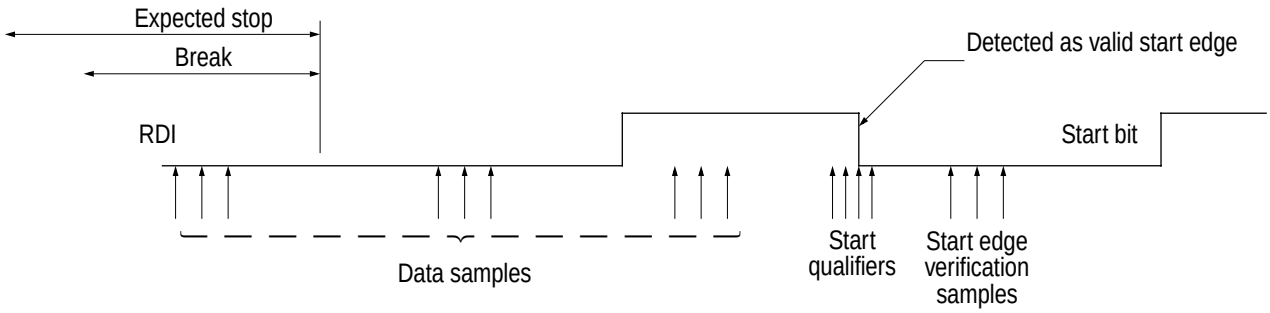


Figure 40. SCI start bit following a break

Transmit data (TDO)

Transmit data is the serial data from the internal data bus that is applied through the serial communications interface to the output line. 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.

SCI registers

Primarily the SCI system is configured and controlled by five registers (BAUD, SCCR1, SCCR2, SCSR, and SCDR). In addition, the serial communications interface baud bit (SCIB) in the PWM mode register (PWMM) provides access to the SCI baud rate register (BAUD) at address $000D (see Figure 8-1).

PWM mode register (PWMM)

Address: $0008

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	SCIB	PWM4	PWM3	PWM2	PWM1	PWM0
Reset:	0	0	1	0	0	0	0	0

Figure 41. PWM Mode Register (PWMM)

SCIB — Serial communications interface baud

SCIB is a read/write control bit that provides access to either the SCI baud rate register or the PWM4 data register at address $000D. This bit is set following reset.

1 = SCI baud rate register accessed at address $000D
0 = SCI PWM4 register accessed at address $000D

The other control bits in this register are discussed in **Pulse Width Modulator (PWM)**.

Serial communications data register (SCDR)

The SCI data register (SCDR) is actually two separate registers. When SCDR is read, the read-only receive data register is accessed and when SCDR is written, the write-only transmit data register is accessed. This address is unaffected by reset.

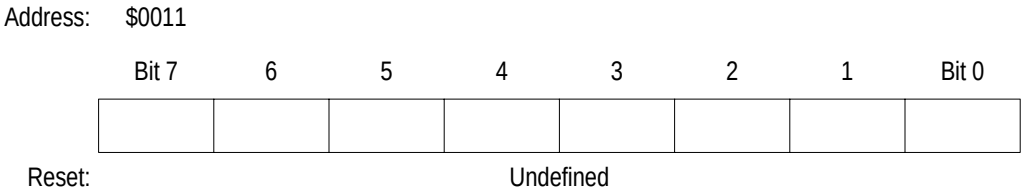


Figure 42. SCI Data Register (SCDR)

Serial communications control register 1 (SCCR1)

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

NOTE: *Bits 0–2 and 5 are not implemented in this register and always read as zeros.*

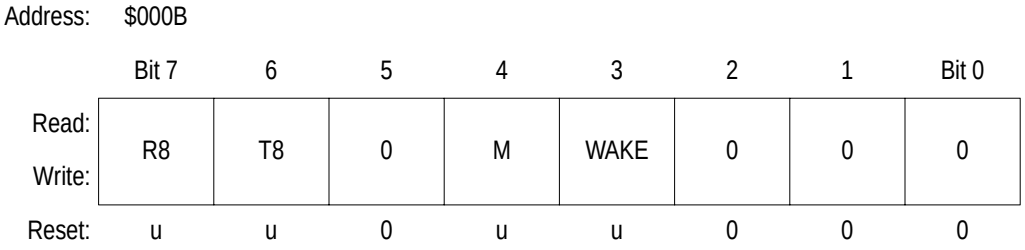


Figure 43. SCI Control Register 1 (SCCR1)

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 0–7) are transferred from the serial receive shifter 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 0–7) 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 shifter.

M — Mode (select character format)

The M bit is a read/write bit which 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 in conjunction with the “address mark” wake-up method. It can also be used as a parity bit.

1 = 1 start bit, 8 data, 9th data bit, 1 stop bit
0 = 1 start bit, 8 data bits, 1 stop bit

WAKE — Wake-up mode select

This read/write bit is used to select the wake-up mode used by the receiver.

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

Serial communications control register 2 (SCCR2)

The SCI control register 2 (SCCR2) contains the control bits that enable/disable individual SCI functions. SCCR2 is cleared during reset and can be read or written at any time.

Address: $000F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 44. SCI Control Register 2 (SCCR2)

TIE — Transmit interrupt enable

This bit allows the processor to be interrupted by the SCI when the contents of the transmit data register are transferred to the transmit shift register, i.e. when the transmit data register is ready to accept the next byte to be transmitted.

1 = Transmit interrupt enabled (interrupt if TDRE = 1)
0 = Transmit interrupt disabled

TCIE — Transmit complete interrupt enable

This bit allows the processor to be interrupted by the SCI when the last bit has been transferred out of the transmit shift register to the TDO line.

1 = Transmit complete interrupt enabled (interrupt if TC = 1)
0 = Transmit complete interrupt disabled

RIE — Receiver interrupt enable

This bit allows the processor to be interrupted by the SCI when the contents of the receive shift register are transferred to the receive data register or when an overrun occurs (i.e. when a byte is ready to be transferred from the receive shift register but the receive data register still contains the previously received byte.

1 = Receiver interrupt enabled (interrupt if RDRF or OR = 1)
0 = Receiver interrupt disabled

ILIE — Idle line interrupt enable

This bit allows the processor to be interrupted by the SCI when the receiver detects an idle line condition.

1 = Idle line interrupt enabled (interrupt if IDLE = 1)
0 = Idle line interrupt disabled

TE — Transmitter enable

When the transmitter 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 can 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, the data direction register control for Port D bit 1 is overridden and the line is forced to be an output.

1 = Transmitter enabled
0 = Transmitter disabled

RE — Receiver enable

When the receiver enable bit is set, the RDI line is applied to the receiver shift register input. When RE is clear, the receiver is disabled and all the status bits associated with the receiver (RDRF, IDLE, OR, NF and FE) are inhibited. While the receiver is enabled, the data direction register control for Port D bit 0 is overridden and the line is forced to be an input. After writing to RE, at least 10 RT cycles must occur before data can be received correctly

1 = Receiver enabled
0 = Receiver disabled

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. If the WAKE bit is cleared, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If the WAKE bit is set, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

1 = Receiver wake-up enabled
0 = Receiver wake-up disabled

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 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.

1 = Send break
0 = Do not send break

Serial communications status register (SCSR)

This read-only register contains all the flag bits that indicate the status of the SCI and are used to generate interrupts from the SCI to the processor.

Address: $0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TDRE	TC	RDRF	IDLE	OR	NF	FE	0
Reset:	1	1	0	0	0	0	0	0

Figure 45. SCI Status Register (SCSR)

TDRE — Transmit data register empty flag

This bit is set when the byte in the transmit data register is 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. Reset sets this bit.

1 = Transmit data register empty
0 = Transmit data register full

TC — Transmission complete flag

This bit is set to indicate that the SCI transmitter has no meaningful information to transmit (no data in shifter, no preamble, no break). When TC is set the serial line will go idle (continuous MARK). Reset sets this bit.

1 = Transmission complete
0 = Transmission not complete

RDRF — Receive data register full flag

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

1 = Receive data register full
0 = Receive data register empty

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. Once cleared, IDLE will not be set again until after RDRF has been set, (until after the line has been active and becomes idle again). Reset clears this bit.

1 = Idle line detected
0 = Idle line not detected

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 this bit is cleared. Reset clears this bit.

1 = Overrun error has occurred
0 = Overrun error has not occurred

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 set during the same cycle as the RDRF bit but does not get set if an overrun (OR) occurs. Reset clears this bit.

1 = Noise error has occurred
0 = Noise error has not occurred

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. It is set during the same cycle as the RDRF bit but does not get set if an overrun (OR) has occurred. The transfer from the receive shifter to the receive data register is also inhibited if an overrun has occurred. The framing error flag inhibits further transfer of data into the receive data register until it is cleared. Reset clears this bit.

1 = Framing error has occurred
0 = Framing error has occurred

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 initialisation, 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 so it is not necessary to use special baud rate crystal frequencies. (See **Figure 35**.)

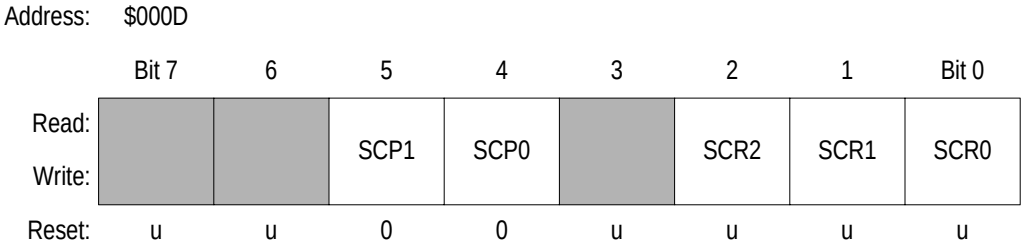


Figure 46. SCI Baud Rate Register (BAUD)

NOTE: *This register shares address $000D with PWM4. Access to the baud rate register is gained by setting the SCIB bit in the PWM mode register to 1. The SCIB bit defaults to 1 on reset.*

SCP1, SCP0 — Serial prescaler select bits

These read/write bits select one of four division ratios for the first prescaler stage (N) shown in **Figure 35**. The bus frequency clock (f_{OP}) is divided by the factor N shown in **Table 15**. This prescaled output provides an input to the second prescaler stage which is controlled by the SCI rate select bits (SCR2–SCR0)

Table 15. First prescaler stage

SCP1	SCP0	SCI prescaler division ratio (N)
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 one of eight division ratios for the second prescaler stage (M) shown in **Figure 35**. The prescaler output described above is divided by the factors shown in **Table 16**.

Table 16. Second prescaler stage

SCR2	SCR1	SCR0	SCI rate select division ratio (M)
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

Pulse Width Modulator (PWM)

Contents

Introduction 99
PWM counter 99
PWM registers. 100
PWM mode register 100
PWM channel data registers (PWM0–PWM4) 102

Introduction

The pulse width modulation unit on the MC68HC05D9 is a self-contained sub-system providing 5 PWM channels of six bits to be used as independent D to A converters. It consists of a 6-bit counter, a PWM mode control register and 5 PWM channel data registers (see **Figure 47**). The PWM output signals, PWM0–4, appear on port D pins 2–5 and 7, respectively, provided they are enabled.

