HC05CC2GRS/D
REV 1.2

68HC05CC2

SPECIFICATION
(General Release)

© February 20, 1995

CSIC System Design Group
Austin, Texas

TABLE OF CONTENTS

Section Title Page

SECTION 1
GENERAL DESCRIPTION

1.1 Features 1-1
1.2 Mask Options 1-3
1.3 Pin Assignments 1-3
1.4 Functional Pin Description 1-4
1.4.1 V_{DD} and V_{SS} 1-4
1.4.2 OSC1 and OSC2 1-5
1.4.3 $\overline{\text{RESET}}$ 1-6
1.4.4 $\overline{\text{IRQ}}$ 1-6
1.4.5 PA0-PA7 1-6
1.4.6 PB0-PB7/DA0-DA7 1-7
1.4.7 PC0-PC6/AD0 and PAMIN 1-7
1.4.8 PD0-PD7/SDIO, SCK, B, G, R, and FBKG 1-7
1.4.9 VSYN and HSYN 1-7
1.4.10 VCO and PLLGND 1-7
1.4.11 VIDEO 1-8

SECTION 2
MEMORY

2.1 Memory Map 2-1
2.2 Registers 2-2
2.3 RAM 2-3
2.4 ROM 2-4
2.4.1 User ROM 2-4
2.4.2 Internal ROM 2-4
2.5 OSD ROM 2-4

SECTION 3
CENTRAL PROCESSING UNIT

3.1 Registers 3-1
3.1.1 Accumulator (A) 3-2
3.1.2 Index Register (X) 3-2
3.1.3 Program Counter (PC) 3-2
3.1.4 Stack Pointer (SP) 3-2
3.1.5 Condition Code Register (CCR) 3-3
3.2 Addressing Modes 3-4
3.2.1 Immediate 3-4
3.2.2 Direct 3-4
3.2.3 Extended 3-4
3.2.4 Relative 3-5

TABLE OF CONTENTS

Section Title Page

3.2.5 Indexed, No Offset 3-5
3.2.6 Indexed, 8-Bit Offset 3-5
3.2.7 Indexed, 16-Bit Offset 3-5
3.2.8 Bit Set/Clear 3-6
3.2.9 Bit Test and Branch 3-6
3.2.10 Inherent 3-6

SECTION 4
INTERRUPTS

4.1 CPU Interrupt Processing 4-1
4.2 Reset 4-2
4.3 Software Interrupt (SWI) 4-3
4.4 External Interrupt 4-3
4.5 Pulse Accumulator Interrupts 4-3
4.6 On-Screen Display Interrupt 4-3
4.7 Closed-Caption Data Slicer Interrupt 4-4
4.8 Synchronous Serial Interface Interrupt 4-4
4.9 Multifunction Timer Interrupts 4-4

SECTION 5
RESETS

5.1 Power-On Reset (POR) 5-1
5.2 Computer Operating Properly (COP) Reset 5-1
5.3 Illegal Address (ILADR) Reset 5-2
5.4 $\overline{\text{RESET}}$ Pin 5-2

SECTION 6
LOW POWER MODES

6.1 STOP 6-1
6.2 WAIT 6-2

SECTION 7
INPUT/OUTPUT PORTS

7.1 Port A 7-1
7.2 Port B 7-1
7.3 Port C 7-1
7.4 Port D 7-1
7.5 Input/Output Programming 7-2

TABLE OF CONTENTS

Section **Title** **Page**

SECTION 8
PULSE ACCUMULATOR

8.1 Features 8-1
8.2 Overview 8-1
8.2.1 Pulse Measurement Mode 8-1
8.2.2 Pulse Counting Mode 8-1
8.3 Programming Guidelines 8-2
8.3.1 Setup 8-2
8.3.2 Interrupt Servicing 8-2
8.4 Input/Output 8-3
8.4.1 PAMIN 8-3
8.5 Registers 8-3
8.5.1 PAM Control Register (PAMCR) 8-3
8.5.2 PAM Status Register (PAMSR) 8-5
8.5.3 PAM Data Register (PAMDR) 8-6
8.6 Low Power Modes 8-6
8.6.1 Operation During WAIT Mode 8-6
8.6.2 Operation During STOP Mode 8-6
8.7 Interrupts and Resets 8-7

SECTION 9
SYNCHRONOUS SERIAL INTERFACE

9.1 Features 9-1
9.2 Overview 9-1
9.2.1 Master Mode 9-1
9.2.2 Slave Mode 9-2
9.3 Programming Guidelines 9-2
9.3.1 Setup 9-2
9.3.2 Interrupt Servicing 9-3
9.4 Input/Output 9-3
9.4.1 SCK 9-3
9.4.2 SDIO 9-3
9.5 Registers 9-4
9.5.1 SSI Control Register (SSICR) 9-4
9.5.2 SSI Status Register (SSISR) 9-7
9.5.3 SSI Data Register (SSIDR) 9-8
9.6 Low Power Modes 9-9
9.6.1 Operation During WAIT Mode 9-9
9.6.2 Operation During STOP Mode 9-9
9.7 Interrupts and Resets 9-9

TABLE OF CONTENTS

Section **Title** **Page**

SECTION 10
MULTIFUNCTION TIMER

10.1 Features 10-1
10.2 Overview 10-1
10.3 Programming Guidelines 10-2
10.3.1 Setup 10-2
10.3.2 Interrupt Servicing 10-2
10.4 Input/Output 10-3
10.5 Registers 10-3
10.5.1 MFT Control and Status Register (MFTCSR) 10-3
10.6 Low Power Modes 10-4
10.6.1 Operation During WAIT Mode 10-4
10.6.2 Operation During STOP Mode 10-4
10.7 Interrupts and Resets 10-5
10.7.1 Computer Operating Properly (COP) Watchdog Reset 10-5
10.7.2 MFT Interrupts 10-5

SECTION 11
ON-SCREEN DISPLAY

11.1 Features 11-1
11.2 Overview 11-2
11.2.1 Synchronization 11-2
11.2.2 Registers and ROM 11-3
11.2.3 OSD Output Logic 11-4
11.2.4 Display Characteristics 11-4
11.3 Programming Guidelines 11-9
11.3.1 Setup 11-9
11.3.2 Interrupt Servicing 11-10
11.3.3 Software Controlled Features 11-11
11.3.4 Character ROM 11-11
11.4 Input/Output 11-11
11.4.1 HSYNC and VSYNC 11-11
11.4.2 PLLGND and VCO 11-12
11.4.3 R, G, and B Outputs 11-12
11.4.4 FBKG 11-13
11.5 Registers 11-13
11.5.1 OSD Character Registers (OSDCH1-OSDCH34) 11-13
11.5.2 OSD Video Control Registers (OSDVCR1-OSDVCR2) 11-14
11.5.3 OSD Enable Control Register (OSDECR) 11-17
11.5.4 OSD Event Line Number Register (OSDELR) 11-19
11.5.5 OSD Event Count Register (OSDEVR) 11-19
11.5.6 OSD Output Control Register (OSDOCR) 11-20

TABLE OF CONTENTS

Section	Title	Page
11.5.7	OSD Horizontal Delay and Character Size Register (OSDHDR)	11-21
11.5.8	OSD Status Register	11-23
11.5.9	OSD Matrix Range Register (OSDMRR)	11-23
11.5.10	OSD Border Control Register (OSDBCR)	11-24
11.6	Low Power Modes	11-25
11.6.1	Operation During WAIT Mode	11-25
11.6.2	Operation During STOP Mode	11-25
11.7	Interrupts and Resets	11-25

SECTION 12
CLOSED-CAPTION DATA SLICER

12.1 Features 12-1
12.2 Overview 12-1
12.2.1 Slicer 12-2
12.2.2 Line and Field Detection 12-3
12.2.3 Data Sampling 12-3
12.3 Programming Guidelines 12-3
12.3.1 Setup 12-3
12.3.2 Interrupt Servicing 12-3
12.3.3 Debugging 12-4
12.4 Input/Output 12-5
12.4.1 Video 12-5
12.4.2 VSYNC, HSYNC, and Composite Video Input Requirements 12-5
12.5 Registers 12-7
12.5.1 DSL Character Registers (DSLCH1-DSLCH2) 12-7
12.5.2 DSL Control Register 1 (DSLCR1) 12-8
12.5.3 DSL Control Register 2 (DSLCR2) 12-9
12.5.4 DSL Status Register (DSLSR) 12-13
12.6 Low Power Modes 12-14
12.6.1 Operation During WAIT Mode 12-14
12.6.2 Operation During STOP Mode 12-14
12.7 Interrupts and Resets 12-14

SECTION 13
PULSE WIDTH MODULATOR

13.1 Features 13-1
13.2 Overview 13-2
13.3 Programming Guidelines 13-3
13.3.1 Setup 13-3

TABLE OF CONTENTS

Section	Title	Page
13.4	Input/Output	13-3
13.4.1	DA0-DA7	13-3
13.5	Registers	13-4
13.5.1	PWM Channel Registers (PWMC0-PWMC7)	13-4
13.5.2	PWM Enable Register (PWMEN)	13-4
13.6	Low Power Modes	13-5
13.6.1	Operation During WAIT Mode	13-5
13.6.2	Operation During STOP Mode	13-5
13.7	Interrupts and Resets	13-5

SECTION 14
A/D CONVERTER

Section	Title	Page
14.1	Features	14-1
14.2	Overview	14-1
14.3	Programming Guidelines	14-2
14.3.1	Setup	14-2
14.3.2	Conversions	14-2
14.4	Input/Output	14-2
14.4.1	ADCIN	14-2
14.5	Registers	14-3
14.5.1	ADC Control/Status Register (ADCCSR)	14-3
14.6	Low Power Modes	14-4
14.6.1	Operation During WAIT Mode	14-4
14.6.2	Operation During STOP Mode	14-4
14.7	Interrupts and Resets	14-4

SECTION 15
ELECTRICAL SPECIFICATIONS

Section	Title	Page
15.1	Maximum Ratings	15-1
15.2	Thermal Characteristics	15-1
15.3	DC Electrical Characteristics	15-2
15.4	Control Timing	15-3
15.5	On-Screen Display Timing	15-3

SECTION 16

MECHANICAL SPECIFICATIONS

Section	Title	Page
16.1	42-Pin SDIP Package (Case 858-01)	16-1
16.2	40-Pin DIP Package (Case 711-03)	16-1

TABLE OF CONTENTS

Section **Title** **Page**

SECTION 17
ORDERING INFORMATION

17.1 MCU Ordering Forms 17-1
17.2 Application Program Media 17-2
17.3 ROM Program Verification 17-3
17.4 ROM Verification Units (RVUs) 17-3
17.5 MC Order Numbers 17-4

LIST OF FIGURES

Figure	Title	Page
1-1	Block Diagram	1-2
1-2	Pin Assignments	1-4
1-3	Oscillator Connections	1-6
2-1	Memory Map	2-2
2-2	Register Organization	2-3
2-3	Registers \$0000-\$000F	2-5
2-4	Registers \$0010-\$001F	2-6
2-5	Registers \$0020-\$002F	2-7
2-6	Registers \$0100-\$010F	2-8
2-7	Registers \$0110-\$011F	2-9
2-8	Registers \$0120-\$12F	2-10
3-1	MC68HC05 Programming Model	3-1
3-2	Stacking Order	3-2
4-1	Interrupt Flowchart	4-5
5-1	Power-On Reset and $\overline{\text{RESET}}$	5-2
6-1	Stop Recovery Timing Diagram	6-1
6-2	STOP/WAIT Flowchart	6-3
7-1	Port I/O Circuitry	7-3
8-1	Pulse Accumulator Block Diagram	8-2
8-2	PAM Control Register (PAMCR)	8-3
8-3	PAM Status Register (PAMSR)	8-5
8-4	PAM Data Register (PAMDR)	8-6
9-1	SSI Block Diagram	9-2
9-2	Serial I/O Port Timing (CPOL=1)	9-4
9-3	Serial I/O Port Timing (CPOL=0)	9-4
9-4	SSI Control Register (SSICR)	9-5
9-5	SSI Status Register (SSISR)	9-7
9-6	SSI Data Register (SSIDR)	9-8
10-1	Multifunction Timer Block Diagram	10-2
10-2	MFT Control and Status Register (MFTCSR)	10-3
11-1	OSD Block Diagram	11-3
11-2	Character Display Styles	11-7
11-3	Display Boundary Conditions	11-8

LIST OF FIGURES

Figure	Title	Page
11-4	PLL Filter Circuit	11-12
11-5	OSD Character Registers (OSDCH1-34)	11-13
11-6	OSD Video Control Registers (OSDVCR1-OSDVCR2)	11-14
11-7	Video Control Code Formats	11-15
11-8	OSD Enable Control Register (OSDECR)	11-17
11-9	OSD Event Line Number Register (OSDELR)	11-19
11-10	OSD Event Count Register (OSDEVR)	11-19
11-11	OSD Output Control Register (OSDOCR)	11-20
11-12	OSD Horizontal Delay and Character Size Register (OSDHDR)	11-21
11-13	OSD Status Register (OSDSR)	11-23
11-14	OSD Matrix Range Register (OSDMRR)	11-23
11-15	OSD Border Control Register (OSDBCR)	11-24
12-1	Data Slicer Block Diagram	12-2
12-2	SLICER Input and Output	12-2
12-3	Video Input Circuit	12-5
12-4	DSL Input Timing	12-6
12-5	DSL Character Registers (DSLCH1-DSLCH2)	12-7
12-6	DSL Control Register 1 (DSLCR1)	12-8
12-7	DSL Control Register 2 (DSLCR2)	12-9
12-8	Data/Sync Slicing	12-10
12-9	Conditions for setting VPD	12-11
12-10	DSL Status Register (DSLSR)	12-13
13-1	PWM Block Diagram	13-2
13-2	PWM Output Waveform Examples	13-3
13-3	PWM Channel Registers (PWMC0-PWMC7)	13-4
13-4	PWM Enable Register (PWMEN)	13-4
14-1	ADC Block Diagram	14-1
14-2	ADC Control/Status Register (ADCCSR)	14-3

LIST OF TABLES

Table **Title** **Page**

4-1 Vector Address for Interrupts and Reset 4-2

7-1 I/O Pin Functions 7-2

8-1 Pulse Accumulator Prescaler Rates 8-5

9-1 SSI SCK Rates 9-7

10-1 RTI and minimum COP Reset Times 10-5

11-1 RBG Color Map 11-16
11-2 OSD Horizontal Delay 11-22

12-1 Line Selection 12-9
12-2 Data Slicer Bias Voltages 12-10
12-3 Minimum Vertical Sync Pulse Width 12-11
12-4 Phase Delay Selection 12-12

SECTION 1
GENERAL DESCRIPTION

The MC68HC05CC2 HCMOS microcomputer is a member of the M68HC05 Family. This part is suitable for applications which require closed-caption decoding and TV or VCR control. This 8-bit microcomputer unit (MCU) contains an on-chip oscillator, central processing unit (CPU), RAM, ROM, serial and parallel input/output (I/O), multifunction timer, analog-to-digital (A/D) converter, pulse-width modulation (PWM) digital-to-analog (D/A) converter, closed-caption data slicer and an on-screen display video controller. A functional block diagram of the MC68HC05CC2 is shown in Figure 1-1.

1.1 Features

- HC05 Core
- 4 MHz CPU Bus Operation
- 42-Pin Shrink Dual-in-Line (SDIP) Package
- 40-Pin Dual-in-Line (DIP) Package (pin-compatible with MC68HC05T2)
- 31 Bidirectional I/O Lines (eight open drain)
- 31,504 Bytes of User ROM
- 928 Bytes of RAM
- On-Screen Display (OSD) Controller with Closed-Caption Capability
- NTSC Closed-Caption Data Slicer (DSL) with Internal Sync Separator
- 28.2/36.3-MHz Phase-Locked Loop (PLL)
- 8-Bit Pulse Accumulator (PAM) for Infrared (IR) Decoding
- Synchronous Serial Interface (SSI) with I^{2}C Master Capability
- Eight Channels of 6-Bit Pulse Width Modulated (PWM) D/A Converter
- One Channel Resistive Ladder A/D Converter (ADC)

- Multifunction Timer (MFT) with 20 ms Periodic Interrupt
- Power Saving STOP and WAIT Modes
- STOP Instruction Disable Mask Option
- COP Watchdog Timer Disable Mask Option
- Edge-Sensitive or Edge-and-Level-Sensitive Interrupt Trigger Mask Option

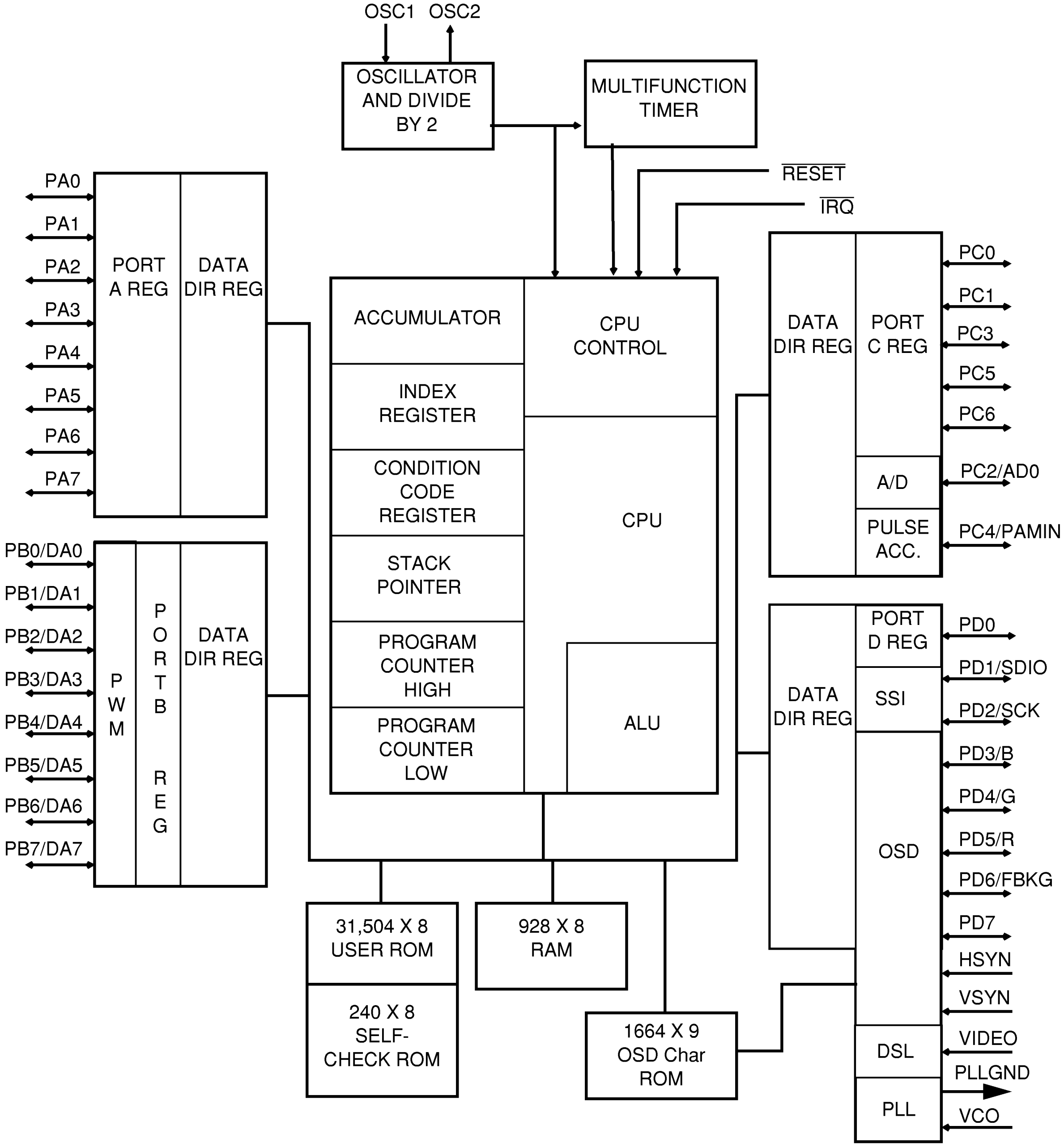

OSC1
OSC2
OSCILLATOR AND DIVIDE BY 2
MULTIFUNCTION TIMER
RESET
IRQ
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PORT A REG
DATA DIR REG
PB0/DA0
PB1/DA1
PB2/DA2
PB3/DA3
PB4/DA4
PB5/DA5
PB6/DA6
PB7/DA7
PWM
PORT B REG
DATA DIR REG
ACCUMULATOR
INDEX REGISTER
CONDITION CODE REGISTER
STACK POINTER
PROGRAM COUNTER HIGH
PROGRAM COUNTER LOW
CPU CONTROL
CPU
ALU
DATA DIR REG
PORT C REG
A/D
PULSE ACC.
PC0
PC1
PC3
PC5
PC6
PC2/AD0
PC4/PAMIN
DATA DIR REG
PORT D REG
SSI
OSD
DSL
PLL
PD0
PD1/SDIO
PD2/SCK
PD3/B
PD4/G
PD5/R
PD6/FBKG
PD7
HSYN
VSYN
VIDEO
PLLGND
VCO
31,504 X 8 USER ROM
240 X 8 SELF-CHECK ROM
928 X 8 RAM
1664 X 9 OSD Char ROM

Figure 1-1. Block Diagram

NOTE

A line over a signal name indicates an active low signal. For example, RESET is active high and $\overline{\mathrm{RESET}}$ is active low. Any reference to voltage, current, or frequency specified in the following sections will refer to the nominal values. The exact values and their tolerance or limits are specified in **SECTION 15 ELECTRICAL SPECIFICATIONS**.

1.2 Mask Options

The following mask options are available:

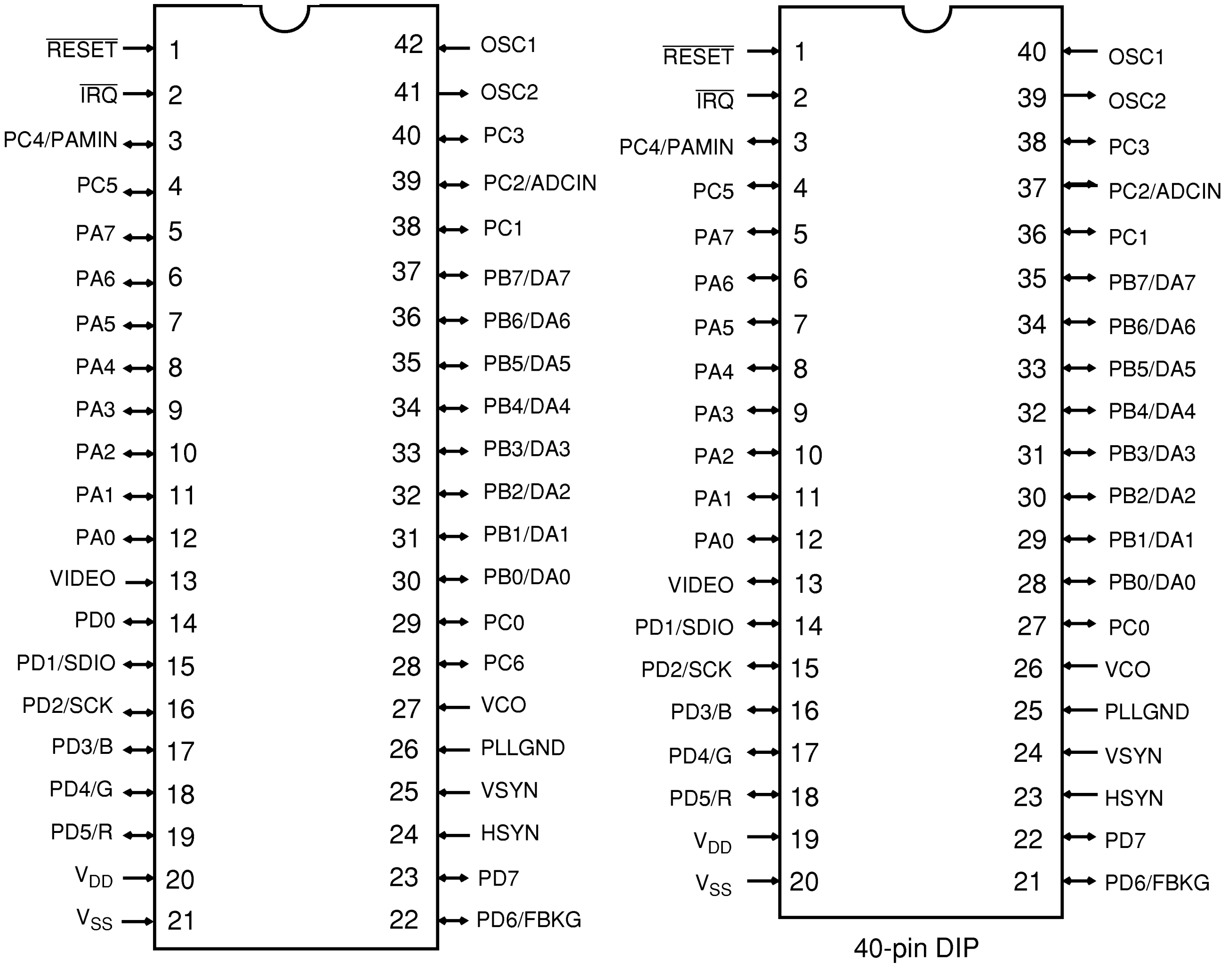
$\overline{\mathrm{IRQ}}$ Interrupt Sensitivity (edge-sensitive only or edge-and-level-sensitive)

Selectable COP Watchdog

Selectable STOP Instruction

1.3 Pin Assignments

The MC68HC05CC2 is available in the 42-pin SDIP and 40-pin DIP packages. The pin assignments for these packages are shown in Figure 1-2.


RESET
1
42
OSC1
IRQ
2
41
OSC2
PC4/PAMIN
3
40
PC3
PC5
4
39
PC2/ADCIN
PA7
5
38
PC1
PA6
6
37
PB7/DA7
PA5
7
36
PB6/DA6
PA4
8
35
PB5/DA5
PA3
9
34
PB4/DA4
PA2
10
33
PB3/DA3
PA1
11
32
PB2/DA2
PA0
12
31
PB1/DA1
VIDEO
13
30
PB0/DA0
PD0
14
29
PC0
PD1/SDIO
15
28
PC6
PD2/SCK
16
27
VCO
PD3/B
17
26
PLLGND
PD4/G
18
25
VSYN
PD5/R
19
24
HSYN
V_{DD}
20
23
PD7
V_{SS}
21
22
PD6/FBKG
RESET
1
40
OSC1
IRQ
2
39
OSC2
PC4/PAMIN
3
38
PC3
PC5
4
37
PC2/ADCIN
PA7
5
36
PC1
PA6
6
35
PB7/DA7
PA5
7
34
PB6/DA6
PA4
8
33
PB5/DA5
PA3
9
32
PB4/DA4
PA2
10
31
PB3/DA3
PA1
11
30
PB2/DA2
PA0
12
29
PB1/DA1
VIDEO
13
28
PB0/DA0
PD1/SDIO
14
27
PC0
PD2/SCK
15
26
VCO
PD3/B
16
25
PLLGND
PD4/G
17
24
VSYN
PD5/R
18
23
HSYN
V_{DD}
19
22
PD7
V_{SS}
20
21
PD6/FBKG

42-pin SDIP

40-pin DIP

Figure 1-2. Pin Assignments

1.4 Functional Pin Description

The following paragraphs give a description of the general function of each pin.

1.4.1 V_{DD} and V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply and V_{SS} is ground. The MCU operates from a single power supply.

Very fast signal transitions occur on the MCU pins. The short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, special care should be taken to provide good power supply bypassing at the MCU by using bypass capacitors with good high-frequency characteristics that are positioned as close to the MCU as possible.

