MC68HC05D9/D
Rev. 2

MC68HC05D9

TECHNICAL DATA

HC05

MC68HC05D9
MC68HC05D24
MC68HC05D32
MC68HC705D32

TECHNICAL
DATA

MOTOROLA

MOTOROLA

MC68HC05D9

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

Conventions

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

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

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

TABLE OF CONTENTS

3
CPU CORE AND INSTRUCTION SET

4
RESETS, INTERRUPTS AND LOW POWER MODES

A
FEATURES SPECIFIC TO THE MC68HC05D24

B
FEATURES SPECIFIC TO THE MC68HC05D32

C
FEATURES SPECIFIC TO THE MC68HC705D32

THIS PAGE LEFT BLANK INTENTIONALLY

LIST OF FIGURES

LIST OF TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1
INTRODUCTION

1.1 General description

The MC68HC05D9, with 16K bytes of masked ROM, is a member of the Motorola 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.

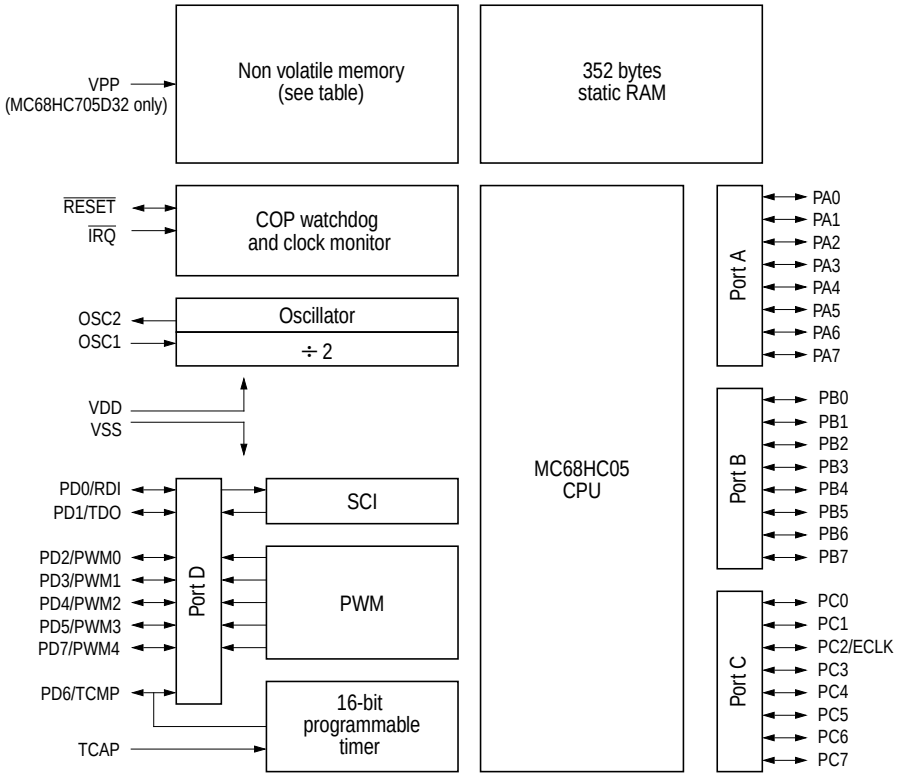
Devices similar to the MC68HC05D9 are descibed in a set of appendices at the back of this book.

Table 1-1 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

1.2 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 1-1 Functional block diagram

THIS PAGE LEFT BLANK INTENTIONALLY

2
OPERATING MODES AND PIN DESCRIPTIONS

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 $\overline{IRQ}$ pin when the device is brought out of reset (see Table 2-1).

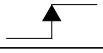
2.1 Single chip mode

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 $\overline{RESET}$ if the voltage level on the $\overline{IRQ}$ pin is within the normal operating range.

Table 2-1 Operating mode entry conditions

$\overline{IRQ}$(1)	$\overline{RESET}$	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 $\overline{IRQ}$ pin should be maintained for at least 7 x t_{CYC} after the rising edge of the reset signal to guarantee proper mode selection.

2.2 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 2-1). To enter self-check mode, a voltage equal to 2x V_{DD} must be present on the $\overline{IRQ}$ 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.

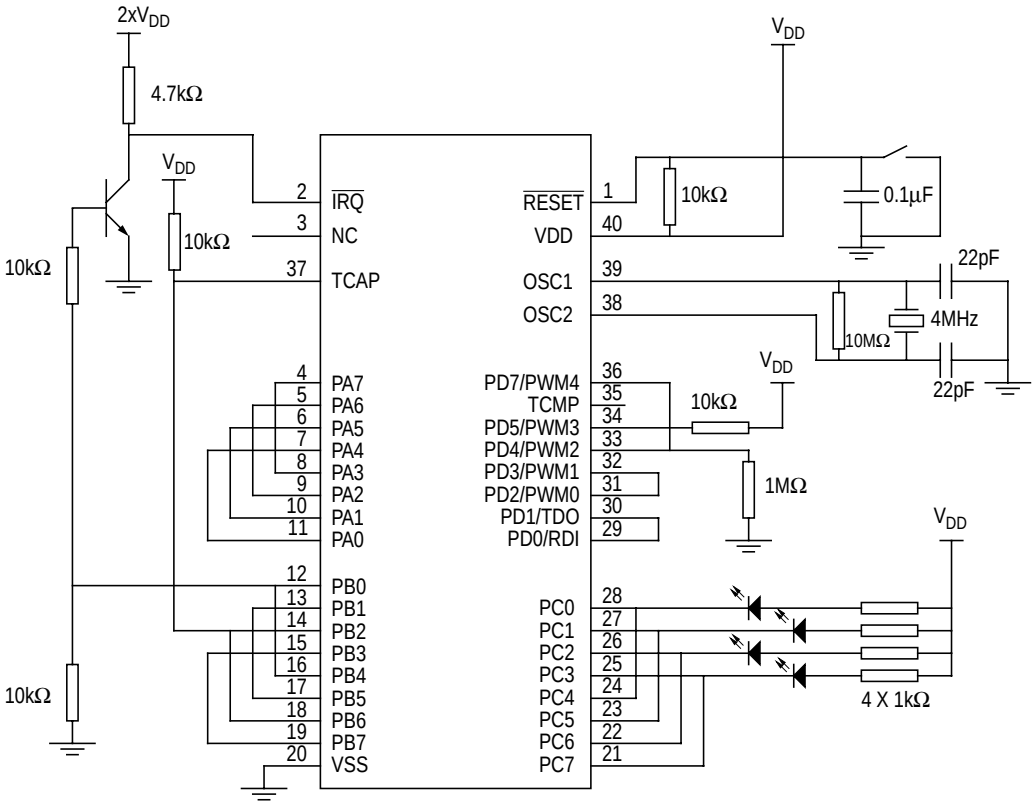


Figure 2-1 Self-check circuit diagram (for 40-pin DIL package)

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 2-2.

Table 2-2 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
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 $\overline{IRQ}$
Flashing				Good device
All other				Bad device

'0' denotes LED on; '1' denotes LED off

2.3 Pin descriptions

Figure 2-2 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).

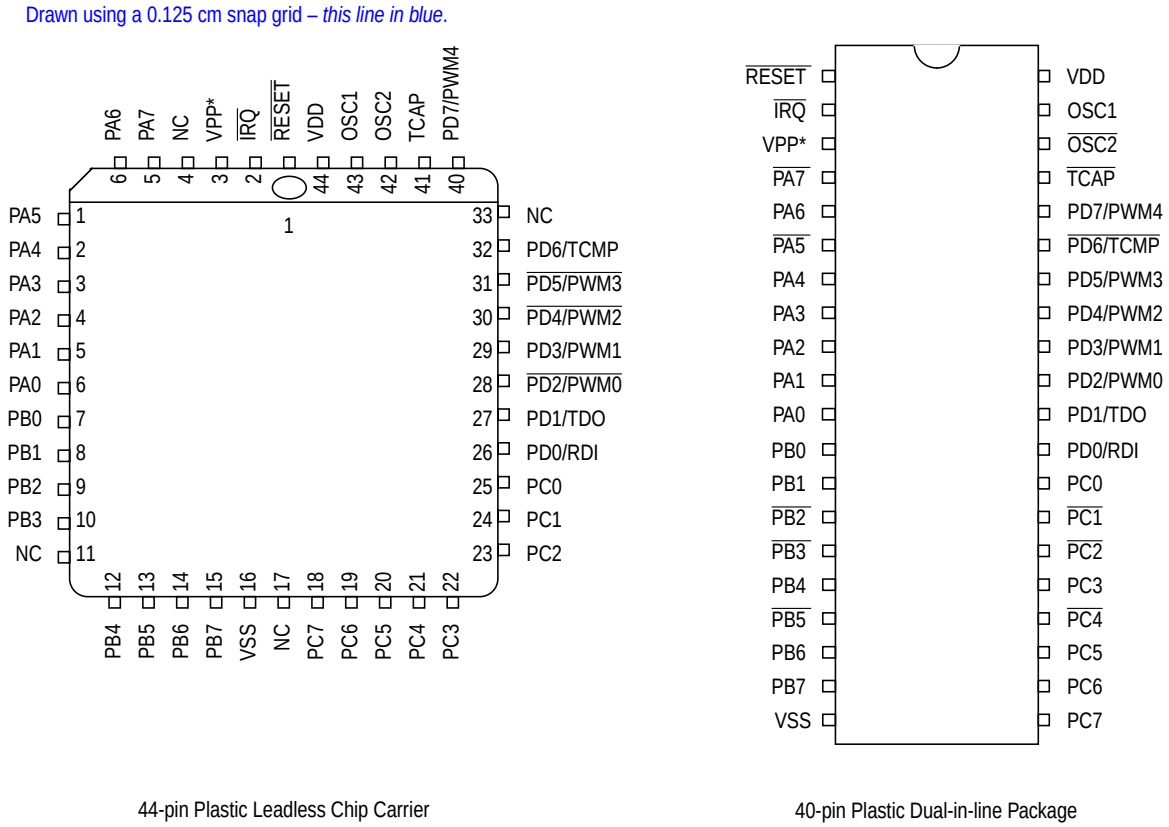
2.3.1 VDD and VSS

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.

2.3.2 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}.



44-pin Plastic Leadless Chip Carrier

40-pin Plastic Dual-in-line Package

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

Figure 2-2 Pin assignments for MC68HC05D9 packages

2.3.3 OSC1/OSC2

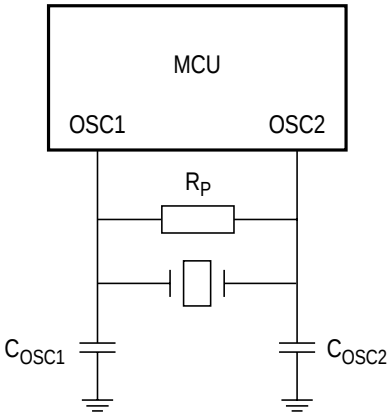
These pins provide control input for an on-chip clock oscillator circuit. A crystal, ceramic resonator or external clock signal connected to these pins provides the oscillator clock. The oscillator frequency is divided by 2 to provide the internal bus frequency.

2.3.3.1 Crystal

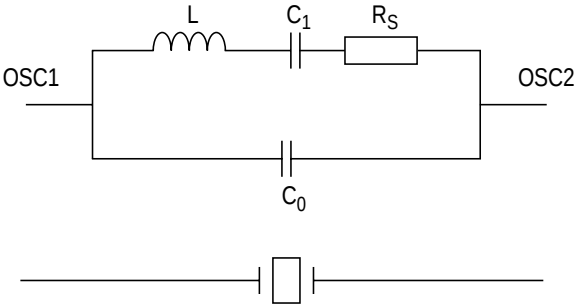
The circuit shown in Figure 2-3(a) is recommended when using a crystal. The internal oscillator is designed to interface with an AT-cut parallel-resonant quartz crystal resonator in the frequency range specified for f_{osc} (see Section 10.4, AC electrical characteristics). Use of an external CMOS oscillator 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.

2.3.3.2 Ceramic resonator

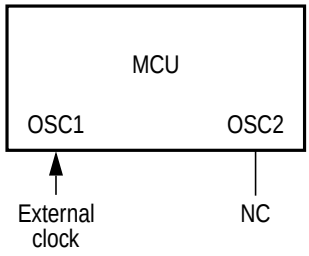
A ceramic resonator may be used instead of a crystal in cost-sensitive applications. The circuit in Figure 2-3(a) is recommended when using a ceramic resonator. Figure 2-3(d) lists the recommended capacitance and feedback resistance values. The manufacturer of the particular ceramic resonator being considered 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 2-3 Oscillator connections

2.3.3.3 External clock

An external clock should be applied to the OSC1 input with the OSC2 pin not connected, as shown in Figure 2-3(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}.

