M68HC05EVM/AD4

M68HC05EVM

Evaluation Module User's Manual

MOTOROLA

May 1992

ADDENDUM

TO

M68HC05EVM

EVALUATION MODULE

USER'S MANUAL

M68HC05EVM/AD4

This addendum corrects text (via change pages) contained in the M68HC05EVM/AD4 user's manual. Attached to this addendum are two change pages (2-5 and 2-6). These pages are to be inserted into your manual. Make certain that the pages you are replacing are removed from your manual.

Information contained in this document applies to REVision (D) M68HC05EVM evaluation modules.

After the change pages are inserted into the user's manual, this page of the addendum should be placed after the user's manual title page and saved as a record of the changes made to the manual. Pages affected by this addendum are:

2-5
2-6

M68HC05EVM/AD4

September 1990

M68HC05EVM

EVALUATION MODULE

USER'S MANUAL

Information contained in this document applies to REVision (D) M68HC05EVM evaluation modules, serial numbers 1000 through 9999.

Fourth Edition

Third Edition February 1989

PREFACE

The user is responsible for installing the MC68HC05L6 MCU surface mount socket (U54) and I/O port connectors (J12 and J24). These components are not supplied with the EVM.

Unless otherwise specified, all address references are in hexadecimal throughout this manual.

An asterisk (*) following the signal name denotes that the signal is true or valid when the signal is low.

TABLE OF CONTENTS

PAGE

CHAPTER 3 OPERATING INSTRUCTIONS

TABLE OF CONTENTS (cont'd)

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

This manual provides general information, hardware preparation, installation instructions, operating instructions, functional description, and support information for the M68HC05EVM Evaluation Module (hereafter referred to as EVM). Appendix A contains EVM downloading S-record information.

1.2 FEATURES

EVM features include:

Economical means of evaluating target systems incorporating M68HC05 HCMOS MCU family devices.

Monitor/debugger firmware

One-line assembler/disassembler

Host computer download capability

Dual 8K-byte memory maps:

8K monitor EPROM
8K/16K User Pseudo ROM

MC68HC805B6 EEPROM MCU programmer

MC68HC705C8 OTPROM/EPROM MCU programmer

MC68HC805C4 EEPROM MCU programmer

28-pin MCU extension I/O port for MC68HC05P1/P7 evaluation

40-pin MCU extension I/O port for MC68HC05A6, HC05C2/C3/C4/C8/C9, HC705C8, HC805C4, HCL05C4/C8, and HSC05C4/C8 evaluation

52-pin MCU extension I/O port for MC68HC05B4/B6 and HC805B6 evaluation

68-pin MCU extension I/O port for MC68HC05L6 evaluation

RS-232C terminal and host computer I/O ports

1.3 SPECIFICATIONS

Table 1-1 lists the EVM specifications.

TABLE 1-1. EVM Specifications

CHARACTERISTICS	SPECIFICATIONS
Internal Clock	1 or 2 MHz bus operation (8 MHz, crystal controlled, divided-by-two, or four) 2 or 4 MHz bus operation (16 MHz, crystal controlled, divided-by-two, or four) for optional HSC operation
External Clock	Up to 8 MHz (2 MHz maximum bus operation) Up to 16 MHz (4 MHz maximum bus operation for optional HSC operation)
Memory size:	
Monitor EPROM	8K bytes
Pseudo ROM	8K/16K bytes
MCU extension I/O ports:	HCMOS compatible
Terminal/host I/O ports	RS-232 compatible
Temperature:	
Operating	+25 degrees C
Storage	-40 to +85 degrees C
Relative humidity	0 to 90% (non-condensing)

TABLE 1-1. EVM Specifications (cont'd)

CHARACTERISTICS	SPECIFICATIONS
Power requirements:	
Module	+5 Vdc @ 1.0 A (max)
	+12 Vdc @ 0.1 A (max)
	-12 Vdc @ 0.1 A (max)
Programmer (VPP):	
MC68HC705C8:	
(Mask 0B67H)	+14.00 Vdc +/- 0.10 Vdc @ 10 mA (max)
(Mask 1B67H)	+14.75 Vdc +/- 0.25 Vdc @ 10 mA (max)
(Mask B44S)	+14.75 Vdc +/- 0.25 Vdc @ 10 mA (max)
(Future Masks)	+14.75 Vdc +/- 0.25 Vdc @ 10 mA (max)
MC68HC805B6:	
(BULK erase)	+19 Vdc @ 2 mA (max)
(Program)	+19 Vdc @ 100 uA (max)
MC68HC805C4	+19 Vdc @ 100 uA (max)
Dimensions:	
Width	12.0 in. (30.5 cm)
Length	9.25 in. (23.5 cm)

1.4 GENERAL DESCRIPTION

The EVM provides a tool for designing, debugging, and evaluating MC68HC05A6, HC05B4/B6, HC05C2/C3/C4/C8/C9, HC05L6, HC05P1/P7, HC705C8, HC805B6, and HC805C4 Microcomputer Unit (MCU) based target system equipment. By providing all of the essential MCU timing and I/O circuitry, the EVM simplifies user evaluation of the prototype hardware/software product. The EVM requires a user supplied power supply and an RS-232C compatible terminal for operation.