1.4.2 OSC1 and OSC2

The OSC1 and OSC2 pins are the connections for the on-chip oscillator. The OSC1 and OSC2 pins can accept the following sets of components:

1. A crystal as shown in Figure 1-3.(a)
2. A ceramic resonator as shown in Figure 1-3.(a)
3. An external clock signal as shown in Figure 1-3.(b)

The frequency, f_{OSC}, of the oscillator or external clock source is divided by two to produce the internal operating frequency, f_{OP}.

1.4.2.1 Crystal Oscillator

The circuit in Figure 1-3.(a) shows a typical oscillator circuit for an AT-cut, parallel resonant crystal. The crystal manufacturer's recommendations should be followed, since the crystal parameters determine the external component values required to provide maximum stability and reliable start-up. The load capacitance values used in the oscillator circuit design should include all stray capacitances. The crystal and components should be mounted as close as possible to the pins for start-up stabilization and to minimize output distortion. An internal start-up resistor of approximately 2 MΩ is provided between OSC1 and OSC2 for the crystal-type oscillator.

1.4.2.2 Ceramic Resonator Oscillator

In cost-sensitive applications, a ceramic resonator can be used in place of the crystal. The circuit in Figure 1-3.(a) is also applicable for a ceramic resonator. The resonator manufacturer's recommendations should be followed, as the resonator parameters determine the external component values required for maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray capacitances. The ceramic resonator and components should be mounted as close as possible to the pins for start-up stabilization and to minimize output distortion. An internal start-up resistor of approximately 2 MΩ is provided between OSC1 and OSC2 for the ceramic resonator type oscillator.

1.4.2.3 External Clock

An external clock from another complementary metal-oxide semiconductor (CMOS) compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in Figure 1-3.(b).

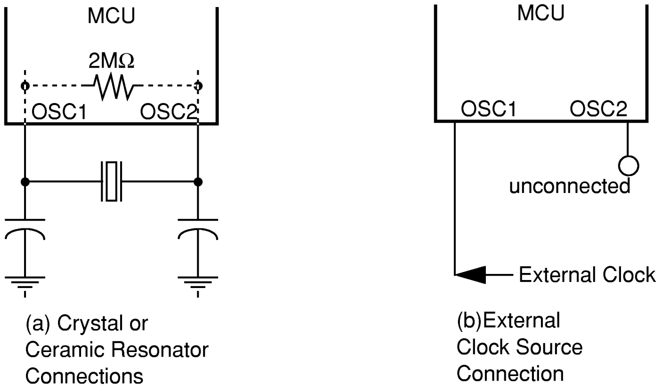

MCU
2MΩ
OSC1
OSC2
(a) Crystal or Ceramic Resonator Connections
MCU
OSC1
OSC2
unconnected
External Clock
(b)External Clock Source Connection

Figure 1-3. Oscillator Connections

1.4.3 $\overline{\text{RESET}}$

This active low input-only pin is used to reset the MCU to a known start-up state. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **SECTION 5 RESETS** for more information.

1.4.4 $\overline{\text{IRQ}}$

This input pin drives the asynchronous IRQ interrupt function of the CPU. The IRQ interrupt function has a mask option to select either negative edge-sensitive triggering or both negative edge-sensitive and low level-sensitive triggering. When the former method is used, the interrupt will be cleared internally; when the latter option is selected, the interrupt will not be cleared until the rising edge of the $\overline{\text{IRQ}}$ pin. If $\overline{\text{IRQ}}$ is not used, it must be tied to the V_{DD} supply. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. See **SECTION 4 INTERRUPTS** for more information.

1.4.5 PA0-PA7

These eight I/O lines comprise port A. The state of any pin is software programmable and all port A lines are configured as inputs during power-on or reset. See **SECTION 7 INPUT/OUTPUT PORTS** for more information.

1.4.6 PB0-PB7 / DA0-DA7

These eight I/O lines comprise port B. The state of any pin is software programmable and all port B lines are configured as inputs during power-on or reset. See **SECTION 7 INPUT/OUTPUT PORTS** for more information. All of the port B pins are shared with the PWM D/A subsystem. See **SECTION 13 PULSE WIDTH MODULATOR** for a detailed description of the PWM. These eight pins have open-drain outputs.

1.4.7 PC0-PC6/AD0 and PAMIN

These seven I/O lines comprise port C. The state of any pin is software programmable and all port C lines are configured as inputs during power-on or reset. See **SECTION 7 INPUT/OUTPUT PORTS** for more information. PC2 is shared with the ADC subsystem. See **SECTION 14 A/D CONVERTER** for a detailed description of the ADC. PC4 is shared with the PAM subsystem. See **SECTION 8 PULSE ACCUMULATOR** for a detailed description of the PAM. PC0, PC1, PC3, PC5, and PC6 are general-purpose I/O pins. PC6 is not externally available on the 40-pin package.

1.4.8 PD0-PD7/SDIO, SCK, B, G, R, and FBKG

These eight I/O lines comprise port D. The state of any pin is software programmable and all port D lines are configured as inputs during power-on or reset. See **SECTION 7 INPUT/OUTPUT PORTS** for a detailed description of I/O programming. PD1 and PD2 are shared with the SSI subsystem. See **SECTION 9 SYNCHRONOUS SERIAL INTERFACE** for a detailed description of the SSI. PD3, PD4, PD5, and PD6 are shared with the OSD subsystem. See **SECTION 11 ON-SCREEN DISPLAY** for a detailed description of the OSD. PD0 and PD7 are general-purpose I/O pins. PD0 is not externally available on the 40-pin package.

1.4.9 VSYN and HSYN

These two input-only pins are used by the OSD subsystem to synchronize to the television. See **SECTION 11 ON-SCREEN DISPLAY** for a detailed description of the OSD. These two pins contain internal Schmitt triggers as part of their inputs to improve noise immunity.

1.4.10 VCO and PLLGND

These two pins are used by the PLL in the OSD subsystem. An external filter should be tied to these pins for proper operation of the PLL. See **SECTION 11 ON-SCREEN DISPLAY** for a detailed description of the OSD and PLL.

1.4.11 VIDEO

This input-only pin is the National Television System Committee (NTSC) composite base-band video input signal for the DSL subsystem. See **SECTION 12 CLOSED-CAPTION DATA SLICER** for a detailed description of the DSL.

SECTION 2
MEMORY

The MC68HC05CC2 has a 32-Kbyte memory map.

2.1 Memory Map

The memory map consists of 96 bytes of registers, 928 bytes of RAM, 31,504 bytes of user ROM and 240 bytes of self-check ROM. The OSD ROM is not available to the CPU. See Figure 2-1.

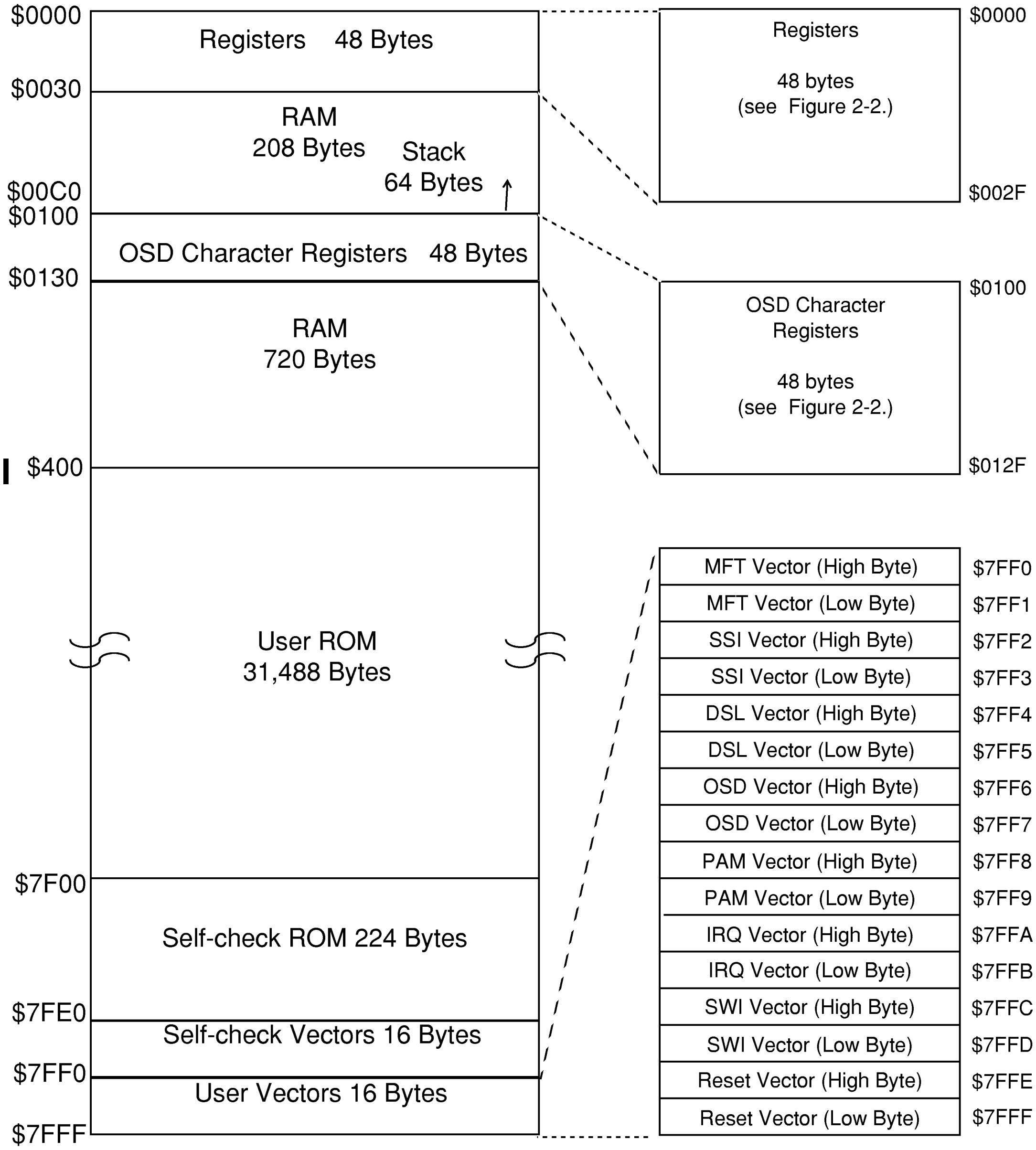

$0000
Registers 48 Bytes
$0030
RAM
208 Bytes
Stack
64 Bytes
$00C0
$0100
OSD Character Registers 48 Bytes
$0130
RAM
720 Bytes
$400
User ROM
31,488 Bytes
$7F00
Self-check ROM 224 Bytes
$7FE0
Self-check Vectors 16 Bytes
$7FF0
User Vectors 16 Bytes
$7FFF
Registers
48 bytes
(see Figure 2-2.)
$0000
$002F
OSD Character
Registers
48 bytes
(see Figure 2-2.)
$0100
$012F
MFT Vector (High Byte) $7FF0
MFT Vector (Low Byte) $7FF1
SSI Vector (High Byte) $7FF2
SSI Vector (Low Byte) $7FF3
DSL Vector (High Byte) $7FF4
DSL Vector (Low Byte) $7FF5
OSD Vector (High Byte) $7FF6
OSD Vector (Low Byte) $7FF7
PAM Vector (High Byte) $7FF8
PAM Vector (Low Byte) $7FF9
IRQ Vector (High Byte) $7FFA
IRQ Vector (Low Byte) $7FFB
SWI Vector (High Byte) $7FFC
SWI Vector (Low Byte) $7FFD
Reset Vector (High Byte) $7FFE
Reset Vector (Low Byte) $7FFF

Figure 2-1. Memory Map

2.2 Registers

The registers reside in locations $0000-$002F and $0100-$012F. The overall organization of these registers is shown in Figure 2-1. The bit assignments for each register are shown in Figure 2-3 through Figure 2-7. Reading from unimplemented bits will return unknown states, and writing to unimplemented bits will be ignored. Reserved spaces should not be read or modified.

Port A Data	$0000
Port B Data	$0001
Port C Data	$0002
Port D Data	$0003
Port A Data Direction	$0004
Port B Data Direction	$0005
Port C Data Direction	$0006
Port D Data Direction	$0007
PWM Channel 0	$0008
PWM Channel 1	$0009
PWM Channel 2	$000A
PWM Channel 3	$000B
PWM Channel 4	$000C
PWM Channel 5	$000D
PWM Channel 6	$000E
PWM Channel 7	$000F
PWM Enable	$0010
unimplemented	$0011
RESERVED	$0012
DSL Control 1	$0013
DSL Control 2	$0014
DSL Status	$0015
DSL Character 1	$0016
DSL Character 2	$0017
PAM Control	$0018
PAM Status	$0019
PAM Data	$001A
SSI Control	$001B
SSI Status	$001C
SSI Data	$001D
unimplemented	$001E
unimplemented	$001F
unimplemented	$0020
unimplemented	$0021
OSD Enable Control	$0022
OSD Event Line	$0023
OSD Output Control	$0024
OSD Horizontal Delay	$0025
OSD Status	$0026
OSD Matrix Range	$0027
OSD Border Control	$0028
OSD Event Count	$0029
RESERVED	$002A
ADC Control/Status	$002B
unimplemented	$002C
MFT Control/Status	$002D
unimplemented	$002E
RESERVED	$002F

$0010 - $002F

OSD Character 1	$0100
OSD Character 2	$0101
OSD Character 3	$0102
OSD Character 4	$0103
OSD Character 5	$0104
OSD Character 6	$0105
OSD Character 7	$0106
OSD Character 8	$0107
OSD Character 9	$0108
OSD Character 10	$0109
OSD Character 11	$010A
OSD Character 12	$010B
OSD Character 13	$010C
OSD Character 14	$010D
OSD Character 15	$010E
OSD Character 16	$010F
OSD Character 17	$0110
OSD Character 18	$0111
OSD Character 19	$0112
OSD Character 20	$0113
OSD Character 21	$0114
OSD Character 22	$0115
OSD Character 23	$0116
OSD Character 24	$0117
OSD Character 25	$0118
OSD Character 26	$0119
OSD Character 27	$011A
OSD Character 28	$011B
OSD Character 29	$011C
OSD Character 30	$011D
OSD Character 31	$011E
OSD Character 32	$011F
OSD Character 33	$0120
OSD Character 34	$0121
OSD Video Control 1	$0122
OSD Video Control 2	$0123
unimplemented	$0124
unimplemented	$0125
unimplemented	$0126
unimplemented	$0127
unimplemented	$0128
unimplemented	$0129
unimplemented	$012A
unimplemented	$012B
unimplemented	$012C
unimplemented	$012D
unimplemented	$012E
unimplemented	$012F

$0100-$012F

Figure 2-2. Register Organization

2.3 RAM

The RAM consists of 928 bytes divided into two segments. The first RAM segment, including the stack area, is from $0030 through $00FF. The second RAM segment is from $0130 through $03FF. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM in the range $00FF to $00C0. See **3.1.4 Stack Pointer (SP)**.

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.

2.4 ROM

The MC68HC05CC2 contains 31-Kbytes of CPU-accessible ROM, which is divided into user ROM and internal ROM.

2.4.1 User ROM

The user ROM consists of 31,504 bytes of ROM: 31,488 bytes of program space from $400 through $7EFF and 16 bytes of user vectors from $7FF0 through $7FFF.

All of the user vectors, from $7FF0 through $7FFF, are dedicated to reset and interrupt vectors.

2.4.2 Internal ROM

The internal, or fixed data, section of the ROM is used for factory testing. The self-check ROM consists of 224 bytes of code from $7F00 through $7FDF and 16 bytes of vectors from $7FE0 through $7FEF.

2.5 OSD ROM

The OSD ROM consists of 128 9 x 13 bit matrices, or 14,976 bits. This ROM is used by the OSD to define the characters which can be displayed on the television screen. The OSD ROM is not accessible by the CPU.

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0000	PORT A DATA PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		W								
$0001	PORT B DATA PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		W								
$0002	PORT C DATA PORTC	R	0	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		W	░							
$0003	PORT D DATA PORTD	R	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
		W								
$0004	PORT A DATA DIRECTION DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
$0005	PORT B DATA DIRECTION DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
$0006	PORT C DATA DIRECTION DDRC	R	0	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W	░							
$0007	PORT D DATA DIRECTION DDRD	R	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		W								
$0008	PWM CHANNEL 0 PWMC0	R	░	░	░	░	░	░	░	░
		W			PW5	PW4	PW3	PW2	PW1	PW0
$0009	PWM CHANNEL 1 PWMC1	R	░	░	░	░	░	░	░	░
		W			PW5	PW4	PW3	PW2	PW1	PW0
$000A	PWM CHANNEL 2 PWMC2	R	░	░	░	░	░	░	░	░
		W			PW5	PW4	PW3	PW2	PW1	PW0
$000B	PWM CHANNEL 3 PWMC3	R	░	░	░	░	░	░	░	░
		W			PW5	PW4	PW3	PW2	PW1	PW0
$000C	PWM CHANNEL 4 PWMC4	R	░	░	░	░	░	░	░	░
		W			PW5	PW4	PW3	PW2	PW1	PW0
$000D	PWM CHANNEL 5 PWMC5	R	░	░	░	░	░	░	░	░
		W			PW5	PW4	PW3	PW2	PW1	PW0
$000E	PWM CHANNEL 6 PWMC6	R	░	░	░	░	░	░	░	░
		W			PW5	PW4	PW3	PW2	PW1	PW0
$000F	PWM CHANNEL 7 PWMC7	R	░	░	░	░	░	░	░	░
		W			PW5	PW4	PW3	PW2	PW1	PW0

UNIMPLEMENTED ░ RESERVED ▨

Figure 2-3. Registers $0000-$000F

ADDR	REGISTER	READ/WRITE	7	6	5	4	3	2	1	0
$0010	PWM ENABLE PWMEN	R	▒	▒	▒	▒	▒	▒	▒	▒
		W	CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0
$0011	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$0012	RESERVED	R	▧	▧	▧	▧	▧	▧	▧	▧
		W								
$0013	DSL CONTROL 1 DSLCR1	R	DSIEN	DSEN	LINE5	LINE4	LINE3	LINE2	LINE1	0
		W								▒
$0014	DSL CONTROL 2 DSLCR2	R	VR1	VR0	PW1	PW0	VPD	PD2	PD1	PD0
		W								
$0015	DSL STATUS DSLSR	R	DSFL	OVFL	FIELD1	INTLC	CSYNC	VPDET	RIC1	RIC0
		W			▒	▒	▒			
$0016	DSL CHARACTER 1 DSLCH1	R	PE	DA6	DA5	DA4	DA3	DA2	DA1	DA0
		W	▒	▒	▒	▒	▒	▒	▒	▒
$0017	DSL CHARACTER 2 DSLCH2	R	PE	DA6	DA5	DA4	DA3	DA2	DA1	DA0
		W	▒	▒	▒	▒	▒	▒	▒	▒
$0018	PAM CONTROL PAMCR	R	PAFIE	PAOVIE	PAPOL	PAEN	PAMODE	0	PAPR1	PAPR0
		W						PARST		
$0019	PAM STATUS PAMSR	R	PAF	PAOVF	PAPIN	0	0	0	0	0
		W	▒		▒	▒	▒	▒	▒	▒
$001A	PAM DATA PAMDR	R	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
		W	▒	▒	▒	▒	▒	▒	▒	▒
$001B	SSI CONTROL SSICR	R	SIE	SE	LSBF	MSTR	CPOL	T/$\overline{R}$	SR1	SR0
		W								
$001C	SSI STATUS SSISR	R	SF	DCOL	0	0	0	0	0	0
		W	▒	▒	▒	▒	▒	▒	▒	▒
$001D	SSI DATA SSIDR	R	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
		W								
$001E	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$001F	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								

UNIMPLEMENTED ▒ RESERVED ▧

Figure 2-4. Registers $0010-$001F

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0020	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W	▒	▒	▒	▒	▒	▒	▒	▒
$0021	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W	▒	▒	▒	▒	▒	▒	▒	▒
$0022	OSD ENABLE CONTROL OSDECR	R	OIEN	OSDEN	XFER	PLLEN	FBKGEN	PLL36	BLNKEN	HMODE
		W								
$0023	OSD EVENT LINE OSDELR	R	ELN7	ELN6	ELN5	ELN4	ELN3	ELN2	ELN1	ELN0
		W								
$0024	OSD OUTPUT CONTROL OSDOCR	R	HINV	VINV	CINV	FBINV	FDINV	REN	BEN	GEN
		W								
$0025	OSD HORIZONTAL DELAY OSDHDR	R	CHHS	CHWS	HD5	HD4	HD3	HD2	HD1	HD0
		W								
$0026	OSD STATUS OSDSR	R	OSDFL	HSYN	VSYN	0	0	0	0	0
		W		▒	▒	▒	▒	▒	▒	▒
$0027	OSD MATRIX RANGE OSDMRR	R	MS3	MS2	MS1	MS0	ME3	ME2	ME1	ME0
		W								
$0028	OSD BORDER CONTROL OSDBCR	R	0	0	0	0	BOS	BOR	BOB	BOG
		W	▒	▒	▒	▒				
$0029	OSD EVENT COUNT OSDEVR	R	EV7	EV6	EV5	EV4	EV3	EV2	EV1	EV0
		W	▒	▒	▒	▒	▒	▒	▒	▒
$002A	RESERVED	R	▨	▨	▨	▨	▨	▨	▨	▨
		W	▨	▨	▨	▨	▨	▨	▨	▨
$002B	ADC CONTROL/STATUS ADCCSR	R	RESULT	ADON	0	AD4	AD3	AD2	AD1	AD0
		W	▒		▒					
$002C	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W	▒	▒	▒	▒	▒	▒	▒	▒
$002D	MFT CONTROL/STATUS MFTCSR	R	TOF	RTIF	TOFE	RTIE	0	0	0	0
		W	▒	▒			TOFC	RTIFC	▒	▒
$002E	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W	▒	▒	▒	▒	▒	▒	▒	▒
$002F	RESERVED	R	▨	▨	▨	▨	▨	▨	▨	▨
		W	▨	▨	▨	▨	▨	▨	▨	▨

UNIMPLEMENTED ▒ RESERVED ▨

Figure 2-5. Registers $0020-$002F

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0100	OSD CHARACTER 1 OSDCH1	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0101	OSD CHARACTER 2 OSDCH2	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0102	OSD CHARACTER 3 OSDCH3	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0103	OSD CHARACTER 4 OSDCH4	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0104	OSD CHARACTER 5 OSDCH5	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0105	OSD CHARACTER 6 OSDCH6	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0106	OSD CHARACTER 7 OSDCH7	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0107	OSD CHARACTER 8 OSDCH8	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0108	OSD CHARACTER 9 OSDCH9	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0109	OSD CHARACTER 10 OSDCH10	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$010A	OSD CHARACTER 11 OSDCH11	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$010B	OSD CHARACTER 12 OSDCH12	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$010C	OSD CHARACTER 13 OSDCH13	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$010D	OSD CHARACTER 14 OSDCH14	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$010E	OSD CHARACTER 15 OSDCH15	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$010F	OSD CHARACTER 16 OSDCH16	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								

UNIMPLEMENTED ▒ RESERVED ▨

Figure 2-6. Registers $0100-$010F

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0110	OSD CHARACTER 17 OSDCH17	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0111	OSD CHARACTER 18 OSDCH18	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0112	OSD CHARACTER 19 OSDCH19	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0113	OSD CHARACTER 20 OSDCH20	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0114	OSD CHARACTER 21 OSDCH21	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0115	OSD CHARACTER 22 OSDCH22	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0116	OSD CHARACTER 23 OSDCH23	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0117	OSD CHARACTER 24 OSDCH24	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0118	OSD CHARACTER 25 OSD CH25	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0119	OSD CHARACTER 26 OSDCH26	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$011A	OSD CHARACTER 27 OSDCH27	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$011B	OSD CHARACTER 28 OSDCH28	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$011C	OSD CHARACTER 29 OSDCH29	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$011D	OSD CHARACTER 30 OSDCH30	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$011E	OSD CHARACTER 31 OSDCH31	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$011F	OSD CHARACTER 32 OSDCH32	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								

UNIMPLEMENTED ▒   RESERVED ▨

Figure 2-7. Registers $0110-$011F

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0120	OSD CHARACTER 33 OSDCH33	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0121	OSD CHARACTER 34 OSDCH34	R	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
		W								
$0122	OSD VIDEO CONTROL 1 OSDVCR1	R	SEL1	SEL0	VC5	VC4	VC3	VC2	VC1	VC0
		W								
$0123	OSD VIDEO CONTROL 2 OSDVCR2	R	SEL1	SEL0	VC5	VC4	VC3	VC2	VC1	VC0
		W								
$0124	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$0125	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$0126	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$0127	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$0128	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$0129	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$012A	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$012B	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$012C	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$012D	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$012E	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								
$012F	unimplemented	R	▒	▒	▒	▒	▒	▒	▒	▒
		W								

UNIMPLEMENTED ▒ RESERVED ▨

Figure 2-8. Registers $0120-$12F

SECTION 3
CENTRAL PROCESSING UNIT

The MC68HC05CC2 has a 32K memory map; therefore, it does not use the most significant bit of the address bus. In the following discussion, bit 15 of the address bus can be ignored.

3.1 Registers

The MCU contains five registers which are hard-wired within the CPU and are not part of the memory map. These five registers are shown in Figure 3-1. The interrupt stacking order is shown in Figure 3-2.

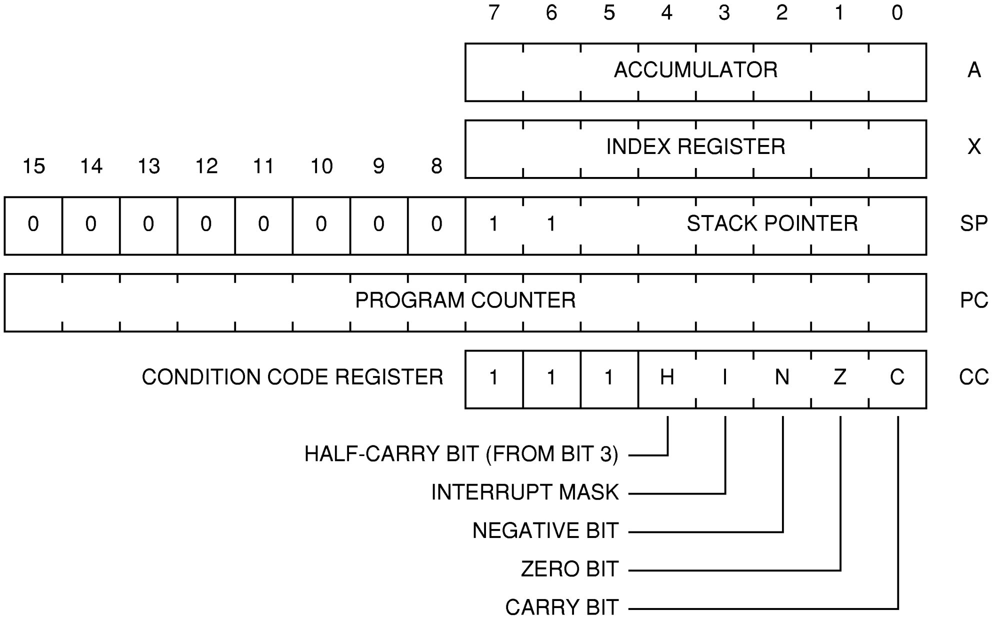
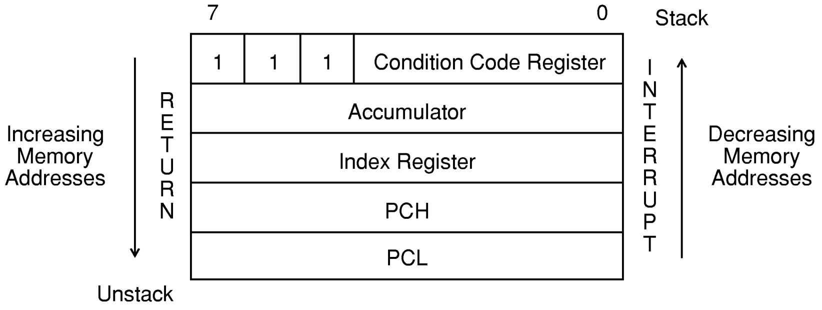

7 6 5 4 3 2 1 0
ACCUMULATOR
A
INDEX REGISTER
X
15 14 13 12 11 10 9 8
0 0 0 0 0 0 0 0 1 1
STACK POINTER
SP
PROGRAM COUNTER
PC
CONDITION CODE REGISTER
1 1 1 H I N Z C
CC
HALF-CARRY BIT (FROM BIT 3)
INTERRUPT MASK
NEGATIVE BIT
ZERO BIT
CARRY BIT

Figure 3-1. MC68HC05 Programming Model



Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 3-2. Stacking Order

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. The accumulator is unaffected by a reset of the device.