2.3.4 $\overline{\text{RESET}}$

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.

2.3.5 $\overline{\text{IRQ}}$

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

2.3.6 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 Section 7, Programmable timer).

2.3.7 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.

2.3.8 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.

2.3.9 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.

2.3.10 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.

2.3.10.1 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.

2.3.10.2 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.

2.3.10.3 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.

2.3.10.4 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.

2.4 Software-selectable options

2.4.1 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 $\overline{IRQ}$.

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

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 (set) – Maps 48 bytes of RAM into the memory map starting at address $0020.

0 (clear) – 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 (set) – Maps 128 bytes of RAM into the memory map starting at address $0100.

0 (clear) – Maps 128 bytes of ROM/EPROM into the memory map starting at address $0100.

IRQ

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

1 (set) – Edge-and-level sensitive interrupt selected.

0 (clear) – Edge sensitive only option is selected.

3
CPU CORE AND INSTRUCTION SET

This section provides a description of the CPU core registers, the instruction set and the addressing modes of the MC68HC05D9.

3.1 Registers

The MCU contains five registers, as shown in the programming model of Figure 3-1. The interrupt stacking order is shown in Figure 3-2.

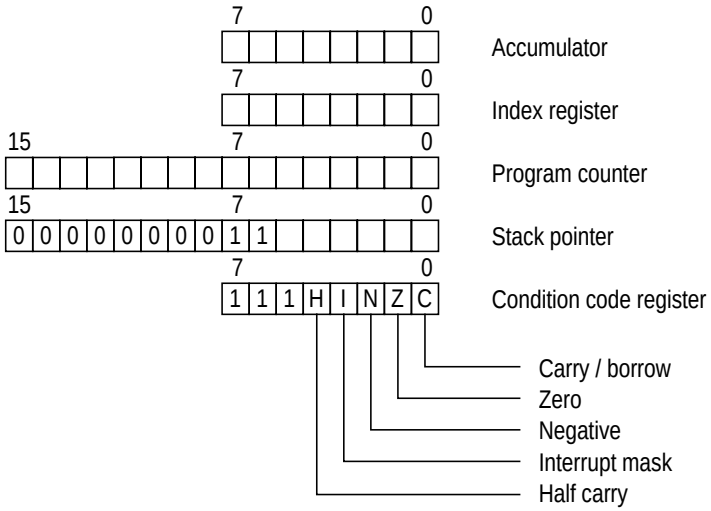


Figure 3-1 Programming model

3.1.1 Accumulator (A)

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

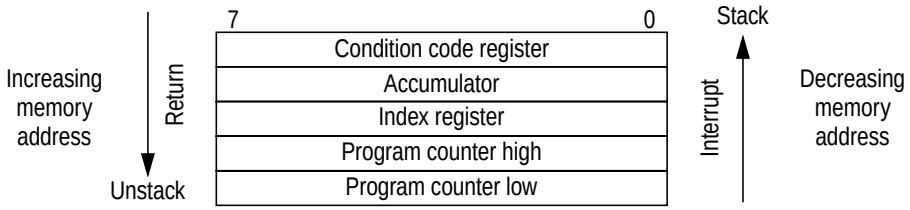


Figure 3-2 Stacking order

3.1.2 Index register (X)

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

3.1.3 Program counter (PC)

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

3.1.4 Stack pointer (SP)

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

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

3.1.5 Condition code register (CCR)

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

Half carry (H)

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

Interrupt (I)

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

Negative (N)

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

Zero (Z)

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

Carry/borrow (C)

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

3.2 Instruction set

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

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

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

This MCU uses all the instructions available in the M146805 CMOS family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown in Table 3-1.

3.2.1 Register/memory Instructions

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 3-2 for a complete list of register/memory instructions.

3.2.2 Branch instructions

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

3.2.3 Bit manipulation instructions

The MCU can set or clear any writable bit that resides in the first 256 bytes of the memory space (page 0). All port data and data direction registers, timer and serial interface registers, control/status registers and a portion of the on-chip RAM reside in page 0. An additional feature allows the software to test and branch on the state of any bit within these locations. The bit set, bit clear, bit test and branch functions are all implemented with single instructions. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 3-4.

3.2.4 Read/modify/write instructions

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to this sequence of reading, modifying and writing, since it does not modify the value. Refer to Table 3-5 for a complete list of read/modify/write instructions.

3.2.5 Control instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 3-6 for a complete list of control instructions.

3.2.6 Tables

Tables for all the instruction types listed above follow. In addition there is a complete alphabetical listing of all the instructions (see Table 3-7 and Table 3-8), and an opcode map for the instruction set of the M68HC05 MCU family (see Table 3-9).

Table 3-1 MUL instruction

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

Table 3-2 Register/memory instructions

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

Table 3-3 Branch instructions

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

Table 3-4 Bit manipulation instructions

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

Table 3-5 Read/modify/write instructions

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

Table 3-6 Control instructions

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

Table 3-7 Instruction set (1 of 2)

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

Address mode abbreviations

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

Not implemented

Condition code symbols

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

Table 3-8 Instruction set (2 of 2)

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

Address mode abbreviations

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

Not implemented

Condition code symbols

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

Table 3-9 M68HC05 opcode map

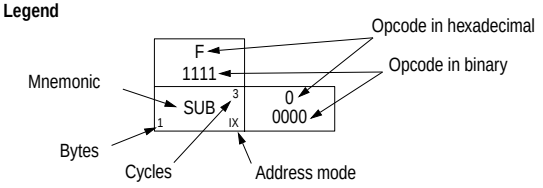
	Bit manipulation		Branch	Read/modify/write					Control		Register/memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
High / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	High / Low
0 0000	BRSET0 5 3 BTB	BSET0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

Abbreviations for address modes and registers

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

Not implemented

Legend



3.3 Addressing modes

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

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

3.3.1 Inherent

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

3.3.2 Immediate

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

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

3.3.3 Direct

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

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

3.3.4 Extended

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

EA = (PC+1):(PC+2); PC ← PC+3
Address bus high ← (PC+1); Address bus low ← (PC+2)

3.3.5 Indexed, no offset

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

EA = X; PC ← PC+1
Address bus high ← 0; Address bus low ← X

3.3.6 Indexed, 8-bit offset

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

EA = X+(PC+1); PC ← PC+2
Address bus high ← K; Address bus low ← X+(PC+1)
where K = the carry from the addition of X and (PC+1)

3.3.7 Indexed, 16-bit offset

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

EA = X+[(PC+1):(PC+2)]; PC ← PC+3
Address bus high ← (PC+1)+K; Address bus low ← X+(PC+2)
where K = the carry from the addition of X and (PC+2)

3.3.8 Relative

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

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

3.3.9 Bit set/clear

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

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

3.3.10 Bit test and branch

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

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

THIS PAGE LEFT BLANK INTENTIONALLY

4

RESETS, INTERRUPTS AND LOW POWER MODES

4.1 Resets

The MCU can be reset in four ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by a COP watchdog timer reset and by an internal clock monitor reset. See Figure 4-1.

4.1.1 Power-on reset

A power-on reset occurs when a positive transition is detected on VDD. The power-on reset function is strictly for power turn-on conditions and should not be used to detect drops in the power supply voltage. The power-on circuitry provides a stabilization delay (t_{PORL}) from when the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of this delay then the processor remains in the reset state until $\overline{\text{RESET}}$ goes high. The user must ensure that the voltage on VDD has risen to a point where the MCU can operate properly by the time t_{PORL} has elapsed. If there is doubt, the external $\overline{\text{RESET}}$ pin should remain low until the voltage on VDD has reached the specified minimum operating voltage. This may be accomplished by connecting an external RC-circuit to this pin to generate a power-on reset (POR). In this case, the time constant must be great enough (at least 100 ms) to allow the oscillator circuit to stabilise.

At power-up, the $\overline{\text{RESET}}$ 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.

4.1.2 $\overline{\text{RESET}}$ pin

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

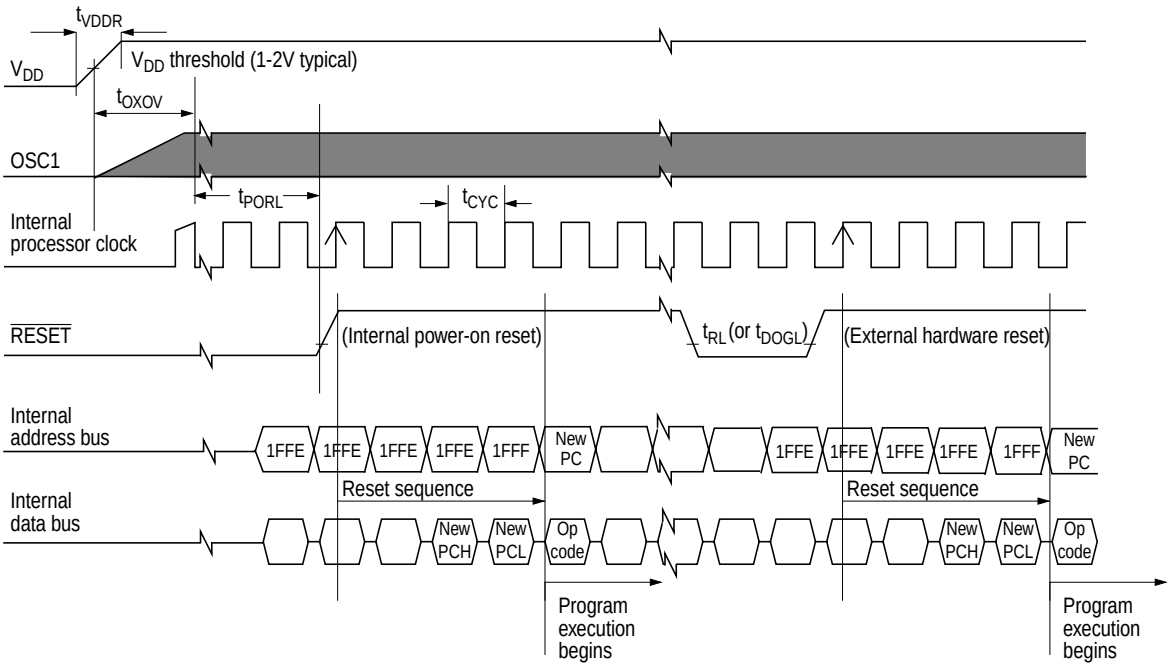


Figure 4-1 Reset timing diagram

4.1.3 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 $\overline{\text{RESET}}$ 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 4-1 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.

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

4.1.5 COP control register (COPCR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
COP control (COPCR)	$001E	0	0	0	COPF	CME	COPE	CM1	CM0	0000 0000

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 (set) – COP or clock monitor reset has occurred

0 (clear) – 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 (set) – Clock monitor enabled

0 (clear) – 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 (set) – COP timeout enabled

0 (clear) – 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 4-1 shows examples of COP timeout periods for several different bus frequencies (f_{OP}).

Table 4-1 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

4.2 Interrupts

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

- External signal on the $\overline{IRQ}$ 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 4-2 shows the relative priority of all the possible interrupt sources. Figure 4-2 shows the interrupt processing flow.

Table 4-2 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

4.2.1 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).

4.2.2 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.

4.2.2.1 External interrupt ($\overline{\text{IRQ}}$)

The interrupt request is latched immediately following the active edge or level on the $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ interrupt will cause the MPU to exit from STOP mode.

4.2.2.2 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).

4.2.2.3 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).

4.2.3 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 4-3.

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 ($\overline{\text{IRQ}}$), 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 ($\overline{\text{IRQ}}$), if enabled, or a timer or SCI interrupt. There are no special wait vectors for these individual interrupts.