NOTE

The EVM cannot emulate the low power capabilities of the MC68HCL05C4/C8 MCU devices. The EVM can emulate the high speed (4 MHz) capabilities of the MC68HSC05C4/C8 MCU devices.

The M68HC05 Family of HCMOS MCU devices are evaluated (emulated) by the EVM resident MC68HC05xx MCU. Several types of resident MCUs can be used. The EVM is shipped with an MC68HC05C9 resident MCU device. This device will evaluate MC68HC05C2/C3/C4/C8/C9, HC805C4, HCL05C4/C8, HSC05C4/C8, and HC05P1/P7 MCUs. For other MC68HC05 MCU devices (e.g., MC68HC05A6, HC705C8, HC05B4/B6, HC805B6, HC805C4, and HC05L6), the resident MC68HC05C9 MCU is replaced by the specific MCU device required for evaluation.

Entering data, program debugging, and EPROM MCU programming is accomplished by the monitor ROM firmware via an external RS-232C compatible terminal connected to the EVM terminal port connector. A fixed 9600 baud rate is provided for the terminal port, and a software selectable 300-19.2K baud rate selection is provided for the host port.

Downloading programs (via Motorola S records) directly from an RS-232C compatible host computer to the EVM is accomplished via either the host or terminal port connector. Downloading is accomplished by the use of the monitor commands.

The EVM simulates the single-chip mode of operation. 28-pin MCU I/O port connector facilitates interconnection of the EVM to the target system for MC68HC05P1/P7 evaluation purposes. 40-pin MCU I/O port connector facilitates interconnection of the EVM to the target system for MC68HC05A6, HC05C2/C3/C4/C8/C9, HC705C8, and HC805C4 MCU evaluation purposes. 60-pin MCU I/O port connector facilitates interconnection of the EVM to the target system for MC68HC05B4/B6 and HC805B6 MCU evaluation purposes. Two 34-pin MCU I/O port connectors A and B (68-pins) facilitate interconnection of the EVM to the target system for MC68HC05L6 MCU evaluation purposes.

MCU code may be generated using the resident one-line assembler/disassembler, or may be downloaded to the user program RAM (pseudo ROM) through the host or terminal port connectors. User code may then be executed using various debugging commands in the monitor. User code may also be executed using the user reset switch. MCU device ROM is simulated by write protecting user program RAM during program execution.

IRQ level/edge sensitivity selection is software programmable. Jumper selectable options such as clock frequency, clock input, and 8K/16K map selection are provided on the EVM, as well as an OTPROM/EPROM/EEPROM MCU programmer. The MCU programmer, under monitor firmware control, allows the user to check, erase, program, verify, and copy the contents of either a 40-pin Dual-In-line Package (DIP) or a 44-/52-lead Plastic Leaded Chip Carrier (PLCC) MCU devices. Switches allow user control of the reset and abort functions, and OTPROM/EPROM/EEPROM MCU programming functions.

1.5 EQUIPMENT REQUIRED

Table 1-2 lists the external equipment requirements for EVM operation.

TABLE 1-2. External Equipment Requirements

EXTERNAL EQUIPMENT
+5, +12, -12 Vdc power supply*
VPP power supply*
Terminal (RS-232C compatible)
Host computer (RS-232C compatible)**
Terminal/host computer - EVM RS-232C cable assembly*
Target system - EVM emulator cable assemblies***

Notes:

(1) * Refer to Chapter 2 for details.

(2) ** Optional - not required for basic operation.

(3)*** DIP cable assemblies fabricated by user.

PLCC cable assembly supplied with EVM.

CHAPTER 2

HARDWARE PREPARATION AND INSTALLATION INSTRUCTIONS

2.1 INTRODUCTION

This chapter provides unpacking instructions, hardware preparation, component implementation, and installation instructions for the EVM.

2.2 UNPACKING INSTRUCTIONS

NOTE

If shipping carton is damaged upon receipt, request carrier's agent be present during unpacking and inspection of the EVM.

Unpack EVM from shipping carton. Refer to packing list and verify that all items are present. Save packing material for storing or reshipping the EVM.

2.3 HARDWARE PREPARATION

This portion of text describes the inspection/preparation of EVM components prior to target system installation. This description will ensure the user that the EVM components are properly configured for target system operation. The EVM has been factory-tested and is shipped with factory-installed jumpers.