3.1.2 Index Register (X)

The index register is an 8-bit register used for the indexed addressing value to create an effective address. The index register may also be used as a temporary storage area. The index register is unaffected by a reset of the device.

3.1.3 Program Counter (PC)

The program counter is a 16-bit register that contains the address of the next instruction or operand byte to be fetched. Normally, the program counter increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch and interrupt operations load the program counter with an address other than that of the next sequential location.

3.1.4 Stack Pointer (SP)

The stack pointer contains the address of the next free location on the stack. During 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 10 most significant bits are permanently set to 0000000011. These 10 bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses 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 condition code register is a 5-bit register in which 4 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. Only the interrupt mask is affected by a reset of the device. The following paragraphs explain the functions of the lower five bits of the condition code register.

3.1.5.1 Half Carry (H)

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

3.1.5.2 Interrupt (I)

When this bit is set, all interrupts (external and internal) with the exception of software interrupt instruction (SWI) are masked (disabled). Interrupts are enabled when the interrupt mask is cleared. When an interrupt occurs, the interrupt mask is automatically set after the CPU registers are saved on the stack, but before the interrupt vector is fetched. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the interrupt is processed as soon as the interrupt mask is cleared.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its state before the interrupt was encountered. After any reset, the interrupt mask is set and can be cleared only by the clear I bit (CLI), STOP, or WAIT instructions.

3.1.5.3 Negative (N)

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

3.1.5.4 Zero (Z)

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

3.1.5.5 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 Addressing Modes

The MCU uses 10 different addressing modes to provide the programmer with an opportunity to optimize the 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 (3 bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or 2-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term effective address (EA) is used in describing the various addressing modes. Effective address is defined as the address at which the argument for an instruction is fetched or stored.

3.2.1 Immediate

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

3.2.2 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 2-byte instruction.

3.2.3 Extended

In the extended addressing mode, the effective address of the argument is contained in the 2 bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-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 shortest form of the instruction.

3.2.4 Relative

The relative addressing mode is used only in branch instructions. In relative addressing, the contents of the 8-bit signed offset byte, which is the last byte of the instruction, is 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 -128 to +127 from the address of the next opcode. 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.

3.2.5 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 1 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.

3.2.6 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 byte. The addressing mode is useful for selecting the Kth element in an n element table. With this 2-byte instruction, K would typically be in X with the address of the beginning of the table in the second byte of the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE). This is the last location which can be accessed in this way.

3.2.7 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 2 unsigned bytes following the opcode byte. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this 3-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.

3.2.8 Bit Set/Clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing 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 2-byte instruction.

3.2.9 Bit Test and Branch

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is 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. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-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 branching is from -128 to +127 from the address of the next opcode. The state of the tested bit is also transferred to the carry bit of the condition code register.

3.2.10 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 and/or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are 1 byte long.

SECTION 4
INTERRUPTS

The MC68HC05CC2 can be interrupted seven ways:

- Non-maskable Software Interrupt Instruction (SWI)
- External Asynchronous Interrupt ($\overline{\text{IRQ}}$)
- Synchronous Serial Interface (SSI)
- On-Screen Display (OSD)
- Closed-Caption Data Slicer (DSL)
- Multifunction Timer (MFT)
- Pulse Accumulator (PAM)

All of the internal and external hardware interrupts are maskable; the SWI instruction is non-maskable.

4.1 CPU Interrupt Processing

Interrupts are used to temporarily suspend the normal flow of processing to immediately service a module or external peripheral. Figure 4-1 shows the sequence of events that occur during interrupt processing.

The SSI, OSD, DSL, MFT, and PAM subsystems are capable of issuing interrupts to the CPU. These interrupts are generated due to a change in status when enabled by corresponding interrupt enable bits.

External peripherals may interrupt the CPU through the $\overline{\text{IRQ}}$ input.

Unlike a reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete. At that time, if interrupts are not masked (CCR I bit clear), the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

The SWI is executed the same as any other instruction, regardless of the I-bit state.

An interrupt, if it is not masked by the CCR I bit, causes the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. If more than one interrupt is pending following the stacking operation, the interrupt with the highest vector location (priority) shown in Table 4-1 will be serviced first.

When the I bit is set, all maskable interrupts (internal and external) are disabled. Clearing the I bit enables interrupts. Unless the CLI instruction is executed within an interrupt service routine, only the SWI will be able to disrupt interrupt servicing.

The RTI instruction is used to signify when an interrupt service routine has been completed. This instruction causes the register contents to be recovered from the stack and normal processing to resume at the next instruction that was to be executed when the interrupt took place. The I bit will be restored to its previous state.

Table 4-1. Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	$\overline{\text{RESET}}$	\$7FFE-\$7FFF
N/A	N/A	Software	SWI	\$7FFC-\$7FFD
N/A	N/A	External Interrupt	$\overline{\text{IRQ}}$	\$7FFA-\$7FFB
PAMSR	PAF	Pulse Accumulator	PAM	\$7FF8-\$7FF9
PAMSR	PAOVF	Pulse Accumulator	PAM	\$7FF8-\$7FF9
OSDSR	OSDFL	On Screen Display	OSD	\$7FF6-\$7FF7
DSLSR	DSFL	CC Data Slicer	DSL	\$7FF4-\$7FF5
SSISR	SF	Sync. Serial Interface	SSI	\$7FF2-\$7FF3
TCSR	TOF	Multifunction Timer	MFT	\$7FF0-\$7FF1
TCSR	RTIF	Multifunction Timer	MFT	\$7FF0-\$7FF1

4.2 Reset

The reset function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in Figure 4-1. A low level input on the $\overline{\text{RESET}}$ pin or the internally generated RST signal causes the program to vector to its starting address which is specified by the contents of memory locations \$7FFE and \$7FFF. The I bit in the condition code register is also set. The MCU is configured to a known state during this type of reset as described in **SECTION 5 RESETS**.

4.3 Software Interrupt (SWI)

The SWI is an executable instruction and a non-maskable interrupt since it is executed regardless of the state of the I bit in the CCR. If the I bit is zero (interrupts enabled), SWI executes after interrupts which were pending before the SWI was fetched, but before interrupts generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations \$7FFC and \$7FFD.

4.4 External Interrupt

The interrupt request is latched immediately following the falling edge of $\overline{\text{IRQ}}$. It is then serviced as specified by the contents of \$7FFA and \$7FFB.

Either an edge-and-level-sensitive trigger or an edge-sensitive-only trigger is available as a mask option. If edge-only sensitivity is selected, the interrupt will be cleared internally when the $\overline{\text{IRQ}}$ interrupt vector is fetched. If edge-and-level sensitivity is selected, the interrupt will not be cleared until the rising edge of $\overline{\text{IRQ}}$.

NOTE

The internal interrupt latch is automatically 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.

4.5 Pulse Accumulator Interrupts

Two pulse accumulator (PAM) interrupt flags will cause an interrupt whenever they are set and enabled. The interrupt flags are located in the PAM status register (PAMSR) and the enable bits are located in the PAM control register (PAMCR). Either of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory locations \$7FF8 and \$7FF9. See **SECTION 8 PULSE ACCUMULATOR** for more information on PAM interrupts.

4.6 On-Screen Display Interrupt

One on-screen display (OSD) interrupt flag will cause an interrupt whenever it is set and enabled. The interrupt flag is located in the OSD status register (OSDSR) and the enable bit is located in the OSD enable control register (OSDECR). This interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations \$7FF6 and \$7FF7. See **SECTION 11 ON-SCREEN DISPLAY** for more information on the OSD interrupt.

4.7 Closed-Caption Data Slicer Interrupt

One closed-caption data slicer (DSL) interrupt flag will cause an interrupt whenever it is set and enabled. The interrupt flag is located in the DSL status register (DSLSR) and the enable bit is located in the DSL control register 1 (DSLCR1). This interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations $7FF4 and $7FF5. See **SECTION 12 CLOSED-CAPTION DATA SLICER** for more information on the DSL interrupt.

4.8 Synchronous Serial Interface Interrupt

One synchronous serial interface (SSI) interrupt flag will cause an interrupt whenever it is set and enabled. The interrupt flag is located in the SSI status register (SSISR) and the enable bit is located in the SSI control register (SSICR). The interrupt will vector to the interrupt service routine located at the address specified by the contents of memory locations $7FF2 and $7FF3. See **SECTION 9 SYNCHRONOUS SERIAL INTERFACE** for more information on the SSI interrupt.

4.9 Multfunction Timer Interrupts

Two multifunction timer (MFT) interrupt flags will cause an interrupt whenever they are set and enabled. The interrupt flags and enable bits are located in the MFT control and status register (MFT CSR). Either of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory locations $7FF0 and $7FF1. See **SECTION 10 MULTIFUNCTION TIMER** for more information on MFT interrupts.

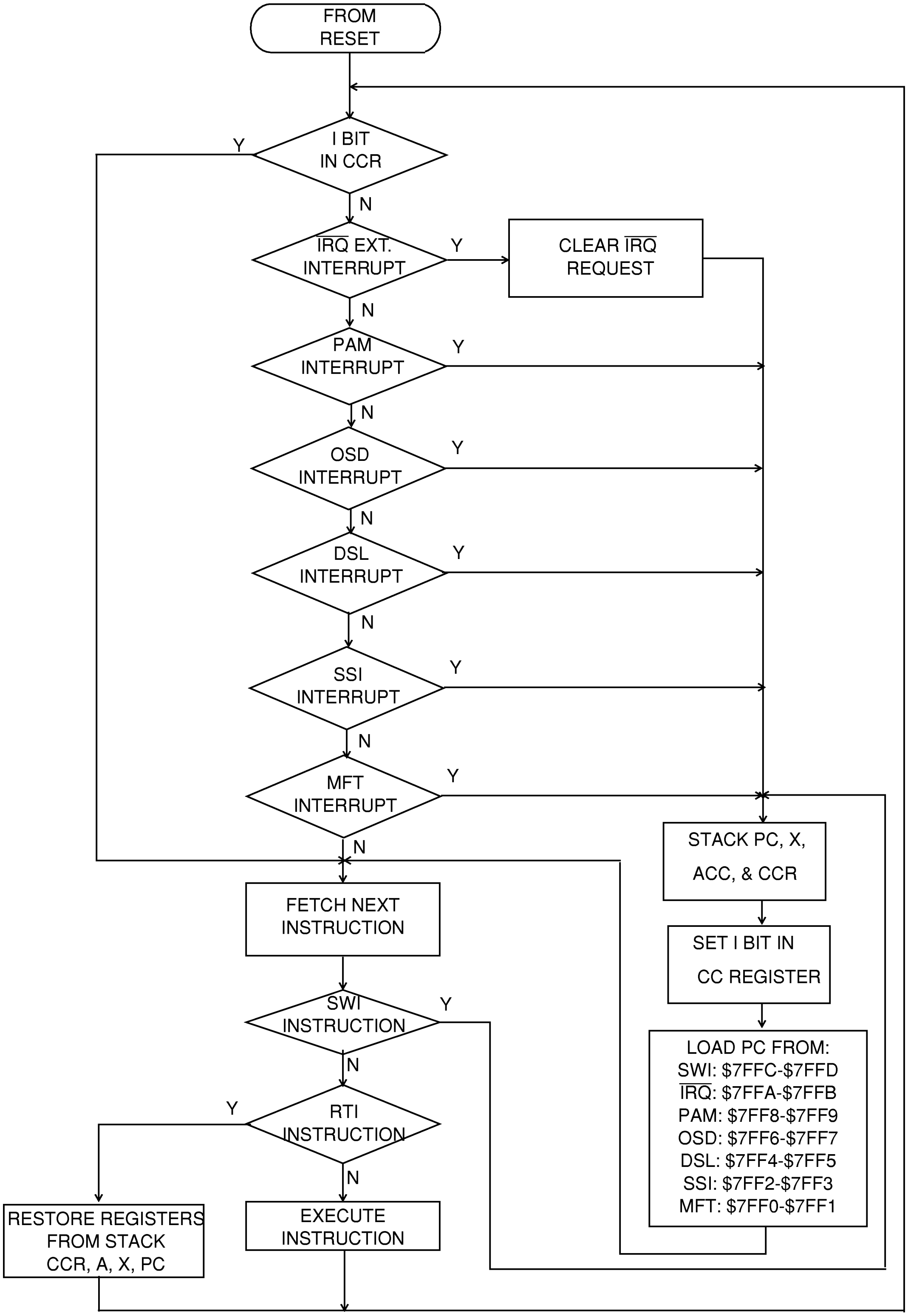

FROM RESET
I BIT IN CCR
Y
N
IRQ EXT. INTERRUPT
Y
CLEAR IRQ REQUEST
N
PAM INTERRUPT
Y
N
OSD INTERRUPT
Y
N
DSL INTERRUPT
Y
N
SSI INTERRUPT
Y
N
MFT INTERRUPT
Y
N
STACK PC, X, ACC, & CCR
SET I BIT IN CC REGISTER
LOAD PC FROM:
SWI: $7FFC-$7FFD
IRQ: $7FFA-$7FFB
PAM: $7FF8-$7FF9
OSD: $7FF6-$7FF7
DSL: $7FF4-$7FF5
SSI: $7FF2-$7FF3
MFT: $7FF0-$7FF1
FETCH NEXT INSTRUCTION
SWI INSTRUCTION
Y
N
RTI INSTRUCTION
Y
N
RESTORE REGISTERS FROM STACK CCR, A, X, PC
EXECUTE INSTRUCTION

Figure 4-1. Interrupt Flowchart

SECTION 5
RESETS

The MC68HC05CC2 can be reset four ways:

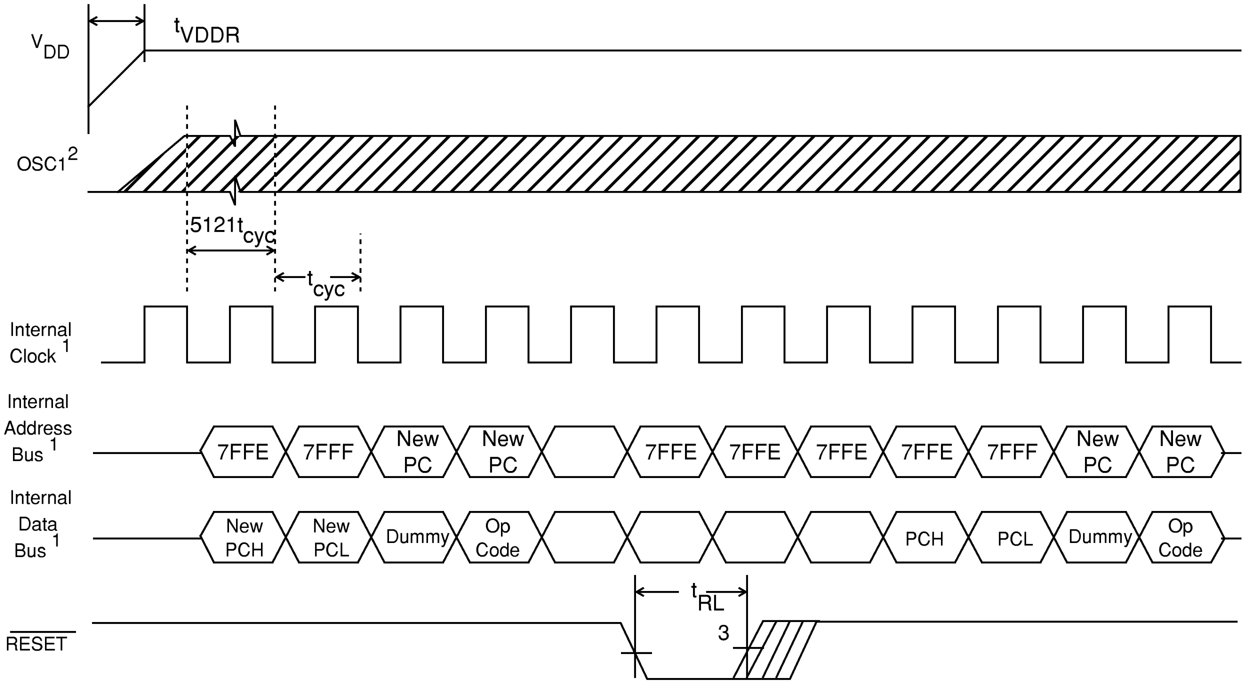
1. Initial Power-on Reset (POR) Function
2. COP Watchdog Timer Reset
3. Illegal Address Reset
4. Active Low Input to the $\overline{\text{RESET}}$ Pin

5.1 Power-On Reset (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 5121 internal processor clock cycle (t_{cyc}) oscillator stabilization delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is held low at the end of this 5121 cycle delay, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ is driven high. See Figure 5-1.

5.2 Computer Operating Properly (COP) Reset

The MCU contains a watchdog timer that when enabled automatically times out if not reset (cleared) within a specific time by a program reset sequence. If the COP watchdog timer is allowed to time-out, a reset is generated to reset the MCU. Because the internal reset signal is used, the MCU comes out of a COP reset in the same operating mode it was in when the COP time-out was generated. See **SECTION 10 MULTIFUNCTION TIMER** for more information on the COP watchdog timer.


VDD
tVDDR
OSC1 2
5121tcyc
tcyc
Internal Clock 1
Internal Address Bus 1
7FFE
7FFF
New PC
New PC
7FFE
7FFE
7FFE
7FFE
7FFF
New PC
New PC
Internal Data Bus 1
New PCH
New PCL
Dummy
Op Code
PCH
PCL
Dummy
Op Code
tRL
3
RESET

NOTES:
1. Internal timing signal and internal bus information not available externally.
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 5-1. Power-On Reset and $\overline{\text{RESET}}$

5.3 Illegal Address (ILADR) Reset

The MCU monitors all opcode fetches. If an illegal address space is accessed during an opcode fetch, an internal reset is generated. Illegal address spaces consist of all external addresses within the memory map and the entire register space. (See **Figure 2-1. Memory Map**.) Because the internal reset signal is used, the MCU comes out of an ILADR reset in the same operating mode it was in when the opcode was fetched.

5.4 $\overline{\text{RESET}}$ Pin

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{cyc}).

SECTION 6
LOW POWER MODES

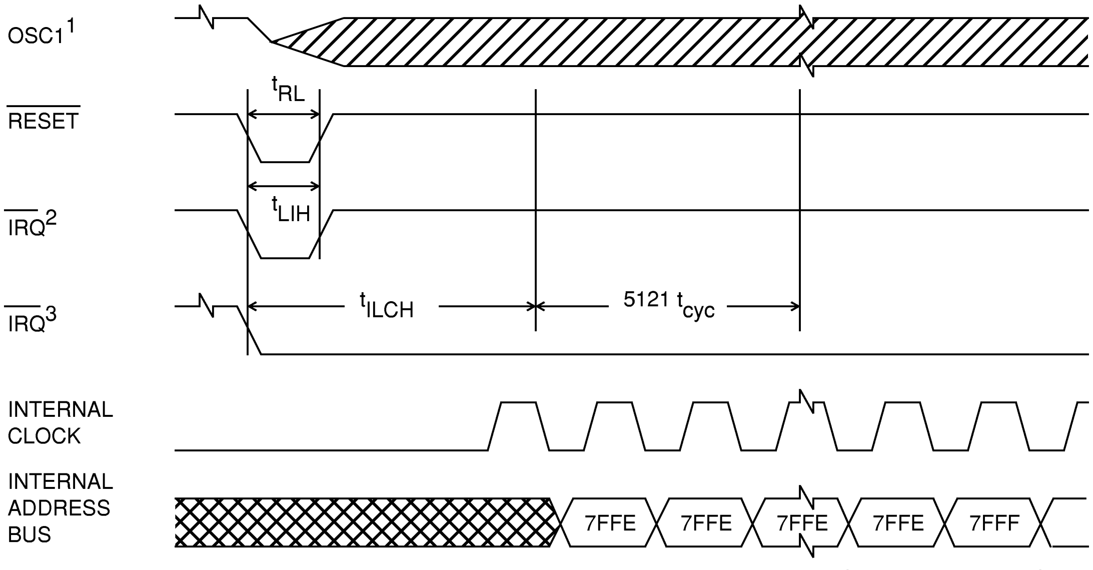
The MC68HC05CC2 has two low power modes:

- STOP mode
- WAIT mode

6.1 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 multifunction timer and COP watchdog timer operation.

The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset. See Figure 6-1.


OSC1¹
RESET
IRQ²
IRQ³
INTERNAL CLOCK
INTERNAL ADDRESS BUS
tRL
tLIH
tILCH
5121 tcyc
7FFE
7FFE
7FFE
7FFE
7FFF
RESET OR INTERRUPT VECTOR FETCH

Notes:
1. Represents the internal gating of the OSC1 pin.
2. $\overline{\text{IRQ}}$ pin edge-sensitive selection.
3. $\overline{\text{IRQ}}$ pin level- and edge-sensitive selection

RESET OR INTERRUPT VECTOR FETCH

Figure 6-1. Stop Recovery Timing Diagram

6.2 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 in WAIT mode. The oscillator, the multifunction timer, the pulse width modulator, the pulse accumulator, the on-screen display (and its PLL), the closed-caption data slicer, and the synchronous serial interface (as a slave) remain active (if enabled). Any interrupt or any form of reset will cause the MCU to exit the WAIT mode. The on-screen display and closed-caption data slicer will not interrupt during WAIT mode, although they remain active (if enabled).

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.

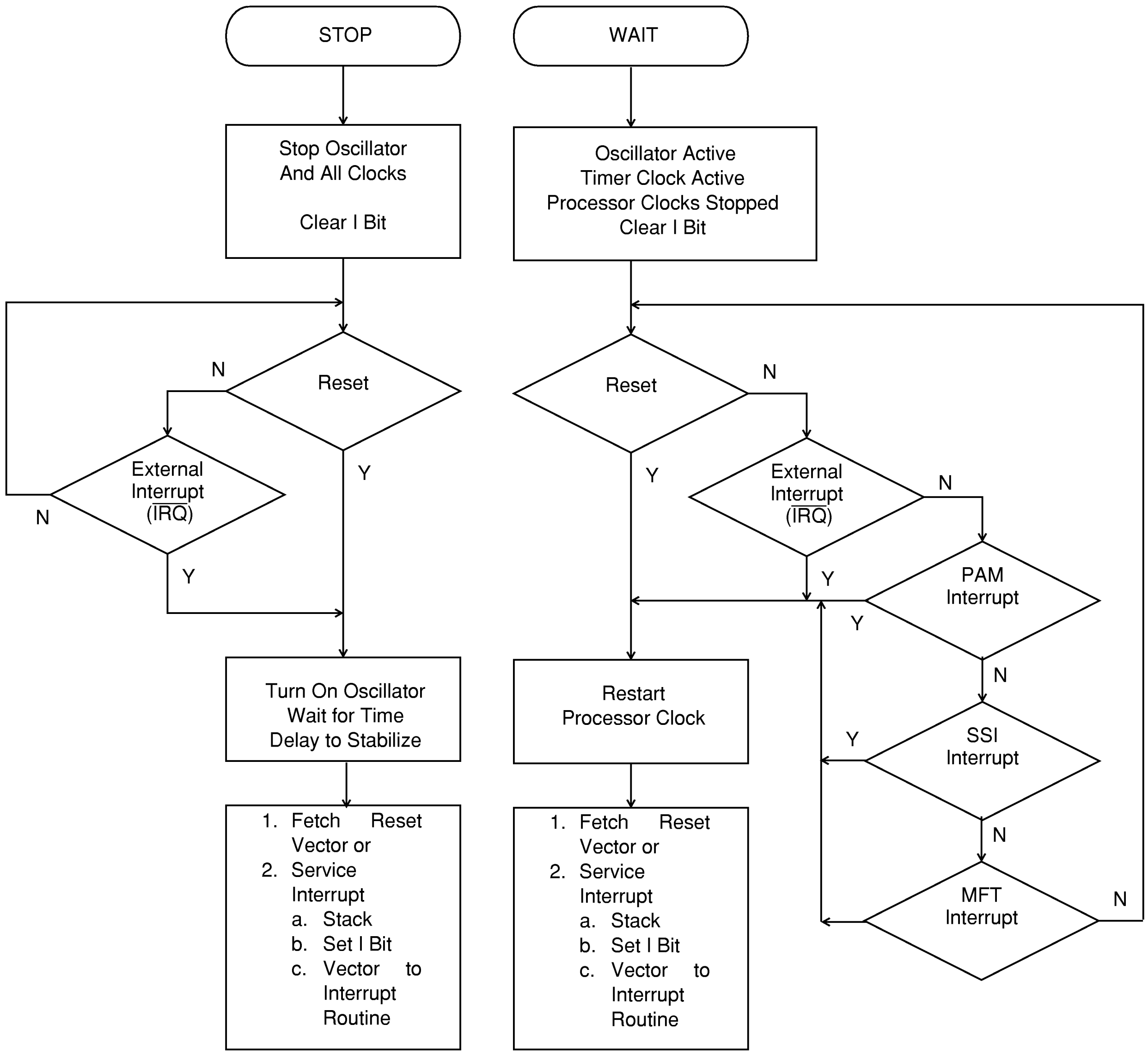

STOP
WAIT
Stop Oscillator And All Clocks
Clear I Bit
Oscillator Active
Timer Clock Active
Processor Clocks Stopped
Clear I Bit
Reset
N
Y
External Interrupt ($\overline{IRQ}$)
N
Y
Reset
N
Y
External Interrupt ($\overline{IRQ}$)
N
Y
PAM Interrupt
Y
N
SSI Interrupt
Y
N
MFT Interrupt
N
Turn On Oscillator Wait for Time Delay to Stabilize
Restart Processor Clock
1. Fetch Reset Vector or
2. Service Interrupt
a. Stack
b. Set I Bit
c. Vector to Interrupt Routine
1. Fetch Reset Vector or
2. Service Interrupt
a. Stack
b. Set I Bit
c. Vector to Interrupt Routine

Figure 6-2. STOP/WAIT Flowchart

SECTION 7
INPUT/OUTPUT PORTS

The 42-pin SDIP version of the MC68HC05CC2 has 31 lines arranged as three 8-bit I/O ports (ports A, B, and D) and one 7-bit I/O port (port C). The 40-pin DIP version of the MC68HC05CC2 has 29 lines arranged as two 8-bit I/O ports (ports A and B), one 7-bit I/O port (port D), and one 6-bit I/O port (port C). The I/O ports are programmable as either inputs or outputs under software control of the data direction registers.

7.1 Port A

Port A is an 8-bit bidirectional port which does not share any of its pins with other subsystems. The address of the port A data register is $00 and the port A data direction register is at address $04.

7.2 Port B

Port B is an 8-bit bidirectional port which shares its pins with the PWM subsystem. See **SECTION 13 PULSE WIDTH MODULATOR** for a detailed description of the PWM. The address of the port B data register is $01 and the port B data direction register is at address $05. When configured as outputs, these pins are open-drain.