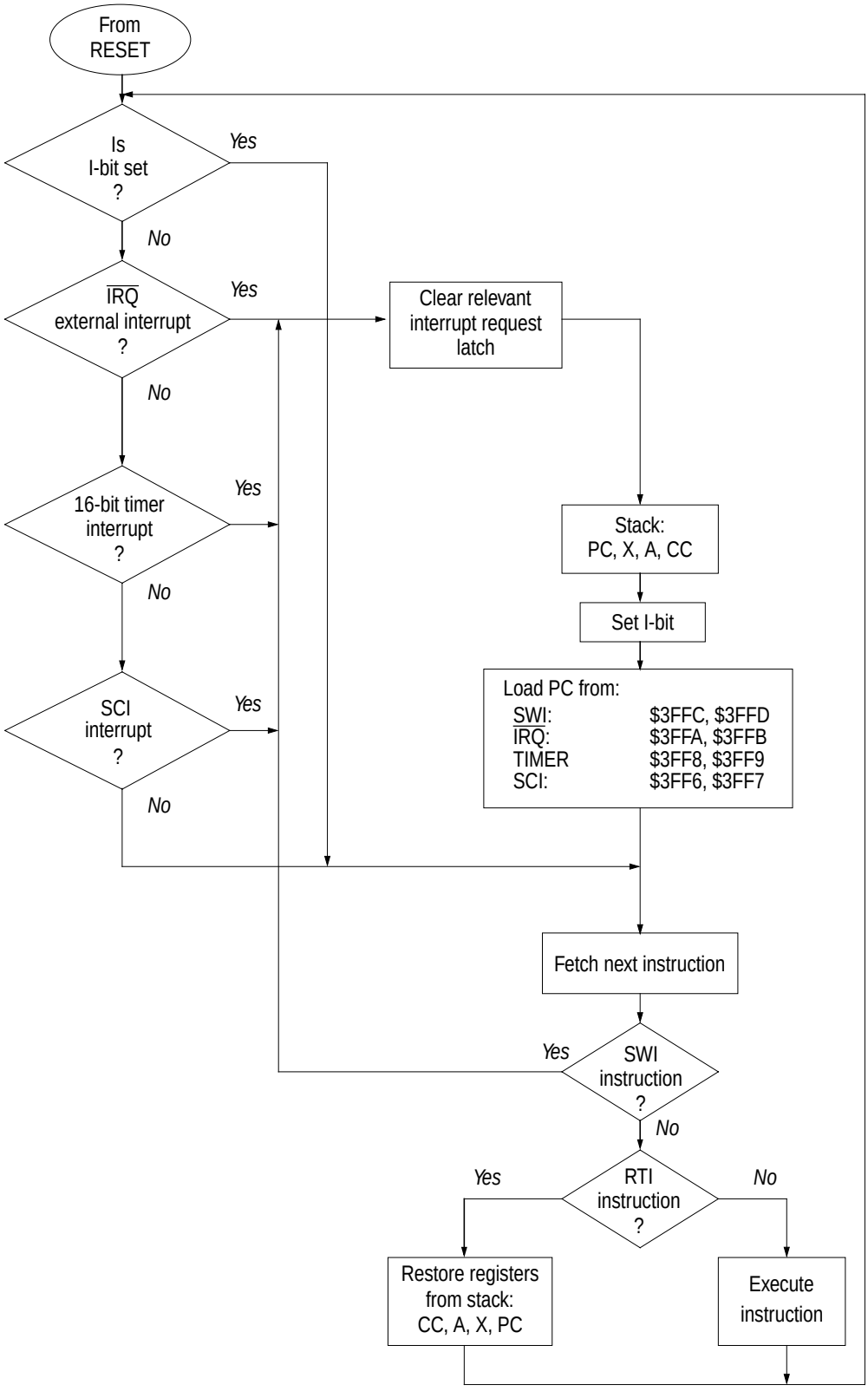


Figure 4-2 Interrupt flowchart

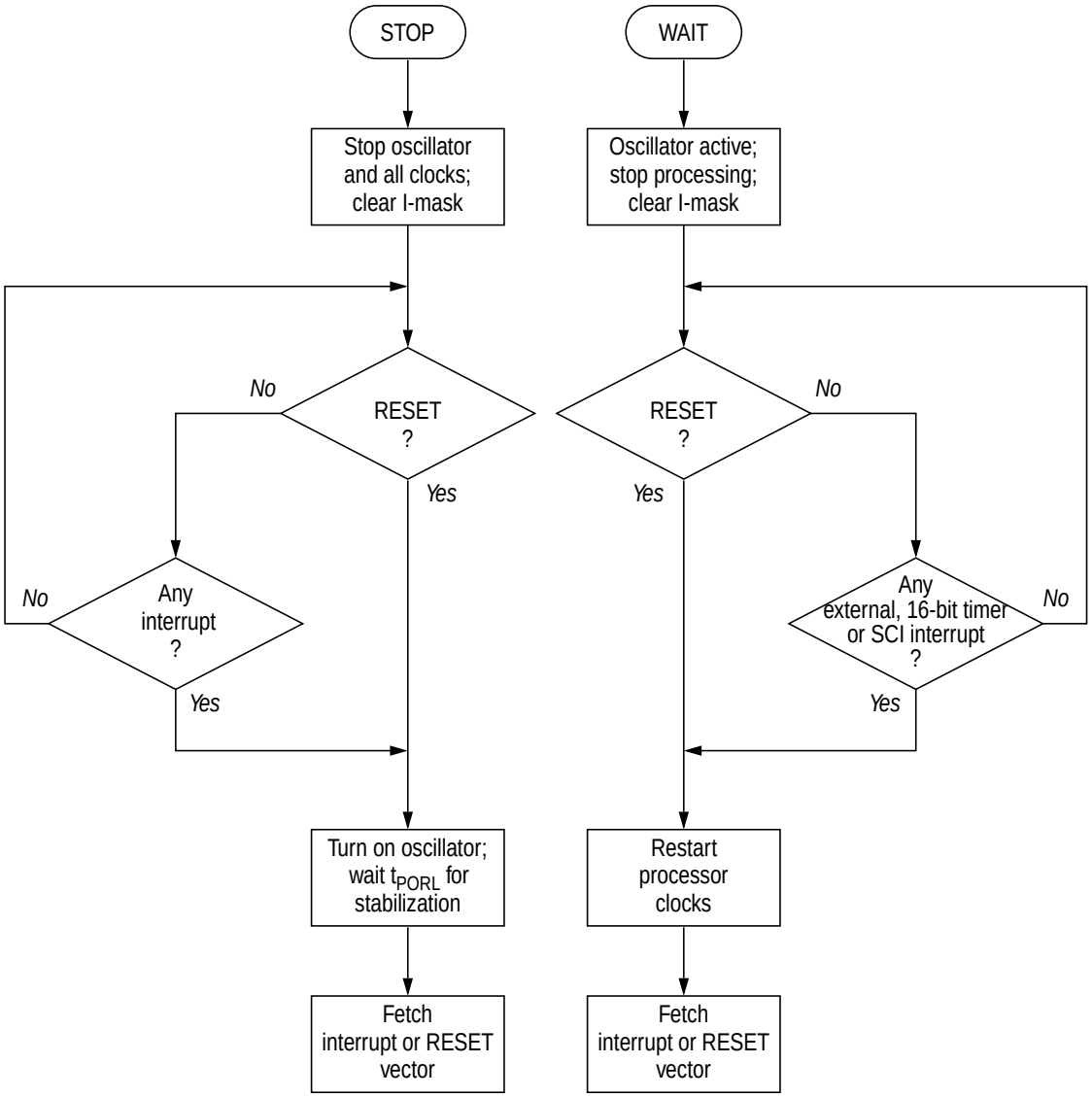


Figure 4-3 STOP and WAIT flowcharts

4.3 Low power modes

4.3.1 STOP

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

During 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 ($\overline{\text{IRQ}}$) or by a reset.

4.3.2 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.

4.3.3 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.

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

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

THIS PAGE LEFT BLANK INTENTIONALLY

5
MEMORY AND REGISTERS

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 5-1.

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).

5.1 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 Section 2.4).

5.2 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 Section 2.4).

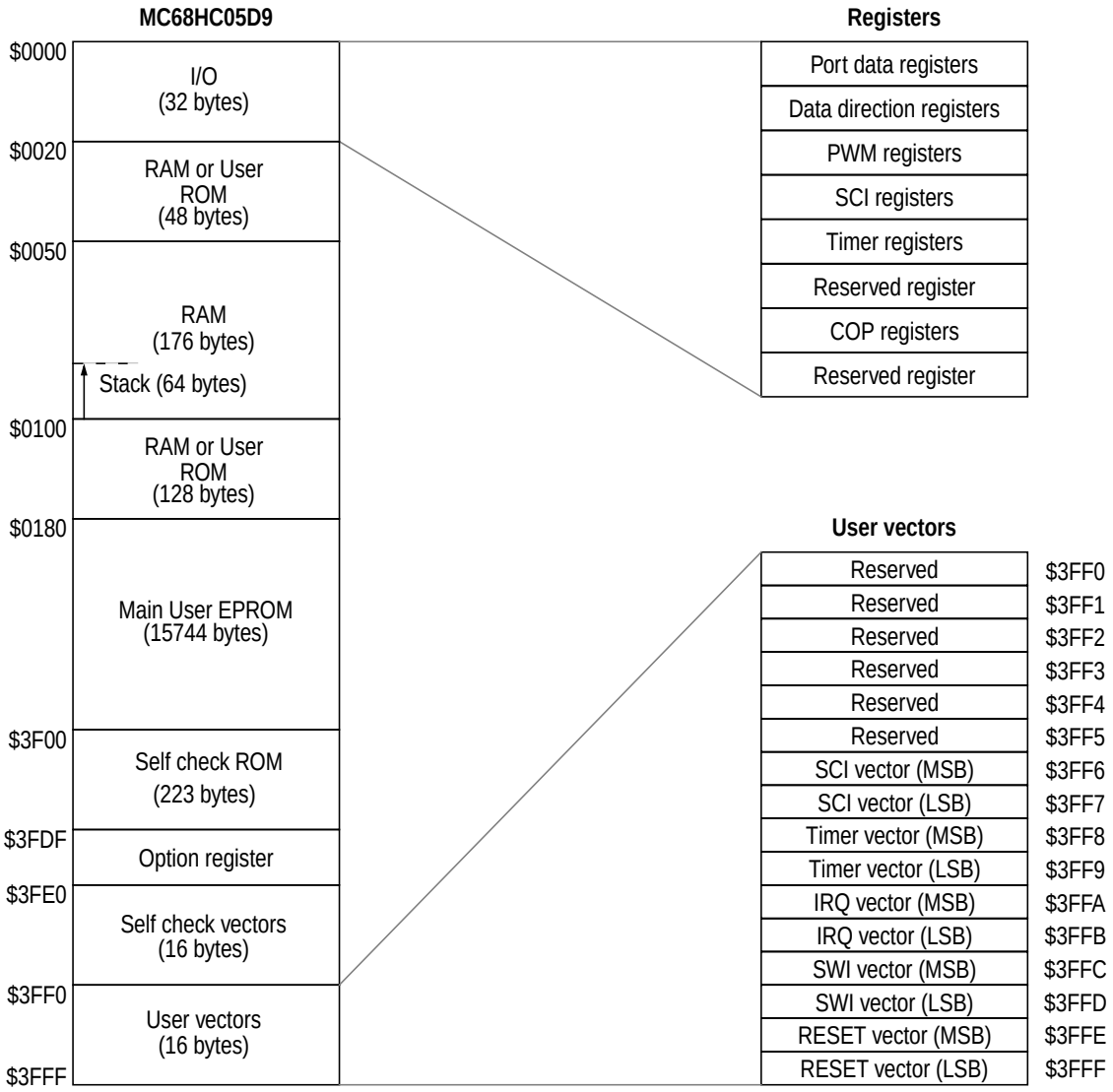


Figure 5-1 Memory map of the MC68HC05D9

5.3 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 5-1.

Table 5-1 MC68HC05D9 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

OPTION	$3FDF	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

6
PARALLEL INPUT/OUTPUT PORTS

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.

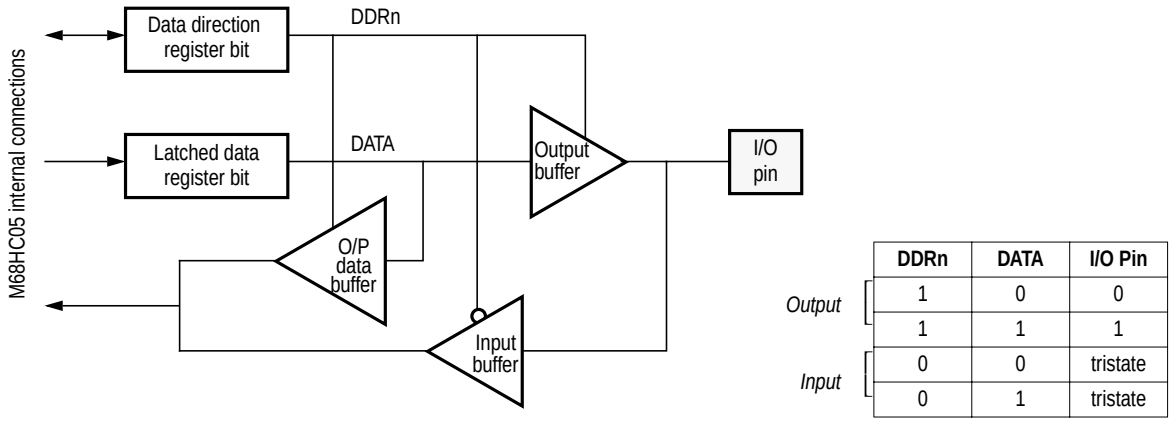
6.1 Input/output programming

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 6-1 and Table 6-1).