PWM counter

The PWM counter, which has a range of $00 to $3F, is driven by the bus frequency clock, f_{op}. The output pulse level on each channel is set to 1 (high) when the counter value equals $00. The counter value is continuously compared with the contents of each data register. When the counter value equals that of the PWM data register, the output pulse is reset to 0 (low).

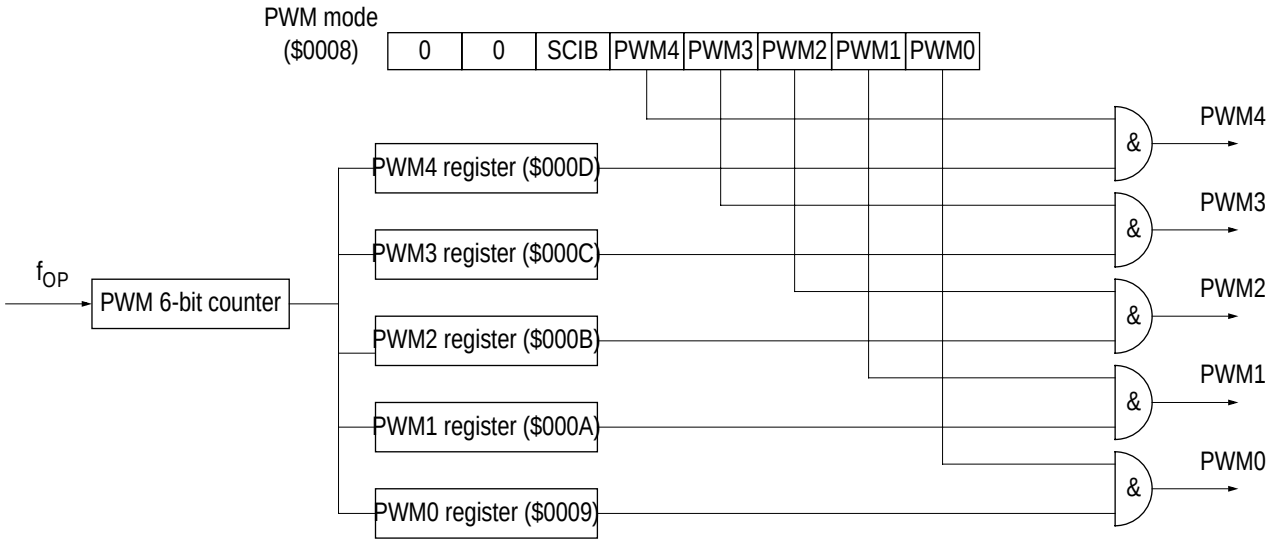


Figure 47. PWM block diagram

PWM registers

PWM mode register

This register contains five bits, PWM0–PWM4, that enable the PWM output waveforms to appear on the associated port D pins and one bit, SCIB, which allows either the PWM4 data register or the SCI baud register to be accessible at address $000D. On reset, the SCI baud register is available at this address.

When not being used for the PWM outputs, the associated pins can be configured as normal I/O pins by clearing the corresponding PWM bit in the PWM mode register, where a 1 dedicates the pin to PWM output. Due to the port D data direction register being set to input on reset, the PWM channels present a high impedance until the PWM mode register is rewritten.

Address: $0008

Bit 7	6	5	4	3	2	1	Bit 0

Figure 48. PWM Mode Register (PWMM)

Read: Write:	0	0	SCIE	PWM4	PWM3	PWM2	PWM1	PWM0
Reset:	0	1	0	0	0	0	0	0

Figure 48. PWM Mode Register (PWMM)

SCIB — Serial communications interface baud

SCIB is a read/write control bit that provides access to either the SCI baud rate register or the PWM4 data register at address $000D. This bit is set following reset.

1 = SCI baud rate register accessed at address $000D
0 = PWM4 register accessed at address $000D

PWM0 – 4 — PWM channel enable bits

These read/write bits allow the corresponding PWM channels to be enabled or disabled.

1 = PWM channel enabled
0 = PWM channel disabled

PWM channel data registers (PWM0–PWM4)

	Address	bit 7	6	5	4	3	2	1	bit 0	Reset
PWM channel 0 (PWM0)	$0009	0	0							0000 0000
PWM channel 1 (PWM1)	$000A	0	0							0000 0000
PWM channel 2 (PWM2)	$000B	0	0							0000 0000
PWM channel 3 (PWM3)	$000C	0	0							0000 0000
PWM channel 4 (PWM4)	$000D	0	0							0000 0000

The PWM4 data register shares its address at $000D with the SCI baud register. Read or write access to the PWM4 register is gained by setting the SCIB bit in the PWM mode register to 0. On reset, this bit defaults to a 1 and the PWM rate is set at $f_{OP}/64$, corresponding to a maximum rate of 31.3 kHz.

A value of $00 loaded into the data registers results in a continuously low output, whereas a value of $20 results in a 50% duty cycle output and so on. The maximum value is $3F which corresponds to an output which is at 1 for 63/64 of the cycle. (See **Figure 49**.)

When the MCU makes a write to a PWM register, the new value will only be picked up by the D to A converters at the end of a conversion cycle. The counter is cleared to $00 on reset and the data latches are set to $00. Therefore, on power-on and on each subsequent reset of the device, the PWM channels resume their zero values.

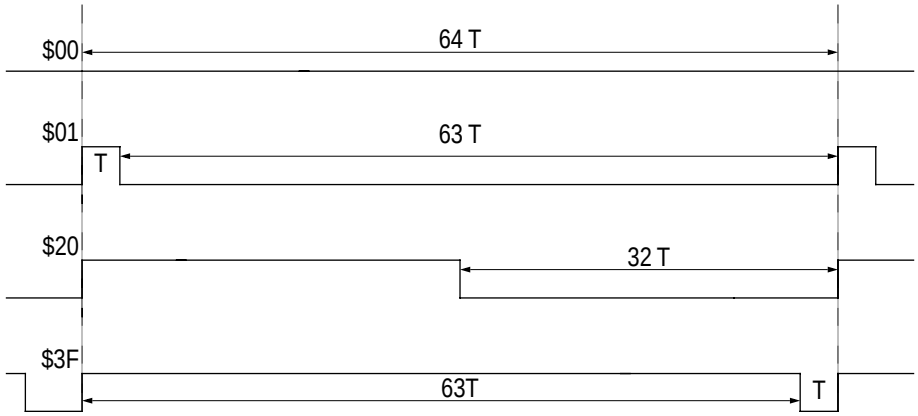


Figure 49. PWM output waveform examples.

Electrical Specifications

Contents

Introduction 103
Maximum ratings 103
Thermal characteristics and power considerations 103
DC electrical characteristics 106
Control timing 107

Introduction

This section contains the electrical specifications and associated timing information for the MC68HC05D9, MC68HC05D24, MC68HC05D32 and MC68HC705D32.

Maximum ratings

Thermal characteristics and power considerations

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

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

where:

T_A = Ambient temperature (°C)

Table 17. Maximum ratings

Rating	Symbol	Value	Unit
Supply voltage	V_{DD}	– 0.3 to +7.0	V
Input voltage (ports, OSC1)	V_{IN}	V_{SS} – 0.3 to V_{DD} + 0.3	V
Input voltage – $\overline{RESET}$, $\overline{IRQ}$	V_{IN}	V_{SS} – 0.3 to 2 x V_{DD} + 0.3	V
Input voltage – VPP – MC68HC705D32	V_{PP}	V_{SS} – 0.3 to 19	V
Operating temperature range	T_A	T_L to T_H –40 to +85 (all devices except MC68HC05D9) –40 to +105 (MC68HC05D9 only)	°C
Storage temperature range	T_{STG}	– 65 to +150	°C
Current drain per pin (note 2) – excluding V_{DD} and V_{SS}	I_D	25	mA

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

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

(3)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}.

θ_{JA} = Package thermal resistance, junction-to-ambient (°C/W)

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

P_{INT} = Internal chip power = $I_{DD} \bullet V_{DD}$ (W)

$P_{I/O}$ = Power dissipation on input and output pins (User determined)

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

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

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

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

where K is a constant for a particular part. K can be determined by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained for any value of T_A by solving the above equations. The package thermal characteristics are shown in **Table 18**.

Table 18. Package thermal characteristics

Characteristics	Symbol	Value	Unit
Thermal resistance – 40-pin plastic DIL package – 44-pin PLCC package	θ_{JA}	50	°C/W

Pins	R1	R2	C
PA0–7, PC0–7, PD0–7	3.26kΩ	2.38kΩ	50pF
PB0–7	520Ω	130Ω	50pF

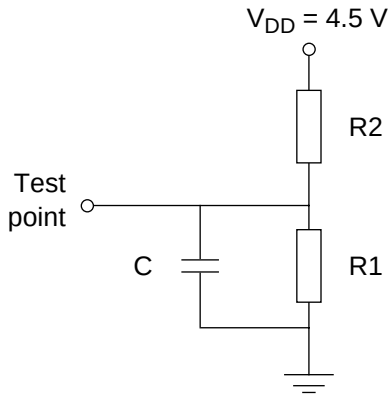


Figure 50. Equivalent test load

DC electrical characteristics

Table 19. DC electrical characteristics (5.0 V operation)

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

Characteristic	Symbol	Min	Typ	Max	Unit
Output voltage I_{LOAD} = –25 µA I_{LOAD} = +25 µA	 V_{OH} V_{OL}	 V_{DD} – 0.1 —	 — —	 — 0.1	 V V
Output high voltage (I_{LOAD} = – 0.8 mA) PA0–7, PB0–7, PC0–7, PD0–7	V_{OH}	V_{DD} – 0.8	—	—	V
Output low voltage (I_{LOAD} = +1.6 mA) PA0–7, PB0–7, PC0–7, PD0–7	V_{OL}	—	—	0.4	V
Output low voltage (I_{LOAD} = +25 mA) PB0–7	V_{OL}	—	—	1.0	V
Input high voltage PA0–7, PB0–7, PC0–7, PD0–7, OSC1, TCAP, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	$0.7V_{DD}$	—	—	V
Input low voltage PA0–7, PB0–7, PC0–7, PD0–7, OSC1, TCAP, $\overline{IRQ}$, $\overline{RESET}$	V_{IL}	V_{SS}	—	$0.2V_{DD}$	V
Supply current (see notes) RUN RUN (MC68HC05D32) WAIT STOP *STOP (MC68HC705D32)*	 I_{DD}	 — — — —	 5.5 — 1 2 —	 9.5 12 4 10 150	 mA mA mA µA *µA*
High-Z leakage current PA0–7, PB0–7, PC0–7, PD0–7	I_{IL}	—	—	±1	µA
Input current OSC1, TCAP, $\overline{IRQ}$, $\overline{RESET}$	I_{IN}	—	—	±1	µA
Total port B sink current to V_{SS}	I_{SS}	—	—	200	mA
Data retention mode voltage	V_{RM}	2.0	—	—	V
Capacitance Ports (as input or output) $\overline{IRQ}$, $\overline{RESET}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
MC68HC705D32 EPROM Programming voltage Programming current Programming time	 V_{PP} I_{PP} t_{PROG}	 14 — 4	 15 2 —	 16 — —	 V mA ms

Table 19. DC electrical characteristics (5.0 V operation)

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

Characteristic	Symbol	Min	Typ	Max	Unit

(1) All I_{DD} measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs (see **VDD and VSS**).