7.3 Port C

Port C is a 7-bit bidirectional port which shares its pins with the ADC and PAM subsystems. See **SECTION 14 A/D CONVERTER** for a detailed description of the ADC, and **SECTION 8 PULSE ACCUMULATOR** for a detailed description of the PAM. The address of the port C data register is $02 and the port C data direction register is at address $06. PC6 is not externally available on the 40-pin DIP package.

7.4 Port D

Port D is an 8-bit bidirectional port which shares its pins with the SSI and OSD subsystems. See **SECTION 9 SYNCHRONOUS SERIAL INTERFACE** for a detailed description of the SSI, and **SECTION 11 ON-SCREEN DISPLAY** for a detailed description of the OSD. The address of the port D data register is $03 and the port D data direction register (DDR) is at address $07. PD0 is not externally available on the 40-pin DIP package.

NOTE

Port lines shared with subsystems will have their direction controlled by the subsystem if enabled, instead of the port line data direction register. However, even with the subsystem enabled, during a read of the port line, the data direction register will still control from where data is read, according to Table 7-1 ($R/\overline{W}$ = 1).

7.5 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). Any I/O port pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero.

In the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin. See Table 7-1 and Figure 7-1.

During a reset, all DDRs are cleared, configuring all port pins as inputs. The data registers are not affected by a reset.

Table 7-1. I/O Pin Functions

$R/\overline{W}$	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.

$R/\overline{W}$ is an internal signal.

NOTE

To avoid a glitch on the output pins, write data to the I/O port data register before writing a one to the corresponding data direction register.

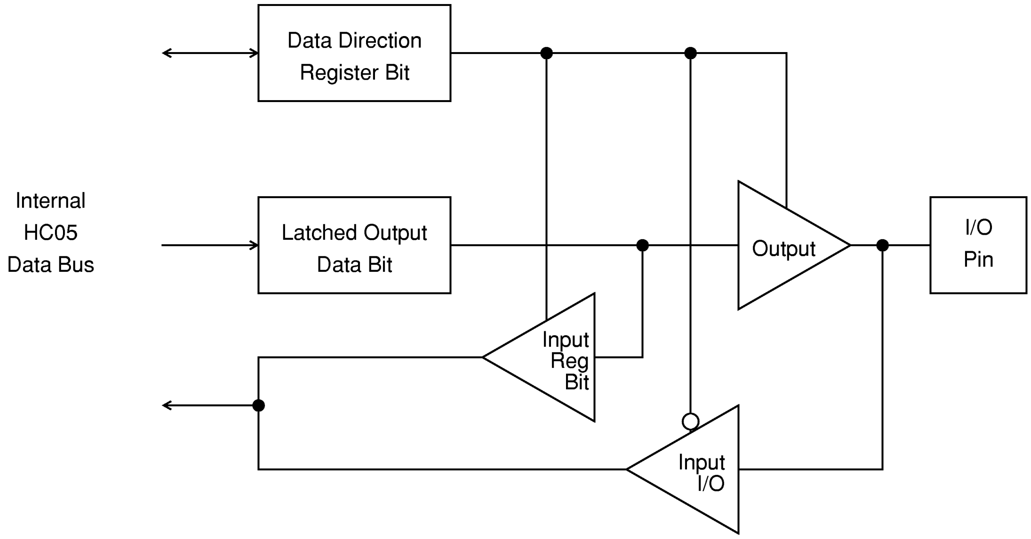

Data Direction
Register Bit
Internal
HC05
Data Bus
Latched Output
Data Bit
Output
I/O
Pin
Input
Reg
Bit
Input
I/O

Figure 7-1. Port I/O Circuitry

SECTION 8
PULSE ACCUMULATOR

The pulse accumulator (PAM) measures or counts pulses of an external input for use in applications such as IR decoding.

8.1 Features

The PAM provides the following features:

- Pulse width or Period Measurement at Any of Four Frequencies
- Pulse Counting at a Maximum Frequency of $f_{op} \div 4$

8.2 Overview

The PAM consists of an 8-bit read-only buffered counter, three registers, and control logic which generates the PAM clock and interrupts. Refer to Figure 8-1. The PAM has two modes of operation: pulse measurement and pulse counting. The PAM mode is determined by the PAMODE bit in the PAMCR.

8.2.1 Pulse Measurement Mode

In this mode, the PAM counts up from $00 at a rate which is a derivative of the internal bus frequency (f_{op}). The PAPOL bit in the PAMCR defines which edge of PAMIN triggers the transfer of the counter contents into the PAMDR and the subsequent clearing of the counter. The counter is also cleared by a reset or by writing to the PARST bit in the PAMCR. The PAM can interrupt upon a transfer to the holding register (PAF in the PAMSR), or an overflow of the counter (PAOVF in the PAMSR). The PAMIN pulse width must be longer than four internal bus clocks.

8.2.2 Pulse Counting Mode

In this mode, the PAM counts up from $00 on the PAPOL-selected edge transitions of the PAMIN input. The counter is cleared by a reset or writing to PARST in the PAMCR. In pulse counting mode, the PAM can interrupt only upon overflow (PAOVF). The PAMIN pulse width must be longer than two internal bus clocks.

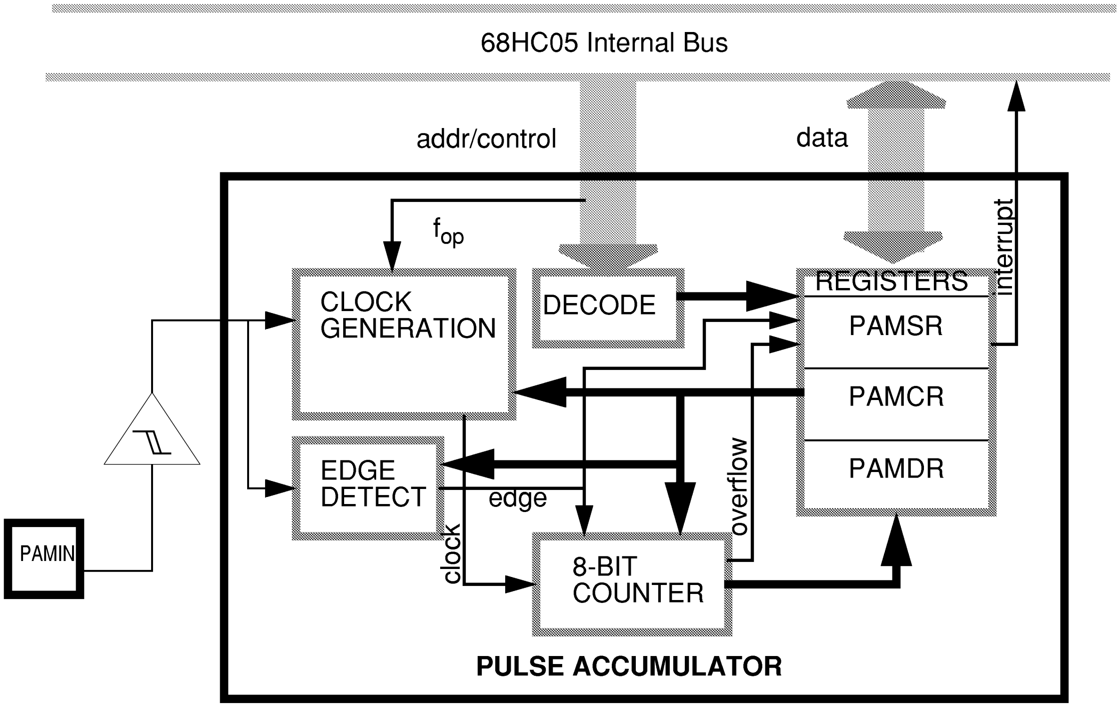

68HC05 Internal Bus
addr/control
data
interrupt
fop
CLOCK GENERATION
DECODE
REGISTERS
PAMSR
PAMCR
PAMDR
EDGE DETECT
edge
overflow
clock
8-BIT COUNTER
PAMIN
PULSE ACCUMULATOR

Figure 8-1. Pulse Accumulator Block Diagram

8.3 Programming Guidelines

Setup and interrupt servicing of the PAM is similar for both modes of operation.

8.3.1 Setup

The PAMCR must be initialized to enable the PAM, establish the mode of operation and frequency of the PAM clock (for pulse measurement mode), select the polarity of interest, and enable the desired interrupts.

8.3.2 Interrupt Servicing

In pulse measurement mode, the PAMSR must be read to determine the source of the interrupt (if both are enabled). If the selected transition has occurred on PAMIN, the PAMDR should be read to clear the interrupt and determine the signal period. If duty cycle measurement is desired instead of period, the PAPOL bit in the PAMCR should be inverted while servicing the PAM. If the reason for the interrupt is overflow, then either a lower frequency can be selected for the PAM internal clock, a RAM location may be used to extend the count, or the external PAMIN signal can be accelerated. The overflow status bit must be cleared before exiting the service routine.

In pulse counting mode, the only possible source of interrupt is overflow. This may be dealt with as described for pulse measurement mode.

If interrupts are disabled, the PAF and PAOVF bits in the PAMSR may be polled to determine when the PAM requires servicing.

8.4 Input/Output

The PAM shares one pin with port C.

8.4.1 PAMIN

The PAMIN input shares PC4 with port C. PAMIN has a separate input path from the pin, which includes an internal Schmitt trigger to improve noise immunity. When the PAEN bit in the PAMCR is a logic zero, this pin is PC4 and follows the port C DDR assignment. When the PAEN bit is a logic one, PC4 becomes the PAMIN input pin. The minimum pulse width of PAMIN varies with the PAM operating mode. Refer to **8.2.1 Pulse Measurement Mode** and **8.2.2 Pulse Counting Mode**.

8.5 Registers

The PAM has three registers.

8.5.1 PAM Control Register (PAMCR)

The PAMCR contains the control and reset bits for the PAM. Figure 8-2 details each bit in the register.

		7	6	5	4	3	2	1	0
PAMCR $18	RD	PAFIE	PAOVIE	PAPOL	PAEN	PAMODE	0	PAPR1	PAPR0
	WR						PARST		
	RST	0	0	0	0	0	0	0	0

Figure 8-2. PAM Control Register (PAMCR)

8.5.1.1 PAFIE - Pulse Accumulator Flag Interrupt Enable

PAFIE determines whether the PAF bit in the PAMSR is enabled to generate interrupt requests to the CPU. A reset clears this bit.

PAFIE = 1: PAF interrupt enabled
PAFIE = 0: PAF interrupt disabled

8.5.1.2 PAOVIE - Pulse Accumulator OVerflow Interrupt Enable

PAOVIE determines whether the PAOVF bit in the PAMSR is enabled to generate interrupt requests to the CPU. A reset clears this bit.

PAOVIE =1: PAOVF interrupt enabled
PAOVIE =0: PAOVF interrupt disabled

8.5.1.3 PAPOL - PAmin POLarity

PAPOL determines whether positive or negative edge transitions will increment the counter in pulse counting mode. PAPOL determines whether positive or negative edge transitions will cause a transfer and reset the counter in pulse measurement mode. A reset clears this bit.

PAPOL = 1: positive edge transitions trigger
PAPOL = 0: negative edge transitions trigger

8.5.1.4 PAEN - Pulse Accumulator ENable

PAEN determines whether the PAM is enabled or disabled. When the PAM is disabled, the PAM clocks are stopped and PAF and PAOVF in the PASR are cleared. A reset clears this bit.

PAEN = 1: PAM enabled
PAEN = 0: PAM disabled

8.5.1.5 PAMODE - Pulse Accumulator MODE

PAMODE determines whether the PAM is operating in pulse counting or pulse measurement mode. A reset clears this bit.

PAMODE = 1: pulse measurement mode
PAMODE = 0: pulse counting mode

8.5.1.6 PARST - Pulse Accumulator ReSeT

PARST is a one-shot bit which, when set, clears the PAM counter. PARST always reads as a logic zero. A reset clears this bit.

8.5.1.7 PAPR1:0 - Pulse Accumulator PRescalar 1:0

PAPR1:0 select the clock prescaler which drives the counter in pulse measurement mode. See Figure 8-1.

Table 8-1. Pulse Accumulator Prescalar Rates

	PRESCALER RATE AT f_{osc} FREQUENCY:		
PAPR1:0	**2.0MHz**	**4.0 MHz**	**8.0 MHz**
00 (E/8)	8 µs	4 µs	2 µs
01 (E/16)	16 µs	8 µs	4 µs
10 (E/80)	80 µs	40 µs	20 µs
11 (E/160)	160 µs	80 µs	40 µs

E = Internal Bus Clock (f_{osc}/2)

8.5.2 PAM Status Register (PAMSR)

The PAMSR contains the PAM interrupt flag bits and provides visibility of the external input. Figure 8-2 details each bit in the register.

PAMSR $19		7	6	5	4	3	2	1	0
	RD	PAF	PAOVF	PAPIN	0	0	0	0	0
	WR								
	RST	0	0	U					

Figure 8-3. PAM Status Register (PAMSR)

8.5.2.1 PAF - PAm interrupt Flag

PAF is a read-only status bit which indicates when the measurement of one phase of PAMIN is complete. PAF is set on the selected edge of PAMIN in the pulse measurement mode; it is inhibited in pulse counting mode. PAF is cleared by reading the PAMSR with PAF set, followed by reading the PAMDR. A reset clears this bit.

8.5.2.2 PAOVF - PAm OVerflow interrupt Flag

PAOVF indicates counter overflow; it is set when the counter rolls over from $FF to $00. PAOVF is cleared by writing a logic zero to it; writing a logic one has no effect. A reset clears this bit.

8.5.2.3 PAPIN - PAmin PIN

PAPIN is a read-only status bit which reflects the state of the PAMIN input after it has gone through the schmitt-triggered buffer. A reset has no effect on this bit.

8.5.3 PAM Data Register (PAMDR)

The PAMDR contains the transferred data bits from the PAM counter. Figure 8-4 details each bit in the register.

PAMDR $1A	RD	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
	WR								
	RST	U	U	U	U	U	U	U	U

Figure 8-4. PAM Data Register (PAMDR)

8.5.3.1 DA7:0 - DAta 7:0

These read-only bits represent the contents of the PAM counter. In pulse counting mode, the holding register is transparent and reflects the contents of the counter. In pulse measurement mode, it is loaded at the selected edge of the PAMIN pin. A reset has no effect on these bits.

8.6 Low Power Modes

The following subsections describe the lower power modes.

8.6.1 Operation During WAIT Mode

The PAM remains active during WAIT mode. If interrupts are enabled, a PAM interrupt can cause the processor to exit the WAIT mode.

8.6.2 Operation During STOP Mode

The PAM halts operation in STOP mode. It is recommended that the PAM be disabled before entering STOP mode.

8.7 Interrupts and Resets

The PAM has two sources of interrupt, the PAF and PAOVF flags in the PAMSR. These interrupts are enabled by the PAFIE and PAOVIE bits, respectively, in the PAMCR. Both interrupts will cause the MCU to vector to the address stored in $7FF8-$7FF9.

SECTION 9
SYNCHRONOUS SERIAL INTERFACE

The synchronous serial interface (SSI) transmits and receives synchronous serial data for communication with other peripherals.

9.1 Features

The SSI provides the following features:

- Master or Slave Operation
- Lease Significant Bit (LSB) or Most Significant Bit (MSB) First Transmission Order
- Limited I^{2}C Master Capability
- Choice of Four Master Mode Transmission Rates
- Programmable Clock Polarity

9.2 Overview

The SSI is a two-wire master/slave system consisting of three registers and control logic which generates the SSI clock and interrupts. Refer to Figure 9-1. The SSI has two modes of operation: master and slave. The SSI mode is determined by the MSTR bit in the SSICR.

9.2.1 Master Mode

In master mode, the user initiates data transfers by writing to the SSIDR and the SSI drives the SCK pin for external slaves. It is up to the user to ensure that only one master exists in the system at any one time.

For an SSI configured as a master, to initiate a transfer, the data register write must occur after the SSI is enabled.

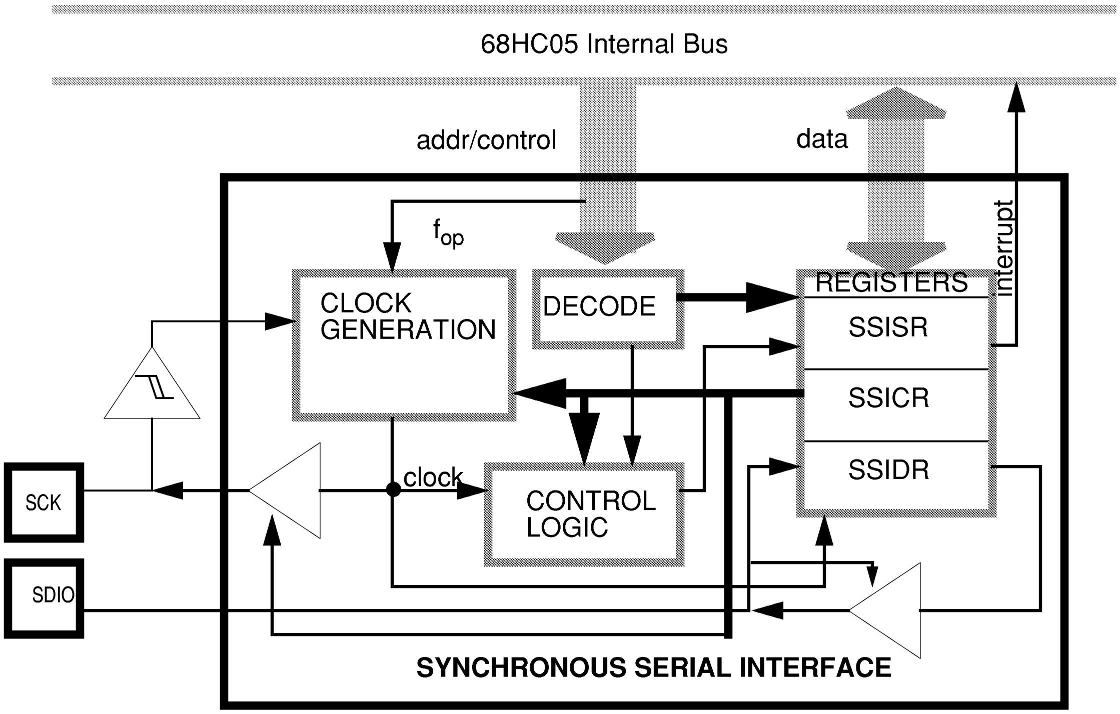

68HC05 Internal Bus
addr/control
data
f_{op}
interrupt
CLOCK GENERATION
DECODE
REGISTERS
SSISR
SSICR
SSIDR
clock
CONTROL LOGIC
SCK
SDIO
SYNCHRONOUS SERIAL INTERFACE

Figure 9-1. SSI Block Diagram

9.2.2 Slave Mode

In slave mode, the SSI responds to the SCK input driven by an external master. Slaves should always be enabled after the master is enabled and disabled before the master is disabled.

9.3 Programming Guidelines

Setup and interrupt servicing of the SSI is somewhat dependent on the mode of operation.

9.3.1 Setup

The SSICR must be initialized to enable the SSI, establish the mode of operation, format data (LSB first or MSB first), and direct data, and enable the interrupt. When in master mode, the clock frequency should also be selected. All devices in the system must operate with the same clock polarity and data rates. When initializing or modifying the other SSICR bits, the SE bit should always be at a logic zero. The slave SSI should always be enabled after the master is enabled to ensure that the slave's SCK input is stable.

9.3.2 Interrupt Servicing

The SF flag in the SSISR generates an interrupt to indicate that eight bits of data have been transferred. Interrupt servicing of the SSI should always clear the SF bit by reading the SSISR followed by reading (or writing) the SSIDR. The master SSI may initiate another transfer by writing to the SSIDR.

If interrupts are not desired, the SF bit in the SSISR may be polled to determine when the SSI requires servicing.

9.4 Input/Output

The SSI shares two pins with port D.

9.4.1 SCK

The SCK (Serial ClocK) input/output shares PD2 with port D. When the SE bit in the SSICR is a logic zero, this pin is PD2 and follows the port D DDR assignment. When the SE bit is a logic one, PD2 becomes the SCK I/O pin. SCK must be stable before the slave SSI is enabled.

In master mode, the SCK pin is an output with four selectable frequencies (SR1:0 bits in SSICR). The SCK pin will idle in the state selected by the CPOL bit in the SSICR.

In slave mode, the SCK pin is an input and the clock must be supplied by an external master with a maximum frequency of $f_{op}/2$. SCK has minimum frequency. The external master should maintain the appropriate SCK idle state between transmissions.

Data is always captured at the SDIO pin on the **rising** edge of SCK. Data is always shifted out and presented at the SDIO pin on the **falling** edge of SCK.

9.4.2 SDIO

The SDIO (Serial Data In/Out) input/output shares PD1 with port D. When the SE bit in the SSICR is a logic zero, this pin is PD1 and follows the port D DDR assignment. When the SE bit is a logic one, PD2 becomes the SDIO I/O pin.

The SDIO pin will either be a high impedance input pin or will idle high, depending on the state of the $T/\overline{R}$ bit in the SSICR. When receiving data in master mode, the $T/\overline{R}$ bit must be low and data must be written to the data register to initiate clock generation. When transmitting data in master mode, the $T/\overline{R}$ bit must be high.

The data will be sent or received in either MSB first or LSB first format, depending on the state of the LSBF bit in the SSICR.

The CPOL bit in the SSICR determines when the first data bit is shifted out to SDIO. Refer to Figure 9-2 and Figure 9-3. If CPOL = 1, the first falling edge of SCK will shift the first data bit out to SDIO. If CPOL = 0, the first data bit will be driven out to SDIO before the first rising edge of SCK. Subsequent falling edges of SCK will shift the remaining data bits out.

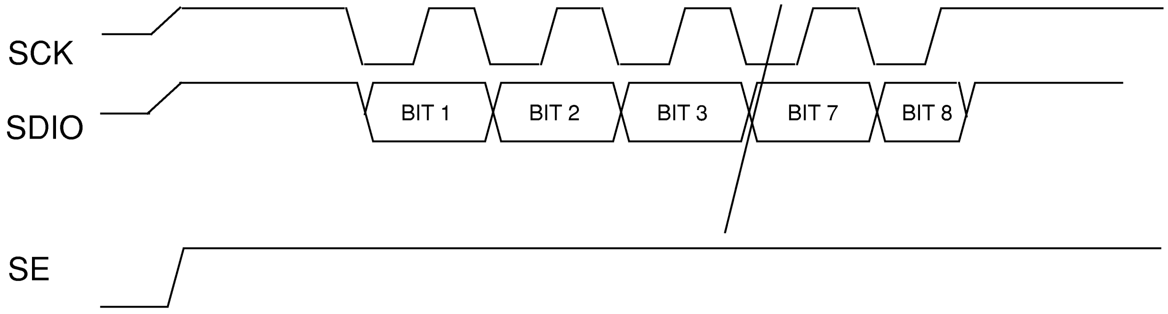


Figure 9-2. Serial I/O Port Timing (CPOL=1)

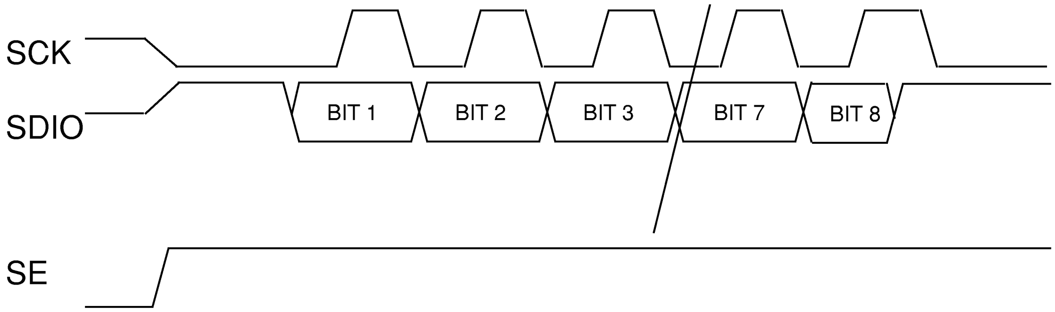


Figure 9-3. Serial I/O Port Timing (CPOL=0)

9.5 Registers

The SSI has three registers.

9.5.1 SSI Control Register (SSICR)

The SSICR contains the control bits for the SSI. Figure 9-4 details each bit in the register.

NOTE

Writes to this register during a transfer should be avoided, with the exception of clearing the SE bit to disable the SSI. In addition, LSBF, CPOL, MSTR and SR1:0 should not be changed while the SSI is enabled or being enabled. Always disable the SSI first before altering these control bits within the SSI control register.

		7	6	5	4	3	2	1	0
SSICR $1B	RD WR	SIE	SE	LSBF	MSTR	CPOL	$T/\overline{R}$	SR1	SR0
	RST	0	0	0	0	0	0	0	0

Figure 9-4. SSI Control Register (SSICR)

9.5.1.1 SIE - Ssi Interrupt Enable

SIE determines whether the SF bit in the SSISR is enabled to generate interrupt requests to the CPU. A reset clears this bit.

SIE = 1: SF interrupt enabled
SIE = 0: SF interrupt disabled

9.5.1.2 SE - Ssi Enable

SE determines whether the SSI is enabled or disabled. When the SSI is enabled, PD2 and PD1 become the SCK and SDIO pins, respectively. When the SSI is disabled, the bit counter is reset, the SSI clocks stop, and pins SCK and SDIO revert to PD2 and PD1. Any transmission in progress will be aborted when the SSI is disabled. A reset clears this bit.

SE = 1: SSI enabled
SE = 0: SSI disabled

9.5.1.3 LSBF - Least Significant Bit First

LSBF determines whether the data is sent and received LSB or MSB first. A reset clears this bit.

LSBF = 1: Least significant bit first format
LSBF = 0: Most significant bit first format

9.5.1.4 MSTR - MaSTeR mode

MSTR determines whether the SSI is operating in master or slave mode. Any transmission in progress will be aborted when the MSTR bit is cleared. A reset clears this bit.
MSTR = 1: Master mode
MSTR = 0: Slave mode

9.5.1.5 CPOL - Clock POLarity

CPOL determines the state of the SCK pin between transmissions. Regardless of the state of CPOL, data will be latched on the rising edge of SCK for serial input, and data will be valid on the rising edge of SCK for serial output. A reset clears this bit.
CPOL = 1: SCK idle state is high
CPOL = 0: SCK idle state is low

NOTE

If the SSI is used as a slave, the SCK input pin must be at the appropriate idle level before enabling the SSI.

9.5.1.6 T/$\overline{\text{R}}$ - Transmit/Receive

T/$\overline{\text{R}}$ determines whether the SSI is transmitting or receiving. When the SSI is transmitting, data will be driven onto the SDIO pin. When the SSI is receiving, the SDIO drivers are disabled. A reset clears this bit.
T/$\overline{\text{R}}$ = 1: SSI transmitting
T/$\overline{\text{R}}$ = 0: SSI receiving

9.5.1.7 SR1:0 - Ssi Rate 1:0

These bits determine the frequency of SCK when in master mode. They have no effect in slave mode. A reset clears these bits. See Table 9-1 for the SCK rate selections.

Table 9-1. SSI SCK Rates

	SCK RATES (Hz) AT f_{osc} FREQUENCY:		
SR1:0	**2.0MHz**	**4.0 MHz**	**8.0 MHz**
00	15.625k	31.25k	62.5k
01	31.25k	62.5k	125k
10	62.5k	125k	250k
11	125k	250k	500k

9.5.2 SSI Status Register (SSISR)

The SSISR contains the SSI interrupt flag and the data collision flag. Figure 9-5 details each bit in the register.