6.2 Ports A, B and C

Ports A, B and C are 8-bit bidirectional ports (see Figure 6-1). 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.



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

Figure 6-1 Standard I/O port structure

Table 6-1 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.

6.3 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 Section 2.3.10). 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.

6.4 Port registers

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

6.4.1 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.

6.4.2 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.

6.5 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 6-2.)

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	

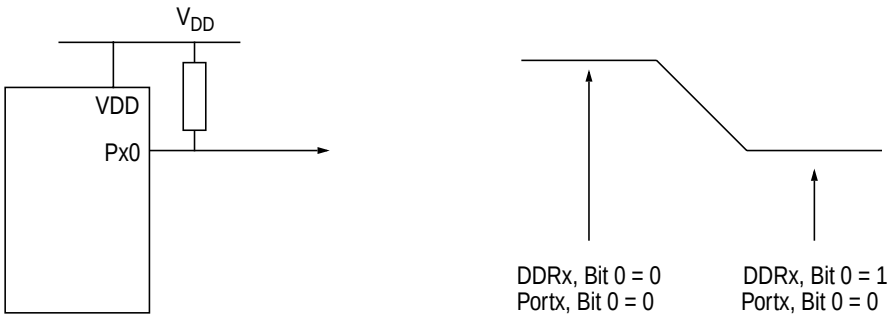


Figure 6-2 Port logic levels

6

THIS PAGE LEFT BLANK INTENTIONALLY

7
PROGRAMMABLE TIMER

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 7-1 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.

7.1 Counter

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

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.

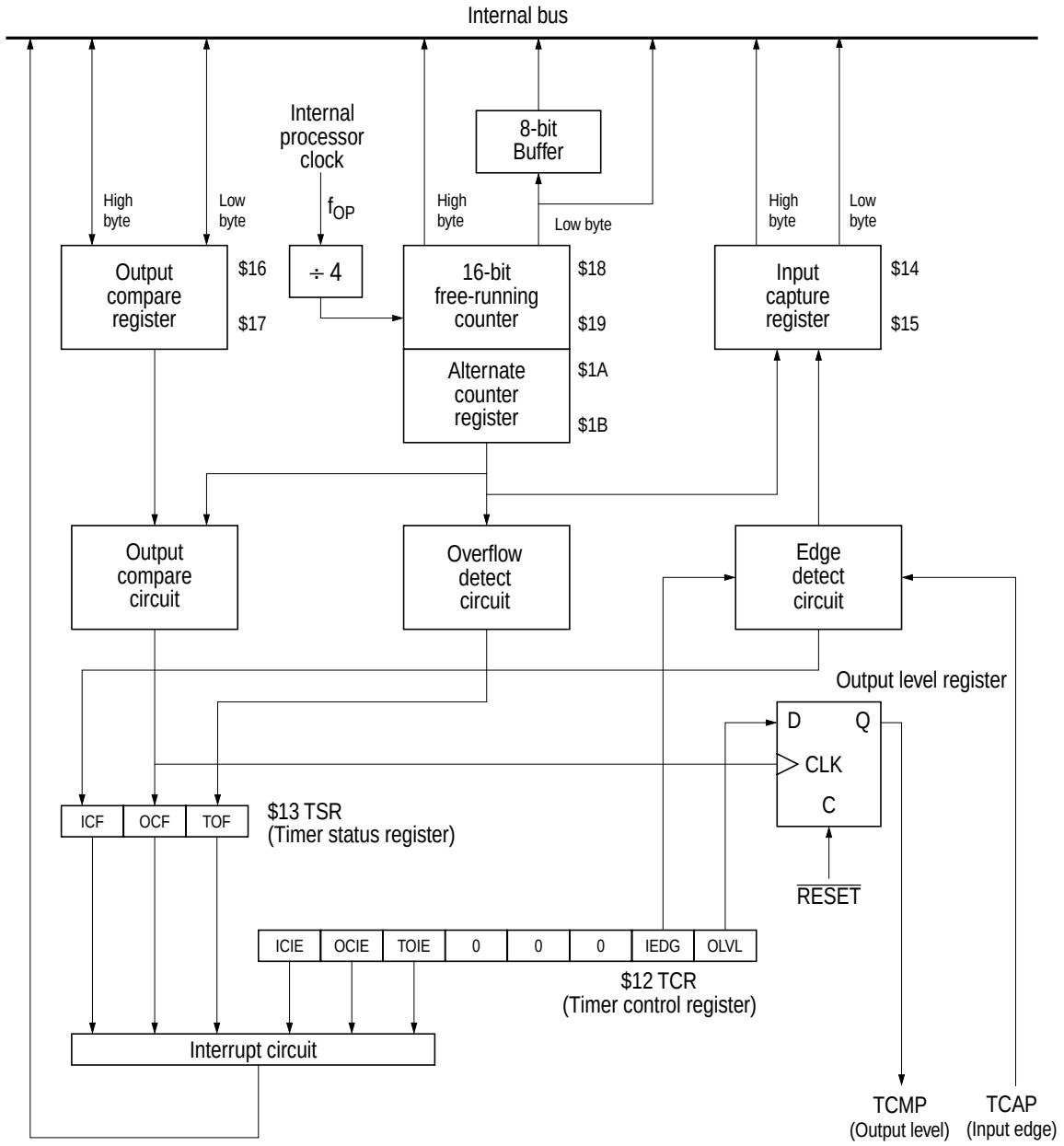


Figure 7-1 16-bit programmable timer block diagram

7.1.1 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.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer counter (TCNT)	$0018									1111 1111
	$0019									1111 1100

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.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Alternate counter (ALTCNT)	$001A									1111 1111
	$001B									1111 1100

7.2 Timer functions

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

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

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 (set) – Input capture interrupt enabled.

0 (clear) – 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 (set) – Output compare interrupt enabled.

0 (clear) – 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 (set) – Timer overflow interrupt enabled.

0 (clear) – 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 (set) – A positive going edge on the TCAP pin will trigger free-running counter transfer to the input capture register.

0 (clear) – 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 (set) – 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 (clear) – 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.

7.2.2 Timer status register (TSR)

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

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

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

ICF — Input capture flag

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 (set) – Input capture has occurred.

0 (clear) – 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 (set) – Output compare has occurred.

0 (clear) – 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 (set) – Timer overflow has occurred.

0 (clear) – 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.

7.2.3 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.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Input capture (ICR)	$0014									undefined
	$0015									undefined

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

The result obtained from an input capture will be one greater than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal 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.

7.2.4 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.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare (OCR)	$0016									undefined
	$0017									undefined

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The contents of the output compare register are 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.

7.3 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.

7.4 Timer during STOP mode

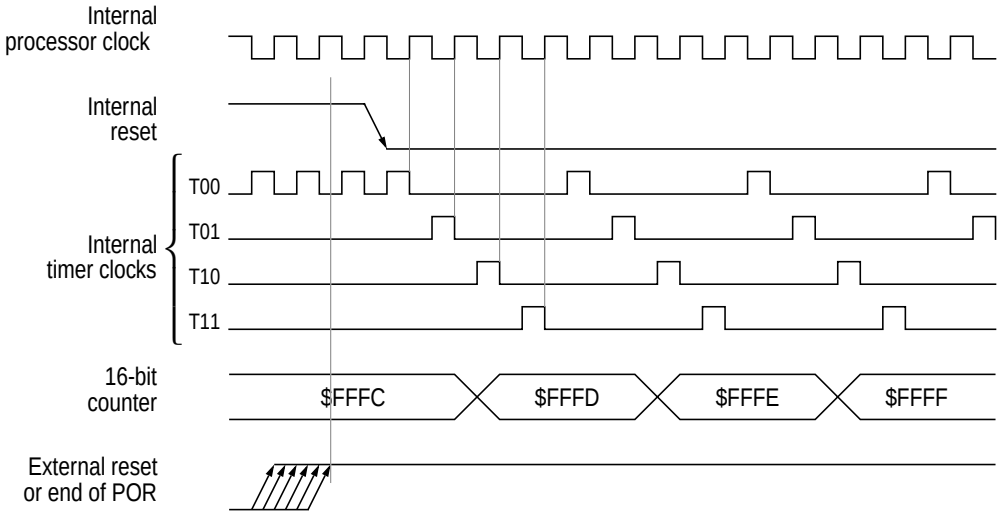
In the STOP mode all MCU clocks are stopped, hence the timer stops counting. If STOP is exited by an interrupt the counter retains the last count value. If the device is reset then the counter is forced to $FFFC.

During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture 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.

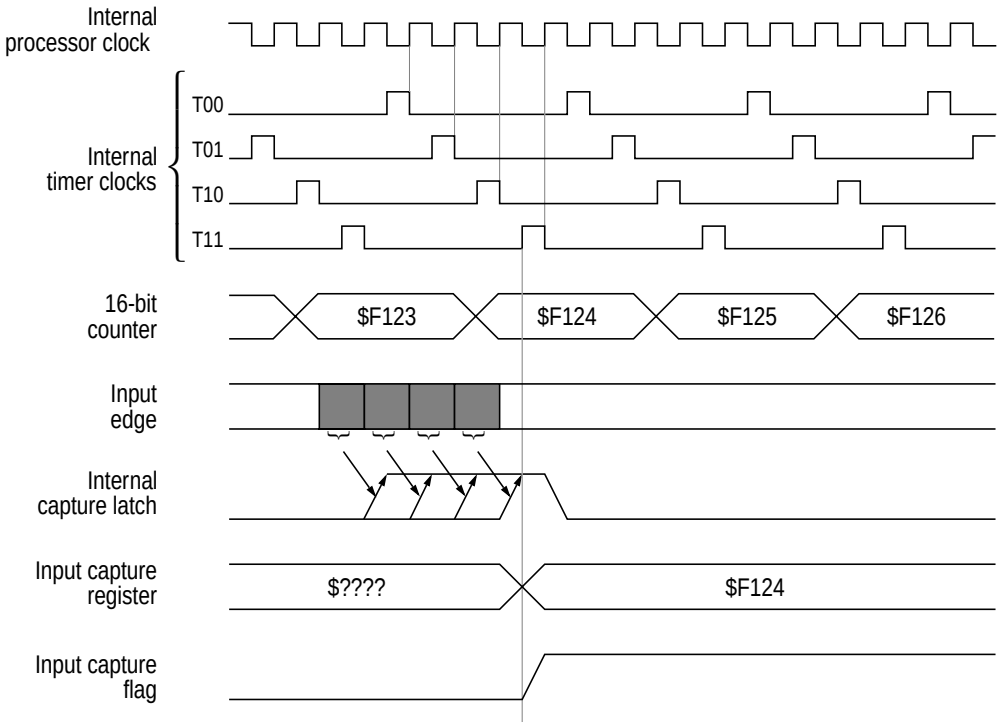
7.5 Timer state diagrams

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



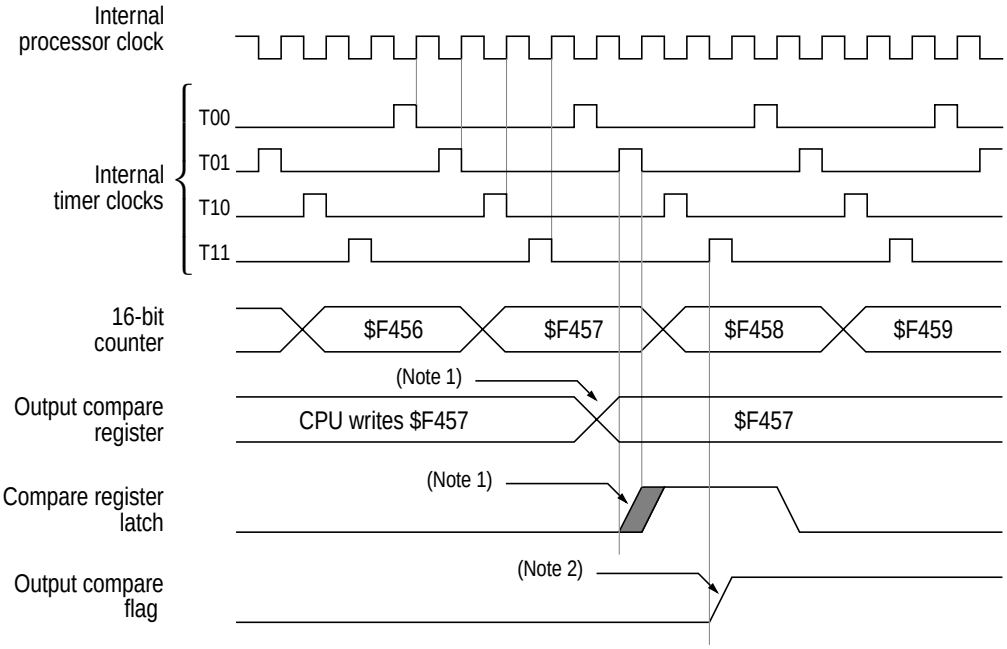
Note: The counter and timer control registers are the only ones affected by power-on or external reset.