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

(3)RUN and WAIT IDD: measured using an external square-wave clock source (fOSC = 4MHz); all inputs 0.2V from rail; no DC loads; maximum load on outputs 50pF (except OSC2 load 20pF).

(4)WAIT IDD: only the timer system active; current varies linearly with the OSC2 capacitance.

(5)WAIT and STOP IDD: all ports configured as inputs; VIL = 0.2V and VIH = VDD – 0.2V.

(6)STOP IDD: measured with OSC1 = VDD.

Control timing

Table 20. DC electrical characteristics (3.3 V operation)

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

Characteristic	Symbol	Min	Typ	Max	Unit
Output voltage I_{LOAD} = –25 µA I_{LOAD} = +25 µA	 V_{OH} V_{OL}	 V_{DD} – 0.1 —	 — —	 — 0.1	 V V
Output high voltage (I_{LOAD} = – 0.8 mA) PA0–7, PB0–7, PC0–7, PD0–7	V_{OH}	V_{DD} – 0.3	—	—	V
Output low voltage (I_{LOAD} = +1.6 mA) PA0–7, PB0–7, PC0–7, PD0–7	V_{OL}	—	—	0.3	V
Output low voltage (I_{LOAD} = +25 mA) PB0–7	V_{OL}	—	—	0.5	V
Input high voltage PA0–7, PB0–7, PC0–7, PD0–7, OSC1, TCAP, $\overline{IRQ}$, $\overline{RESET}$	V_{IH}	$0.7V_{DD}$	—	—	V
Input low voltage PA0–7, PB0–7, PC0–7, PD0–7, OSC1, TCAP, $\overline{IRQ}$, $\overline{RESET}$	V_{IL}	V_{SS}	—	$0.2V_{DD}$	V
Supply current (see notes) RUN WAIT STOP	I_{DD}	 — — —		 3 1.4 5	 mA mA µA
High-Z leakage current PA0–7, PB0–7, PC0–7, PD0–7	I_{IL}	—	—	±1	µA
Input current OSC1, TCAP, $\overline{IRQ}$, $\overline{RESET}$	I_{IN}	—	—	±1	µA
Total port B sink current to V_{SS}	I_{SS}	—	—	200	mA
Data retention mode voltage	V_{RM}	2.0	—	—	V
Capacitance Ports (as input or output) $\overline{IRQ}$, $\overline{RESET}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
MC68HC705D32 EPROM Programming voltage Programming current Programming time	 VPP IPP t_{PROG}	 14 — 4	 15 2 —	 16 — —	 V mA ms

(1) All I_{DD} measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs (see **VDD and VSS**).

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

(3)RUN and WAIT IDD: measured using an external square-wave clock source (fOSC = 4MHz); all inputs 0.2V from rail; no DC loads; maximum load on outputs 50pF (except OSC2 load 20pF).

(4)WAIT IDD: only the timer system active; current varies linearly with the OSC2 capacitance.

(5)WAIT and STOP IDD: all ports configured as inputs; VIL = 0.2V and VIH = VDD – 0.2V.

(6)STOP IDD: measured with OSC1 = VDD.

Table 21. Control timing (5.0 V operation)

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

Characteristic	Symbol	Min	Max	Unit
Frequency of operation Crystal External clock	 f_{OSC} f_{OSC}	 — dc	 4.0 4.0	 MHz MHz
Internal operating frequency Crystal (f_{OSC} /2) External clock (f_{OSC} /2)	 f_{OP} f_{OP}	 — dc	 2.0 2.0	 MHz MHz
Processor cycle time	t_{CYC}	500	—	ns
Crystal oscillator start-up time	t_{OXOV}	—	100	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	8	—	t_{CYC}
Power-on reset delay	t_{PORL}	3968	3968	t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	125	—	ns
Interrupt pulse period	t_{ILIL}	(1)	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	100	—	ns

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

Table 22. Control timing (3.0 V operation)

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

Characteristic	Symbol	Min	Max	Unit
Frequency of operation Crystal External clock	 f_{OSC} f_{OSC}	 — dc	 2.0 2.0	 MHz MHz
Internal operating frequency Crystal (f_{OSC} /2) External clock (f_{OSC} /2)	 f_{OP} f_{OP}	 — dc	 1.0 1.0	 MHz MHz
Processor cycle time	t_{CYC}	1000	—	ns
Crystal oscillator start-up time	t_{OXOV}	—	100	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	8	—	t_{CYC}
Power-on reset delay	t_{PORL}	3968	3968	t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	250	—	ns
Interrupt pulse period	t_{ILIL}	(1)	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	200	—	ns

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

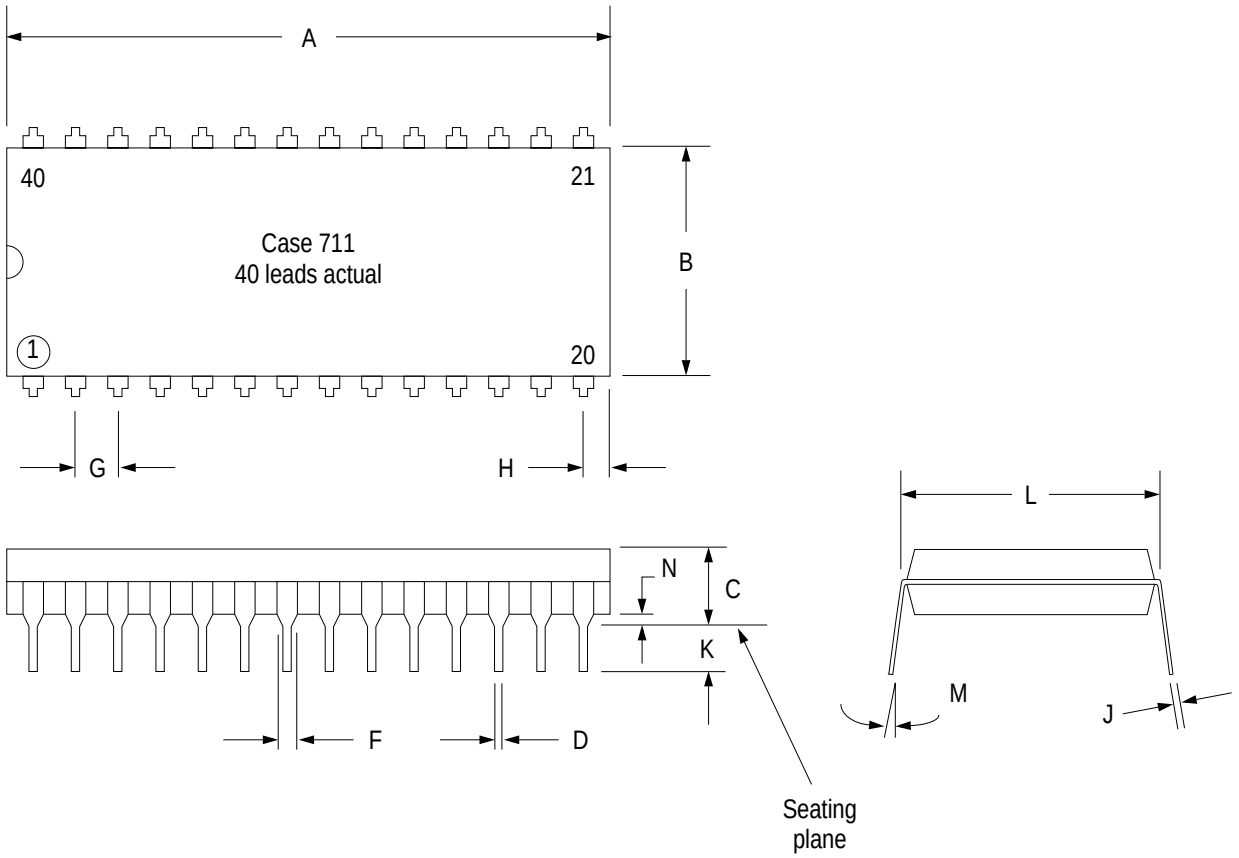
Electrical Specifications

Mechanical Data and Ordering Information

Contents

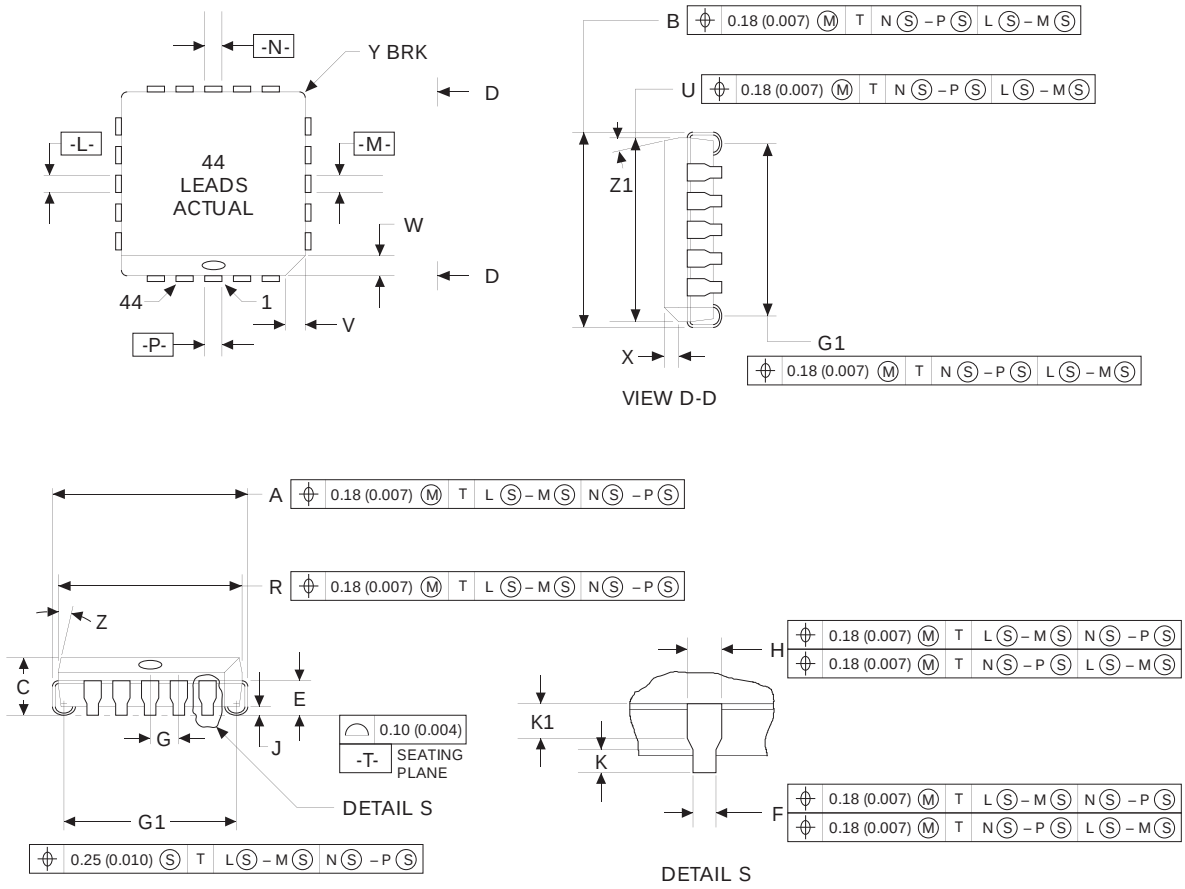
Mechanical Data 112
Ordering information 115
EPROMs 115
Verification media 115