SSISR $1C	RD	SF	DCOL	0	0	0	0	0	0
	WR								
	RST	0	0						

Figure 9-5. SSI Status Register (SSISR)

9.5.2.1 SF - Ssi Flag

This read-only bit is set upon occurrence of the last rising clock edge, and indicates that a data transfer has taken place. If the SSI is in slave mode and its interrupt is disabled, SF has no effect on any further transmissions, although SF must be cleared in order to write the SSIDR. If the SSI interrupt is enabled, SF must be cleared in order to clear the interrupt. In master mode the SF flag must always be cleared between transfers. The SF flag can be cleared three ways: (1) by reading the SSR with SF set, followed by a read or write of the SSIDR, (2) by a reset, or (3) by disabling the SSI. If SF is cleared before the last edge of the next byte, it will be set again.

9.5.2.2 DCOL - Data COLlision

This is a read-only status bit which indicates that an invalid access to the data register has been made. An invalid access is:

- An access of the SSIDR in the middle of a transfer (after the first falling edge of SCK and before SF is set), or
- An access of the SSIDR made before an access of the SSISR (after SF is set).

DCOL is cleared by reading the status register with SF set, followed by a read or write of the SSIDR. A reset clears this bit.

9.5.3 SSI Data Register (SSIDR)

The SSIDR contains the eight transmit/receive data bits. Figure 9-6 details each bit in the register.

		7	6	5	4	3	2	1	0
SSIDR $1D	RD WR	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
	RST	0	0	0	0	0	0	0	0

Figure 9-6. SSI Data Register (SSIDR)

9.5.3.1 DA7:0 - DAta 7:0

These bits are the eight data bits which have been received or are to be transferred by the SSI. These bits are not double-buffered, but writes to this register are masked during transfers and will not destroy the previous contents. These bits should be written to only upon completion of a transfer after the SF flag has been cleared. This register may be read at any time, but the results may be ambiguous if a transfer is in progress. These bits could be altered whenever the CPOL bit in the SSICR is altered. A reset clears these bits.

9.6 Low Power Modes

The following subsections describe low power modes.

9.6.1 Operating During WAIT Mode

The SSI remains active in WAIT mode. If interrupts are enabled, an SSI interrupt will cause the processor to exit the WAIT mode.

9.6.2 Operation During STOP Mode

In STOP mode, the SSI halts operation. The SDIO and SCK pins will maintain their states.

If the SSI is nearing completion of a transfer when STOP mode is entered, it is possible for the SSI to generate an interrupt request and thus cause the processor to exit STOP mode immediately. To prevent this occurrence, the programmer should ensure that all transfers are complete before entering STOP mode.

If the SSI is configured as a slave, further care should be taken in entering STOP mode. The SCK pin will still accept a clock from an external master, allowing potentially unwanted transfers to take place and power consumption to be increased. The SSI will not generate interrupt requests in this situation but, on exiting STOP mode through some other means, the SF flag may be found to be set and an interrupt request will be generated if enabled.

To avoid these potential problems, it is safer to disable the SSI completely before entering STOP mode.

9.7 Interrupts and Resets

The SSI has one source of interrupt, the SF flag in the SSISR. This interrupt is enabled by the SIE bit in the SSICR. This interrupt will cause the MCU to vector to the address stored in $7FF2-$7FF3.

SECTION 10
MULTIFUNCTION TIMER

The multifunction timer (MFT) is a 15-stage multifunction ripple counter.

10.1 Features

The MFT provides the following features:

- Real Time Interrupt (RTI) and Timer Overflow (TOF)
- Power-On Reset (POR)
- Computer Operating Properly (COP) Watchdog

10.2 Overview

As seen in Figure 10-1, the MFT begins with a fixed divide-by-five prescaler which drives a 7-bit ripple counter. A timer overflow occurs on the last stage of this counter, providing a periodic interrupt at the rate of $f_{op}/640$. This circuit is followed by a divide-by-125 stage, with the resulting clock ($f_{op}/80000$) driving the real time interrupt circuit. This results in a maximum interrupt period of 20 ms at a bus rate of 4 MHz. The output of the RTI circuit is further divided by 15 to drive the mask optional COP watchdog timer circuit. The MFT also supplies POR and STOP recovery timing with the $f_{op}/5121$ signal and shares outputs of the prescaler and divide-by-128 with the PWM.

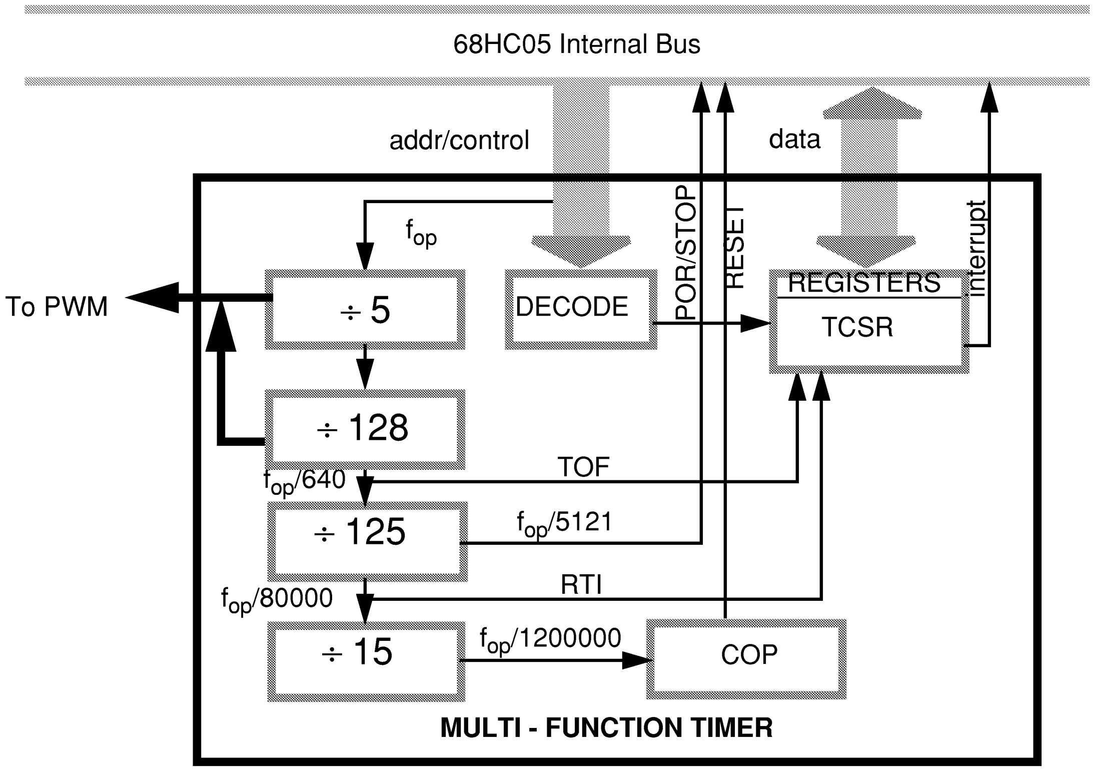

68HC05 Internal Bus
addr/control
data
To PWM
f_{op}
÷ 5
DECODE
POR/STOP
RESET
REGISTERS
TCSR
interrupt
÷ 128
$f_{op}/640$
TOF
÷ 125
$f_{op}/5121$
$f_{op}/80000$
RTI
÷ 15
$f_{op}/1200000$
COP
MULTI - FUNCTION TIMER

Figure 10-1. Multifunction Timer Block Diagram

10.3 Programming Guidelines

Setup and interrupt servicing of the MFT is very simple.

10.3.1 Setup

The MFTCSR must be initialized to enable the TOF and RTI interrupts.

10.3.2 Interrupt Servicing

If enabled, the TOF flag in the MFTCSR generates an interrupt when the 7-bit ripple counter rolls over from $FF to $00. For a 4 MHz internal bus, TOF would interrupt every 40 µs. TOF is cleared by writing a logic one to TOFC in the MFTCSR.

If enabled, the RTIF flag in the MFTCSR generates an interrupt when the output of the divide-by-125 goes active. For a 4 MHz internal bus, RTIF would interrupt every 20 ms. RTIF is cleared by writing a logic one to RTIFC in the MFTCSR.

If the COP is enabled, it must be serviced regularly also. Refer to **10.7.1 Computer Operating Properly (COP) Watchdog Reset**.

10.4 Input/Output

The MFT does not require any input or output.

10.5 Registers

The MFT has one register.

10.5.1 MFT Control and Status Register (MFTCSR)

The MFTCSR contains all control and status bits for the MFT. Figure 10-2 details each bit in the register.

		7	6	5	4	3	2	1	0
MFTCSR $2D	RD	TOF	RTIF	TOFE	RTIE	0	0	0	0
	WR					TOFC	RTIFC		
	RST	0	0	0	0				

Figure 10-2. MFT Control and Status Register (MFTCSR)

10.5.1.1 TOF - Timer Overflow Flag

TOF is a read-only status bit that is set when the 7-bit ripple counter rolls over from $FF to $00. A CPU interrupt request will be generated if TOFE is set. Clearing the TOF is done by writing a logic one to TOFC. Writing to TOF has no effect. A reset also clears this bit.

10.5.1.2 RTIF - Real Time Interrupt Flag

RTIF is a read-only status bit that is set when the output of the RTI circuit goes active. A CPU interrupt request will be generated if RTIE is set. Clearing the RTIF is done by writing a logic one to RTIFC. Writing to RTIF has no effect. A reset also clears this bit.

10.5.1.3 TOFE - Timer Overflow Flag Enable

TOFE determines whether the TOF bit is enabled to generate interrupt requests to the CPU. A reset clears this bit.
TOFE = 1: TOF interrupt enabled
TOFE = 0: TOF interrupt disabled

10.5.1.4 RTIE - Real Time Interrupt Enable

RTIE determines whether the RTIF bit is enabled to generate interrupt requests to the CPU. A reset clears this bit.
RTIE = 1: RTIF interrupt enabled
RTIE = 0: RTIF interrupt disabled

10.5.1.5 TOFC - Timer Overflow Flag Clear

TOF is cleared by writing a logic one to TOFC. Writing a logic zero to TOFC has no effect on TOF. This bit always reads as logic zero. A reset has no effect on this bit.

10.5.1.6 RTIFC - Real Time Interrupt Flag Clear

RTIF is cleared by writing a logic one to RTIFC. Writing a logic zero to RTIFC has no effect on RTIF. This bit always reads as logic zero. A reset has no effect on this bit.

10.6 Low Power Modes

The following subsections describe low power modes.

10.6.1 Operation During WAIT Mode

The MFT remains active in WAIT mode. If interrupts are enabled, an MFT interrupt will cause the processor to exit the WAIT mode.

10.6.2 Operation During STOP Mode

The MFT is cleared when going into STOP mode. In addition, the interrupt flags (TOF and RTIF) and the interrupt enable bits (TOFE and RTIE) in the TCSR are cleared by internal hardware to remove any pending interrupt requests and to disable any further interrupts.When STOP is exited by an external interrupt or an external $\overline{\text{RESET}}$, the internal oscillator will resume, followed by an internal processor oscillator stabilization delay (stop recovery time). The MFT is then cleared and operation resumes.

10.7 Interrupts and Resets

The following subsections describe interrupts and resets.

10.7.1 Computer Operating Properly (COP) Watchdog Reset

The COP watchdog timer function is implemented on this device by using the output of the RTI circuit and further dividing it by 15.

If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched. Preventing a COP time-out is done by writing a logic zero to bit 0 of address $7FF0 at the minimum reset rate. When the COP is cleared, only the final divide-by-15 stage (output of the RTI) is cleared.

The minimum COP reset rates are listed in Table 10-1. Because it is not readily possible to determine the state of the divider chain ahead of the COP circuit, the COP should be reset within a period equivalent to **fourteen** real-time interrupts, rather than the 15 that might be expected.

This function is enabled/disabled with a mask option.

Table 10-1. RTI and minimum COP Reset Times

	RTI AND MINIMUM COP RATES AT f_{osc} FREQUENCY SPECIFIED:		
	2.0 MHz	**4.0 MHz**	**8.0 MHz**
RTI	80 ms	40 ms	20 ms
COP	1.12 s	560 ms	280 ms

10.7.2 MFT Interrupts

The MFT has two sources of interrupt, the TOF and RTIF flags in the MFTCSR. These interrupts are enabled by the TOFE and RTIE bits, respectively, in the MFTCSR. Both interrupts will cause the MCU to vector to the address stored in $7FF0-$7FF1.

SECTION 11
ON-SCREEN DISPLAY

The on-screen display (OSD) module converts programmed character addresses and control information into digital color and blanking outputs to display user-defined characters on a television screen for on-screen programming and closed-captioning applications.

11.1 Features

The OSD provides the following features:

- NTSC Timing (262.5 Line/60 Hz Interlaced, 525 Line/60 Hz Non-interlaced)
- FCC Closed-Caption Display Support
- 127 User-Defined 9 X 13 Characters and One Fixed Border Space Character
- Display: Programmable Character Display in Three Sizes:
  - Standard, 34 Columns by 15 Rows
  - One-and-a-Half Width/Double Height, 24 Columns by Seven Rows
  - Double Width/Double Height, 13 Columns by Seven Rows
- Software Selectable Character Attributes (Can Be Changed Mid-Row):
  - Rounding
  - Black Outline
  - Underline
  - Italics (By Slanting)
  - Foreground Color (One of Eight, Including Black and White)
  - Background Color (One of Nine, Including Black, White and Transparent)

- Software Selectable Border (Area Outside Characters and Background):
    - Nine Border Colors (Including Black, White and Transparent)
- Software Controlled Features:
    - Soft Scrolling
    - Blinking
- Programmable Horizontal Positioning
- Outputs
    - RGB and Fast Luminance Blanking (FBKG) Outputs
    - All Output Polarities are Software Programmable
    - Outputs Can be Put in High Impedance State or Used as I/O Port Pins
- Intercolumn Spacing = 0 dots; Interrow Spacing = 0 lines, so Continuous Horizontal and Vertical Lines may be Formed for Linear Scales

11.2 Overview

The OSD utilizes a single character row architecture instead of a full screen display RAM to minimize die area and address space requirements. Refer to Figure 11-1.

11.2.1 Synchronization

The OSD horizontal timing is based on an independent on-chip oscillator which is phase locked to the rising edge of the internal horizontal sync pulse. The OSD vertical timing is synchronized with the rising edge of the internal vertical sync pulse.

The time base generation logic feeds a line-rate signal into a scan line counter (OSDEVR). This line counter is continuously compared to the target scan line (OSDELR), except during writes to the OSDELR. When a match occurs, all the external rank is (optionally) loaded into the internal rank and an interrupt request is (optionally) generated to notify the CPU of the availability of the external rank for the next row of characters to be displayed.

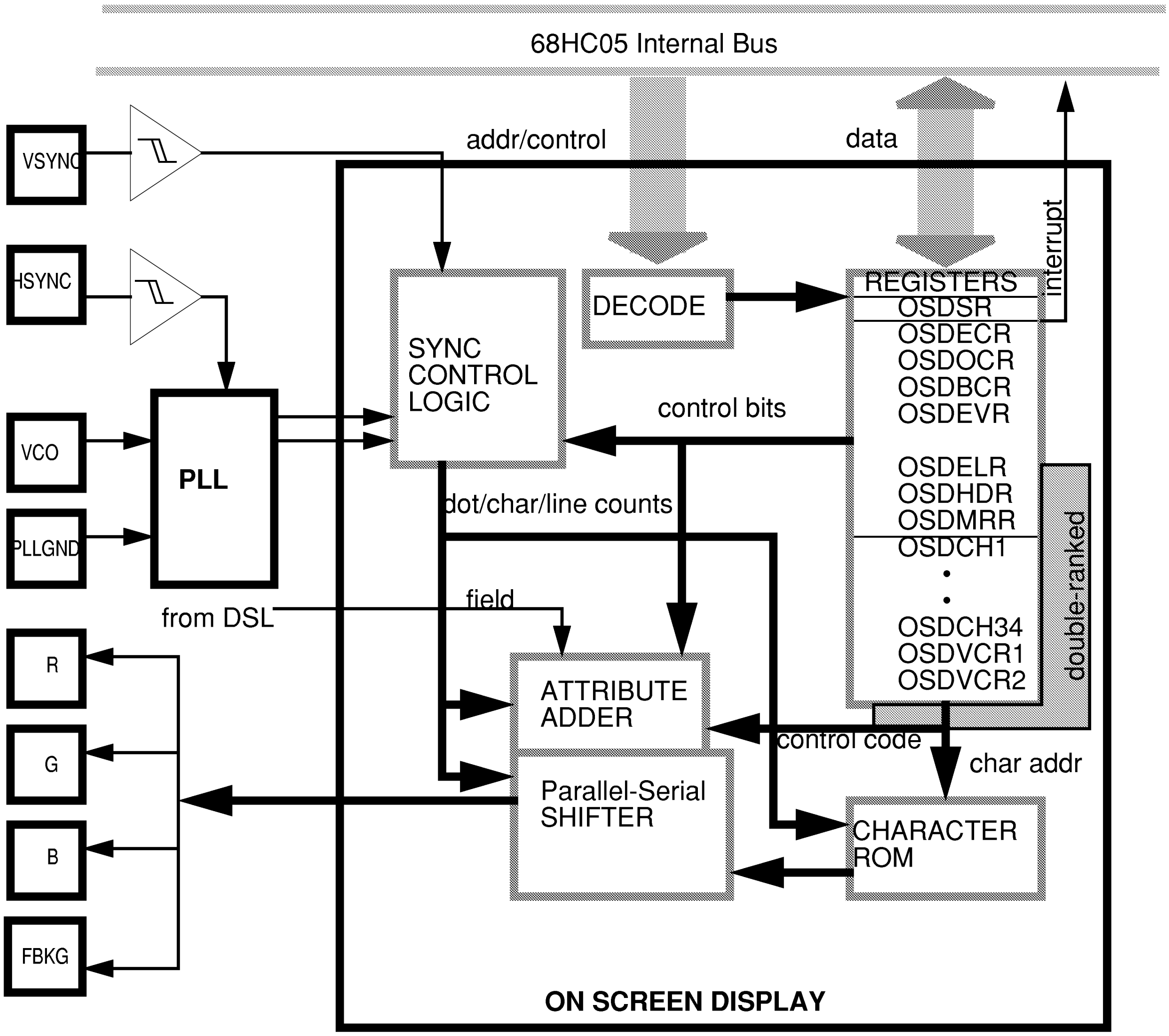

68HC05 Internal Bus
addr/control
data
interrupt
VSYNC
HSYNC
VCO
PLLGND
PLL
SYNC CONTROL LOGIC
DECODE
REGISTERS
OSDSR
OSDECR
OSDOCR
OSDBCR
OSDEVR
OSDELR
OSDHDR
OSDMRR
OSDCH1
OSDCH34
OSDVCR1
OSDVCR2
double-ranked
control bits
dot/char/line counts
field
from DSL
R
G
B
FBKG
ATTRIBUTE ADDER
Parallel-Serial SHIFTER
control code
char addr
CHARACTER ROM
ON SCREEN DISPLAY

Figure 11-1. OSD Block Diagram

11.2.2 Registers and ROM

A majority of the OSD registers are dual-ranked because the OSD must read the values for the current row while the CPU writes the values for the next row. The external rank consists of a set of registers visible to the CPU containing the character codes and control information relevant to a single row on the display screen. It appears in the register space and its bits are readable as well as writable.The internal rank is parallel shifted from the external rank whenever the OSD is ready to display the next row. The internal rank feeds the character generation logic of the OSD.

Character codes are read sequentially from the internal rank of the character registers during each scan line. Each code, along with the scan line number within the character row, is used as an address to fetch the appropriate 9-bit-wide row pattern from the character ROM. The 14,976-bit ROM contains 127 user-defined 9 x 13 dot matrix characters, and one hardwired border space character. Character codes may alternately be interpreted as control codes which alter display attributes mid-row.

11.2.3 OSD Output Logic

Programmable character attributes include color (foreground and background), rounding, black edging, underlining, italics and blinking. To perform character rounding and black edging for the standard-size characters in 1H mode, the display utilizes interlaced video to take advantage of the odd and even fields. The odd/even field indicator is extracted by the closed-caption data slicer, which must be enabled to provide field information to the OSD.

Two double-ranked non-visible registers contain the defaults for these attributes for a character row. Control codes, which appear as a background color space, inserted mid-row may change these attributes. The bit pattern data from the character ROM passes through circuitry which adds the selected attributes of rounding, italics, underlining and black outline. The luminance signal passes through a color encoder and is converted into R, G and B signals.

Outputs are available for R, G, and B signals, and for blanking and windowing external video. The outputs are all standard HCMOS drivers carrying video-frequency data at logic levels.

11.2.4 Display Characteristics

For this section, refer to Figure 11-2 and Figure 11-3.

11.2.4.1 Character Foreground

The location of foreground color within a character is defined by the character ROM, but may be modified by the rounding, black outline, underline and italics attributes. Rounding pixels, which are generated to smooth diagonals, are 1/4 the size of a normal ROM pixel and are displayed in the foreground color. Black outline pixels, also 1/4 the size of a normal ROM pixel, are displayed in black to outline a character. Black outline will not work properly unless rounding is enabled. The underline attribute will generate a line of pixels which are the same color and size as ROM pixels. Underline will affect all characters with the exception of the border space ($00) and control codes; the underline itself will not be black outlined.The Italics attribute does not generate any additional pixels, but will slant the display of the character.

11.2.4.2 Character Background

Character background is the region of a character where there is no ROM, rounding, black outline or underline pixels. Background may be a solid color or transparent. If background is transparent, the video will show through (regardless of border color selection).

11.2.4.3 Border

Border is displayed wherever there is no foreground or background. Border may be a solid color or transparent. If the border is transparent, the video will show through. Border video can be specified for a character within a displayed row by using the border space character.

11.2.4.4 Character Size

Character size is determined by the horizontal position delay and character size register (OSDHDR). There are three possible sizes, 1W, 1.5W and 2W, where W refers to the width; the heights of 1.5W and 2W are twice that of the 1W size. Size is fixed for an entire row. The 1W size character is 18 dots in width and 26 scan lines in height. A maximum of 34 1W characters may be displayed per row. The 1.5W size character is 18 dots (at 2/3 the 1W dot clock frequency) and 52 scan lines in height and has a maximum of 24 characters per row. The 2W size character is 18 dots (at 1/2 the 1W dot clock frequency) and 52 scan lines in height, and has a maximum of 18 characters per row.

11.2.4.5 Character Position

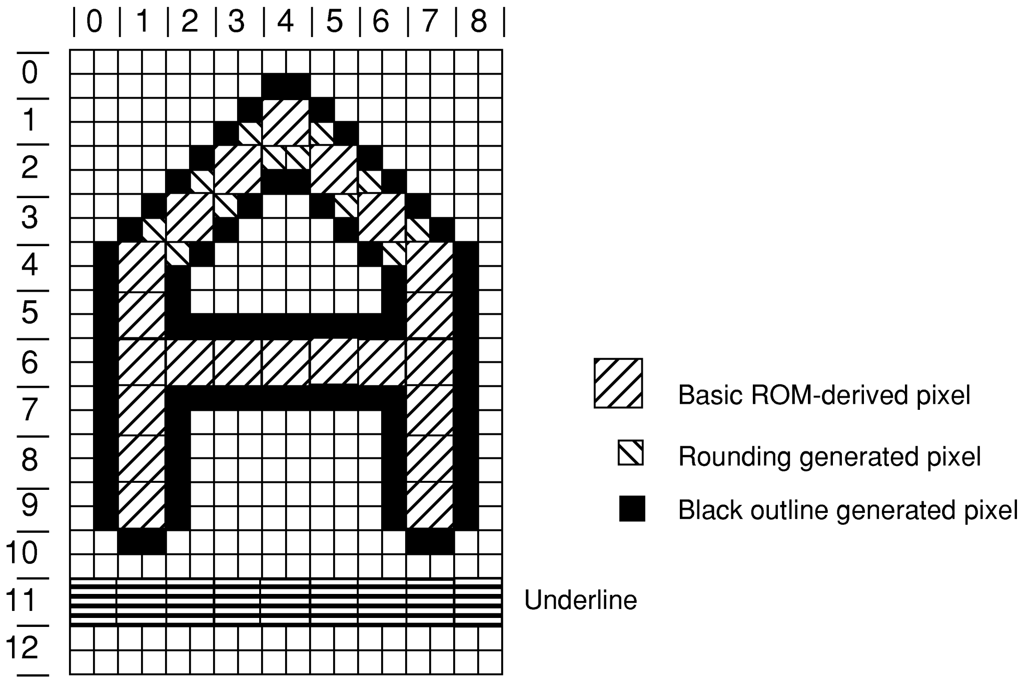
Horizontal positioning of a row of characters is determined by the horizontal delay value specified in the OSDHDR. The border space character will provide further indentation at a character width resolution.

Vertical positioning is indicated in the OSDELR. It is compared with the counter which maintains the current scan line number, which is incremented every scan line in 1H mode and every other scan line in 2H mode.

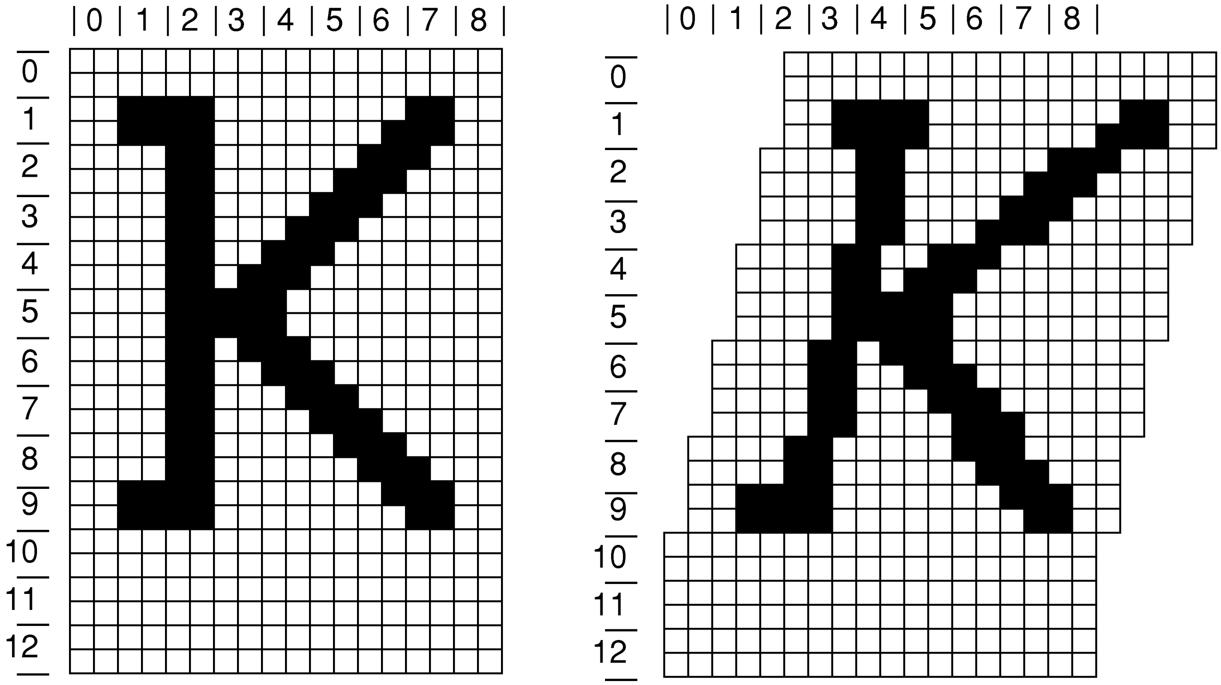
Scrolling can be accomplished in software by modifying the vertical positioning of a row, in conjunction with specifying the subset of horizontal lines of the row to display. The matrix range register (OSDMRR) contents indicate the first and last lines of a row to be displayed. All bit lines before and after the lines displayed are filled in with border.The matrix counter increments according to the character size (1W or 1.5W/2W) and mode (1H or 2H), so that use of the OSDMRR is independent of both.