Figure 7-2 Timer state timing diagram for reset



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

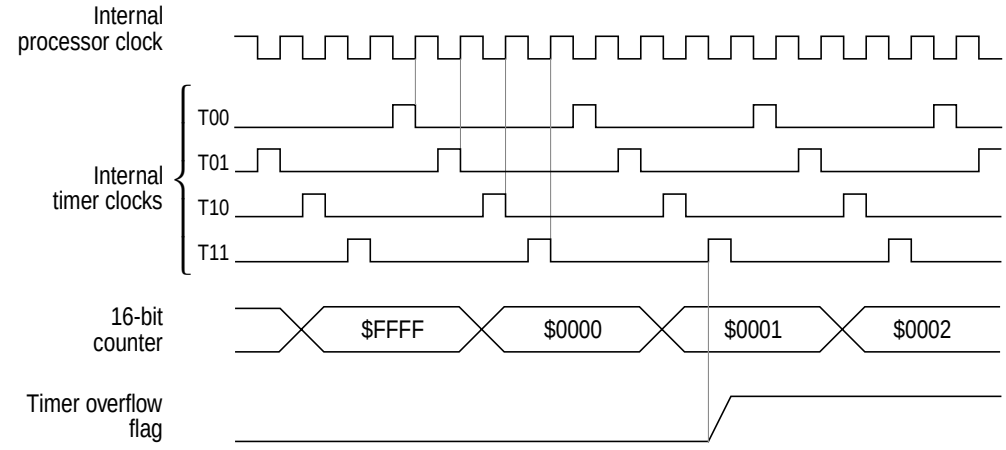
Figure 7-3 Timer state timing diagram for input capture



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

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

Figure 7-4 Timer state timing diagram for output compare



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

Figure 7-5 Timer state timing diagram for timer overflow

8
SERIAL COMMUNICATIONS INTERFACE

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.

8.1 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.

8.1.1 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

8.1.2 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

8.1.3 SCI transmitter features

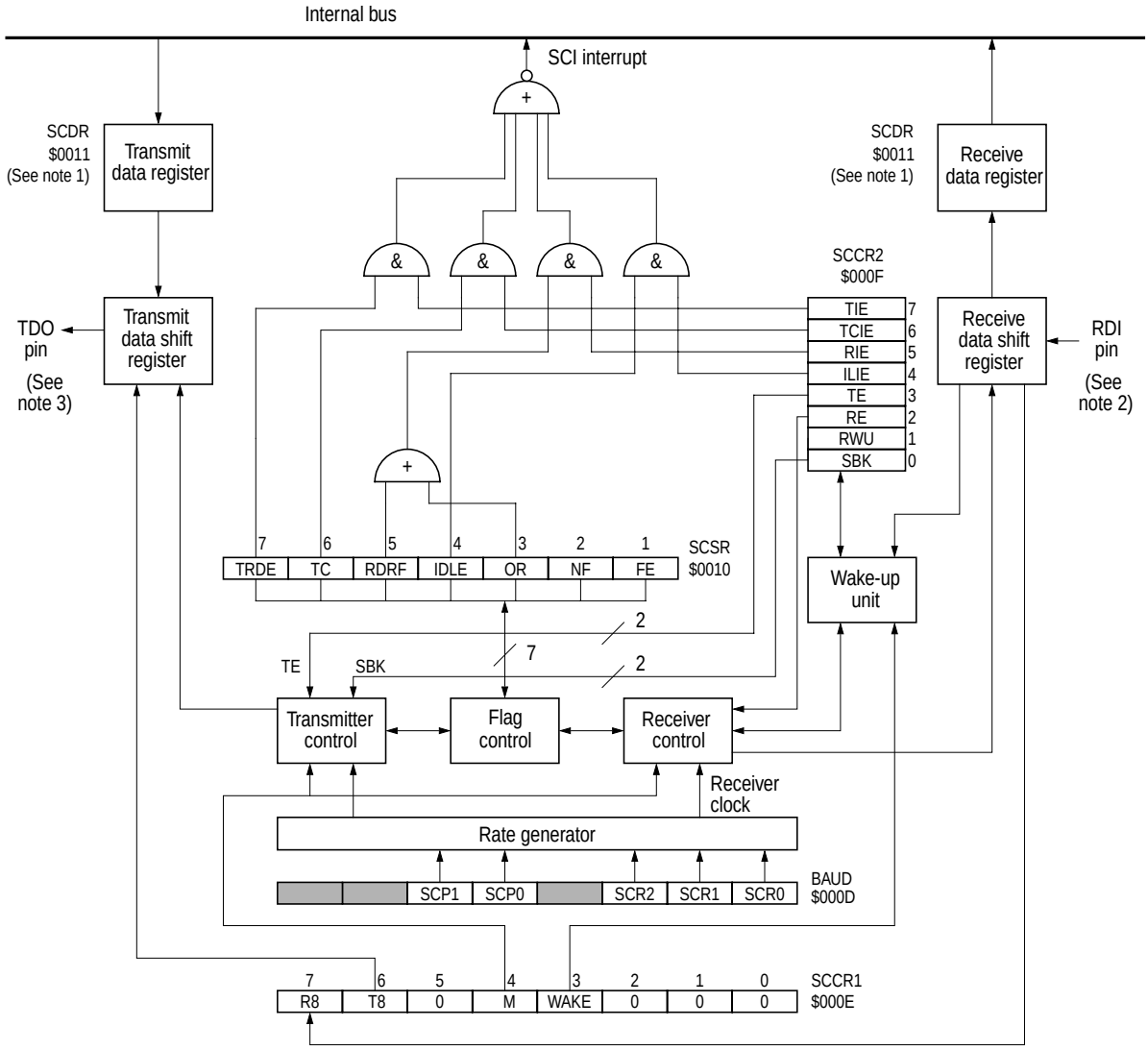
- Transmit data register empty flag
- Transmit complete flag
- Send break

8

8.2 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 8-2). 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.



Note 1: 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.

Note 2: Receive data in (RDI) is served by pin 0 of port B (PD0)

Note 3: Transmit data out (TDO) is served by pin 1 of port B (PD1)

Figure 8-1 Serial communications interface block diagram

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 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.

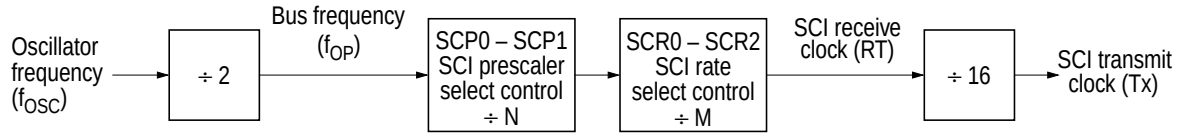


Figure 8-2 Rate generator division

8.3 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.

.

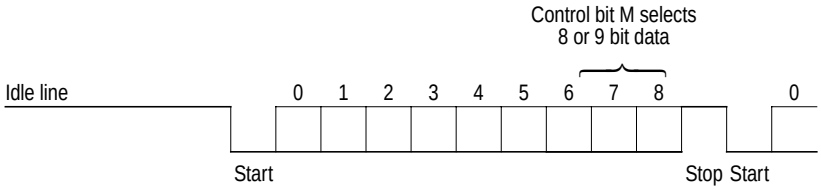


Figure 8-3 Data format

8.4 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.

8.4.1 Idle line wake-up

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

8.4.2 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.

8.5 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 8-4. The value of the bit is determined by voting logic which takes the value of the majority of the samples.

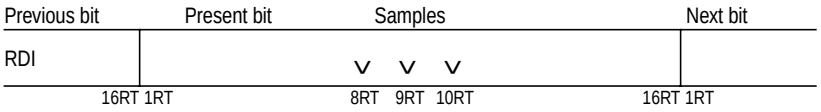


Figure 8-4 SCI sampling technique used on all bits

8.6 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 8-5). If at least two of these three verification samples detect a logic zero, a valid start bit has been detected, otherwise the line is assumed to be idle. A noise flag is set if 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 8-5) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see Figure 8-6); 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 8-7).

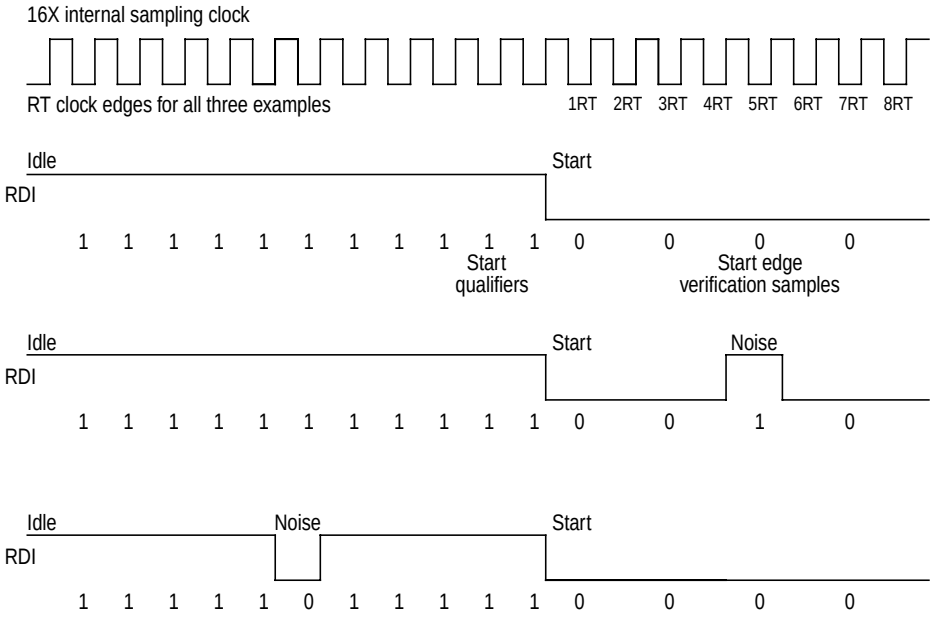
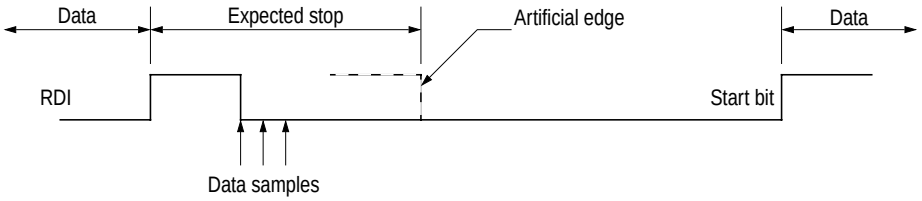
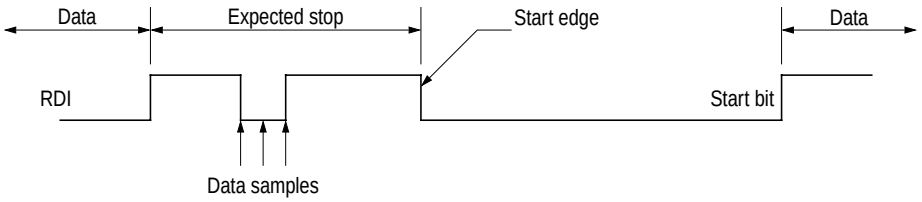


Figure 8-5 SCI examples of start bit sampling technique



a) Case 1, receive line low during artificial edge



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

Figure 8-6 Artificial start following a framing error

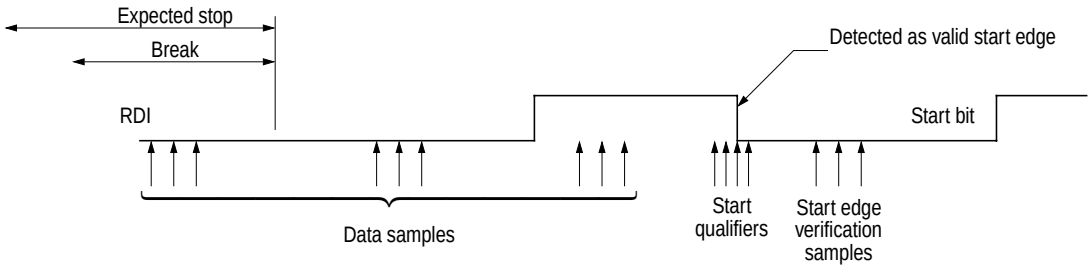


Figure 8-7 SCI start bit following a break

8.7 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.

8

8.8 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).