Mechanical Data



Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	51.69	52.45	1.Due to space limitations, case no. 711 shall be represented by a general case outline, rather than one showing all the leads. 2.All dimensions in mm. 3.Positional tolerance of leads (‘D’) shall be within 0.25 mm at maximum material condition, in relation to seating plane and to each other. 4.Dimension ‘L’ is to centre of leads when formed parallel. 5.Dimension ‘B’ does not include mould protrusion.	H	1.65	2.16
B	13.72	14.22		J	0.20	0.38
C	3.94	5.08		K	2.92	3.43
D	0.36	0.56		L	15.24 BSC	
F	1.02	1.52		M	0°	15°
G	2.54 BSC			N	0.51	1.02

Figure 51. 40-pin PDIP mechanical dimensions



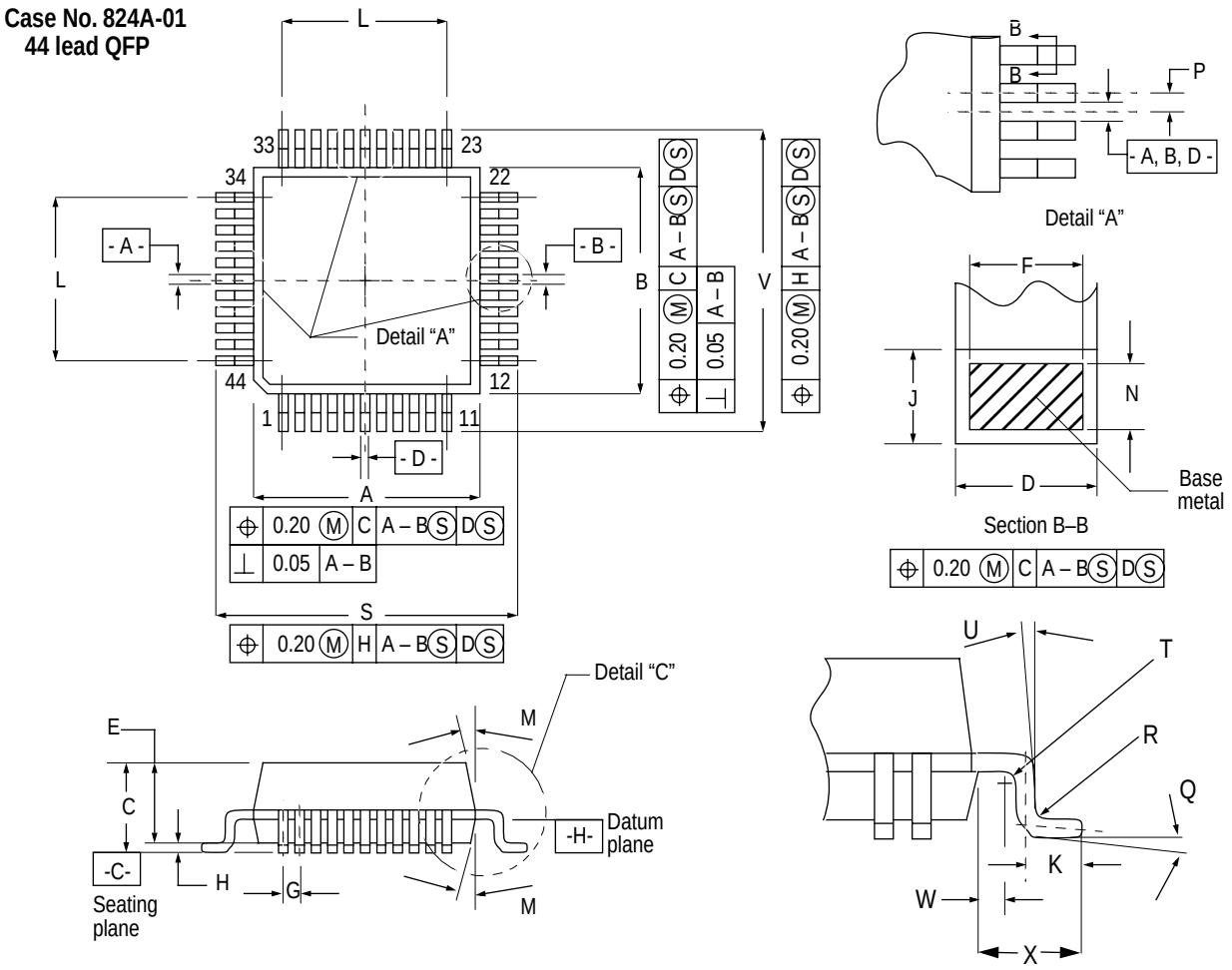
Dim	Millimeters		Inches		Dim	Millimeters		Inches	
	Min	Max	Min	Max		Min	Max	Min	Max
A	17.40	17.65	0.685	0.695	U	16.51	16.66	0.650	0.656
B	17.40	17.65	0.685	0.695	V	1.07	1.21	0.042	0.048
C	4.20	4.57	0.165	0.180	W	1.07	1.21	0.042	0.048
E	2.29	2.79	0.090	0.110	X	1.07	1.42	0.042	0.056
F	0.33	0.48	0.013	0.019	Y	—	0.50	—	0.020
G	1.27 BSC		0.050 BSC		Z	2°	10°	2°	10°
H	0.66	0.81	0.026	0.032	G1	15.50	16.00	0.610	0.630
J	0.51	—	0.020	—	K1	1.02	—	0.040	—
K	0.64	—	0.025	—	Z1	2°	10°	2°	10°
R	16.51	16.66	0.650	0.656					

1. Due to space limitation, case 777-02 shall be represented by a general (smaller) case outline drawing rather than showing all 44 leads.
2. Datums -L-, -M-, -N- and -P- determine where top of lead shoulder exits plastic body at mould parting line.
3. Dim G1: True position to be measured at datum -T-, seating plane.
4. Dim R and U do not include mould protrusion. Allowable mould protrusion is 0.25 (0.010) per side.
5. Dimensioning and tolerancing per ANSI Y14.5M, 1982.
6. Controlling dimension: inch.
7. 777-01 is obsolete; new standard 777-02.

Figure 52. 44-pin PLCC mechanical dimensions

Mechanical Data and Ordering Information

Case No. 824A-01
44 lead QFP



Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	9.90	10.10	1.Datum plane –H– is located at bottom of lead and is coincident with the lead where the lead exits the plastic body at the bottom of the parting line.	M	5°	10°
B	9.90	10.10		N	0.130	0.170
C	2.10	2.45		Q	0°	7°
D	0.30	0.45	2.Datums A–B and –D to be determined at datum plane –H–.	R	0.13	0.30
E	2.00	2.10	3.Dimensions S and V to be determined at seating plane –C–.	S	12.95	13.45
F	0.30	0.40	4.Dimensions A and B do not include mould protrusion. Allowable mould protrusion is 0.25mm per side. Dimensions A and B do include mould mismatch and are determined at datum plane –H–.	T	0.13	—
G	0.80 BSC			U	0°	—
H	—	0.250	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.	V	12.95	13.45
J	0.130	0.230		W	0.40	—
K	0.65	0.95		X	1.6 REF	
L	8.00 REF		6.Dimensions and tolerancing per ANSI Y 14.5M, 1982. 7.All dimensions in mm.			

Figure 53 44-pin QFP mechanical dimensions

Ordering information

This section describes the information needed to order the MCU.

To initiate a ROM pattern for the MCU, it is necessary first to contact your local field service office, local sales person or Freescale 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.

NOTE: *The MC68HC705D32 has no customer specific ROM, or options, and may therefore be ordered as a standard part.*

EPROMs

For the MC68HC05D9, a 16K byte EPROM programmed with the customer's software (positive logic for address and data) should be submitted for pattern generation. All unused bytes should be programmed to zeros. For the MC68HC05D24 and the MC68HC05D32, a 32K byte EPROM should be used.

NOTE: *The EPROM must be programmed such that the user's software is mapped exactly as it will appear in the masked ROM, e.g. the reset vector must be located at $3FFE and $3FFF in the EPROM submitted for an MC68HC05D32.*

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

Verification media

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

Mechanical Data and Ordering Information

Features Specific to the MC68HC05D24

Contents

Introduction 117
Memory map and registers 117
Programming model 117

Introduction

The MC68HC05D24 is similar to the MC68HC05D9, but with 24K bytes of masked ROM instead of 16K bytes. Apart from the extended ROM and the resulting changes to the memory map and register addresses, which are discussed in this appendix, the MC68HC05D24 is identical to the MC68HC05D9 (see **Figure 54**).

Memory map and registers

The memory map for this device is shown in **Figure 54**. The registers are shown in more detail in **Table 23**. Note that the address locations of the self-check ROM, the option register and the self-check and user vectors are $4000 higher than in the MC68HC05D9.