11.2.4.6 Boundary Conditions

Boundaries between border and character display are always vertical. If italics are not enabled, the width of a border space character is the same as the width of a ROM character. If italics are enabled, the boundaries of the border space character are not slanted, but the width occupied by this space is smaller than the width of a ROM character by five pixels. However, if the preceding character is a control code or another border space, the border space character size is not reduced.


| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
0
1
2
3
4
5
6
7
8
9
10
11
12
Underline
Basic ROM-derived pixel
Rounding generated pixel
Black outline generated pixel

Character with Rounding, Underline, and Black Edge


| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
0
1
2
3
4
5
6
7
8
9
10
11
12
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
0
1
2
3
4
5
6
7
8
9
10
11
12

Normal with Rounding

Italics with Rounding

Figure 11-2. Character Display Styles

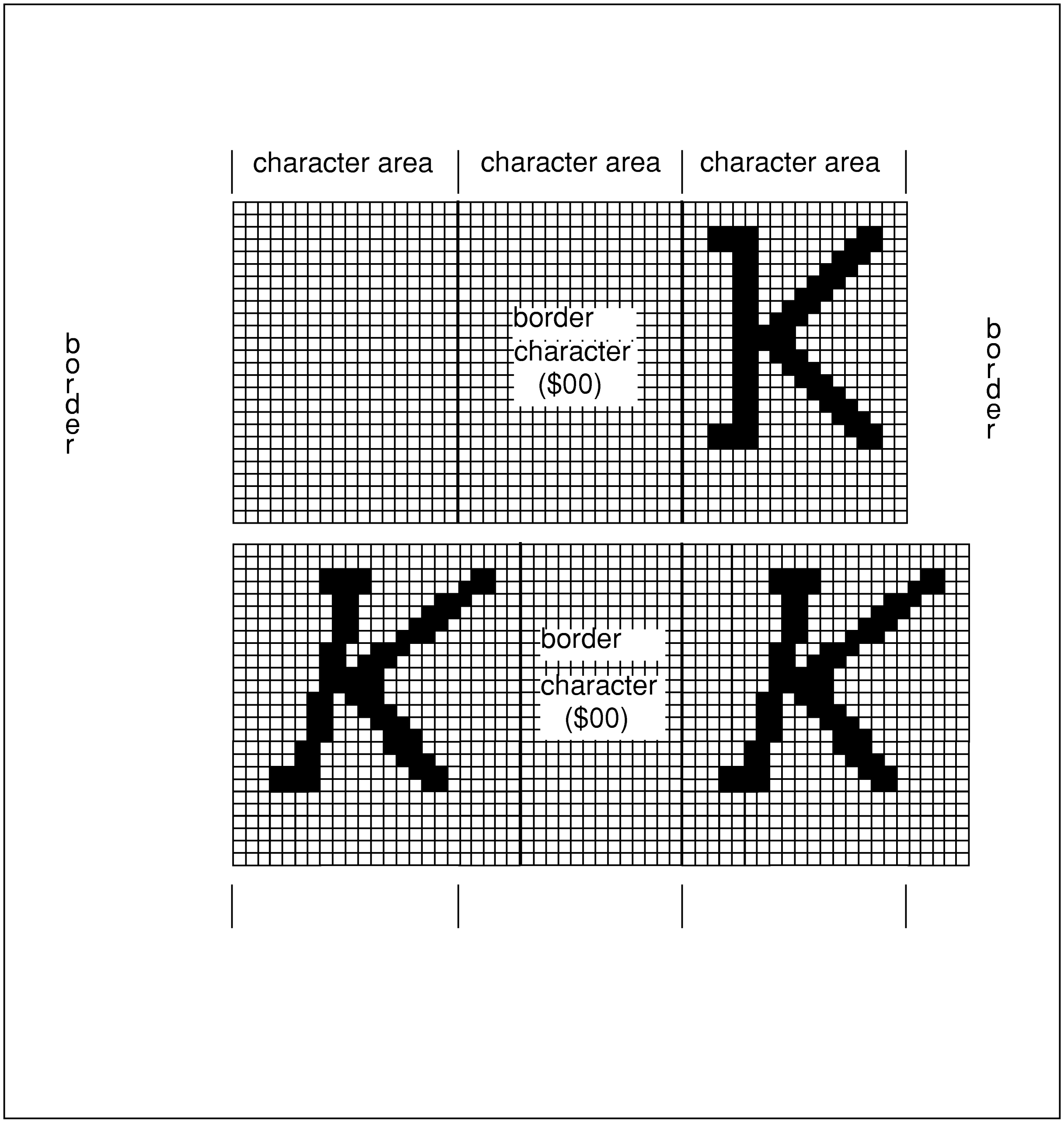

character area
character area
character area
border
border
border
character
($00)
border
character
($00)

Figure 11-3. Display Boundary Conditions

11.3 Programming Guidelines

The following information is available to the programmer for characterization of the displayed row:

- A String of up to 34 Characters (Each One Selectable from 127 Symbols, a Border Space, or 101 Control Codes)
- The Scan Line Number on which the Character Row is to Start
- The Horizontal Placement of the Character Row
- The Character Size from Three Choices
- The Attributes (Black Outline, Rounding, Underline, Italics and Blinking)
- The Region of Scan Lines of a Character Row to Display
- The Foreground Color from Eight Choices
- The Background Color from Nine Choices Including Transparent

In addition, several control registers govern when and how the OSD interacts with the rest of the system.

11.3.1 Setup

Initializing the OSD involves enabling the appropriate modules, defining how interrupts are to be used, enabling outputs and defining the active levels of both inputs and outputs, and setting up the default display characteristics.

For the OSD to operate, the OSD, PLL and DSL must be enabled. The PLL provides clock signals synchronized to the television chassis horizontal, and the DSL provides the field information for interlacing. The PLL has a start-up time which must elapse before the OSD should be used for display. The PLL36 and HMODE bits, which will determine the PLL frequency, in the OSDECR should be defined when the PLLEN bit is set.

The OSD will generate an interrupt, if the OIEN bit in the OSDECR is set, whenever the target scan line defined by the user in the OSDELR matches the current scan line count in the OSDEVR. The user can specify, with the XFER bit in the OSDECR, whether the OSD transfers the external rank of registers to the internal rank when the interrupt occurs. If transferring is not enabled, then the OSD is interrupting only to indicate a scan line match. Transferring must be enabled to display more than one row of characters.

The active levels of the HSYNC and VSYNC inputs, and the FBKG, R, G and B outputs are all programmable. The outputs may be enabled independently of one another.

The character registers and display attributes must be initialized, since most of these are undefined upon a reset. When the OSD is disabled (OSDEN=0), writing to the external rank of registers also writes to the internal rank. The character registers should be set to $00 to indicate border space characters if no display is required upon start-up. The border color must be defined in the OSDBCR.

11.3.2 Interrupt Servicing

When the target scan line (OSDELR) matches the current scan line (OSDEVR), the OSDFL flag is set in the OSD status register. If the OIEN bit is set in the OSD control register, an interrupt to the CPU will also be generated. This allows the program to load the next row of character addresses, attributes and display positioning information into the external rank of latches.

The information that needs to be updated for each row consists of: the target scan line, the region of scan lines to display, the horizontal positioning, character size, and the character addresses and attributes. The target scan line is defined in the OSDELR, and the OSDMRR contents indicate the region of scan lines to display for that row. The horizontal positioning and character size are established in the OSDHDR. The character addresses and mid-row control codes are stored in the OSDCH1-34 character registers, and the default attributes for the row are defined in the OSDVCR1-2 video control registers.

The worst-case time available to the program for updating the registers extends from the beginning of the previous displayed character row (when the OSDFL is set) to the end of that row. In a 525-line 15,734 Hz line-rate system, this interval is approximately 826 µs. The CPU program will commonly require about 468 cycles (117 µs at 4 MHz bus rate) to load 34 bytes of data and five attribute registers into the (external-rank) registers, leaving the remainder of the character row display time for the CPU to do other tasks. This constraint holds only if the next row to be displayed is immediately below the previous row. The time is longer if the next row is separated from the previous row by one or more non-display scan lines.

The OSD interrupt must be cleared in the interrupt service routine by writing a logic zero to the OSDFL bit in the OSDSR.

11.3.3 Software Controlled Features

The following subsections describe features controlled by software.

11.3.3.1 Scrolling

Soft scrolling is accomplished by manipulating the OSDELR and OSDMRR registers on a periodic basis. The OSDELR contains the target scan line and should be gradually decreased to scroll a display up the screen. The OSDMRR contains the starting and ending lines of a row to display. When scrolling up onto the bottom of the screen, the range of lines should be gradually increased from the top line of the row. When scrolling up off the top of the screen, the range of lines should be gradually decreased to the bottom line of the row. The OSD may provide the time base for the scrolling speed or the MFT may be used.

11.3.3.2 Blinking

Blinking text is achieved by utilizing mid-row control codes and the BLINKEN bit in the OSDECR. Blinking text is designated by a proceeding control code character with BLINK set, and a following control code character with BLINK cleared. The designated text will be replaced with border space characters whenever the BLINKEN bit in the OSDECR is set. The BLINKEN bit can be toggled at a rate determined by the OSD or the MFT.

11.3.4 Character ROM

The character ROM is user-definable; its contents are specified at the time of product order along with the contents of the program ROM. This allows the programmer to form graphics elements as well as alphanumerics, allowing bar graphs, boxes, and an icon-oriented operator interface.

11.4 Input/Output

The OSD has four dedicated pins and shares four pins with port D.The R, G, B and FBKG outputs from the OSD are at HCMOS logic levels, intended for connection to external analog video circuitry between a source of external analog video (typically, the detector output or auxiliary input of a receiver) and a display device (usually a CRT).

11.4.1 HSYNC and VSYNC

These two dedicated inputs provide the horizontal and vertical timing reference from the external video system. Both pins have internal Schmitt triggers to improve noise immunity. The HINV and VINV bits in the OSDOCR select the polarity of the HSYNC and VSYNC inputs.

11.4.2 PLLGND and VCO

These two dedicated pins are used to tailor the 28.2 MHz PLL loop filter and center frequency. A typical filter circuit for the PLL is shown in Figure 11-4.

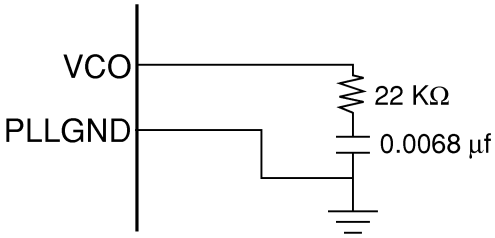


Figure 11-4. PLL Filter Circuit

NOTE

The PLLGND pin is connected internally to chip V_{SS}. Do **NOT** connect PLLGND to V_{SS} externally.

11.4.3 R, G, and B Outputs

The R, G, and B outputs are the red, green and blue color-encoded pixel signals which form the characters to be displayed.

The R, G, and B outputs share PD5, PD4 and PD3 with port D. When the REN, GEN, and BEN bits in the OSDOCR are at a logic zero, these pins become PD5, PD4, and PD3 and follow the port D DDR assignments. When the REN, GEN, and BEN bits are at a logic one, these pins become R, G, and B outputs regardless of the port D DDR assignments. The R, G, and B outputs can be enabled independently. The R, G, and B output polarity is determined by the CINV bit in the OSDOCR. The OSDEN bit in the OSD enable control register has no effect on these outputs.

During horizontal and vertical flyback, the FBKG and R, G, and B outputs are driven to their inactive state.

11.4.4 FBKG

The FBKG (Fast BlanKinG) signal is intended to blank the external video source so that the combination of OSD and external video is non-additive. FBKG is asserted in five places:

1. Character foreground (regardless of programmed color)
2. Character rounding (if enabled)
3. Character black outline (if enabled)
4. Character background (if not transparent)
5. Border (if not transparent).

The FBKG output shares PD6 with port D. When the FBKGEN bit in the OSDECR is a logic zero, this pin becomes PD6 and follows the port D DDR assignment. When the FBKGEN bit is a logic one, this pin becomes the FBKG output regardless of the port D DDR assignment. FBKG can be enabled independently of the R, G, and B outputs. The FBKG output polarity is determined by the FBINV bit in the OSDOCR. The OSDEN bit in the OSD enable control register has no effect on this output.

11.5 Registers

The OSD has 44 registers, 39 of which are double-ranked to prevent CPU access from disturbing the video display.

11.5.1 OSD Character Registers (OSDCH1-OSDCH34)

The 34 double-ranked character registers, OSDCH1 through OSDCH34, each contain a character address or a video control code. A character address specifies a character from the OSD ROM for display; a video control code, which appears as a background space, is used to modify display attributes **between** words. Figure 11-5 details each bit in the registers.

		7	6	5	4	3	2	1	0
OSDCH1-34 $100-$121	RD	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
	WR								
	RST	U	U	U	U	U	U	U	U

Figure 11-5. OSD Character Registers (OSDCH1-34)

11.5.1.1 CH7 - CHaracter code 7

CH7 determines whether CH6:0 are interpreted as a character address or a video control code. A reset has no effect on this bit.
CH7 = 1: CH6:0 are interpreted as a video control code
CH7 = 0: CH6:0 are interpreted as a character address

11.5.1.2 CH6:0 - CHaracter code 6:0

When CH7 = 0, CH6:0 address one of the 128 characters in the character ROM. Code $00 represents a fixed border space character.

When CH7 = 1, CH6 is used to select one of two video control code formats. CH5:0 are interpreted by the hardware as described in Figure 11-7.

A reset has no effect on these bits.

11.5.2 OSD Video Control Registers (OSDVCR1-OSDVCR2)

The two double-ranked registers, OSDVCR1 and OSDVCR2, each contain a video control code. These two registers allow character row attributes to be established **prior** to display. Figure 11-6 details each bit in the registers.

		7	6	5	4	3	2	1	0
OSDVCR1-2 $122-$123	RD	VC7	VC6	VC5	VC4	VC3	VC2	VC1	VC0
	WR								
	RST	U	U	U	U	U	U	U	U

Figure 11-6. OSD Video Control Registers (OSDVCR1-OSDVCR2)

11.5.2.1 VC7 - Video Control 7

VC7 determines whether VC6:0 are interpreted as a video control code. A reset has no effect on this bit.
VC7 = 1: VC6:0 are interpreted as a video control code
VC7 = 0: VC6:0 are ignored

11.5.2.2 VC6 - Video Control 6

When VC7 = 1, VC6 is used to select one of two video control code formats. VC5:0 are interpreted by the hardware as described in Figure 11-7. Either video control code may be modified at either address. A reset has no effect on this bit.

11.5.2.3 VC5:0 - Video Control 5:0

These six bits contain a video control code. The interpretation of VC5:0 depends on the values of VC7 and VC6, as shown in Figure 11-7. A reset has no effect on these bits.

	VC7/CH7	VC6/CH6	VC5/CH5	VC4/CH4	VC3/CH3	VC2/CH2	VC1/CH1	VC0/CH0
FORMAT A	1	0	BLINK	FITAL	FUL	FGR	FGB	FGG
FORMAT B	1	1	RNDEN	BEEN	BKS	BKR	BKB	BKG

Figure 11-7. Video Control Code Formats

Control code commands are executed as follows:

1. A background color change command occurs midway through the control code space.
2. Underline, foreground color, black outline, and italics changes occur at the beginning of the characters to follow the control code space.

BLINK - BLINK foreground

BLINK determines, in conjunction with the BLINKEN bit in the OSDECR, which text is blinking on the screen. The BLINKEN bit is toggled, through software, at the desired blinking frequency. Foreground text (including underline) which is preceded by a video control code with BLINK set will flash at the BLINKEN frequency.

BLINK = 1: Subsequent foreground will be replaced by background color when BLINKEN=1
BLINK = 0: Subsequent foreground will not blink

RNDEN - character RouNDing ENable

RNDEN determines whether foreground characters are rounded. An example of rounding is shown in Figure 11-2.

RNDEN = 1: Subsequent foreground will be rounded
RNDEN = 0: Subsequent foreground will not be rounded

FITAL - Foreground ITALics

FITAL determines when characters are displayed in italics by slanting. An example of italics is shown in Figure 11-2.

FITAL = 1: Subsequent foreground will be italicized
FITAL = 0: Subsequent foreground will not be italicized

BEEN - Black Edge ENable

BEEN determines whether foreground characters are black outlined. An example of black outline is shown in Figure 11-2. RNDEN should always be set when BEEN is set.

BEEN = 1: Subsequent foreground is black outlined
BEEN = 0: Subsequent foreground is not black outlined

FUL - Foreground UnderLine

FUL determines when character codes $01-$7F are displayed with an underline. An example of underline is shown in Figure 11-2.

FUL = 1: Subsequent foreground character codes $01-$7F are underlined
FUL = 0: Subsequent foreground is not underlined

BKS - BacKground Solid

BKS determines whether the background is transparent or visible.

BKS = 1: the background is a solid color (On) determined by BKR, BKB and BKG
BKS = 0: the background is transparent (Off)

FGR, FGB, FGG - ForeGround RBG

These bits define the foreground color for the subsequent characters being displayed, according to Table 11-1.

BKR, BKB, BKG - Background Color

These bits define the background color for the subsequent characters being displayed, according to Table 11-1. BKS must be set to display a background color.

Table 11-1. RBG Color Map

R	B	G	Color
0	**0**	**0**	**Black**
0	0	1	Green
0	1	0	Blue
0	1	1	Cyan
1	0	0	Red
1	0	1	Yellow
1	1	0	Magenta
1	1	1	White

11.5.3 OSD Enable Control Register (OSDECR)

The OSDECR contains enable and control bits for the OSD. Figure 11-8 details each bit in this register.

		7	6	5	4	3	2	1	0
OSDECR $22	RD	OIEN	OSDEN	XFER	PLLEN	FBKGEN	PLL36	BLINKEN	HMODE
	WR								
	RST	0	0	0	0	0	0	0	0

Figure 11-8. OSD Enable Control Register (OSDECR)

11.5.3.1 OIEN - Osd Interrupt ENable

OIEN determines whether the OSDFL bit in the OSDSR is enabled to generate interrupt requests to the CPU. A reset clears this bit.

OIEN = 1: OSDFL interrupt enabled
OIEN = 0: OSDFL interrupt disabled

11.5.3.2 OSDEN - OSD ENable

OSDEN determines whether the OSD is enabled or disabled. The PLLEN bit must also be set for OSD operation, and the DSL must be enabled to provide field information for interlacing. A reset clears this bit.

OSDEN = 1: OSD enabled
OSDEN = 0: OSD disabled

11.5.3.3 XFER - eXternal to internal rank transFER enable

XFER determines whether the external rank of registers is transferred to the internal rank upon an event line match. A reset clears this bit.

XFER = 1: transfers enabled
XFER = 0: transfers disabled

11.5.3.4 PLLEN - PLL ENable

PLLEN determines whether the phase-locked loop oscillator is enabled. When enabling the PLL, the program must wait for the PLL to stabilize before activating the OSD to achieve a stable display. The stabilization time depends on the external components chosen to implement the PLL loop filter on the VCO pin. A reset clears this bit.

PLLEN = 1: PLL enabled
PLLEN = 0: PLL disabled

NOTE

The PLL should be disabled before STOP mode is entered; in addition, it may be useful to stop the PLL whenever OSD data are not currently being displayed, to eliminate ingress of 28.2 MHz interference to the RF, IF or baseband portions of the external video chain.

11.5.3.5 FBKGEN - Fast BlanKinG ENable

FBKGEN determines whether the FBKG signal is enabled to output on PD6. Clearing this bit after using FBKG as an OSD output restores the pin to its previous port D condition. A reset clears this bit.

FBKGEN = 1: FBKG output is enabled and controls PD6
FBKGEN = 0: FBKG output is disabled and PD6 is under the control of port D

11.5.3.6 PLL36 - PLL 36MHz

PLL36 determines whether the PLL should run at 28.2 MHz or 36.3 MHz. A reset clears this bit.

PLL36 = 1: PLL will run at 36.3 MHz
PLL36 = 0: PLL will run at 28.2 MHz

11.5.3.7 BLINKEN - BLINK ENable

BLINKEN determines whether text following a control code with BLINK set will be displayed. BLINKEN should be toggled in software to establish the desired blinking frequency. A reset clears this bit.

BLINKEN = 1: Text following a control code with BLINK set will not be displayed
BLINKEN = 0: Text following a control code with BLINK set will be displayed

11.5.3.8 HMODE - 1H or 2H MODE line frequency

HMODE selects one of two chassis horizontal frequencies. A reset clears this bit.
HMODE = 1: 2H mode, chassis horizontal frequency of 31.468 KHz
HMODE = 0: 1H mode, chassis horizontal frequency of 15.734 KHz

11.5.4 OSD Event Line Number Register (OSDELR)

The OSDELR contains the target 8-bit field scan line address for the next row to be displayed. Figure 11-9 details each bit in the register.

		7	6	5	4	3	2	1	0
OSDELR $23	RD WR	ELN7	ELN6	ELN5	ELN4	ELN3	ELN2	ELN1	ELN0
	RST	0	0	0	0	0	0	0	0

Figure 11-9. OSD Event Line Number Register (OSDELR)

11.5.4.1 ELN7:0 - Event Line Number 7:0

ELN7:0 define where to display the first scan line of the next character row. This value is compared to the internal 9-bit scan line counter with ELN8 = 0. The usable range of scan lines is between 0 and 255. A reset clears these bits.

11.5.5 OSD Event Count Register (OSDEVR)

The OSDEVR contains the current 8-bit field scan line address. Figure 11-10 details each bit in the register.

		7	6	5	4	3	2	1	0
OSDEVR $29	RD	EV7	EV6	EV5	EV4	EV3	EV2	EV1	EV0
	WR	▨	▨	▨	▨	▨	▨	▨	▨
	RST	U	U	U	U	U	U	U	U

Figure 11-10. OSD Event Count Register (OSDEVR)

11.5.5.1 EV7:0 - EVent count 7:0

These read-only bits provide the user with visibility of the internal 9-bit field scan line counter. EV7:0 represent only the 8 least significant bits. A reset has no effect on these bits.

11.5.6 OSD Output Control Register (OSDOCR)

The OSDOCR contains input and output control bits for the OSD. Figure 11-11 details each bit in the register.

		7	6	5	4	3	2	1	0
OSDOCR $24	RD WR	HINV	VINV	CINV	FBINV	FDINV	REN	BEN	GEN
	RST	0	0	0	0	0	0	0	0

Figure 11-11. OSD Output Control Register (OSDOCR)

11.5.6.1 HINV - Hsync INVert

HINV determines whether the HSYNC input is active low or active high. A reset clears this bit.

HINV = 1: HSYNC input is active low and is inverted internally
HINV = 0: HSYNC input is active high

11.5.6.2 VINV - Vsync INVert

VINV determines whether the VSYNC input is active low or active high. A reset clears this bit.

VINV = 1: VSYNC input is active low and is inverted internally
VINV = 0: VSYNC input is active high

11.5.6.3 CINV - Color INVert

CINV determines whether the R, G, and B outputs are active low or active high. A reset clears this bit.

CINV = 1: R, G and B outputs are active low
CINV = 0: R, G and B outputs are active high

11.5.6.4 FBINV - Fast Blanking INVert

FBINV determines whether the FBKG output is active low or active high. A reset clears this bit.

FBINV = 1: FBKG output is active low
FBINV = 0: FBKG output is active high

11.5.6.5 FDINV - FielD INVert

FDINV determines whether the field indicator from the DSL is inverted. The odd/even field indicator is used to correctly position the odd field-related scan data relative to the even field scan line data in interlaced display. A reset clears this bit.
FDINV = 1: Odd/Even Field indicator from the data slicer is inverted
FDINV = 0: Odd/Even Field indicator from the data slicer is not inverted

11.5.6.6 REN, BEN, GEN - Red, Blue and Green output ENable

REG, BEN and GEN determine whether the R, B and G outputs are enabled to use PD5, PD3 or PD4, respectively. Clearing these bits after using R, B and G as OSD outputs restores the pins to their previous port D condition. A reset clears these bits.

REN, BEN or GEN = 1:	R, B, or G output is enabled, using a port D pin
REN, BEN or GEN = 0:	R, B or G output is disabled, returning control of the pin to port D

11.5.7 OSD Horizontal Delay and Character Size Register (OSDHDR)

The double-ranked OSDHDR contains OSD output control bits. Figure 11-12 details each bit in this register.

		7	6	5	4	3	2	1	0
OSDHDR $25	RD	CHHS	CHWS	HD5	HD4	HD3	HD2	HD1	HD0
	WR								
	RST	0	0	0	0	0	0	0	0

Figure 11-12. OSD Horizontal Delay and Character Size Register (OSDHDR)

11.5.7.1 CHHS - CHaracter Height Select

CHHS determines whether characters are displayed at standard height and width, or at double height with width selected by CHWS. Character height is selected for an entire row. A reset clears this bit.
CHHS = 1: 2X height characters, CHWS will determine the character width
CHHS = 0: standard size (height and width) characters

11.5.7.2 CHWS - CHaracter Width Select

CHWS determines the character width when CHHS = 1. Character width is selected for an entire row. A reset clears this bit.
CHWS = 1: 2X width characters are selected
CHWS = 0: 1.5X width characters are selected

11.5.7.3 HD5:0 - Horizontal Delay 5:0

The six horizontal position delay bits, HD5:0, determine the horizontal positioning of the OSD characters with respect to the rising edge of the internal (conditioned with HINV bit) HSYNC. Each increment provides one character bit of shift, at the character size selected by the CHHS and CHWS bits; 63 increments are provided by the 6-bit field. In addition to the specified delay, there is a built-in delay as shown in Table 11-2.

Table 11-2. OSD Horizontal Delay

	Standard Width		1.5X Width		2X Width	
	HD = 0	**HD = 63**	**HD = 0**	**HD = 63**	**HD = 0**	**HD = 63**
1H MODE	7.2µs	15.9µs	8.9µs	22.0µs	10.9µs	28.4µs
2H MODE	3.6µs	8.0µs	4.5µs	11.0µs	5.5µs	14.2µs

11.5.8 OSD Status Register

The OSDSR contains the OSD interrupt flag bit and provides visibility of the external input. Figure 11-13 details each bit in the register.

		7	6	5	4	3	2	1	0
OSDSR $26	RD	OSDFL	HSYN	VSYN	0	0	0	0	0
	WR		▨	▨	▨	▨	▨	▨	▨
	RST	0	▨	▨	0	0	0	0	0

Figure 11-13. OSD Status Register (OSDSR)

11.5.8.1 OSDFL - OSD FLag

This bit is set when the current field scan line (OSDEVR) matches the target field scan line (OSDELR). It will cause an interrupt if the OIEN bit is set in the OSDECR. Writing a logic zero to this bit will clear it. A reset also clears this bit.

11.5.8.2 HSYN - Horizontal SYNc pulse

This read-only bit provides visibility to the HSYNC input. A logic one indicates the presence of a horizontal sync pulse. A reset has no effect on this bit.

11.5.8.3 VSYN - Vertical SYNc pulse

This read-only bit provides visibility to the VSYNC input. A logic one indicates the presence of a vertical sync pulse. A reset has no effect on this bit.

11.5.9 OSD Matrix Range Register (OSDMRR)

The double-ranked OSDMRR contains OSD output control bits used for software soft scrolling. Figure 11-14 details each bit in this register.

		7	6	5	4	3	2	1	0
OSDMRR $27	RD WR	MS3	MS2	MS1	MS0	ME3	ME2	ME1	ME0
	RST	0	0	0	0	0	0	0	0

Figure 11-14. OSD Matrix Range Register (OSDMRR)

11.5.9.1 MS3:0 - Matrix Start line 3:0

These bits set the starting scan line within the character row. MS3:0 = 0 means that the row display starts at the first scan line. Valid start numbers are 0 through 12. A reset clears these bits.