8.8.1 PWM mode register (PWMM)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PWM mode (PWMM)	$0008	0	0	SCIB	PWM4	PWM3	PWM2	PWM1	PWM0	0010 0000

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 (set) – SCI baud rate register accessed at address $000D

0 (clear) – SCI PWM4 register accessed at address $000D

The other control bits in this register are discussed in Section 9.

8.8.2 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.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI data (SCDR)	$0011									undefined

8.8.3 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.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI control 1 (SCCR1)	$000E	R8	T8	0	M	WAKE	0	0	0	uu0u u000

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 (set) – 1 start bit, 8 data, 9th data bit, 1 stop bit

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

8.8.4 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	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI control 2 (SCCR2)	$000F	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000

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 (set) – Transmit interrupt enabled (interrupt if TDRE = 1)

0 (clear) – 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 (set) – Transmit complete interrupt enabled (interrupt if TC = 1)

0 (clear) – 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 (set) – Receiver interrupt enabled (interrupt if RDRF or OR = 1)

0 (clear) – 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 (set) – Idle line interrupt enabled (interrupt if IDLE = 1)

0 (clear) – 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 (set) – Transmitter enabled

0 (clear) – 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 (set) – Receiver enabled

0 (clear) – 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 (set) – Receiver wake-up enabled

0 (clear) – 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 (set) – Send break

0 (clear) – Do not send break

8.8.5 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	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI status (SCSR)	$0010	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	1100 0000

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 (set) – Transmit data register empty

0 (clear) – 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 (set) – Transmission complete

0 (clear) – 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 (set) – Receive data register full

0 (clear) – 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 (set) – Idle line detected

0 (clear) – 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 (set) – Overrun error has occurred

0 (clear) – 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 (set) – Noise error has occurred

0 (clear) – 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 (set) – Framing error has occurred

0 (clear) – Framing error has occurred

8.8.6 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 8-2.)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI baud rate (BAUD)	$000D			SCP1	SCP0		SCR2	SCR1	SCR0	uu00 uuuu

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 8-2. The bus frequency clock (f_{OP}) is divided by the factor N shown in Table 8-1. This prescaled output provides an input to the second prescaler stage which is controlled by the SCI rate select bits (SCR2–SCR0)

Table 8-1 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 8-2. The prescaler output described above is divided by the factors shown in Table 8-2.

Table 8-2 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

9
PULSE WIDTH MODULATION UNIT

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 9-1). The PWM output signals, PWM0–4, appear on port D pins 2–5 and 7, respectively, provided they are enabled.

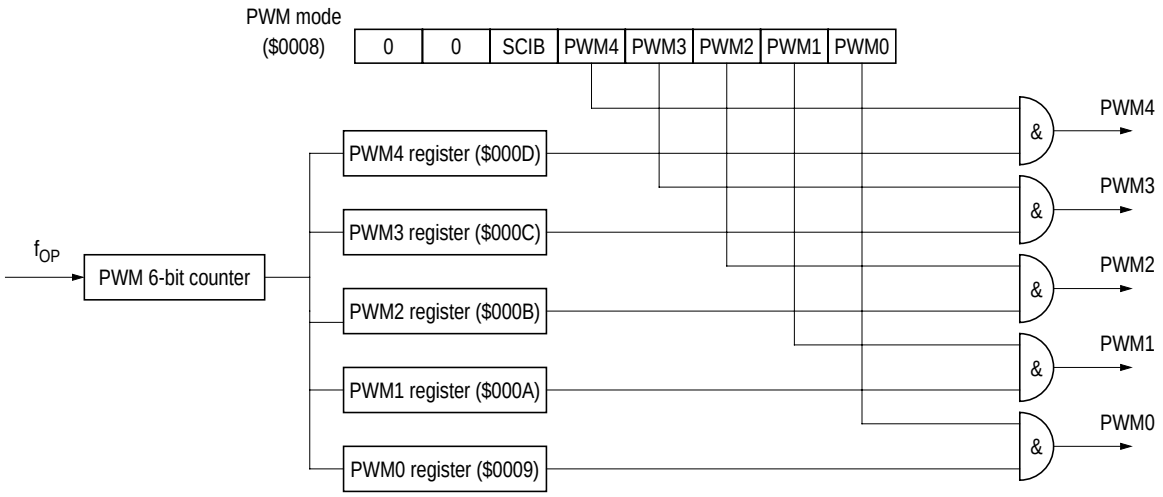


Figure 9-1 PWM block diagram

9.1 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).

9.2 PWM registers

9.2.1 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	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PWM mode (PWMM)	$0008	0	0	SCIB	PWM4	PWM3	PWM2	PWM1	PWM0	0010 0000

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 (set) – SCI baud rate register accessed at address $000D

0 (clear) – 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 (set) – PWM channel enabled

0 (clear) – PWM channel disabled

9.2.2 PWM channel data registers (PWM0–PWM4)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on 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 9-2.)

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.

9

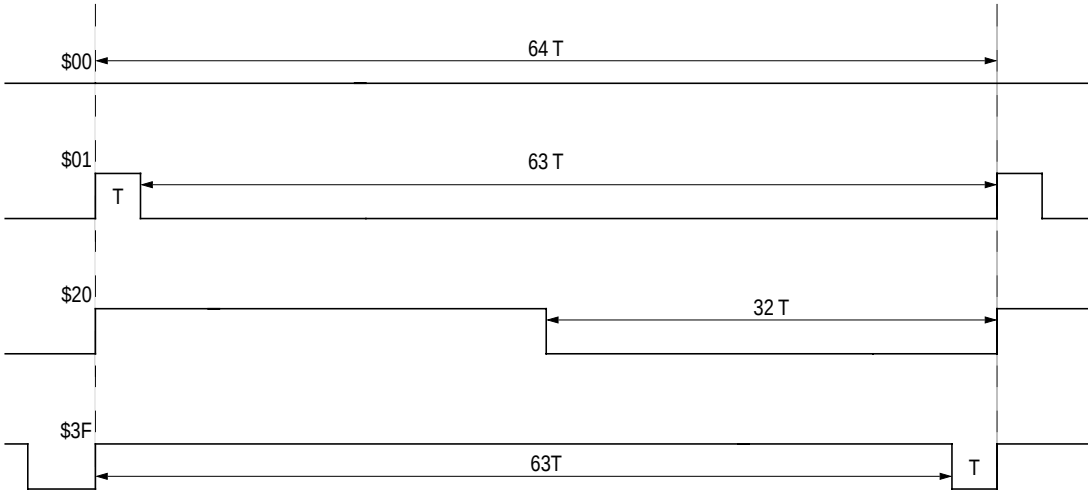


Figure 9-2 PWM output waveform examples.

9

THIS PAGE LEFT BLANK INTENTIONALLY

10
ELECTRICAL SPECIFICATIONS

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

10.1 Maximum ratings

Table 10-1 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 – $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$	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}.

10.2 Thermal characteristics and power considerations

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

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

where:

T_A = Ambient temperature (°C)

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

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

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

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

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

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

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

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

where K is a constant for a particular part. K can be determined by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained for any value of T_A by solving the above equations. The package thermal characteristics are shown in Table 10-2.

10

Table 10-2 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

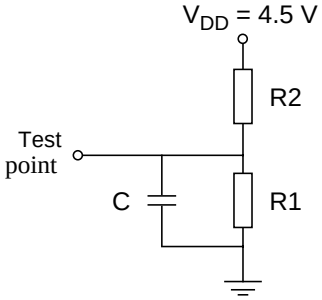


Figure 10-1 Equivalent test load

10.3 DC electrical characteristics

Table 10-3 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 WAIT STOP *STOP (*MC68HC705D32*)*	I_{DD}	 — — — —	 5.5 1 2 –	 9.5 4 10 *150*	 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

(1) All I_{DD} measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs (see Section 2.3.1).

(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 10-4 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 Section 2.3.1).

(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.

10.4 Control timing

Table 10-5 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 10-6 Control timing (3.0 V operation)

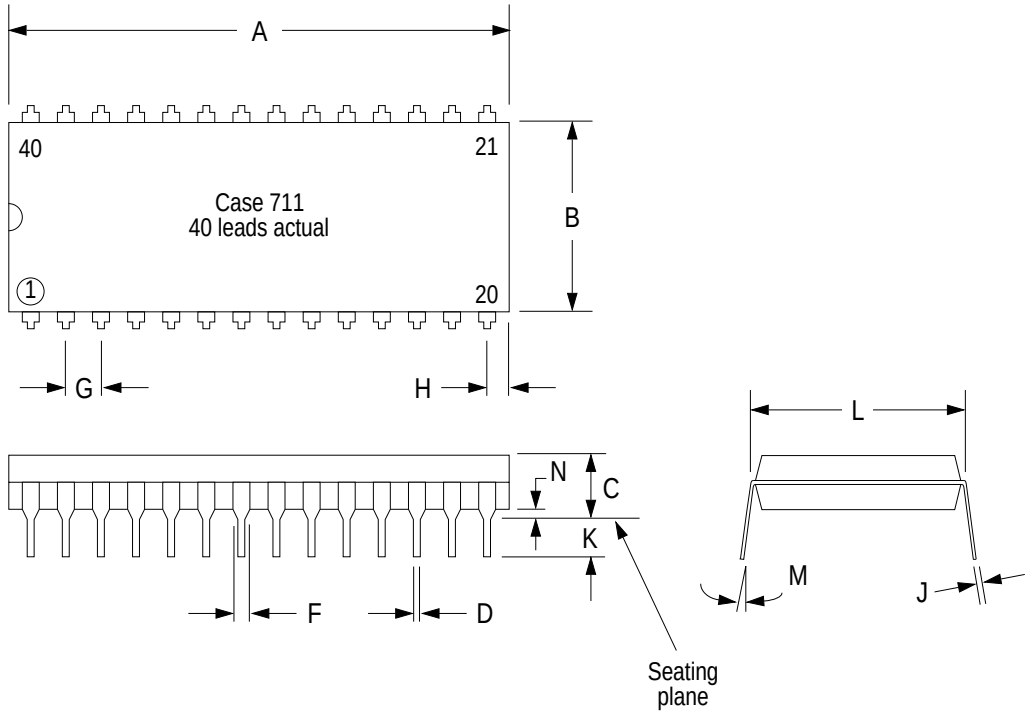
(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

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}.

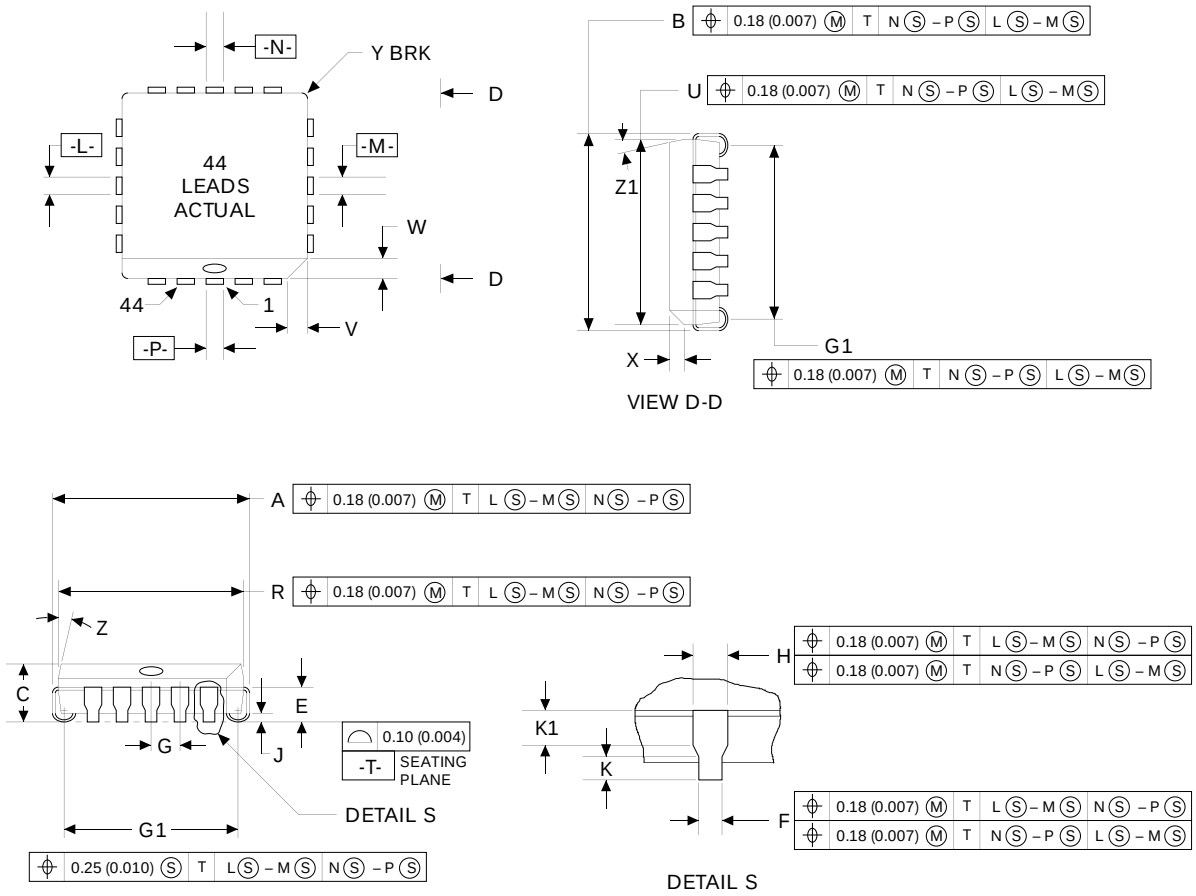
THIS PAGE LEFT BLANK INTENTIONALLY

11
MECHANICAL DATA AND ORDERING INFORMATION



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 11-1 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					

Notes:

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 11-2 44-pin PLCC mechanical dimensions

11.1 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 Motorola representative. Please note that you will need to supply details such as mask option selections, temperature range, oscillator frequency, package type, electrical test requirements and device marking details, so that an order can be processed and a customer specific part number allocated.