Programming model

The programming model of the MC68HC05D24 is identical to that of the MC68HC05D9, except for the program counter (see **Figure 55**). Only bit 15 of the program counter (PC) in the MC68HC05D24 is permanently

Features Specific to the MC68HC05D24

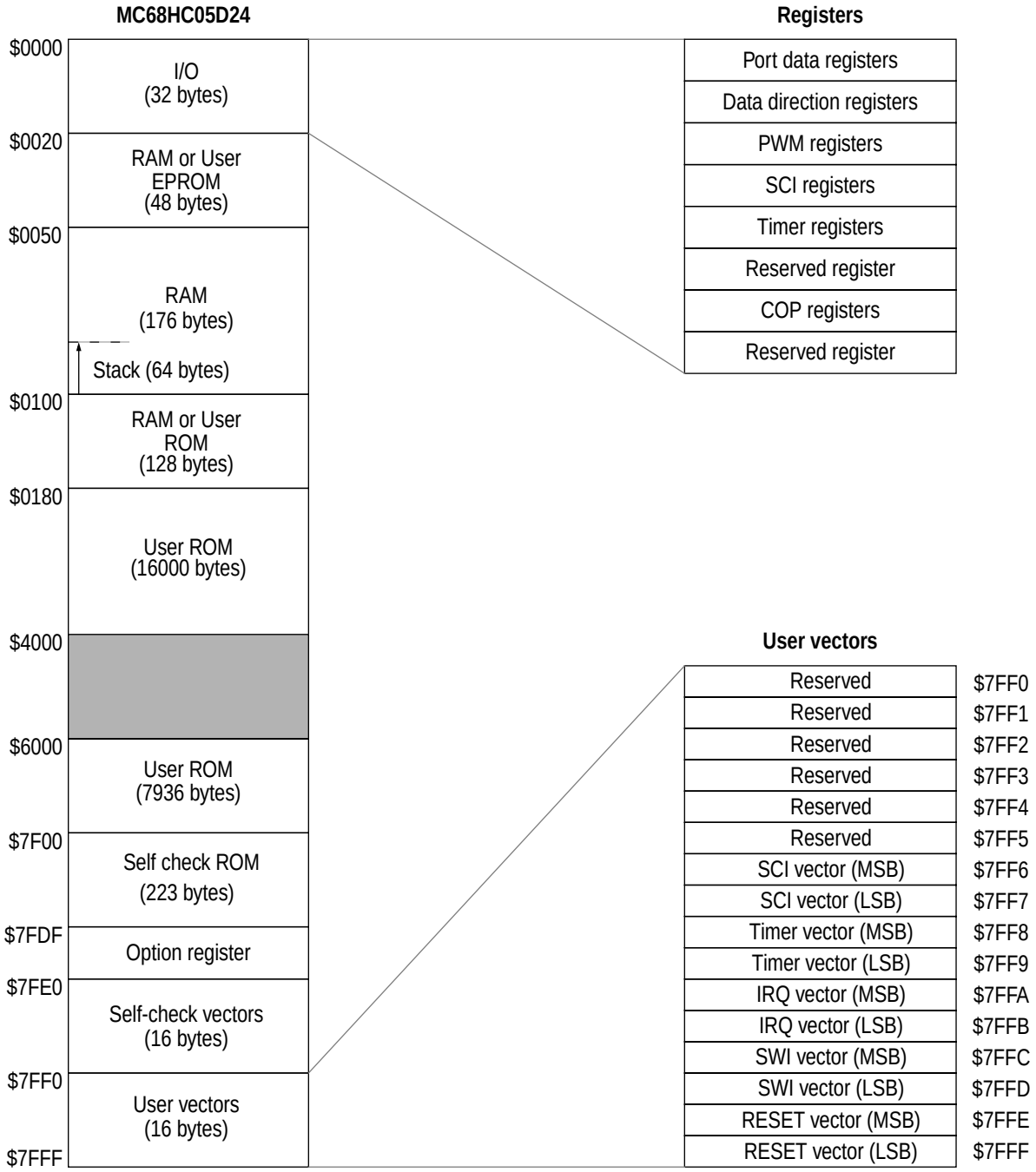


Figure 54. Memory map of the MC68HC05D24

Table 23. MC68HC05D24 register assignment

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000									undefined
Port B data (PORTB)	$0001									undefined
Port C data (PORTC)	$0002									undefined
Port D data (PORTD)	$0003									undefined
Data direction A (DDRA)	$0004									0000 0000
Data direction B (DDRB)	$0005									0000 0000
Data direction C (DDRC)	$0006									0000 0000
Data direction D (DDRD)	$0007									0u00 0000
PWM mode (PWMM)	$0008	0	0	SCIB	PWM4	PWM3	PWM2	PWM1	PWM0	0010 0000
PWM channel 0 (PWM0)	$0009	0	0							0000 0000
PWM channel 1 (PWM1)	$000A	0	0							0000 0000
PWM channel 2 (PWM2)	$000B	0	0							0000 0000
PWM channel 3 (PWM3)	$000C	0	0							0000 0000
PWM channel 4 (PWM4)	$000D (1)	0	0							0000 0000
SCI baud rate (BAUD)				SCP1	SCP0		SCR2	SCR1	SCR0	uu00 uuuu
SCI control 1 (SCCR1)	$000E	R8	T8	0	M	WAKE	0	0	0	uu0u u000
SCI control 2 (SCCR2)	$000F	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI status (SCSR)	$0010	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	1100 0000
SCI data (SCDR)	$0011									undefined
Timer control (TCR)	$0012	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	0000 0000
Timer status (TSR)	$0013	ICF	OCF	TOF	0	0	0	0	0	uuu0 0000
Input capture (ICR)	$0014									undefined
	$0015									undefined
Output compare (OCR)	$0016									undefined
	$0017									undefined
Timer counter (TCNT)	$0018									1111 1111
	$0019									1111 1100
Alternate counter (ALTCNT)	$001A									1111 1111
	$001B									1111 1100
Reserved	$001C									undefined
COP reset (COPRST)	$001D	ICAF	ICBF	OCAF	TOF	0	0	0	0	0000 0000
COP control (COPCR)	$001E	0	0	0	COPF	CME	COPE	CM1	CM0	0000 0000
Reserved	$001F									undefined

u = undefined

Table 23. MC68HC05D24 register assignment

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
OPTION	$7FDF	RAM0	RAM1	0	0	0		IRQ	0	0000 0u10

u = undefined

1. If SCIB = 0, the PWM4 register is available at location $000D
   If SCIB = 1, the SCI baud rate generator register is available at location $000D

set to zero, thus restricting the address range to $0000–$7FFF (32K bytes).

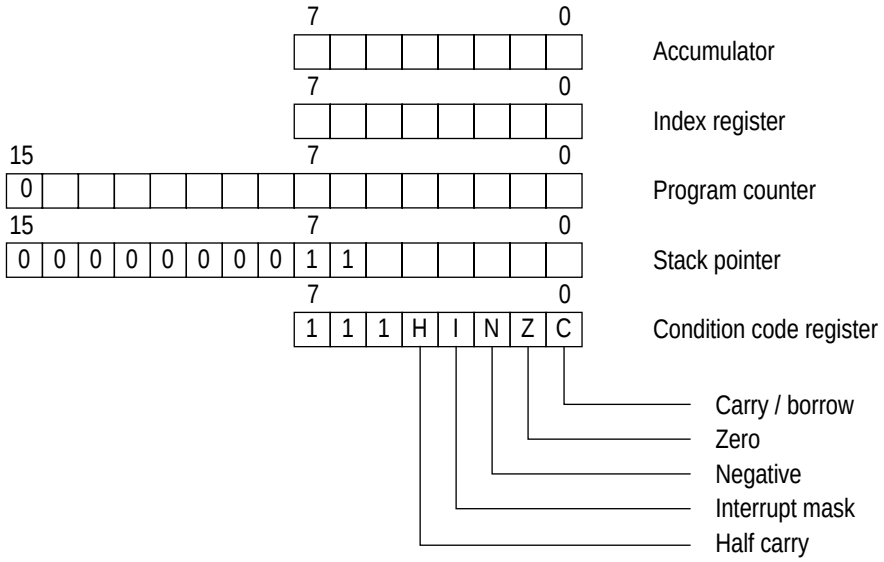


Figure 55. Programming model of the MC68HC05D24

Features Specific to the MC68HC05D32

Contents

Introduction 121
Memory map and registers 121
Programming model 121

Introduction

The MC68HC05D24 is similar to the MC68HC05D9, but with 32K bytes of masked ROM instead of 16K bytes. Apart from the extended ROM and the resulting changes to the memory map and register addresses, which are discussed in this appendix, the MC68HC05D24 is identical to the MC68HC05D9 (see **Figure 56**).

Memory map and registers

The memory map for this device is shown in **Figure 56**. The registers are shown in more detail in **Table 24**. Note that the address locations of the self-check ROM, the option register and the self-check and User vectors are $4000 higher than in the MC68HC05D9.

Programming model

The programming model of the MC68HC05D24 is identical to that of the MC68HC05D9, except for the program counter (see **Figure 57**). Only bit 15 of the program counter (PC) in the MC68HC05D24 is permanently

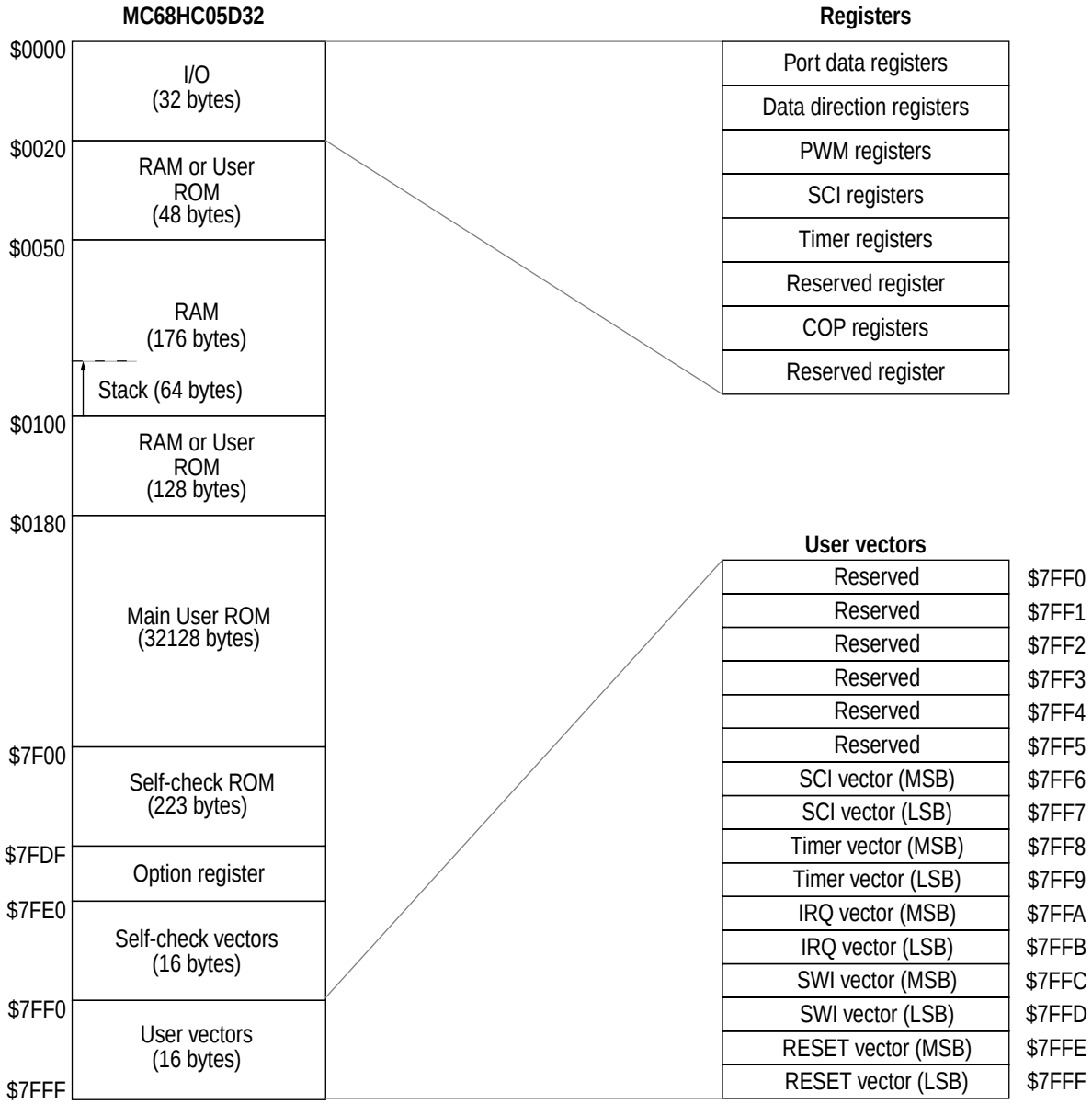


Figure 56. Memory map of the MC68HC05D24

set to zero, thus restricting the address range to $0000–$7FFF (32K bytes).