11.5.9.2 ME3:0 - Matrix End line 3:0

These bits set the ending scan line of the character row (non-inclusive). ME3:0 = 13 means that the row is displayed through its last scan line. Valid end numbers are 1 through 13. A reset clears these bits.

11.5.10 OSD Border Control Register (OSDBCR)

The OSDBCR contains four OSD output control bits. Figure 11-15 details each bit in the register.

		7	6	5	4	3	2	1	0
OSDBCR $28	RD	0	0	0	0	BOS	BOR	BOB	BOG
	WR								
	RST	0	0	0	0	0	0	0	0

Figure 11-15. OSD Border Control Register (OSDBCR)

11.5.10.1 BOS - BOrder Solid

BOS determines whether the border is transparent or visible. A reset clears this bit.

BOS = 1: border is a solid color (On) determined by BOR, BOB, and BOG
BOS = 0: border is transparent (Off)

11.5.10.2 BOR, BOB, BOG - BOrder Red, Blue, Green

These bits define the color of the character border according to Table 11-1. A reset clears these bits.

11.6 Low Power Modes

Low power modes are discussed in the following subsections.

11.6.1 Operation During WAIT Mode

The OSD remains active during WAIT mode; however, it will be unable to interrupt the CPU and bring it out of WAIT mode. It is recommended that the OSD and PLL be disabled during WAIT mode, unless a single row of fixed video output is desired.

11.6.2 Operation During STOP Mode

The OSD remains active during STOP mode; however, it will be unable to interrupt the CPU and bring it out of STOP mode. It is recommended that the OSD and PLL be disabled during STOP mode.

11.7 Interrupts and Resets

The OSD has one source of interrupt, the OSDFL flag in the OSDSR. This interrupt is enabled by the OIEN bit in the OSDECR. This interrupt will cause the MCU to vector to the address stored in $7FF6-$7FF7.

SECTION 12
CLOSED-CAPTION DATA SLICER

The closed-caption data slicer (DSL) extracts FCC closed-caption compatible data from an NTSC composite video signal for closed-caption and extended data services applications.

12.1 Features

The DSL provides the following features:

- FCC Line 21 Format Data Extraction on both Field 1 and Field 2
- Software Programmable Line Selection
- Hardware Parity Checking
- Software Programmable Data Slicing Level

12.2 Overview

The closed-caption data slicer (DSL) extracts FCC closed-caption compatible data from an NTSC composite video signal. The DSL accomplishes this by slicing sync and data information from the incoming video; the sync information is used to locate fields and lines to trigger data sampling, and the sampled sliced data is stored in registers for CPU access. A block diagram of the DSL is shown in Figure 12-1.

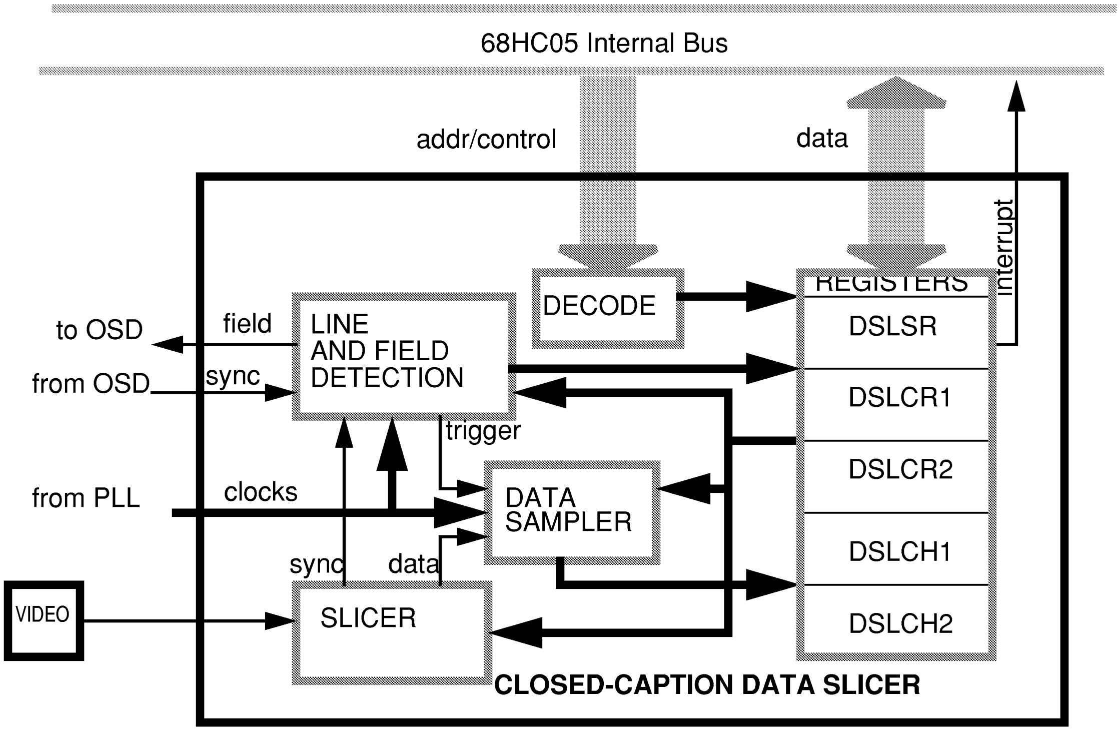

68HC05 Internal Bus
addr/control
data
interrupt
DECODE
REGISTERS
DSLSR
DSLCR1
DSLCR2
DSLCH1
DSLCH2
to OSD
field
LINE
AND FIELD
DETECTION
from OSD
sync
trigger
from PLL
clocks
DATA
SAMPLER
sync
data
VIDEO
SLICER
CLOSED-CAPTION DATA SLICER

Figure 12-1. Data Slicer Block Diagram

12.2.1 Slicer

The slicer circuitry compares the composite video input with internal reference levels to determine when sync pulses or data bits are being received. DSL control bits allow the user to select one of four reference levels for slicing data. Refer to Figure 12-2.

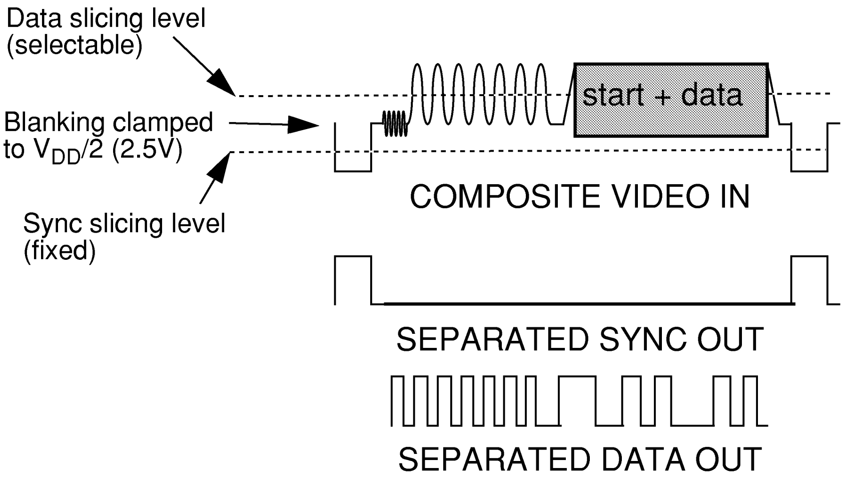

Data slicing level
(selectable)
Blanking clamped
to $V_{DD}/2$ (2.5V)
Sync slicing level
(fixed)
start + data
COMPOSITE VIDEO IN
SEPARATED SYNC OUT
SEPARATED DATA OUT

Figure 12-2. : SLICER Input and Output

12.2.2 Line and Field Detection

The line and field detection circuitry uses the separated sync output of the slicer section, along with signals derived from the HSYNC and VSYNC inputs, to determine line and field timing for the DSL and field timing for the OSD. DSL control registers enable the user to select one of several lines from which to extract data in closed caption format, and also provide flexibility for adapting to differences in chassis sync and video signal timing.

12.2.3 Data Sampling

The data sampling circuitry uses the frequency reference from the PLL (phase locked to the HSYNC input) to sample data bits from the separated data output of the slicer section. The sampling circuitry performs parity checking and stores the data in parallel format in registers which the CPU can access. The DSL sets a status register bit (DSFL) and optionally will generate an interrupt to indicate when data is available.

12.3 Programming Guidelines

The DSL can be used for extracting data in the FCC closed-caption format from both fields in the composite video signal. The DSL may be configured to search for data on any line from five to 36 (VSYNC timing may limit this range in some applications), although it cannot search on multiple lines. The DSL does not perform any closed-caption decoding or display; these functions must be provided in software with support of the OSD module.

12.3.1 Setup

The DSL control registers 1 and 2 must be initialized for proper operation of the DSL. The DSEN bit in DSLCR1 should be set to enable the DSL, and the DSIEN bit should be set if the DSFL will be serviced through interrupts rather than polling. The desired line for decoding data should be specified in DSLCR1, and the data slicing voltage reference, vertical pulse width and delay, and 2H clock phase delay should be initialized in DSLCR2. Some of these control bits may require adjustment during operation if video input conditions change.

12.3.2 Interrupt Servicing

In both the interrupt and polling methods, the DSFL flag in the DSLSR indicates that 2 bytes are available for reading. The FIELD1 bit in the DSLSR indicates which field the 2 bytes have been extracted from. The OVFL bit should also be checked to ensure that overflow has not occurred. Within the service routine, DSFL should be cleared by writing to the DSLSR.

Since the two character registers are shared for both field 1 and field 2, the user has approximately one video vertical scan (16.68ms) to unload 2 bytes of data per field. The 2 bytes each contain a 7-bit code and a parity error flag. The parity error flags indicate that the data received may be invalid.

12.3.3 Debugging

The DSLSR status bits can be helpful when using the DSL in a new application. If the DSL does not correctly slice data, check the following status bits in this order:

1. CSYNC
   This bit provides visibility of the separated sync output of the slicer section. If CSYNC is not set during the time that a sync pulse is present in the incoming video, then the input level of the video signal needs adjustment to allow proper sync slicing.

2. VPDET
   This bit indicates that a wide vertical sync pulse has been detected in the composite video. If VPDET is not set after the vertical blanking interval, this indicates that either the wide vertical sync pulses in the composite video are shorter than the pulse width defined by PW1:0, or the VSYNC input was not active during the vertical blanking interval, or the selected decoding line was encountered before the vertical sync was seen in composite video or VSYNC.

3. INTLC
   The decoding circuitry is not dependent on INTLC, but INTLC=0 may indicate a problem. In field 1, there should be an even number of .5H (H=horizontal period) intervals from the rising edge of the first wide vertical sync pulse to the the rising edge of the target line's horizontal sync pulse. In field 2, there should be an odd number of .5H intervals. If INTLC=0, the timing from the vertical sync pulse to the horizontal sync does not look like interlaced video.

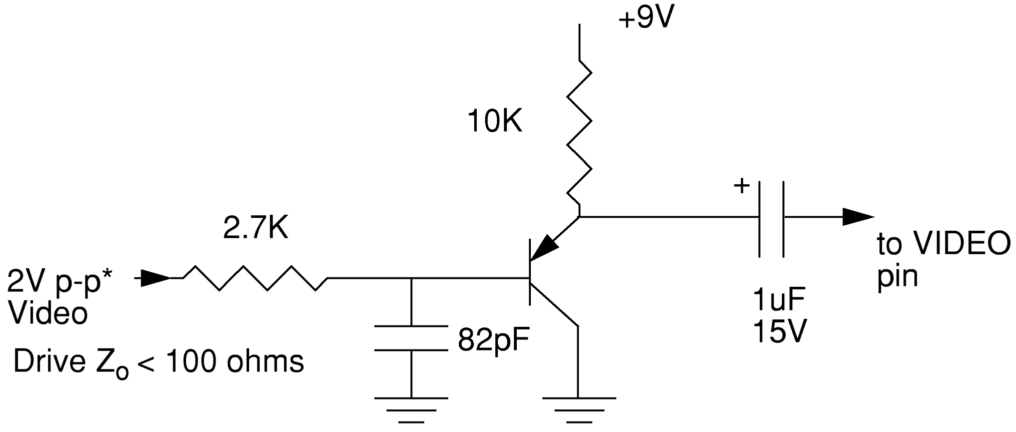
4. RIC1:0
   These two bits are the output of a counter which counts up to three rising edges in the sliced data during the run-in-clock window. If the count is less than three, this may indicate that the incorrect line is being decoded, or that there is too much delay between composite video and HSYNC.

12.4 Input/Output

The DSL has one dedicated pin. It also uses the VSYNC and HSYNC inputs indirectly.

12.4.1 Video

This dedicated input provides the NTSC composite video signal from the external video system. The closed-caption data and sync information is extracted from this signal. A typical input circuit for the closed-caption video input is shown in Figure 12-3. The video input uses a sync versus blanking level duty cycle clamp to set the blanking level of the incoming video to $V_{DD}/2$.



* negative sync tip to 100 IRE white

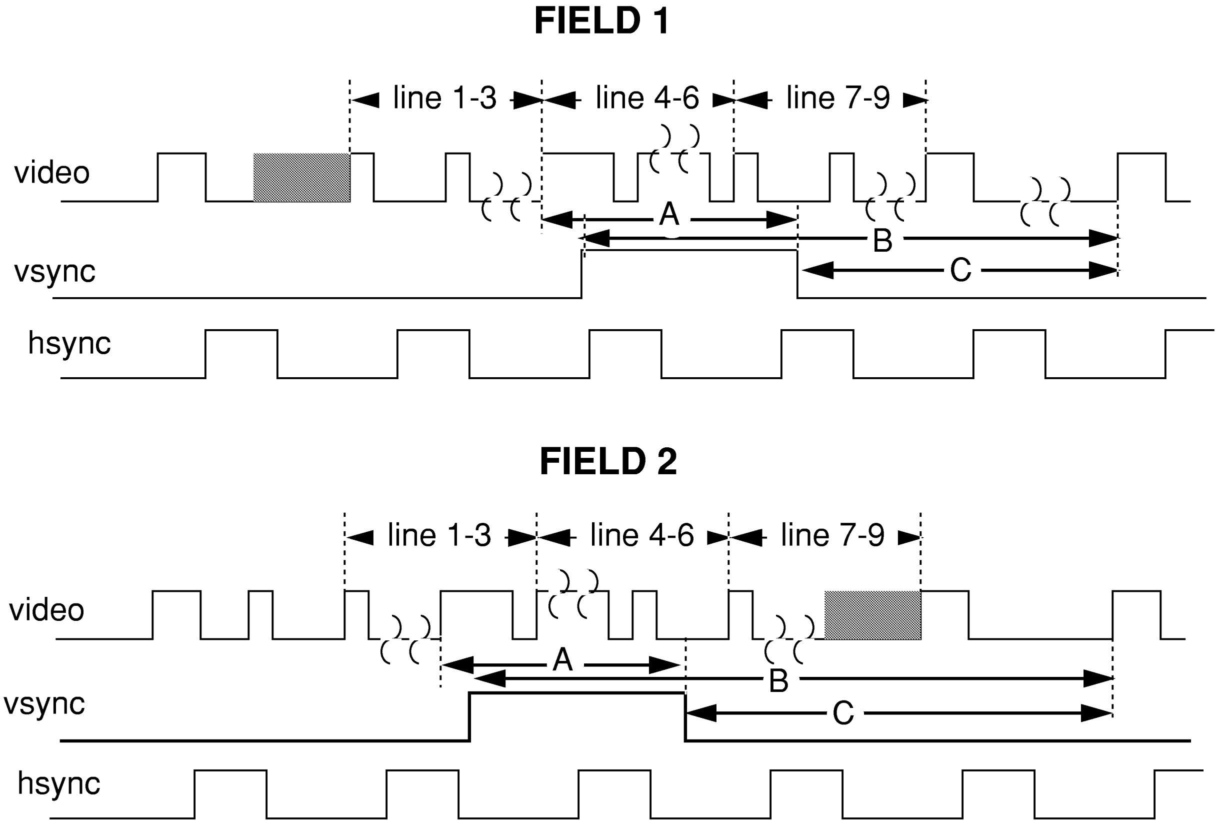
Figure 12-3. Video Input Circuit

12.4.2 VSYNC, HSYNC, and Composite Video Input Requirements

The DSL uses both VSYNC and HSYNC indirectly and puts some restrictions on the relationship between these two signals and composite video. Refer to Figure 12-4.

The external VSYNC input is provided by the OSD to the DSL as an active-high signal. VSYNC is used by the DSL to:

- detect a valid vertical blanking interval
- disable the video input clamp during vertical blanking
- generate field information for the DSL and OSD
- verify an interlaced composite video input, and
- latch an overflow condition.



A: rising edge CSYNC first wide vertical pulse to falling edge VSYNC - min. 32 µs
B: rising edge VSYNC to rising edge CSYNC of line to be detected - min. 64 µs
C: falling edge VSYNC to rising edge CSYNC of line to be detected - min. 32 µs

Note: If the rising edge of VSYNC is delayed from CSYNC by more than the width of a wide vertical pulse, the target line count may have to be adjusted.

Figure 12-4. : DSL Input Timing

A synthesized version (50% duty cycle) of external HSYNC is provided by the PLL to the DSL. The DSL also uses higher frequency PLL derivatives of HSYNC for most of its synchronization. The horizontal frequency is used to generate field information for the OSD.

The VIDEO input is separated into sync and data components. Composite sync (both vertical and horizontal sync information) is used by the DSL to:

- detect a valid vertical blanking interval
- generate field information for the DSL

- verify an interlaced composite video input, and
- indicate when the desired decoding line has been found.

Composite data is used for detecting the run-in-clock, the start bit, and sampling the data.

12.5 Registers

The DSL has six registers.

12.5.1 DSL Character Registers (DSLCH1-DSLCH2)

These registers, DSLCH1 and DSLCH2, each contain seven data bits and a parity error flag. Figure 12-5 details each bit in these registers.

		7	6	5	4	3	2	1	0
DSLCH1-2 $16-17	RD	PE	DA6	DA5	DA4	DA3	DA2	DA1	DA0
	WR								
	RST	0	U	U	U	U	U	U	U

Figure 12-5. : DSL Character Registers (DSLCH1-DSLCH2)

12.5.1.1 PE - Parity Error

This read-only bit is set when odd parity is not detected on the byte received. Writing to the DSL status register clears this bit. A reset also clears this bit.

12.5.1.2 DA6:0 - closed-caption DAta 6:0

These read-only bits are sampled (LSB first) from the data slicer, which extracts data from the VIDEO input pin. These seven bits will define a closed-caption character. A reset has no effect on these bits.

12.5.2 DSL Control Register 1 (DSLCR1)

DSLCR1 contains eight control bits. Figure 12-6 details each bit in the register.

		7	6	5	4	3	2	1	0
DSLCR1 $13	RD	DSIEN	DSEN	LINE5	LINE4	LINE3	LINE2	LINE1	0
	WR								▨
	RST	0	0	0	0	0	0	0	0

Figure 12-6. : DSL Control Register 1 (DSLCR1)

12.5.2.1 DSIEN - Data Slicer Interrupt ENable

DSIEN determines whether the DSFL bit in the DSLSR is enabled to generate interrupt requests to the CPU. A reset clears this bit.

DSIEN = 1: DSFL interrupt enabled
DSIEN = 0: DSFL interrupt disabled

12.5.2.2 DSEN - Data Slicer ENable

DSEN determines whether the DSL is enabled. When the DSL is enabled, the data slicer voltage reference is turned on and the PLL clock outputs are input to the DSL circuitry. The PLL must be enabled (PLLEN bit in the OSDECR) to operate the DSL. The DSL provides field information to the OSD and should not be disabled if the OSD is in use. A reset clears this bit.

DSEN = 1: DSL enabled
DSEN = 0: DSL disabled

12.5.2.3 LINE5:1 - closed-caption LINE 5:1

These five bits allow the user to specify the line to be used for closed-caption decoding, according to Table 12-1. A reset clears these bits.

Table 12-1. Line Selection

LINE5:1	Target Line
00000	5
00001	6
00010	7
00011	8
00100	9
:	:
01111	20
10000	21
10001	22
:	:
11101	34
11110	35
11111	36

12.5.3 DSL Control Register 2 (DSLCR2)

DSLCR2 contains eight control bits. Figure 12-7 details each bit in the register.

		7	6	5	4	3	2	1	0
DSLCR2 $14	RD WR	VR1	VR0	PW1	PW0	VPD	PD2	PD1	PD0
	RST	0	0	0	0	0	0	0	0

Figure 12-7. DSL Control Register 2 (DSLCR2)

12.5.3.1 VR1:0 - Voltage Reference 1:0

These bits select the reference bias voltage for slicing data according to Table 12-2. As seen in Figure 12-8, the input blanking level is clamped to 2.5 V, and the sync slicing level is fixed at 2.31 V. A reset clears these bits.

Table 12-2. Data Slicer Bias Voltages

VR1	VR0	Nominal Bias
0	0	3.12V
0	1	2.98V
1	0	2.84V
1	1	2.70V

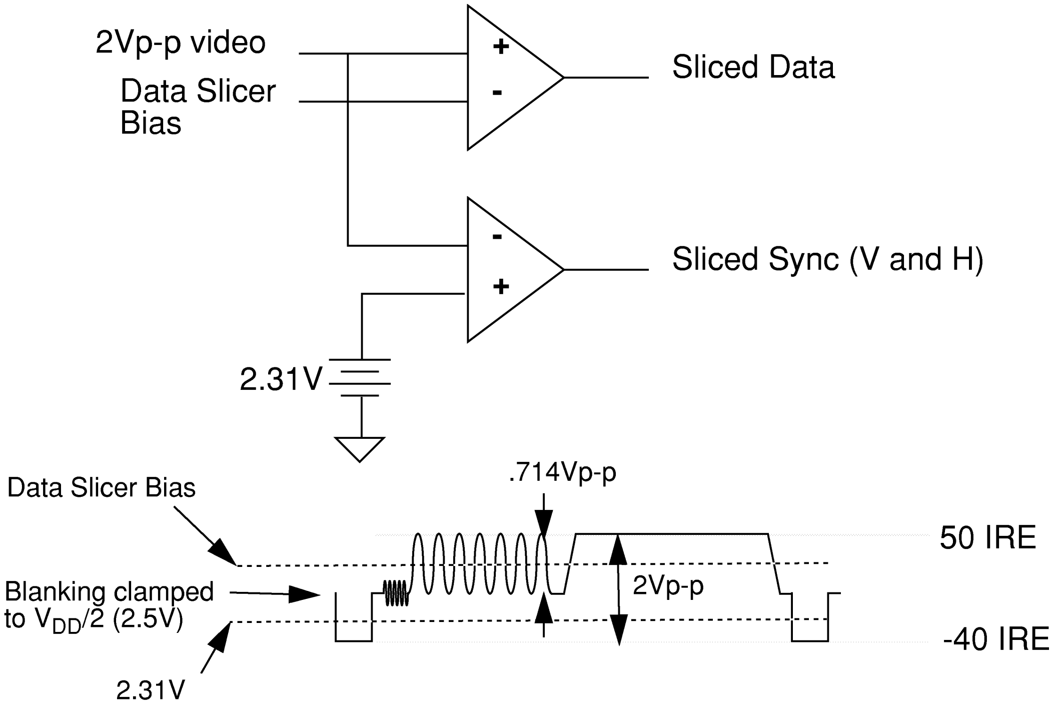

2Vp-p video
Data Slicer Bias
+
-
Sliced Data
-
+
Sliced Sync (V and H)
2.31V
Data Slicer Bias
.714Vp-p
50 IRE
Blanking clamped to $V_{DD}/2$ (2.5V)
2Vp-p
-40 IRE
2.31V

Figure 12-8. Data / Sync Slicing

12.5.3.2 PW1:0 - vertical sync Pulse Width 1:0

These bits define, according to Table 12-3, the minimum width of an extracted sync signal pulse required to identify it as a vertical sync pulse. The standard video vertical sync interval pulse width is 29.2 µs; the DSL requires at least 8 µs to recognize a vertical sync pulse. A reset clears these bits.

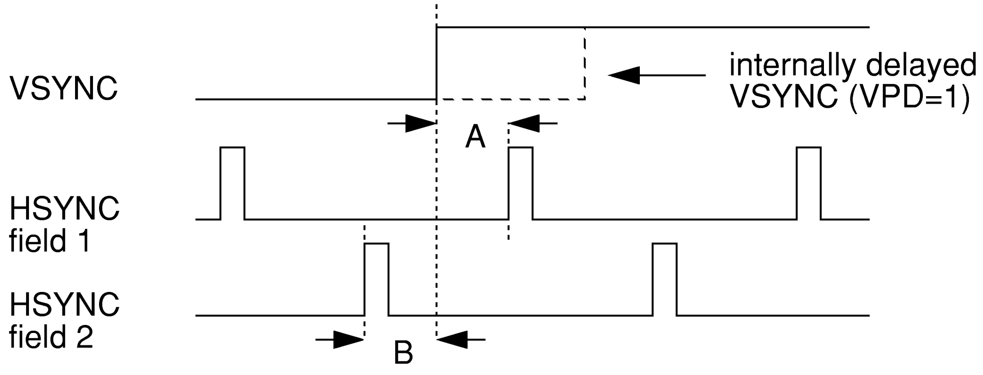
Table 12-3. : Minimum Vertical Sync Pulse Width

PW1	PW0	Pulse Width
0	0	8 µs
0	1	10 µs
1	0	12 µs
1	1	14 µs

12.5.3.3 VPD - Vertical Pulse Delay

VPD determines whether the VSYNC input is delayed before field detection. VPD should be set when the VSYNC active edge is within +/-4 µs of the HSYNC active edge for either field. See Figure 12-9 for clarification. A reset clears this bit.

VPD = 1: VSYNC input is delayed by 8 to 24 µs before field detection
VPD = 0: VSYNC input is not delayed



If |A| < 4µs, or |B| < 4µs, set VPD = 1

Figure 12-9. Conditions for setting VPD

12.5.3.4 PD2:0 - Phase Delay of line counter clock 2:0

These bits specify the delay added to the clock input to the line counter in the line detection circuitry. The phase delay selection is a delay of the internal 2H output of the PLL. It is intended to be used in 2H mode to compensate for inherent delays in the video processing circuits relative to the incoming video. Since the internal 2H signal has a period of 32µs, the allowable phase delay selections are in 4µs increments from 0 to 28µs to allow up to one full period of shift. Although the internal 2H signal is not available externally, the internal 2H signal is phase locked to the 2H (HSYNC) input. By comparing the leading edge of the HSYNC signal to the leading edge of the 2H equalization pulses during the vertical sync interval, the video processing delay can be determined. The closest 4µs increment of the phase delay can then be selected. The phase delay control has a secondary function to allow optimization of the DSL with signals from VCRs and signals with certain types of copyguard protection. Reset clears these bits.

Table 12-4. Phase Delay Selection

PD2	PD1	PD0	Delay
0	0	0	0 µs
0	0	1	4 µs
0	1	0	8 µs
0	1	1	12 µs
1	0	0	16 µs
1	0	1	20 µs
1	1	0	24 µs
1	1	1	28 µs

12.5.4 DSL Status Register (DSLSR)

The DSLSR contains the DSL interrupt flag and status bits and provides visibility to the slicer sync output. Figure 12-10 details each bit in this register.

		7	6	5	4	3	2	1	0
DSLSR $15	RD	DSFL	OVFL	FIELD1	INTLC	CSYNC	VPDET	RIC1	RIC0
	WR			▨	▨	▨			
	RST	0	0	0	0	▨	0	0	0

Figure 12-10. DSL Status Register (DSLSR)

12.5.4.1 DSFL - Data Sampled FLag

This bit is set to indicate that the DSL has sampled 16 bits of closed-caption data (including parity) and placed them in DSLCH1 and DSLCH2. It will cause an interrupt if the DSIEN bit in DSLCR1 is set. It is cleared by writing to the DSLSR. A reset also clears this bit.