The EVM should be inspected/prepared for jumper placements prior to target system installation. Figure 2-1 illustrates the EVM connector, switch, and jumper header locations.

Connectors J1, J15, J17 through J20, J26, J28, and J29 facilitate interconnection of external equipment to the EVM. Switches S1 through S5 provide user control of the EVM. Refer to Chapter 3 for switch descriptions.

Jumper header locations J5, J16, J21, J22, and J24 provide mode selection capabilities as follows:

a. Mode select (J22 and J24)

b. Normal/test select (J5) ***Factory use only.***

c. MC68HC705C8/MC68HC805C4 program select (J16)

d. Evaluation/program select (J21)

Jumper header locations J7 through J14 provide specific I/O port configuration capabilities as follows:

a. PC2/E select (J7) ***MC68HC05B4/B6 operation only.***

b. PB select (J9 thru J11) ***MC68HC05P1 operation only.***

c. PD select (J12 thru J14) ***MC68HC05P7 operation only.***

d. TCAP' select (J8) ***MC68HC05P1 operation only.***

Jumper header locations J2 through J4, J6, J23, J25, and J27 provide general selection capabilities that are applicable for all modes of operation as follows:

a. Host port CTS select (J2)

b. Clock frequency select (J6)

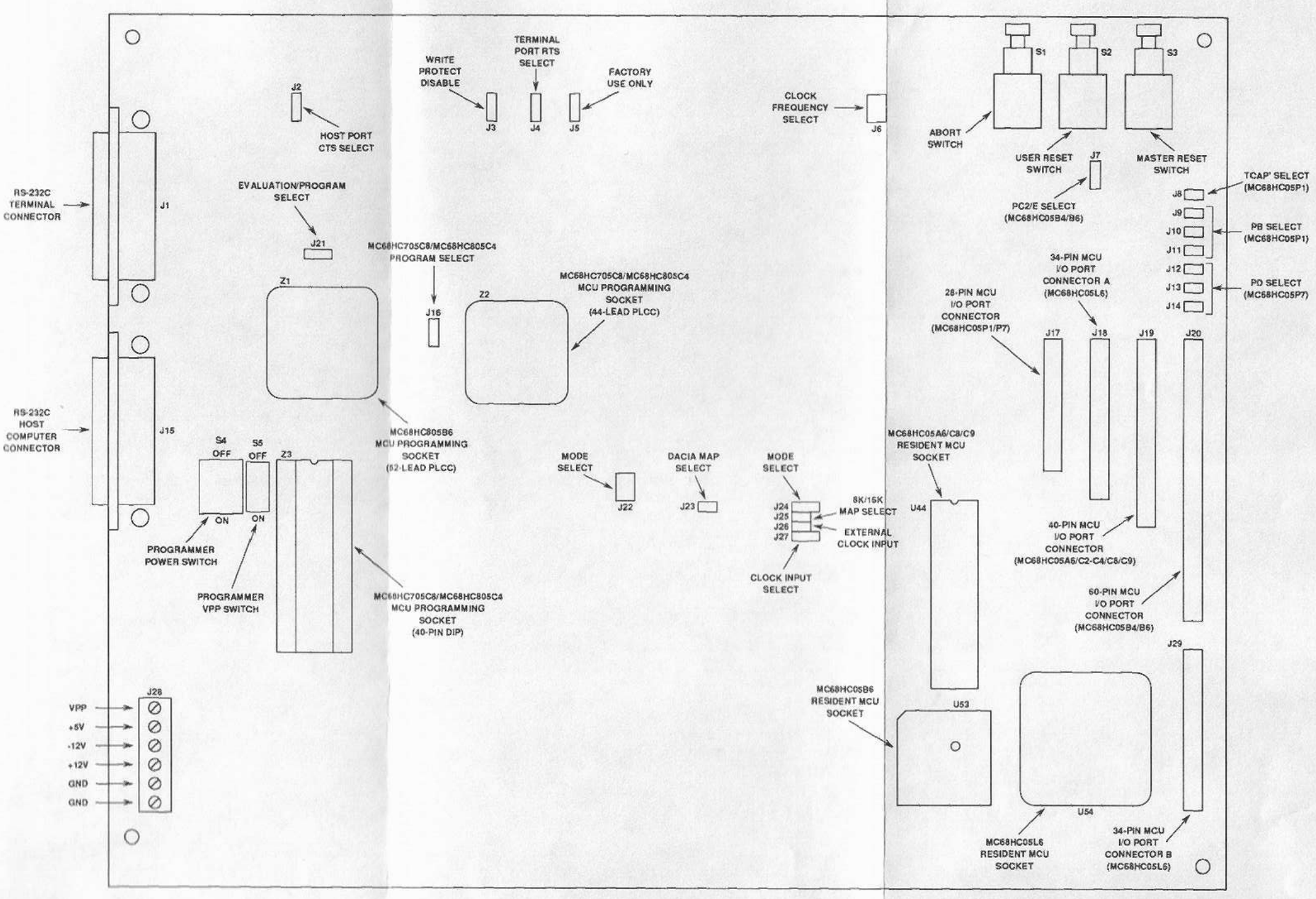
c. Terminal port RTS select (J4)

d. DACIA map select (J23)

e. Clock input select (J27)

f. 8K/16K map select (J25)