Table 24. MC68HC05D24 register assignment

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000									undefined
Port B data (PORTB)	$0001									undefined
Port C data (PORTC)	$0002									undefined
Port D data (PORTD)	$0003									undefined
Data direction A (DDRA)	$0004									0000 0000
Data direction B (DDRB)	$0005									0000 0000
Data direction C (DDRC)	$0006									0000 0000
Data direction D (DDRD)	$0007									0u00 0000
PWM mode (PWMM)	$0008	0	0	SCIB	PWM4	PWM3	PWM2	PWM1	PWM0	0010 0000
PWM channel 0 (PWM0)	$0009	0	0							0000 0000
PWM channel 1 (PWM1)	$000A	0	0							0000 0000
PWM channel 2 (PWM2)	$000B	0	0							0000 0000
PWM channel 3 (PWM3)	$000C	0	0							0000 0000
PWM channel 4 (PWM4)	$000D(1)	0	0							0000 0000
SCI baud rate (BAUD)				SCP1	SCP0		SCR2	SCR1	SCR0	uu00 uuuu
SCI control 1 (SCCR1)	$000E	R8	T8	0	M	WAKE	0	0	0	uu0u u000
SCI control 2 (SCCR2)	$000F	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI status (SCSR)	$0010	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	1100 0000
SCI data (SCDR)	$0011									undefined
Timer control (TCR)	$0012	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	0000 0000
Timer status (TSR)	$0013	ICF	OCF	TOF	0	0	0	0	0	uuu0 0000
Input capture (ICR)	$0014									undefined
	$0015									undefined
Output compare (OCR)	$0016									undefined
	$0017									undefined
Timer counter (TCNT)	$0018									1111 1111
	$0019									1111 1100
Alternate counter (ALTCNT)	$001A									1111 1111
	$001B									1111 1100
Reserved	$001C									undefined
COP reset (COPRST)	$001D	ICAF	ICBF	OCAF	TOF	0	0	0	0	0000 0000
COP control (COPCR)	$001E	0	0	0	COPF	CME	COPE	CM1	CM0	0000 0000
Reserved	$001F									undefined

u = undefined

Table 24. MC68HC05D24 register assignment

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
OPTION	$7FDF	RAM0	RAM1	0	0	0		IRQ	0	0000 0u10

u = undefined

1. If SCIB = 0, the PWM4 register is available at location $000D
   If SCIB = 1, the SCI baud rate generator register is available at location $000D

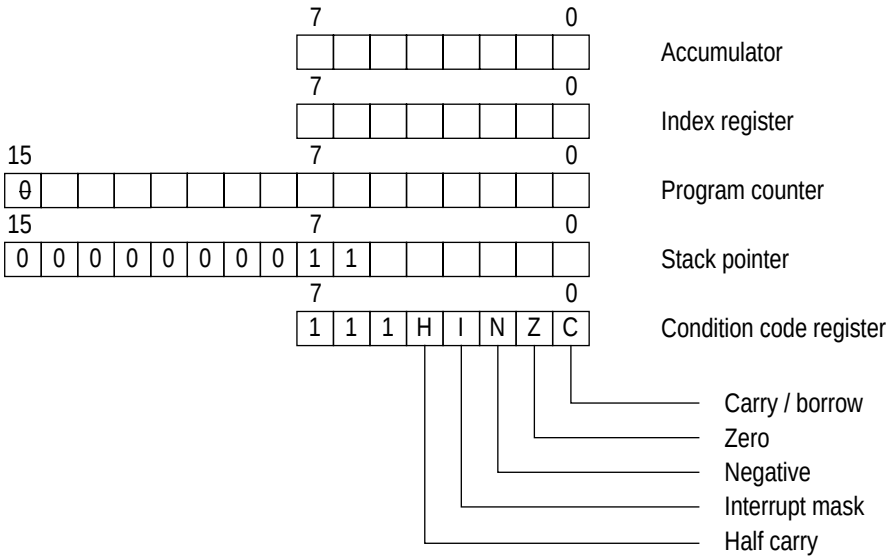


Figure 57. Programming model of the MC68HC05D24

Features Specific to the MC68HC705D32

Contents

Introduction 125
Operating mode selection 125
Pin descriptions 126
VPP 126
OSC1/OSC2 127
Crystal 127
External clock 127
Memory 127
MC68HC705D32 EPROM description 130
Programming sequence 130
EPROM security 130
Program control register (PCR) 131
EPROM mask option register (MOR) 131
Bootloader mode 132
Bootloader functions 132

Introduction

The MC68HC705D32 is similar to the MC68HC05D32, but with 32K bytes of UV-erasable EPROM instead of masked ROM. The MC68HC705D32 is identical to the MC68HC05D32 (see **Figure 58**) apart from the EPROM replacing ROM.

Operating mode selection

Single chip mode on the MC68HC705D32 is identical to single chip mode on the MC68HC05D32. However, on the MC68HC705D32 a bootloader ROM replaces the self-check ROM on the MC68HC05D32.

The bootloader mode provides for self-programming of the EPROM array and allows software to be loaded into, and run from, the on-board RAM. The voltage level on the $\overline{IRQ}$ pin determines whether single chip mode or bootloader mode is selected (see **Table 25**).

NOTE: *The TCAP pin must be tied to V_{DD} to ensure correct selection of the bootloader mode. Failure to do so could result in unpredictable operation.*

CAUTION: *For the MC68HC705D32, all vectors are fetched from the EPROM (locations $7FF6–$7FFF) in single chip mode; therefore, the EPROM must be programmed (via the bootloader mode) before the device is powered up in single chip mode.*

Table 25. Operating mode entry conditions

IRQ	RESET	TCAP	Mode
V_{SS} to V_{DD}		Don’t care	Single chip
2 V_{DD}		V_{DD}	Bootloader

Note: The voltage level on the $\overline{IRQ}$ pin should be maintained for at least 7x t_{cyc} after the rising edge of the reset signal to guarantee proper mode selection.

Pin descriptions

VPP

Programming power is supplied to the EPROM array on the MC68HC705D32 via this pin. The nominal programming voltage is 15 Volts. The voltage level on the VPP pin should never be allowed to fall below V_{DD}.

OSC1/OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal or an external clock signal connected to these pins provides the oscillator clock. The oscillator frequency is divided by 2 to provide the internal bus frequency.

Crystal

The circuit shown in **Figure 5**(b) is recommended when using a crystal. 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 **Control timing**). Use of an external CMOS oscillator is recommended when crystals outside the specified ranges are to be used. The crystal and components should be mounted as close as possible to the input pins to minimise output distortion and start-up stabilizing time.

External clock

An external clock should be applied to the OSC1 input with the OSC2 pin not connected, as shown in **Figure 5**(d). The t_{OXOV} specification 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}.

NOTE: *A ceramic resonator should not be used with the MC68HC705D32.*

Memory

The memory map for this device is shown in **Table 25**. The registers are shown in more detail in **Table 26**. Note that the MC68HC705D32 has an additional register to control the programming of the EPROM array.The circuit diagram for this self-programming mode is shown in **Figure 58**.