12.5.4.2 OVFL - data OVerFLow

This bit is set at the beginning of the field if the DSFL is still set from the previous field. This condition does NOT cause an interrupt. It is cleared by writing to the DSLSR. A reset also clears this bit.

12.5.4.3 FIELD1 - FIELD1 indicator

This bit indicates which field the closed-caption data in DSLCH1 and DSLCH2 was extracted from. A logic one indicates that the data in the status and character registers correspond to information retrieved from the specified line in field 1. A logic zero indicates that the register contents correspond to information retrieved from the specified line in field 2. A reset clears this bit.

12.5.4.4 INTLC - INTerLaCe flag

This bit indicates whether a normal video signal consisting of 525 lines in two consecutive fields has been detected at the VIDEO input. A logic one indicates that a normal video signal is present; a logic zero indicates that an interlaced video signal is not present. A reset clears this bit.

12.5.4.5 CSYNC - Composite separated SYNC

This bit reflects the current state of the separated sync signal. If CSYNC = 1, it indicates the presence of a vertical or horizontal sync pulse in the composite video input. If CSYNC = 0, it indicates that a sync pulse is not currently present in the composite video input. A reset has no effect on this bit.

12.5.4.6 VPDET - Vertical sync Pulse DETect

This bit is set when a vertical sync pulse is sampled in the separated sync signal according to the PW1:0 bits in DSL control register 2 and qualified by VSYNC being active between the first wide vertical sync pulse and the line to be decoded. This bit is cleared by writing to the DSLSR. A reset clears this bit.

12.5.4.7 RIC1:0 - Run-In Clock cycle count 1:0

These bits reflect the current state of a 2-bit run-in clock cycle counter. This counter triggers off of the positive transitions of the video signal in an 8µs wide window starting 12 µs after the horizontal sync pulse of the specified line. If both bits are set, this indicates that at least three positive transitions were detected. These bits are cleared by writing to the DSLSR. A reset also clears these bits.

12.6 Low Power Modes

Low power modes are discussed in the following subsections.

12.6.1 Operation During WAIT Mode

The DSL remains active during WAIT mode; however, it will be unable to interrupt the CPU and bring it out of WAIT mode. It is recommended that the DSL be disabled during WAIT mode.

12.6.2 Operation During STOP Mode

The DSL remains active during STOP mode; however, it will be unable to interrupt the CPU and bring it out of STOP mode. It is recommended that the DSL be disabled during STOP mode.

12.7 Interrupts and Resets

The DSL has one source of interrupt, the DSFL flag in the DSLSR. This interrupt is enabled by the DSIEN bit in the DSLCR1. This interrupt will cause the MCU to vector to the address stored in $7FF4-$7FF5.

SECTION 13
PULSE WIDTH MODULATOR

The pulse width modulated D/A converter (PWM) converts digital values into variable duty-cycle square waves which can be filtered for applications needing analog control.

13.1 Features

The PWM provides the following features:

- Staggered Channel Switching for Noise Reduction
- Repetition Rate of 6250 Hz
- Eight Open-Drain Channel Outputs
- Six-Bit PWM Resolution

13.2 Overview

The pulse width modulated D/A converter (PWM) system works in conjunction with the multifunction timer to execute eight 6-bit PWM D/A conversions.

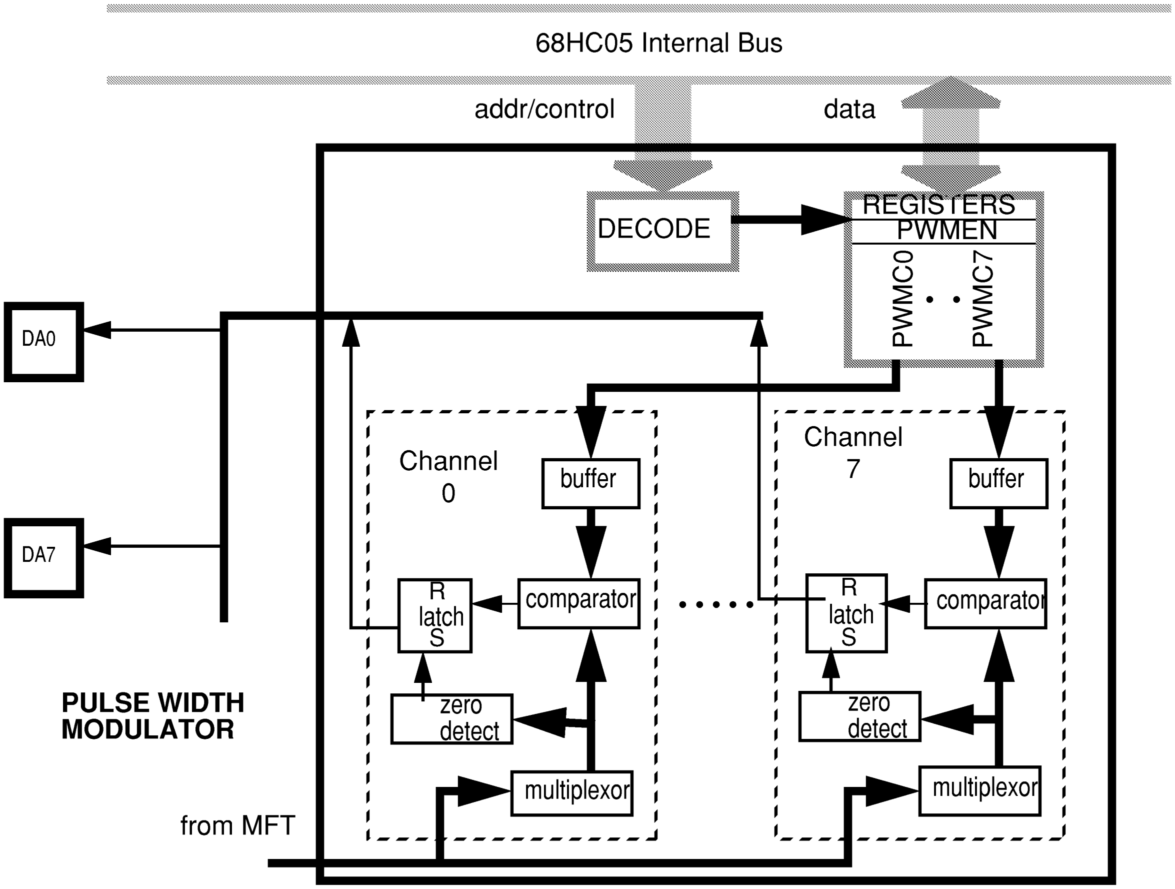

68HC05 Internal Bus
addr/control
data
DECODE
REGISTERS
PWMEN
PWMC0
PWMC7
DA0
DA7
Channel 0
Channel 7
buffer
comparator
R
latch
S
zero detect
multiplexor
PULSE WIDTH MODULATOR
from MFT

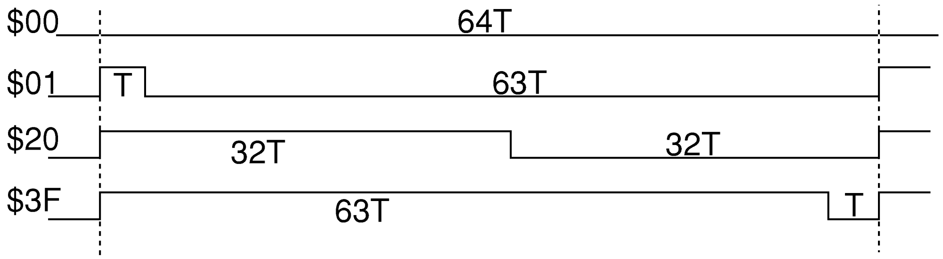
Figure 13-1. PWM Block Diagram

Each output is a pulse width modulated signal whose duty cycle varies according to the value set into its data register. The duty cycle is expressed with 6 bits of resolution. The signal can be used directly as a PWM, or it may be filtered to obtain an average value for general-purpose analog output.

Each output pin is shared with a general programmable port I/O bit on port B. The choice of function is expressed by the value of a DAC output control bit for each DAC channel. When the DAC output control bit is set to one, the pin functions as a PWM output; otherwise, the pin functions as an I/O bit of port B. All port B pins are open-drain.

The repetition rate is equal to the multifunction timer overflow rate (repetition rate for an 8.00-MHz crystal is 6250 Hz). When the MCU writes to register DAC X, the new value will be picked up by the PWM only at the end of a complete cycle of conversion. This results in a monotonic change of the DC component of the output without overshoots or vicious starts. (A vicious start is an output which gives totally erroneous PWM during the initial period following an update of the PWM data register). This feature is achieved by double buffering of the PWM data registers.

To reduce the peak current drawn by the MCU during the start of the 8 PWM channels, each channel is staggered from the one before it by 1 full bus cycle (250 ns for 8.00 MHz crystal).



1/64T = 6250 Hz, and T = 1/(6250Hz x 64) = 2.5 µs, so T = 10 CPU clocks

Figure 13-2. PWM Output Waveform Examples

13.3 Programming Guidelines

Programming guidelines are discussed in the following subsections.

13.3.1 Setup

To begin operation of a PWM channel, the pulse width must be defined in the corresponding channel register, and then the appropriate channel enable bit should be set.

13.4 Input/Output

The PWM shares eight outputs with port B.

13.4.1 DA0-DA7

These eight outputs share PB0 through PB7 with port B. All eight outputs are open-drain, both under PWM and port B control. When CHENn in the PWMEN is a logic zero, the corresponding pin becomes the PBn input/output and follows the port B DDR assignments. When CHENn is a logic one, this pin becomes the DAn PWM D/A output, regardless of the port B DDR assignments. Each of the eight outputs can be enabled separately.

13.5 Registers

The PWM has nine write-only registers.

NOTE

Do not use read/modify/write instructions to write to these registers because they are not readable.

13.5.1 PWM Channel Registers (PWMC0-PWMC7)

These write-only registers, PWMC0-PWMC7, each contain 6 bits of PWM control data. Figure 13-3 details each bit in these registers.

		7	6	5	4	3	2	1	0
PWMC0-7 $08-$0F	RD								
	WR			PW5	PW4	PW3	PW2	PW1	PW0
	RST			0	0	0	0	0	0

Figure 13-3. PWM Channel Registers (PWMC0-PWMC7)

13.5.1.1 PW5:0 - channel Pulse Width 5:0

These write-only bits define the pulse width of the corresponding PWM channel output. A value of $00 results in a continuous low output on the corresponding PWM output pin. A value of $20 results in a 50% duty cycle output, and so on, to the maximum value, $3F which corresponds to an output which is at logic level one for 63/64 of the cycle. A reset clears these bits.

13.5.2 PWM Enable Register (PWMEN)

The PWMEN contains 8 PWM control bits. Figure 13-4 details each bit in the register.

		7	6	5	4	3	2	1	0
PWMEN $10	RD								
	WR	CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0
	RST	0	0	0	0	0	0	0	0

Figure 13-4. PWM Enable Register (PWMEN)

13.5.2.1 CHEN7:0 - CHannel ENable 7:0

Each bit corresponds to a PWM channel, and determines whether that channel is enabled. When enabled, the PWM channel takes control of the corresponding port B pin; when disabled, control of the pin returns to port B. These bits are write-only. A reset clears these bits.

CHENn = 1: PWM channel n is enabled
CHENn = 0: PWM channel n is disabled

13.6 Low Power Modes

The following subsections describe low power modes.

13.6.1 Operation During WAIT Mode

The PWM remains active during WAIT mode. For lowest power consumption, it is recommended that the PWM registers be cleared before entering WAIT mode.

13.6.2 Operation During STOP Mode

The PWM is inactive during STOP mode.

13.7 Interrupts and Resets

The PWM does not interrupt or reset the MCU.

SECTION 14
A/D CONVERTER

The analog-to-digital converter (ADC) performs individual analog comparisons which can be used with a successive approximation software algorithm to obtain an analog-to-digital conversion.

14.1 Feature

The ADC provides the following feature:

- 5-bit D/A

14.2 Overview

The ADC system consists of a single 5-bit D/A converter and comparator with continuous conversion. A result flag indicates if the comparator output is above or below the analog input, ADCIN.

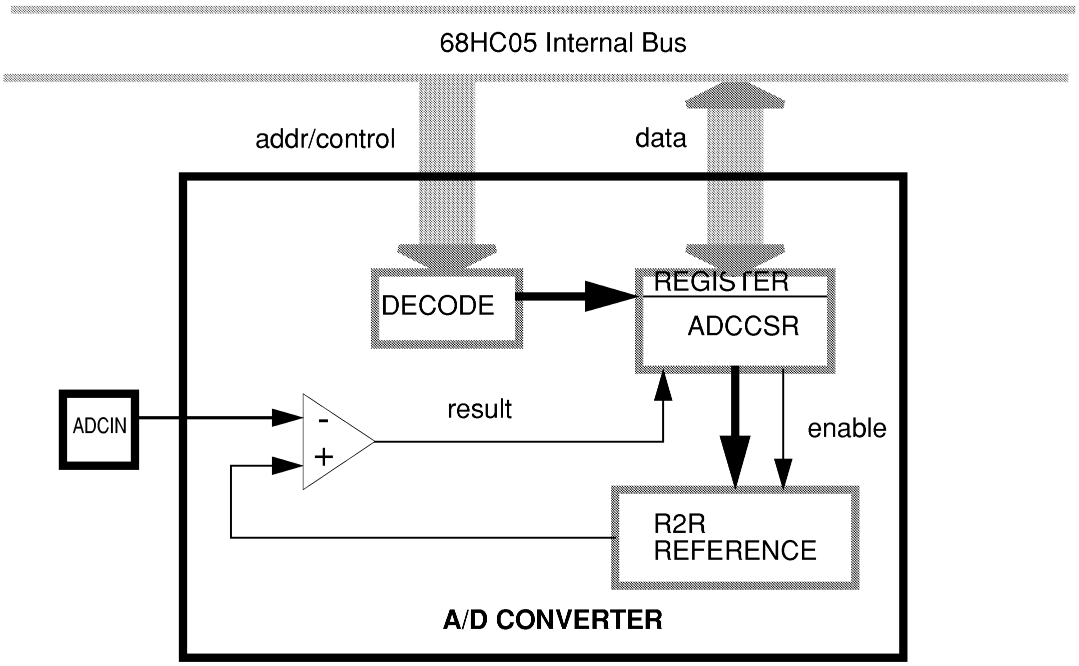

68HC05 Internal Bus
addr/control
data
DECODE
REGISTER
ADCCSR
ADCIN
-
+
result
enable
R2R
REFERENCE
A/D CONVERTER

Figure 14-1. ADC Block Diagram

14.3 Programming Guidelines

The following subsections describe programming guidelines.

14.3.1 Setup

The ADC must be enabled by setting the ADON bit in the ADCCSR.

14.3.2 Conversions

An A/D conversion is performed through a successive approximation algorithm. The following is a conversion example code segment:

```
        LDA     #$40
        STA     ADCCSR, ATD
        INC     ADCCSR
        LDA     ADCCSR
        AND     #$1F
        CMP     #$1F
        BLS     DTA

; out of range

ATD   ...          ; analog value in ADC.
                   ;ANALOG IN = (ADC +1)*0.15625V AS VDD = 5V
```

14.4 Input/Output

The ADC shares one pin with port C.

14.4.1 ADCIN

This analog input shares PC2 with port C. When ADON in the ADCCSR is a logic zero, this pin becomes the PC2 input/output and follows the port C DDR assignment. When ADON is a logic one, this pin becomes the ADCIN analog input, regardless of the port C DDR assignment. A digital read of PC2 when ADON is set results in a logic zero.

14.5 Registers

The ADC has one register.

14.5.1 ADC Control/Status Register (ADCCSR)

The ADCCSR contains all of the ADC status and control bits. Figure 14-2 details each bit in the register.

		7	6	5	4	3	2	1	0
ADCCSR $2B	RD	RESULT	ADON	0	AD4	AD3	AD2	AD1	AD0
	WR	▨		▨					
	RST	U	0	▨	0	0	0	0	0

Figure 14-2. ADC Control/Status Register (ADCCSR)

14.5.1.1 RESULT - comparator RESULT bit

This bit indicates the relationship of the analog input to the analog version of the AD4:0 value. A reset has no effect on this bit.
RESULT = 1: D/A output ≥ ANALOG IN
RESULT = 0: D/A output ≤ ANALOG IN

14.5.1.2 ADON - ADc ON

This bit indicates whether the ADC is enabled. When enabled, the ADC controls PC2 and forces it to be an input. Enabling the ADC also supplies power to the D/A resistive ladder. When disabled, PC2 returns to the control of port C. A reset clears this bit.
ADON = 1: ADC enabled
ADON = 0: ADC disabled

NOTE

If not in use, the input should be tied to V_{SS} and a value of $00 should be written to the ADCCSR.

14.5.1.3 AD4:0 - ADC comparison value 4:0

These bits are controlled by the user to perform a successive approximation conversion in software. When a value causes the RESULT bit to change state from the value immediately before or after it, AD4:0 are considered to be the digital equivalent of the analog input. A reset clears these bits.

14.6 Low Power Modes

Low power modes are discussed in the following subsections.

14.6.1 Operation During WAIT Mode

The ADC will function in WAIT mode, but to achieve lowest power consumption it is recommended that the ADC be disabled before entering WAIT mode.

14.6.2 Operation During STOP Mode

The ADC will function in STOP mode, but to achieve lowest power consumption it is recommended that the ADC be disabled before entering STOP mode.

14.7 Interrupts and Resets

The ADC does not interrupt or reset the MCU.

SECTION 15
ELECTRICAL SPECIFICATIONS

15.1 Maximum Ratings

(Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} - 0.3 to V_{DD} + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05CC1	T_A	T_L to T_H 0 to +70	°C
Storage Temperature Range	T_{stg}	0 to +125	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}).

15.2 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic	θ_{JA}	60	°C/W

15.3 DC Electrical Characteristics

V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA I_{Load} = -10.0 µA	V_{OL} V_{OH}	— V_{DD}-0.1	— —	0.1 —	V
Output High Voltage (I_{Load} = -0.8 mA) PA0-7, PB0-7, PC0-1, PC3-6, PD0-7	V_{OH}	V_{DD}-0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-7, PB0-7, PC0-1, PC3-6, PD0-7	V_{OL}	—	—	0.4	V
Input High Voltage Ports A, B, C, and D, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, VSYN, HSYN	V_{IH}	0.7×V_{DD}	—	V_{DD}	V
Input Low Voltage Ports A, B, C, and D, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, VSYN, HSYN	V_{IL}	V_{SS}	—	0.2×V_{DD}	V
Supply Current (see Notes) Run I_{DD} (PLL Running) Run I_{DD} (PLL Stopped) Wait Stop 25°C 0°C to +70°C	I_{DD} I_{DD} I_{DD} I_{DD} I_{DD}	— — — — —	12 5 0.7 1 7	35 24 4 100 100	mA mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-7, PB0-7, PC0-1, PC3-6, PD0-7	I_{OZ}	—	—	10	µA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, OSC1, PAMIN, PC2/AD0	I_{in}	—	—	1	µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, PAMIN, PC2/AD0	C_{out} C_{in}	— —	— —	12 8	pF pF
Hysteresis $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, SCK, PAMIN	V_{HYST}	—	1.0	—	V

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only
3. Wait I_{DD}: Only timer system active.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 8.4 MHz), all inputs 0.2V from rail; no dc loads, less than 50 pF on all outputs, C_L = 100 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD}-0.2 V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

15.4 Control Timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{osc} f_{osc}	 — dc	 8.4 8.4	 MHz MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{op} f_{op}	 — dc	 4.2 4.2	 MHz MHz
Cycle Time	t_{cyc}	240	—	ns
Crystal Oscillator Start-up Time	t_{OXOV}	—	100	ms
Stop Recovery Start-up Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	50	—	ns
A/D Comparator Stabilization Time	t_{settle}	—	10	us
A/D Comparator Conversion Time	t_{conv}	2	3	t_{cyc}

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

15.5 On-Screen Display Timing

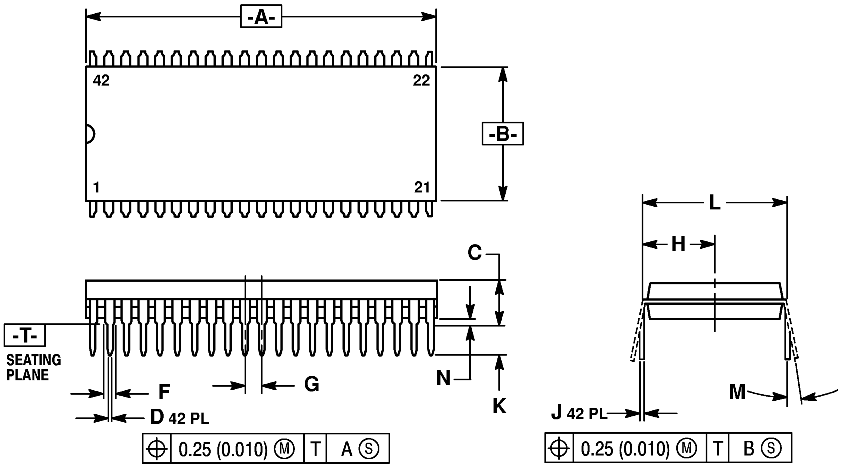
(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Rise Time (load = 2kΩ, 12 pF; .1 to .9 V_{DD} R, G, B, FBKG	t_R	—	5	—	ns
Fall Time (load = 2kΩ, 12 pF; .9 to .1 V_{DD} R, G, B, FBKG	t_F	—	5	—	ns
Dot Clock Frequency (master PLL)	f_{OSD}	—	28.2	28.2	MHz

SECTION 16
MECHANICAL SPECIFICATIONS

The MC68HC05CC2 is available in a 42-pin shrink dual-in-line (SDIP) package and a 40-pin dual-in-line (DIP) package. Package dimensions are provided in this section.

16.1 42-Pin SDIP Package (Case 858-01)

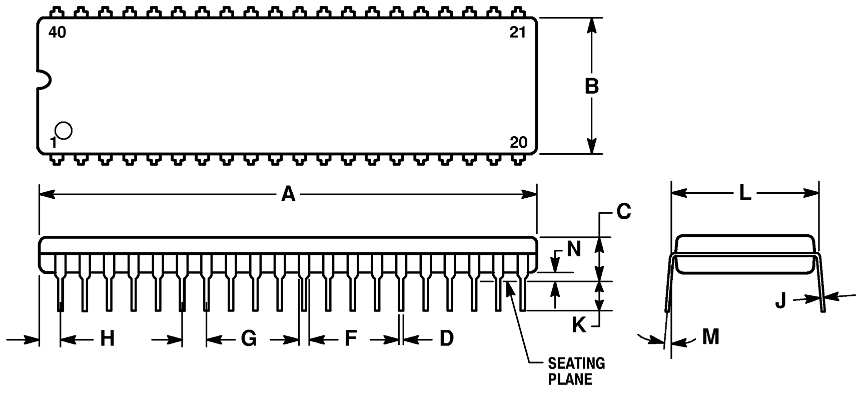


NOTES:

1. DIMENSIONS AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSIONS A AND B DO NOT INCLUDE MOLD FLASH. MAXIMUM MOLD FLASH 0.25 (0.010).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.435	1.465	36.45	37.21
B	0.540	0.560	13.72	14.22
C	0.155	0.200	3.94	5.08
D	0.014	0.022	0.36	0.56
F	0.032	0.046	0.81	1.17
G	0.070 BSC		1.778 BSC	
H	0.300 BSC		7.62 BSC	
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.600 BSC		15.24 BSC	
M	0°	15°	0°	15°
N	0.020	0.040	0.51	1.02

16.2 40-Pin DIP Package (Case 711-03)



NOTES:

1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

SECTION 17
ORDERING INFORMATION

This section contains instructions for ordering custom-masked ROM MCUs.

17.1 MCU Ordering Forms

To initiate an order for a ROM-based MCU, first obtain the current ordering form for the MCU from a Motorola representative. Submit the following items when ordering MCUs:

- A current MCU ordering form that is **completely filled out** (Contact your Motorola sales office for assistance.)
- A copy of the customer specification if the customer specification deviates from the Motorola specification for the MCU
- Customer's application program on one of the media listed in **17.2 Application Program Media**

The current MCU ordering form is also available through the Motorola Freeware Bulletin Board Service (BBS). The telephone number is (512) 891-FREE. After making the connection, type `bbs` in lower-case letters. Then press the return key to start the BBS software.

17.2 Application Program Media

Please deliver the application program to Motorola in one of the following media:

- Macintosh®[1] 3 1/2-inch diskette (double-sided 800K or double-sided high-density 1.4M)
- MS-DOS®[2] or PC-DOS™[3] 3 1/2-inch diskette (double-sided 720K or double-sided high-density 1.44M)
- MS-DOS® or PC-DOS™ 5 1/4-inch diskette (double-sided double-density 360K or double-sided high-density 1.2M)

Use positive logic for data and addresses.

When submitting the application program on a diskette, clearly label the diskette with the following information:

- Customer name
- Customer part number
- Project or product name
- File name of object code
- Date
- Name of operating system that formatted diskette
- Formatted capacity of diskette

On diskettes, the application program must be in Motorola's S-record format (S1 and S9 records), a character-based object file format generated by M6805 cross assemblers and linkers.

NOTE

Begin the application program at the first user ROM location. Program addresses must correspond exactly to the available on-chip user ROM addresses as shown in the memory map. **Write $00 in all non-user ROM locations or leave all non-user ROM locations blank**. Refer to the current MCU ordering form for additional requirements. Motorola may request pattern re-submission if non-user areas contain any non-zero code.

---

[1] Macintosh is a registered trademark of Apple Computer, Inc.
[2] MS-DOS is a registered trademark of Microsoft Corporation.
[3] PC-DOS is a trademark of International Business Machines Corporation.

If the memory map has two user ROM areas with the same address, then write the two areas in separate files on the diskette. Label the diskette with both file names.

In addition to the object code, a file containing the source code can be included. Motorola keeps this code confidential and uses it only to expedite ROM pattern generation in case of any difficulty with the object code. Label the diskette with the file name of the source code.

17.3 ROM Program Verification

The primary use for the on-chip ROM is to hold the customer's application program. The customer develops and debugs the application program and then submits the MCU order along with the application program.

Motorola inputs the customer's application program code into a computer program that generates a listing verify file. The listing verify file represents the memory map of the MCU. The listing verify file contains the user ROM code and may also contain non-user ROM code, such as self-check code. Motorola sends the customer a computer printout of the listing verify file along with a listing verify form.

To aid the customer in checking the listing verify file, Motorola will program the listing verify file into customer-supplied blank preformatted Macintosh or DOS disks. All original pattern media are filed for contractual purposes and are not returned.

Check the listing verify file thoroughly, then complete and sign the listing verify form and return the listing verify form to Motorola. The signed listing verify form constitutes the contractual agreement for the creation of the custom mask.

17.4 ROM Verification Units (RVUs)

After receiving the signed listing verify form, Motorola manufactures a custom photographic mask. The mask contains the customer's application program and is used to process silicon wafers. The application program cannot be changed after the manufacture of the mask begins. Motorola then produces 10 MCUs, called RVUs, and sends the RVUs to the customer. RVUs are usually packaged in unmarked ceramic and tested to 5 Vdc at room temperature. RVUs are not tested to environmental extremes because their sole purpose is to demonstrate that the customer's user ROM pattern was properly implemented. The 10 RVUs are free of charge with the minimum order quantity. These units are not to be used for qualification or production. RVUs are not guaranteed by Motorola Quality Assurance.

17.5 MC Order Numbers

Table 17-1 shows the MC order numbers for the available package types.

Table 17-1. MC Order Numbers

MC Order Number	Operating Temperature Range
MC68HC05CC2B	0° to 70°C
MC68HC05CC2P	0° to 70°C

NOTE: B = 42-pin shrink dual-in-line (SDIP) package
P = 40-pin dual-in-line (DIP) package

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

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

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

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