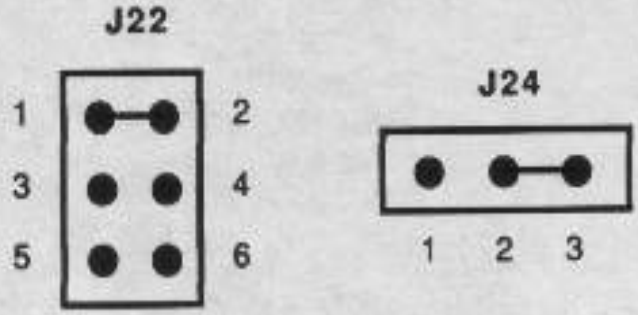
g. Write protect disable (J3)


WRITE PROTECT DISABLE
TERMINAL PORT RTS SELECT
FACTORY USE ONLY
J2
HOST PORT CTS SELECT
J3
J4
J5
CLOCK FREQUENCY SELECT
J6
S1
S2
S3
ABORT SWITCH
USER RESET SWITCH
MASTER RESET SWITCH
J7
PC2/E SELECT (MC68HC05B4/B6)
TCAP' SELECT (MC68HC05P1)
J8
J9
J10
J11
PB SELECT (MC68HC05P1)
J12
J13
J14
PD SELECT (MC68HC05P7)
RS-232C TERMINAL CONNECTOR
J1
EVALUATION/PROGRAM SELECT
J21
MC68HC705C8/MC68HC805C4 PROGRAM SELECT
J16
Z1
Z2
MC68HC705C8/MC68HC805C4 MCU PROGRAMMING SOCKET (44-LEAD PLCC)
34-PIN MCU I/O PORT CONNECTOR A (MC68HC05L6)
28-PIN MCU I/O PORT CONNECTOR (MC68HC05P1/P7)
J17
J18
J19
J20
RS-232C HOST COMPUTER CONNECTOR
J15
S4 OFF
S5 OFF
ON
ON
Z3
MC68HC805B6 MCU PROGRAMMING SOCKET (52-LEAD PLCC)
MODE SELECT
J22
DACIA MAP SELECT
J23
MODE SELECT
J24
J25
J26
J27
8K/16K MAP SELECT
EXTERNAL CLOCK INPUT
CLOCK INPUT SELECT
MC68HC05A6/C8/C9 RESIDENT MCU SOCKET
U44
PROGRAMMER POWER SWITCH
PROGRAMMER VPP SWITCH
MC68HC705C8/MC68HC805C4 MCU PROGRAMMING SOCKET (40-PIN DIP)
40-PIN MCU I/O PORT CONNECTOR (MC68HC05A6/C2-C4/C8/C9)
60-PIN MCU I/O PORT CONNECTOR (MC68HC05B4/B6)
J29
J28
VPP
+5V
-12V
+12V
GND
GND
MC68HC05B6 RESIDENT MCU SOCKET
U53
U54
MC68HC05L6 RESIDENT MCU SOCKET
34-PIN MCU I/O PORT CONNECTOR B (MC68HC05L6)

REV D

2.3.1 Mode Select Headers (J22 and J24)

Configure jumper headers J22 and J24 for EVM evaluation of a specific MCU operation. J22 and J24 are factory configured for evaluating the MC68HC05C4 MCU (shown below). MCU device limitations must be considered when configuring the EVM for a specific MCU operation (refer to page 2-18).

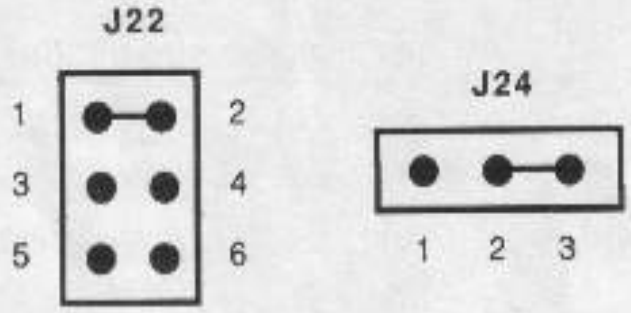


Jumper header J22 and J24 basic configurations for the M68HC05 family of devices are provided in the table below.