Features Specific to the MC68HC705D32

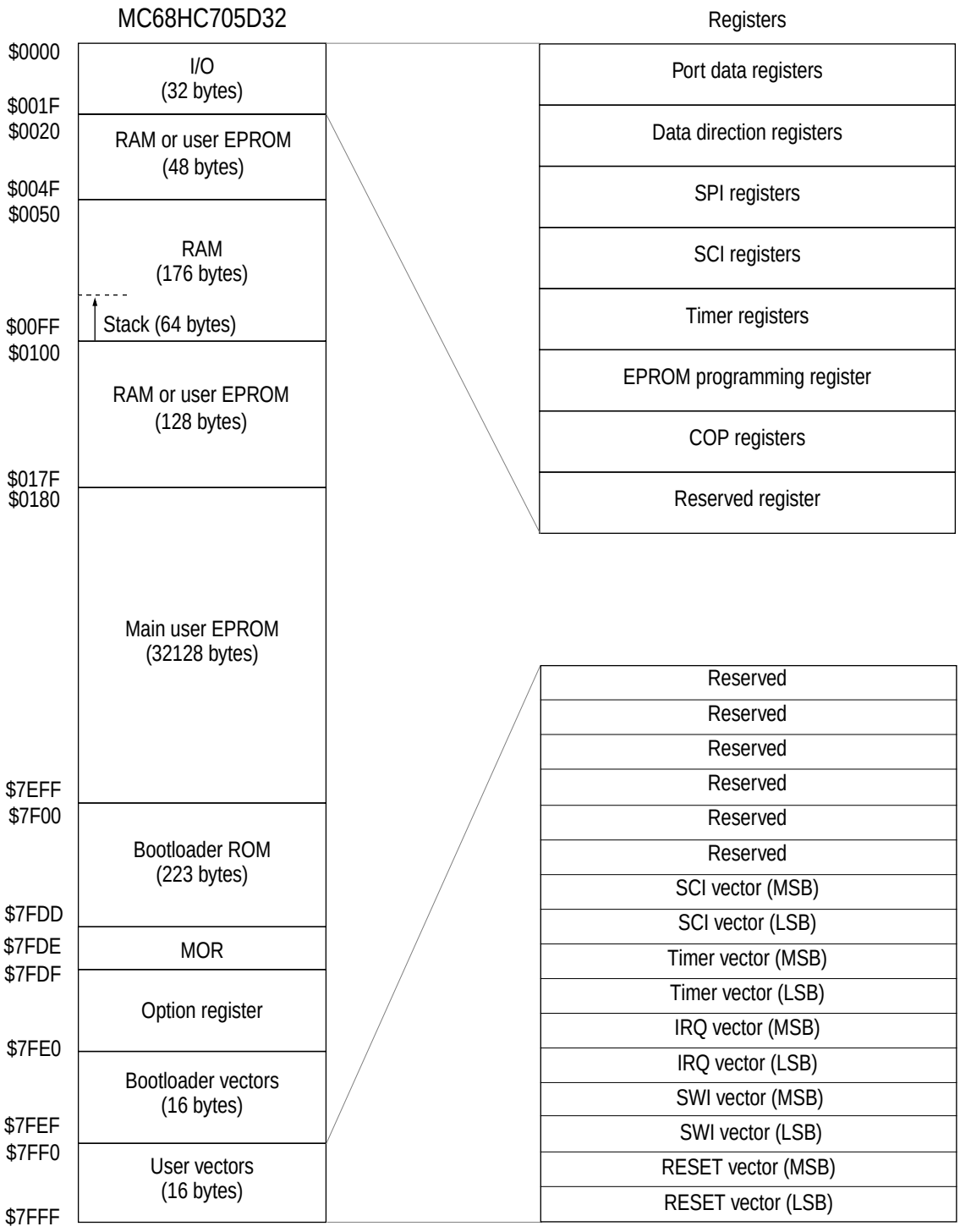


Figure 58. Memory map of the MC68HC705D32

Table 26. MC68HC705D32 register assignment

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									Unaffected
Port B data (PORTB)	$0001									Unaffected
Port C data (PORTC)	$0002									Unaffected
Port D data (PORTD)	$0003									Unaffected
Data direction A (DDRA)	$0004									0000 0000
Data direction B (DDRB)	$0005									0000 0000
Data direction C (DDRC)	$0006									0000 0000
Data direction D (DDRD)	$0007									0u00 0000
Unused	$0008									
Unused	$0009									
SPCR SPI control register	$000A	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	0000 uuuu
SPSR SPI status register	$000B	SPIF	WCOL		MODF					0000 uuuu
SPDR SPI data register	$000C	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	Unaffected
SCI baud rate (BAUD)	$000D*	—	—	SCP1	SCP0	—	SCR2	SCR1	SCR0	uu00 uuuu
SCI control 1 (SCCR1)	$000E	R8	T8	0	M	WAKE	0	0	0	uu0u u000
SCI control 2 (SCCR2)	$000F	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI status (SCSR)	$0010	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	1100 0000
SCI data (SCDR)	$0011									Unaffected
Timer control (TCR)	$0012	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	0000 0000
Timer status (TSR)	$0013	ICF	OCF	TOF	0	0	0	0	0	uuu0 0000
Input capture (ICR)	$0014	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	Unaffected
	$0015	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Unaffected
Output compare (OCR)	$0016	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	Unaffected
	$0017	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Unaffected
Timer counter (TCNT)	$0018	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	1111 1111
	$0019	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	1111 1100
Alternate counter (ALTCNT)	$001A	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	1111 1111
	$001B	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	1111 1100
EPROM programming register	$001C						LATCH		EPGM	0000 0000
COP reset (COPRST)	$001D	ICAF	ICBF	OCAF	TOF	0	0	0	0	0000 0000
COP control (COPCR)	$001E	0	0	0	COPF	CME	COPE	CM1	CM0	0000 0000
Reserved	$001F									
OPTION	$7FDF	RAM0	RAM1	0	0	0	—	IRQ	0	0000 0u10

Note: If SCIE = 0, the PWM4 register is available at location $000D.
If SCIE = 1, the SCI Baud Rate Generator register is available at this location.

u = undefined

MC68HC705D32 EPROM description

The 32k byte EPROM is positioned at locations $0020 to $004F and $0100 to $7EFF, with 16 bytes of the EPROM located at $7FF0 to $7FFF for user vectors. Location $7FDE is reserved for the mask option register. The erased state of EPROM reads as $00 and EPROM power is supplied by the VPP pin and the VDD pin.

The program control register (PCR) is provided for EPROM programming and testing.

Access to the EPROM in bootloader and test mode is controlled by a security bit in the mask option register called SEC. For further information on EPROM security, see **EPROM security**.

Programming sequence

The sequence includes:

- Setting the ELAT bit
- Writing the data to the address to be programmed
- Setting the PGM bit
- Delaying for an appropriate amount of time
- Clearing the ELAT and PGM bit

It is important to remember that an external programming voltage must be applied to the VPP pin while programming, but it should be equal to V_{DD} during normal operations.

EPROM security

A security feature has been incorporated into the MC68HC705D32 to prevent externally accessing the contents of the EPROM in any non-user mode of operation. This feature, once enabled, can only be disabled by completely erasing the EPROM.

NOTE: *For OTP (plastic) packaged parts, once the security feature has been enabled it cannot be disabled*

Program control register (PCR)

The program control register is provided for EPROM programming in BOOT modes. This register is available only in the MC68HC705D32 (EPROM device).

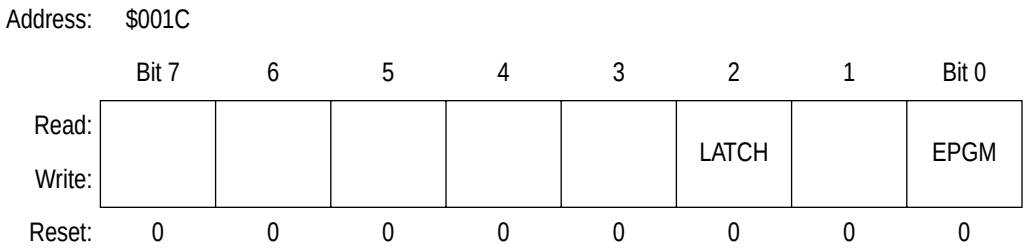


Figure 59. Program Control Register (PCR)

LATCH — EPROM latch control

1 = EPROM address and data bus configured for programming (writes to EPROM cause address and data to be latched). EPROM is in programming mode and cannot be read if LATCH is set. This bit should not be set if no programming voltage is being applied to the VPP pin. Reset clears this bit.

0 = EPROM address and data bus configured for normal reads.

EPGM — EPROM program command

1 = Programming power is switched ON to EPROM array. This bit can be set only if the LATCH bit has been previously set. Reset clears this bit.

0 = Programming power is switched OFF from EPROM array.

EPROM mask option register (MOR)

The option register MOR is implemented as EPROM bits in the main EPROM array at address $7DFE.

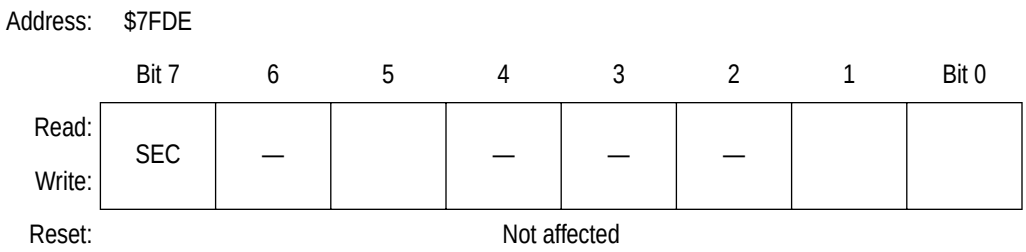


Figure 60. EPROM Mask Option Register (MOR)(1)

1. This register is implemented in EPROM, therefore reset has no effect on the state of the individual bits.

SEC — Security bit

This bit is used to control the security of the EPROM code when the device is in test or bootloader mode.

1 = EPROM contents not available in bootloader/test mode

0 = EPROM contents available in bootloader/test mode

Bootloader mode

Bootloader mode is entered upon the rising edge of $\overline{\text{RESET}}$ if the VPP pin is at V_{TST} and the TCAP pin is at logic one. The bootloader code and vectors reside in the ROM from $7F00 to $7FEF. This program handles copying of user code from an external EPROM into the on-chip EPROM. The bootloader function has to be done from an external EPROM. The bootloader performs one programming pass at 2 ms per byte then does a verify pass.

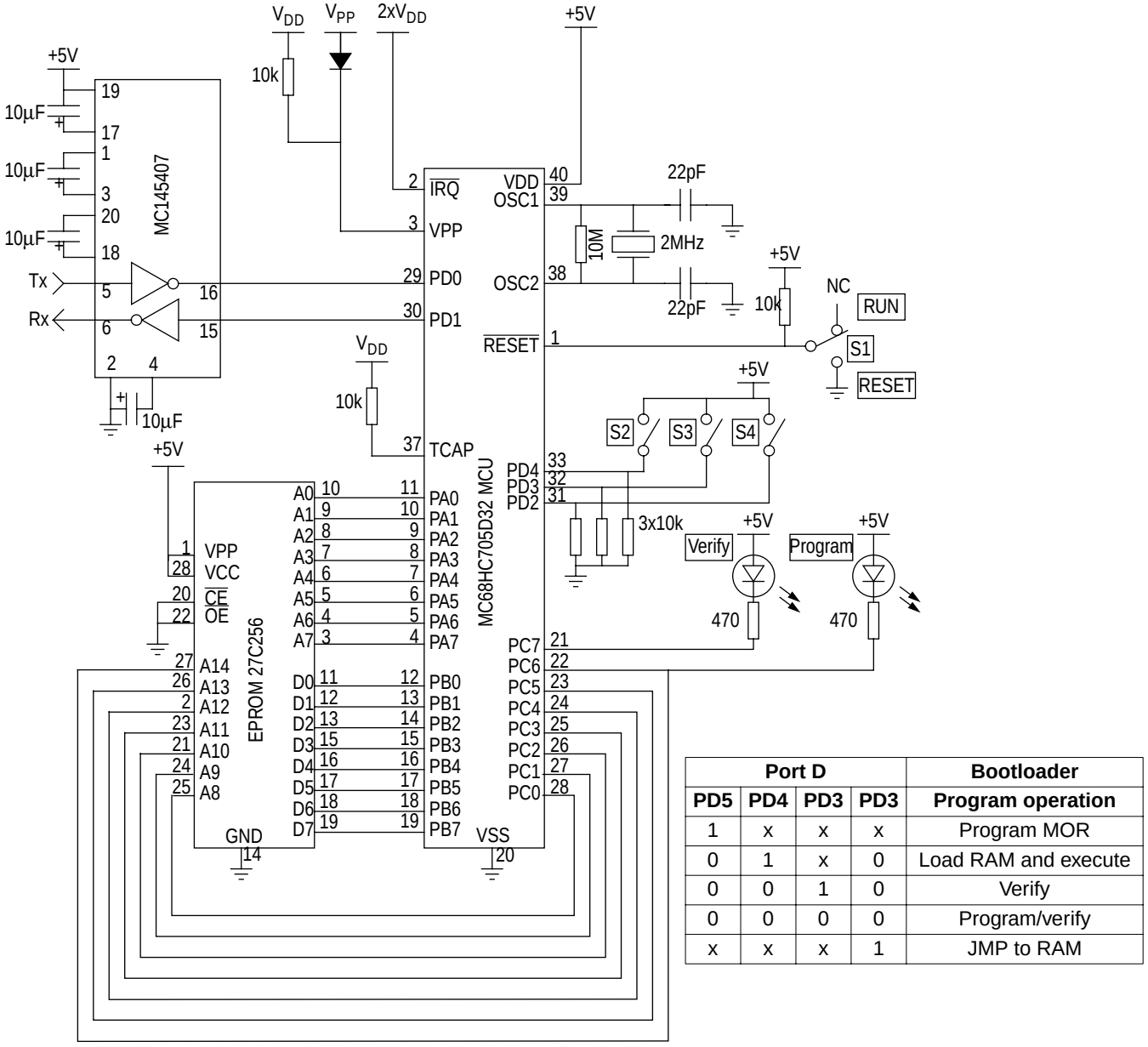
The user code must be a one-to-one correspondence with the internal EPROM addresses.The designer MUST disable the COP hardware in bootloader mode. In the erase state the COP is disabled.