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

11.1.1 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.

11.1.2 Verification media

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

11.1.3 ROM verification units (RVUs)

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

11.1.4 Order numbers

Table 11-1 Order numbers

Package type	Operating range	Order number
40-pin DIP	–40 to +85°C (P suffix)	MC68HC05D9P MC68HC05D24P MC68HC05D32P MC68HC705D32P*
40-pin DIP	–40 to +85°C (VP suffix)	MC68HC05D9VP
44-pin PLCC	–40 to +105°C (FN suffix)	MC68HC05D9FN MC68HC05D24FN MC68HC05D32FN MC68HC705D32FN*
44-pin PLCC	–40 to +105°C (VFN suffix)	MC68HC05D9VFN

Note: Both packages in the table above are windowless plastic packages. Consequently the MC68HC705D32 device (marked with an asterisk) is a one time programmable (OTP) device only.

A
FEATURES SPECIFIC TO THE MC68HC05D24

A.1 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 A-1).

A.2 Memory map and registers

The memory map for this device is shown in Figure A-1. The registers are shown in more detail in Table A-1. 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.

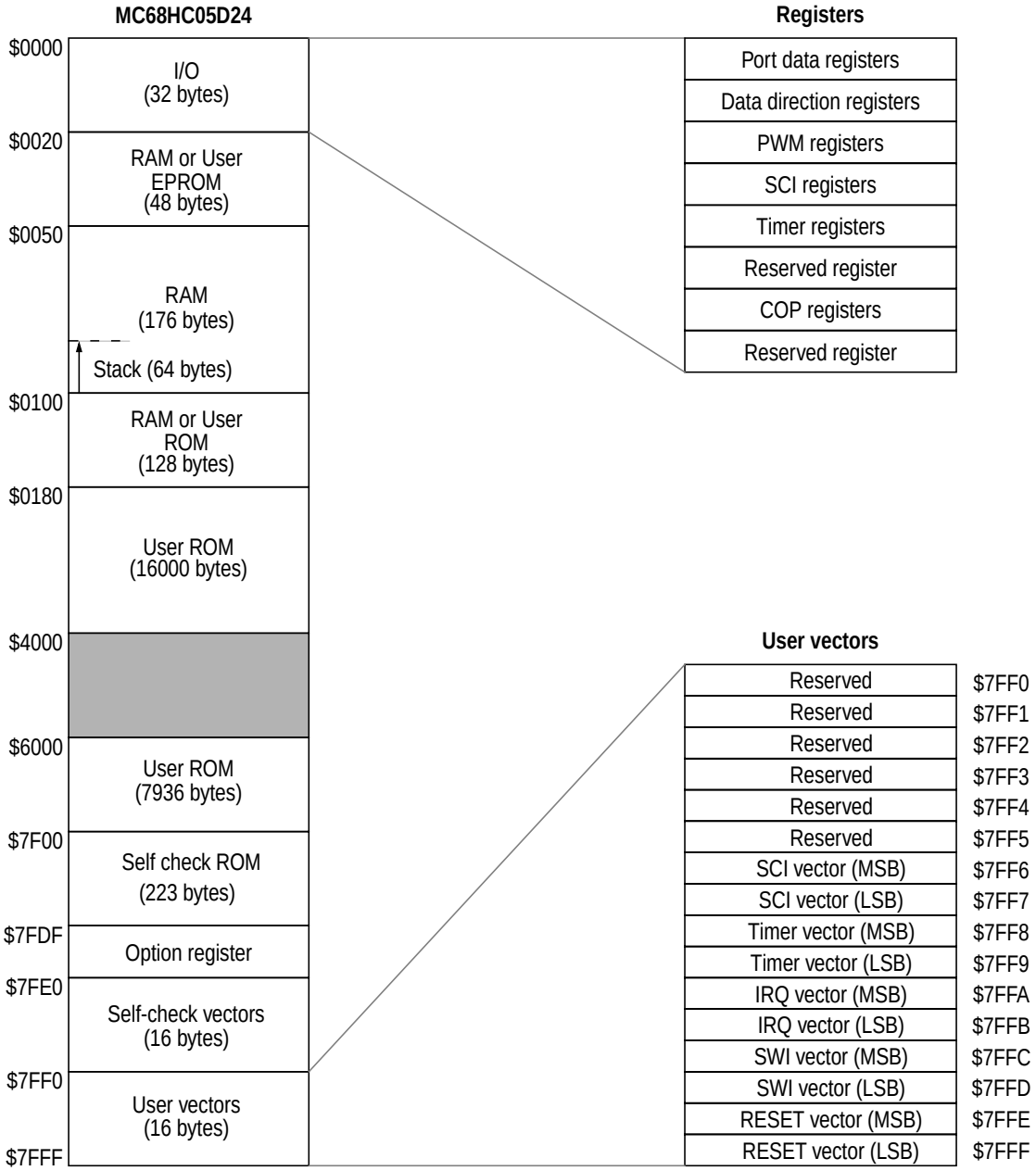


Figure A-1 Memory map of the MC68HC05D24

A

Table A-1 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

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

A

A.3 Programming model

The programming model of the MC68HC05D24 is identical to that of the MC68HC05D9, except for the program counter (see Figure A-2). Only bit 15 of the program counter (PC) in the MC68HC05D24 is permanently set to zero, thus restricting the address range to $0000–$7FFF (32K bytes).

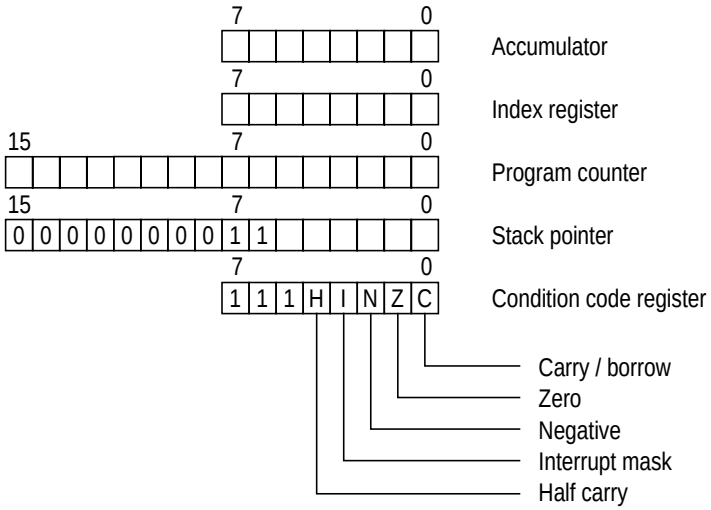


Figure A-2 Programming model of the MC68HC05D24

A

B
FEATURES SPECIFIC TO THE MC68HC05D32

B.1 Introduction

The MC68HC05D32 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 MC68HC05D32 is identical to the MC68HC05D9 (see Figure B-1).

B.2 Memory map and registers

The memory map for this device is shown in Figure B-1. The registers are shown in more detail in Table B-1. 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.

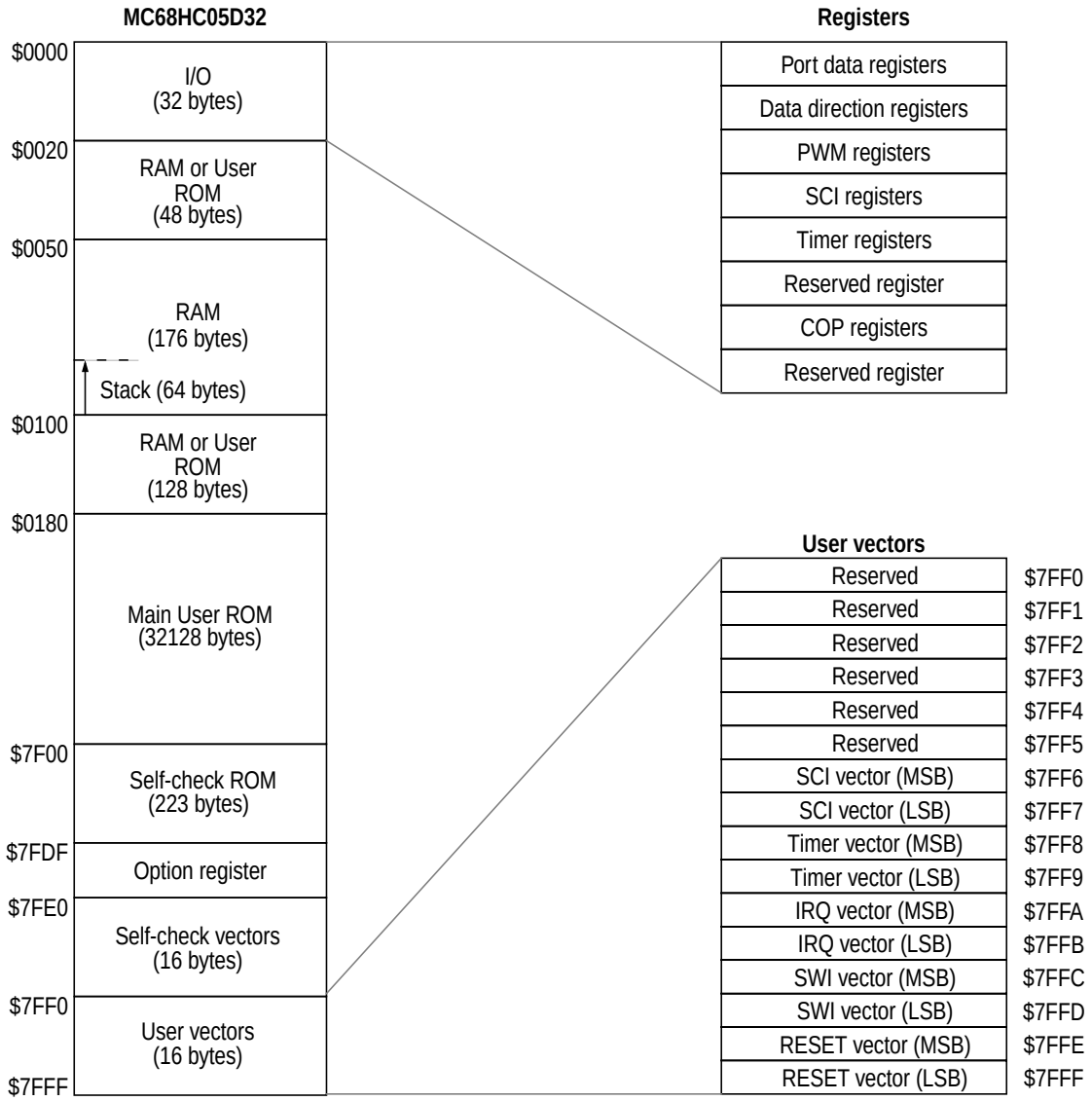


Figure B-1 Memory map of the MC68HC05D32

B

Table B-1 MC68HC05D32 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

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

B

B.3 Programming model

The programming model of the MC68HC05D32 is identical to that of the MC68HC05D9, except for the program counter (see Figure B-2). Only bit 15 of the program counter (PC) in the MC68HC05D32 is permanently set to zero, thus restricting the address range to $0000–$7FFF (32K bytes).