RESIDENT MCU	EVALUATION MCU(1)	J22	J24
MC68HC05A6	MC68HC05A6 w/ Internal EEPROM (2)	5-6	2-3
	MC68HC05A6 w/ Pseudo EPROM	3-5	2-3
MC68HC05B6	MC68HC05B6/805B6 w/ Internal EEPROM	2-4	2-3
	MC68HC05B6/805B6 w/ Pseudo EPROM	1-2	2-3
MC68HC05B4	MC68HC05B4, MC68HC05B6, MC68HC805B6	1-2	2-3

2.3.1 Mode Select Headers (J22 and J24)

Jumper headers J22 and J24 are use to configure the EVM for a specific microcontroller unit (MCU) evaluation operation. The EVM is factory configured and shipped for MC68HC05C4 evaluation operation as shown below.



Jumper header J22 and J24 basic configurations for the M68HC05 family of devices are provided in the table shown below.

BASIC CONFIGURATIONS	JUMPER HEADER	
MODE OF OPERATION	J22	J24
MC68HC05A6	5-6	3-5
MC68HC05B4/B6/805B6	2-4	3-5
MC68HC05C2-C4/C8	1-2	1-2
MC68HC05C9*	1-2	3-5
MC68HC705C8	3-5	1-2
MC68HC805C4	3-5	3-5
MC68HC05L6	1-2	1-2
MC68HC05P1/P7	1-2	1-2
MC68HCL05C4/C8	1-2	1-2
MC68HSC05C4/C8	1-2	1-2

NOTE: * Denotes 16K memory map configured via J25 for C9 configuration. All other configurations require 8K memory map (see para. 2.3.14).

Specific MCU device limitations (refer to page 2-18) must be considered when configuring the EVM for a specific MCU operation. The following page describes specific configurations for specific device evaluation operations.

RESIDENT MCU	EVALUATION MCU	J22	J24
MC68HC05C9	MC68HC05A6, MC68HC05C2, MC68HC05C3, MC68HC05C4, MC68HC05C8, MC68HC05L6 MC68HC05P1, MC68HC05P7, MC68HCL05C4, MC68HCL05C8, MC68HSC05C4, MC68HSC05C8	1-2	2-3
MC68HC05C9(3)	MC68HC05C9 w/ 16K memory map(4)	1-2	2-3
MC68HC705C8, or MC68HC805C4	MC68HC05A6, MC68HC05C2, MC68HC05C3, MC68HC05C4, MC68HC05C8, MC68HC05L6, MC68HC05P1, MC68HC05P7, MC68HCL05C4, MC68HCL05C8, MC68HSC05C4, MC68HSC05C8	3-5	2-3
MC68HC705C8(5), or MC68HC805C4(5)	MC68HC805C4 MC68HC705C8	3-5 3-5	2-3 1-2

1. Except where noted all emulation MCUs require at lest 8K for the memory map.
2. Option register is readable and writeable.
3. Write option register at $3FDF.
4. Install a jumper on jumper header J25 to enable the 16K memory map.
5. Write option register at $1FDF.

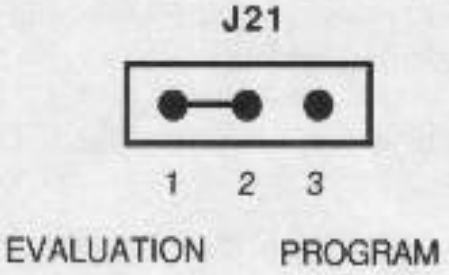
EVM Configurations for Specific Evaluation Operations

EVALUATION MODE	RESIDENT MCU	CONFIGURATION (J22)	(J24)	(J25)	NOTES	
MC68HC05A6	MC68HC05A6	5-6	2-3	Removed	Internal EEPROM	(1.)
MC68HC05A6		3-5	2-3	Removed	Pseudo EPROM	
MC68HC05B6	MC68HC05B6	1-2	2-3	Removed	Pseudo EPROM	
	or	2-4	2-3	Removed	Internal EEPROM	
	MC68HC805B6					
MC68HC05C2	MC68HC05C9	1-2	1-2	Removed	8K memory map	(2.)
MC68HC05C3		1-2	1-2	Removed	8K memory map	(2.)
MC68HC05C4		1-2	1-2	Removed	8K memory map	(2.)
MC68HC05C8		1-2	1-2	Removed	8K memory map	(2.)
MC68HC05P1		1-2	1-2	Removed	8K memory map	(2.)
MC68HC05P7		1-2	1-2	Removed	8K memory map	(2.)
MC68HC05A6		1-2	1-2	Removed	8K memory map	(2.)
MC68HC05C2	MC68HC705C8	3-5	2-3	Removed	8K memory map	(3.)
MC68HC05C3	or	3-5	2-3	Removed	8K memory map	(3.)
MC68HC05C4	MC68HC805C4	3-5	2-3	Removed	8K memory map	(3.)
MC68HC05C8		3-5	2-3	Removed	8K memory map	(3.)
MC68HC05P1		3-5	2-3	Removed	8K memory map	(3.)
MC68HC05P7		3-5	2-3	Removed	8K memory map	(3.)
MC68HC05A6		3-5	2-3	Removed	8K memory map	(3.)
MC68HC05C9	MC68HC05C9	1-2	2-3	Installed	16K memory map	
MC68HC705C8	MC68HC705C8	3-5	1-2	Removed	8K memory map	
MC68HC805C4	MC68HC805C4	3-5	2-3	Removed	8K memory map	
MC68HC05B4	MC68HC05B4	1-2	2-3	Removed	8K memory map	
	MC68HC05B6	1-2	2-3	Removed	8K memory map	
	MC68HC805B6	1-2	2-3	Removed	8K memory map	