Bootloader functions

Table 27. Bootloader functions

Port D				Bootloader
Pin 5	**Pin 4**	**Pin 3**	**Pin 3**	**Program operation**
1	x	x	x	Program MOR
0	1	x	0	Load RAM and execute
0	0	1	0	Verify
0	0	0	0	Program/verify
x	x	x	1	JMP to RAM

The EPROM must be erased before performing a program cycle



Port D				Bootloader
PD5	PD4	PD3	PD3	Program operation
1	x	x	x	Program MOR
0	1	x	0	Load RAM and execute
0	0	1	0	Verify
0	0	0	0	Program/verify
x	x	x	1	JMP to RAM

Figure 61. Self-programming via the bootstrap loader (40-pin PDIP)

Features Specific to the MC68HC705D32

Glossary

A — See “accumulator (A).”

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

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

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

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

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

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

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

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

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

binary — Relating to the base 2 number system.

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

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

234 (decimal) = 0010 0011 0100 (BCD)

Glossary

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

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

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

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

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

bus — A set of wires that transfers logic signals.

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

byte — A set of eight bits.

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

CCR — See “condition code register.”

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

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

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

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

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

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

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

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

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

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

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

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

CPU — See “central processor unit (CPU).”

CPU08 — The central processor unit of the M68HC08 Family.

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

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

Glossary

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

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

CSIC — customer-specified integrated circuit

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

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

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

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

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

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

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

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

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

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

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

firmware — Instructions and data programmed into nonvolatile memory.

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

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

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

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

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

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

illegal address — An address not within the memory map

illegal opcode — A nonexistent opcode.

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

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

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

Glossary

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

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

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

I/O — See “input/output (I/0).”

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

jitter — Short-term signal instability.

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

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

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

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

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

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

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

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

M68HC08 — A Freescale family of 8-bit MCUs.

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

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

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

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

MCU — Microcontroller unit. See “microcontroller.”

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

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

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

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

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

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

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

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

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

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

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

Glossary

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

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

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

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

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

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

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

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

PC — See “program counter (PC).”

peripheral — A circuit not under direct CPU control.

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

PLL — See "phase-locked loop (PLL)."

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

register — A circuit that stores a group of bits.

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

reset — To force a device to a known condition.

Glossary

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

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

serial — Pertaining to sequential transmission over a single line.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

VCO — See "voltage-controlled oscillator."

Glossary

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

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

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

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

word — A set of two bytes (16 bits).

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

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

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

Index

Numerics

16-bit timer interrupt 49
40-pin PDIP
  mechanical dimensions 112
  pinout 18
44-pin PLCC
  mechanical dimensions 113
  pinout 18

A

accumulator 24
addressing modes 28
ALU — arithmetic/logic unit 27

B

BAUD – baud rate register 96
  SCP1, SCP0 – serial prescaler select bits 96
  SCR2, SCR1, SCR0 – SCI rate select bits 97
bit manipulation instructions 35
block diagrams
  40-pin PDIP pinout 18
  44-pin PLCC pinout 18
  MC68HC05D9 11
  PWM 100
bootloader
  functions 132
  mode 132
  self-programming 133

C

carry/borrow flag 27
C-bit in CCR 27
ceramic resonator 19
clock monitor reset 45
condition code register (CCR) 26
control instructions 35
COP 45
COPCR – COP control register 46
  CM1, CM0 – COP mode bits 46
  CME – clock monitor enable 46
  COPE – COP enable 46
  COPF – COP failure 46
CPU
  programming model 120, 124
crystal 19, 126

D

direct addressing mode 29

E

ELAT bit in PCR 131
EPGM bit in PCR 131
EPROM
  MOR – EPROM mask option register 131
  PCR – program control register 130
  programming sequence 130
  test features 130
EXCOL, EXROW bits in PCR 131
extended addressing mode 29

Index

F

features
- MC68HC05D9 10

flowcharts
- interrupt 51
- STOP and WAIT 52

H

half-carry flag 26
H-bit in CCR 26

I

I/O port structure 58
immediate addressing mode 29
index register 25
indexed addressing mode 29
inherent addressing mode 28
Input 57
instruction types 31
interrupt priorities 48
interrupts 47
- interrupt flowchart 51
- interrupt priorities 48
- maskable hardware interrupts 48
- SWI – non-maskable software interrupt . 48

IRQ 21
IRQ – external interrupt 49

J

jump/branch instructions 33

L

literature distribution centers 151
low power modes 50
- data retention mode 50
- STOP 50
- WAIT 50

M

maskable hardware interrupts 48
- 16-bit timer interrupt 49
- IRQ – external interrupt 49
- SCI interrupt 49

MC68HC05D24
- block diagram 11
- memory map 118
- register assignment 119

MC68HC05D32
- block diagram 11
- memory map 122
- register assignment 123

MC68HC05D9
- block diagram 11

MC68HC705D32
- block diagram 11
- memory map 128
- register assignment 129

MC68HC705D9
- block diagram 11

memory
- EPROM 130–131
- memory map 55
- RAM 53
- ROM 54

memory map
- MC68HC05D24 118
- MC68HC05D32 122
- MC68HC705D32 128

modes of operation
- self-check 14
- single chip 14

N

N-bit in CCR 27
negative flag 27

O

OCF 69
opcode map 42
OPTION — option register
  IRQ 16
  RAM0 16
  RAM1 16
ordering information
  literature distribution centers 151
  Mfax 152
  Web server 152
  Web site 152
OSC1, OSC2 19
oscillator connections 20

P

PA0–PA7 21
packages
  QFP 114
PB0–PB7 21
PC0–PC7 22
PCR
  ELAT – EPROM latch control 131
  EPGM – EPROM program command . 131
  EXCOL, EXROW 131
PD0–PD7 22
pins
  IRQ 21
  OSC1, OSC2 . . . . . 19, 126
  PA0–PA7 21
  PB0–PB7 21
  PC0–PC7 22
  PD0–PD7 22
  RESET 21
  TCAP 21
  VDD, VSS 14, 19
  VPP 14, 19, 126
port registers
  data direction registers 61
  port data registers 60
ports
  I/O port structure 58
  I/O programming 57
  port D 59
  ports A, B and C 59
power-on reset 43, 53
program counter 26
programmable timer
  block diagram 64
  counter 65
  during STOP mode 73
  during WAIT mode 72
  input capture register 69–70
  output compare register 71
  TCR – timer control register 67
  timer state diagrams 73–78
  TSR – timer status register 68
programming model 24
PWM
  block diagram 100
  counter 99
  output waveforms 102
  PWM0–PWM4 – PWM channel data registers 102
  PWMM – PWM mode register 100
PWMM
  PWM0–4 – PWM channel enable bits . 100
  SCIB – serial communications interface baud 100

Index

R

RAM 53
read-modify-write instructions 32
register assignment 56
register summary 129
register/memory instructions 31
relative addressing mode 30
RESET 21
RESET pin 44
resets 43, 53
  clock monitor reset 45
  COP 45
  power-on reset 43, 53
  RESET pin 44
  reset timing diagram 44
  timing diagram 44
ROM 54

S

SCCR1
  M – mode 90
  R8 – receive data bit 8 90
  T8 – transmit data bit 8 90
  WAKE – wake-up mode select 91
SCCR2
  ILIE – idle line interrupt enable 92
  RE – receiver enable 92
  RIE – receiver interrupt enable 92
  RWU – receiver wake up 92
  SBK – send break 92
  TCIE – transmit complete interrupt enable 92
  TE – transmitter enable 92
  TIE – transmit interrupt enable 92
SCI 80
  BAUD – baud rate register 96
  block diagram 82
  data format 83
  RDI – receive data 85
  receiver 80
  registers 89–97
  SCCR1 – serial communications control register1 90
  SCCR2 – serial communications control register2 91
  SCDR – serial communications data register 90
  SCSR – serial communications status register 94
  start bit detection 86
  TDO – transmit data 89
  transmitter 80
  two-wire system 80
SCI interrupt 49
SCSR – serial communications status register
  FE – framing error flag 95
  IDLE – idle line detected flag 94
  NF – noise error flag 95
  OR – overrun error flag 95
  RDRF - receive data register full flag . . 94
  TC – transmission complete flag 94
  TDRE – transmit data register empty flag . 94
self-check mode 14
single chip mode 14
stack pointer 25
start bit detection 86

T

TCAP 21
TCR – timer control register
  ICIE – input capture interrupt enable . . . 67
  IEDG – input edge 68
  OCIE – output compare interrupt enable . . 67
  OLVL – output level 68
  TOIE – timer overflow interrupt enable . 67
TSR – timer status register
  ICF – input capture flag 69
  TOF – timer overflow flag 69

V

VDD, VSS 14
VPP 14, 19
VSS, VSS 19

W

Web server 152
Web site 152

Z

Z-bit in CCR. 27
zero flag. 27

Index

Literature Updates

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

Literature Distribution Centers

Order literature by mail or phone.

USA/Europe

Freescale Literature Distribution
P.O. Box 5405
Denver, Colorado, 80217
Phone 1-303-675-2140

US & Canada only

http://freescale.com

Japan

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

MC68HC05D9 Rev 3.0

Literature Updates

Hong Kong

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

Customer Focus Center

1-800-521-6274

:

World Marketing World Wide Web Server

Use the Internet to access Freescale's World Wide Web server. Use the following URL: www.freescale.com

Microcontroller Division's Web Site

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

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

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

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

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

freescale™ semiconductor

MC68HC05D9/D

Freescale Semiconductor, Inc.