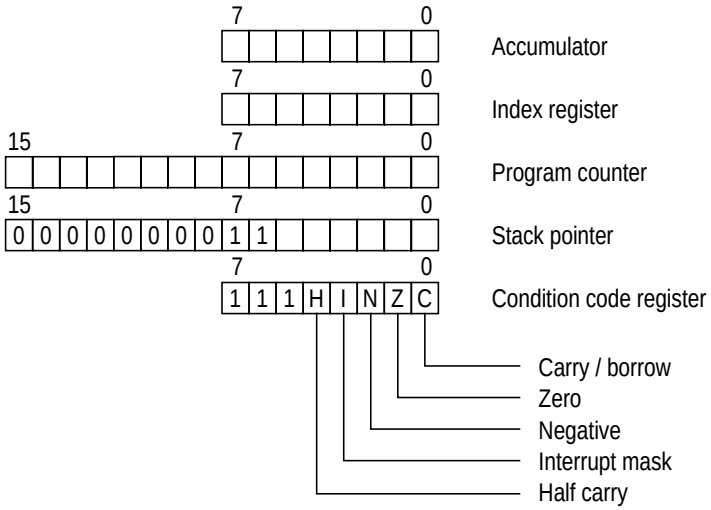


Figure B-2 Programming model of the MC68HC05D32

B

C
FEATURES SPECIFIC TO THE MC68HC705D32

C.1 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 C-1) apart from the EPROM replacing ROM.

C.2 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 C-1).

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 C-1 Operating mode entry conditions

$\overline{IRQ}$	$\overline{RESET}$	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.

C.3 Pin descriptions

C.3.1 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}.

C.3.2 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.

C.3.3 Crystal

The circuit shown in Figure 2-3(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 Section 10.4). 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.

C

C.3.4 External clock

An external clock should be applied to the OSC1 input with the OSC2 pin not connected, as shown in Figure 2-3(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.

C.4 Memory

The memory map for this device is shown in Figure C-1. The registers are shown in more detail in Table C-2. 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 C-2.

C

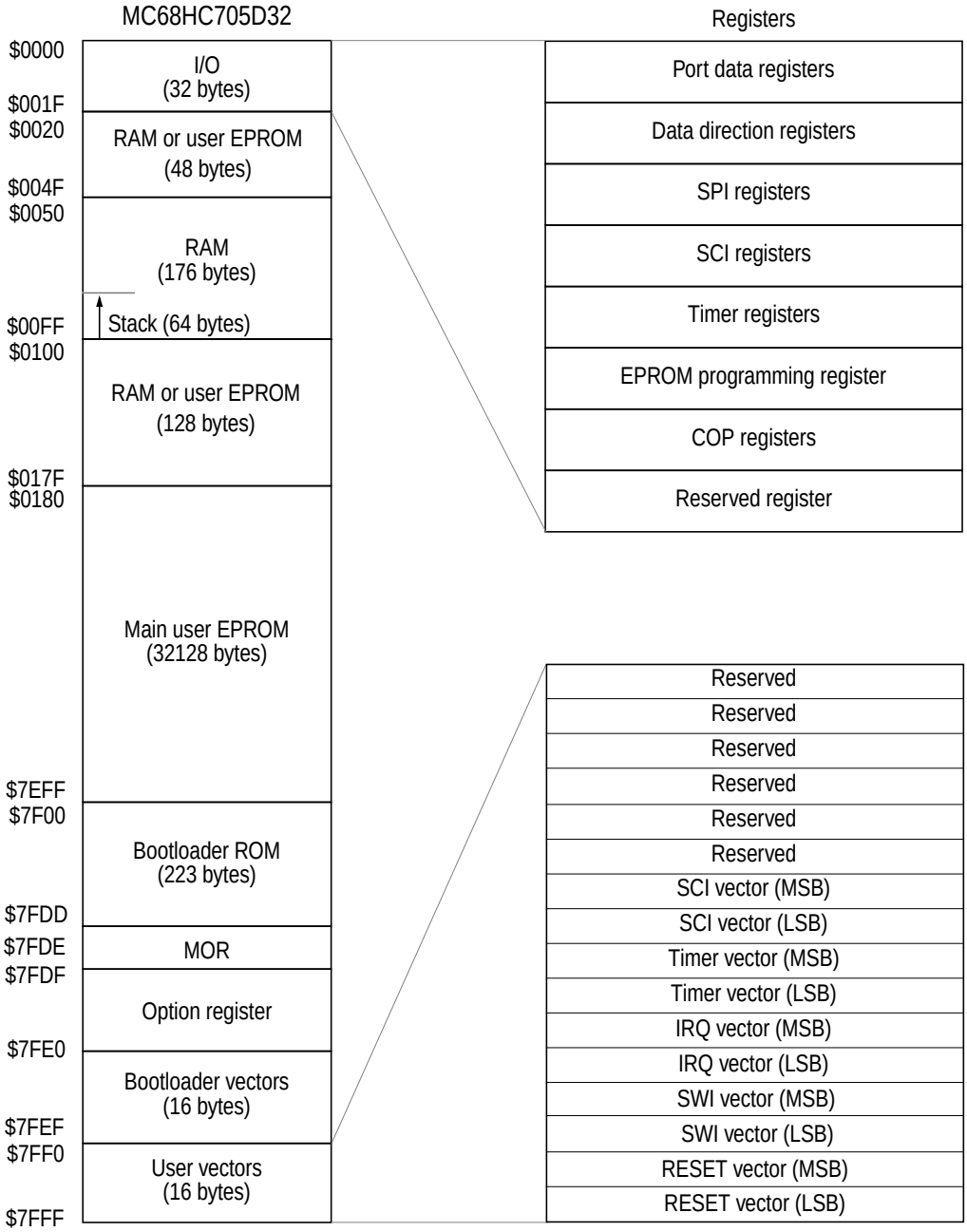


Figure C-1 Memory map of the MC68HC705D32

Table C-2 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

C

Table C-2 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
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

C.5 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 Appendix C.5.2.

C.5.1 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.

C

C.5.2 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

C.5.3 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).

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Program control register (PCR)	$001C						LATCH		EPGM	0000 0000

LATCH — EPROM latch control

1 (set) – 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 (clear) – EPROM address and data bus configured for normal reads.

EPGM — EPROM program command

1 (set) – 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 (clear) – Programming power is switched OFF from EPROM array.

C.5.4 EPROM mask option register (MOR)

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

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EPROM mask option register (MOR)(1)	$7FDE	SEC	—		—	—	—			Not affected

(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 (set) – EPROM contents not available in bootloader/test mode

0 (clear) – EPROM contents available in bootloader/test mode

C.6 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.

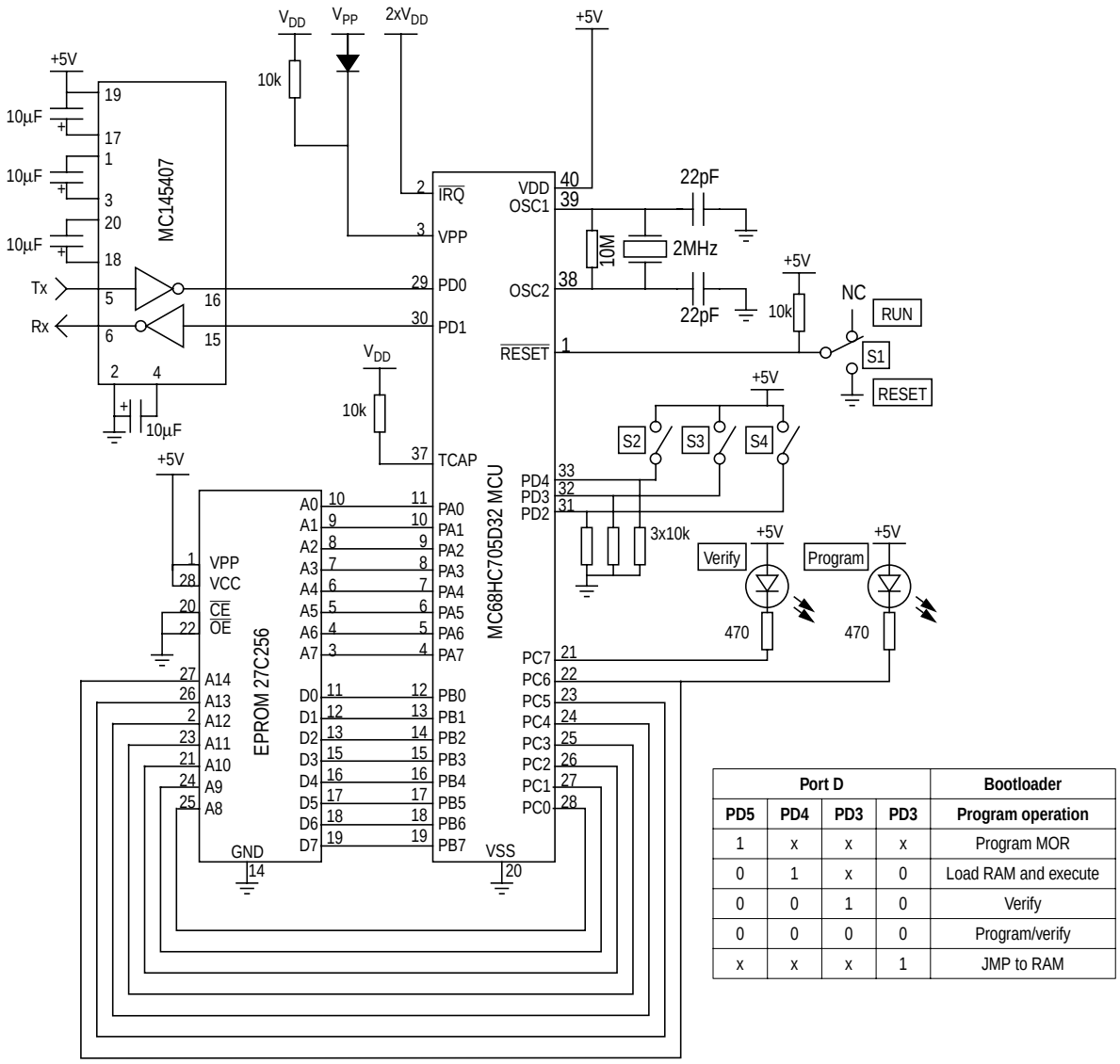
C.6.1 Bootloader functions

Table C-3 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

C



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 C-2 Self-programming via the bootstrap loader (40-pin PDIP)

C

THIS PAGE LEFT BLANK INTENTIONALLY

C

GLOSSARY

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

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

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

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

THIS PAGE LEFT BLANK INTENTIONALLY

INDEX

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

L

M

O

P

R

S

T

V

X

THIS PAGE LEFT BLANK INTENTIONALLY

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC05D9/D Rev. 2)

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

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

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

Comments: _____

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

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

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

Completely			Not at all
❑	❑	❑	❑

Comments: _____

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

Easy			Difficult
❑	❑	❑	❑

Comments: _____

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

	Too little detail				Too much detail
SECTION 1 INTRODUCTION	❑	❑	❑	❑	❑
SECTION 2 OPERATING MODES AND PIN DESCRIPTIONS	❑	❑	❑	❑	❑
SECTION 3 CPU CORE AND INSTRUCTION SET	❑	❑	❑	❑	❑
SECTION 4 RESETS, INTERRUPTS AND LOW POWER MODES	❑	❑	❑	❑	❑
SECTION 5 MEMORY AND REGISTERS	❑	❑	❑	❑	❑
SECTION 6 PARALLEL INPUT/OUTPUT PORTS	❑	❑	❑	❑	❑
SECTION 7 PROGRAMMABLE TIMER	❑	❑	❑	❑	❑
SECTION 8 SERIAL COMMUNICATIONS INTERFACE	❑	❑	❑	❑	❑
SECTION 9 PULSE WIDTH MODULATION UNIT	❑	❑	❑	❑	❑
SECTION 10 ELECTRICAL SPECIFICATIONS	❑	❑	❑	❑	❑
SECTION 11 MECHANICAL DATA AND ORDERING INFORMATION	❑	❑	❑	❑	❑
SECTION A FEATURES SPECIFIC TO THE MC68HC05D24	❑	❑	❑	❑	❑
SECTION B FEATURES SPECIFIC TO THE MC68HC05D32	❑	❑	❑	❑	❑
SECTION C FEATURES SPECIFIC TO THE MC68HC705D32	❑	❑	❑	❑	❑

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

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

– Cut along this line to remove –

8. How could we improve this document? _____

_____

_____

9. How would you rate Motorola's documentation?

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

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

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

12. How many years have you worked with microprocessors?

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

– Second fold back along this line –

By air mail
Par avion

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

NE PAS AFFRANCHIR

NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

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

F.A.O. Technical Publications Manager
(re: MC68HC05D9/D Rev. 2)

MOTOROLA
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

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

_____

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

Name: _____ Phone No: _____

Position: _____ FAX No: _____

Department: _____

Company: _____

Address: _____

_____

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

– Finally, tuck this edge into opposite flap –

How to reach us:

Mfax™: RMFAX0@email.sps.mot.com – TOUCHTONE (602) 244-6609

INTERNET: http://www.mot.com/SPS/

USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405; Denver, Colorado 80217. 303-675-2140

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki, 6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 81-3-3521-8315

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