NOTES:
1. Option register readable and writeable.
2. Write option register at $3FDF.
3. Write option register at $1FDF.

2.3.4 Evaluation/Program Select Header (J21)

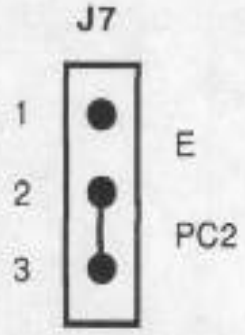
Jumper header J21 is use to configure the EVM for either the evaluation mode of operation or the programming mode of operation. The EVM is factory configured and shipped for evaluation mode of operation as shown below.



To select the programming mode of operation, reinstall J21 fabricated jumper between pins 2 and 3.

2.3.5 PC2/E Select Header (J7) *(MC68HC05B4/B6/805B6 operation only.)*

Jumper header J7 is used only for MC68HC05B4/B6/805B6 MCU evaluation. Jumper header J7 is used to select either PC2 for normal I/O operations, or E for the CPU E-clock out option. The EVM is factory configured and shipped for normal I/O evaluation operation as shown below.

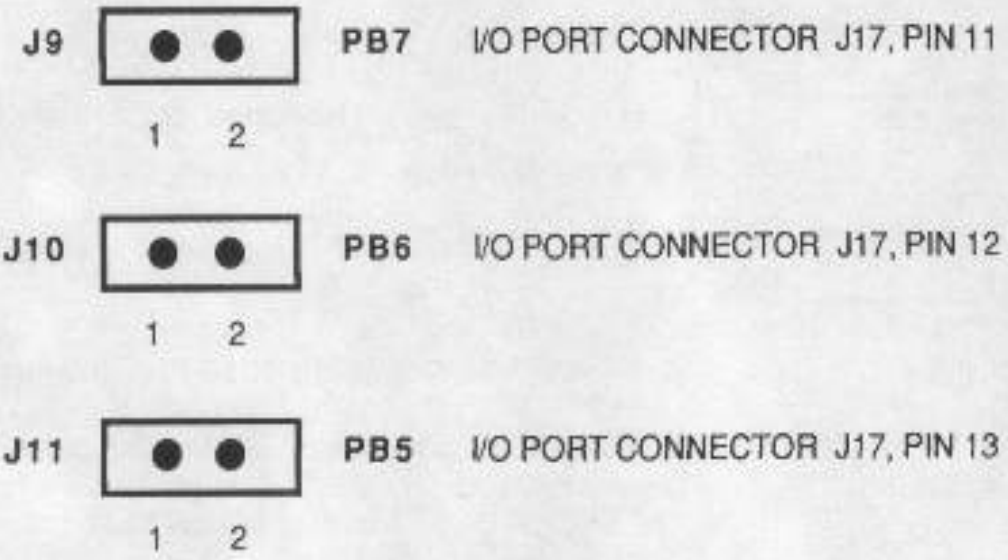


To select the CPU E-clock out option, reinstall J7 fabricated jumper between pins 1 and 2.

2.3.6 PB Select Headers (9 thru J11) *(MC68HC05P1 operation only.)*

Jumper headers J9 through J11 are used only for MC68HC05P1 MCU evaluation. Jumper headers J9 through J11 are used to select PB5 through PB7 signal lines for MC68HC05P1 I/O port operations. When the fabricated jumpers are installed, PB5 through PB7 signal lines are routed to the 28-pin I/O port connector J17 pins 11 through 13, respectively. The EVM is factory configured and shipped with jumpers not installed as shown below.

Prior to installing fabricated jumpers on jumper headers J9 through J11, ensure that fabricated jumpers are removed from jumper headers J12 through J14. Conversely, prior to installing fabricated jumpers on jumper headers J12 through J14, ensure that fabricated jumpers are removed from jumper headers J9 through J11.



CAUTION

MCU damage may occur if jumpers are installed on both PB and PD select headers.

If fabricated jumpers are installed on both the PB (J9 thru J11) and PD (J12 thru J14) select headers simultaneously, erratic MCU I/O port and/or EVM operation will result. To avoid erratic port and/or EVM operation, only one group of select headers (either PB or PD) must be jumpered for a specific port operation (MC68HC05P1 or MC68HC05P7).

2.3.7 PD Select Headers (J12 thru J14) *(MC68HC05P7 operation only.)*

Jumper headers J12 through J14 are used only for MC68HC05P7 MCU evaluation. Jumper headers J12 through J14 are used to select PD2 through PD4 signal lines for MC68HC05P7 I/O port operations. When the fabricated jumpers are installed, PD2 through PD4 signal lines are routed to the 28-pin I/O port connector J17 pins 11 through 13, respectively. The EVM is factory configured and shipped with jumpers not installed as shown below.

J12 **PD4*** **(SCK)** I/O PORT CONNECTOR J17, PIN 11

1 2

J13 **PD3*** **(SDI)** I/O PORT CONNECTOR J17, PIN 12

1 2

J14 **PD2*** **(SDO)** I/O PORT CONNECTOR J17, PIN 13

1 2

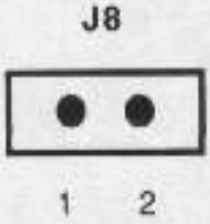
NOTE

Above asterisks (*) denote if parallel I/O is required instead of SPI, J9 thru J11 jumpers must be installed instead of J12 thru J14.

Prior to installing fabricated jumpers on jumper headers J12 through J14, ensure that fabricated jumpers are removed from jumper headers J9 through J11. Conversely, prior to installing fabricated jumpers on jumper headers J9 through J11, ensure that fabricated jumpers are removed from jumper headers J12 through J14.

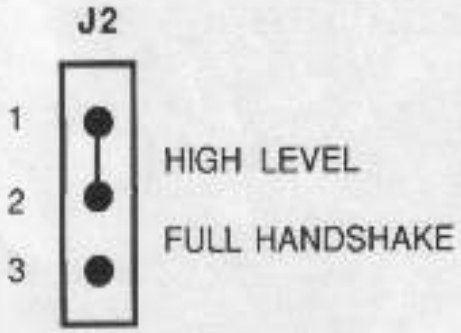
2.3.8 TCAP' Select Header (J8) *(MC68HC05P1 operation only.)*

Jumper header J8 is used only for MC68HC05P1 MCU evaluation. Jumper header J8 is used to generate the TCAP' signal for the resident MCU. When the fabricated jumper is installed, TCAP' signal is generated via the PD7 signal line. The EVM is factory configured and shipped with jumpers not installed as shown below.



2.3.9 Host Port CTS Select Header (J2)

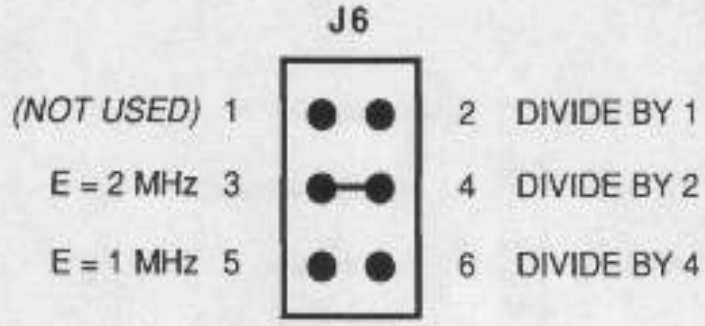
Jumper header J2 is used to configure the RS-232C host computer I/O port for semi- or full-handshaking operation. The EVM is factory-configured for semi-handshaking (high level) operation. This semi-handshaking configuration (high level) is accomplished by the installation of a fabricated jumper on jumper header J2 as shown below.



Should the host computer or modem require full-handshake capability, the jumper is repositioned between pins 2 and 3. Refer to the schematic diagram located in Chapter 5 (Figure 5-2, sheet 11 of 11) for CTS signal wiring information.

2.3.10 Clock Frequency Select Header (J6)

Jumper header J6 is used to select the internal EVM clock frequency. The EVM is shipped with an internal 8 MHz crystal, and factory configured for 4 MHz clock (E = 2 MHz) operation as shown below.



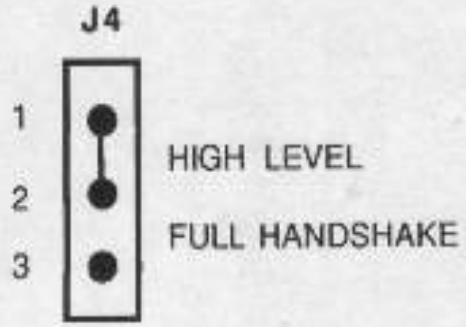
The divide by one option (pins 1 and 2) is not used. Do not attempt to operate the EVM with low speed devices (e.g., A6, B6, L6, etc.) above 4 MHz clock (E = 2 MHz). Improper EVM operation (device evaluation) will result.

NOTE

For HSC operation, the user is required to replace the 8 MHz crystal (Y1) with a 16 MHz crystal. When the 16 MHz crystal is installed, the divide by 2 and divide by 4 options become E = 4 MHz and E = 2 MHz, respectively.

2.3.11 Terminal Port RTS Select Header (J4)

Jumper header J4 is used to configure the RS-232C terminal I/O port for semi- or full-handshaking operation. The EVM is factory-configured for semi-handshaking (high level) operation. This semi-handshaking configuration (high level) is accomplished by the installation of a fabricated jumper on jumper header J4 as shown below.



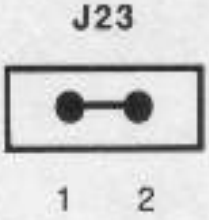
Should the host computer or modem require full-handshake capability, the jumper is repositioned between pins 2 and 3. Refer to the schematic diagram located in Chapter 5 (Figure 5-2, sheet 11 of 11) for RTS signal wiring information.

2.3.12 DACIA Map Select Header (J23)

NOTE

Jumper header J23 is for factory use only.

Jumper header J23 is for factory use only. The installed fabricated jumper (shown below) should not be removed during normal EVM operations.



2.3.13 Clock Input Select Header (J27)

Jumper header J27 is used to select either internal or external clock source to be used by the EVM. The internal clock source is an 8 MHz crystal. The EVM is factory-configured and shipped with the clock input selected to internal clock operation as shown below. To select an external clock source, reinstall jumper between pins 1 and 2.

J27

1 2 3

EXTERNAL INTERNAL

When the external clock source is selected, connector J26 (2-pin header) is used to facilitate interconnection of the external clock source.

J26

1 2

INPUT GND

2.3.14 8K/16K Map Select Header (J25)

Jumper header J25 is used to select either the 16K memory map which is required for MC68HC05C9 operations, or the 8K memory map which is applicable for all other operations. The EVM is factory-configured and shipped with the 16K memory map enabled. This is accomplished by the installation of a fabricated jumper on pins 1 and 2 as shown below. To enable the 8K memory map, remove jumper from pins 1 and 2.

J25

1 2

NOTE

Jumper is removed for all evaluation operations except when evaluating the MC68HC05C9 MCU.

2.3.15 Write Protect Disable Header (J3)

Jumper header J3 is used to write protect the user program space during program execution (pseudo ROM). The EVM is factory-configured and shipped with the write protect function enabled. This is accomplished by the installation of a fabricated jumper on pins 1 and 2 as shown below. To disable the write protect function, remove jumper from pins 1 and 2, and reinstall jumper on pins 2 and 3.

J3

WRITE PROTECT ENABLE

WRITE PROTECT DISABLE

2.4 COMPONENT IMPLEMENTATION

This portion of text describes the components required for MC68HC05L6 evaluation, specifies resident MCUs for specific target system evaluation, and illustrates proper resident MCU installation techniques. One time programmable read only memory (OTPROM), erasable programmable read only memory (EPROM), and electrically erasable programmable read only memory (EEPROM) MCU installation techniques for programming operations are also provided.

2

2.4.1 Resident MC68HC05L6 Evaluation Circuitry

As shown in Figure 2-1; the resident MC68HC05L6 MCU, MCU socket (U54), and MCU I/O port connectors (J18 and J29) are supplied and installed by the user. PCB feed-thru holes and conductive lands (traces) are provided on the EVM for the user to install the PLCC socket and two 34-pin I/O port connectors. A 68-lead PLCC surface mount socket (Plastronics # P2068SP) must be installed at socket location U54. Two 34-pin connectors (Aptronics # 929715-01-17) must be installed at connector locations J18 and J29. These connectors consist of two 17-pin headers, double row post. Care in socket and connector installation/soldering will save many hours of troubleshooting and repair after assembly.

2.4.2 Resident MCU Sockets

There are three resident MCU sockets (U44, U53, and U54) available on the EVM. Only one resident MCU may be installed on the EVM for a specific evaluation. Multiple resident MCUs will cause MCU damage and/or erratic EVM operation. On the following page is a list of recommended resident MCUs used for evaluation purposes.

When required, the resident 40-pin DIP MCU must be inserted into resident socket location U44. Inspect socket locations U53, U54, Z1, Z2, and Z3. Remove MCUs (if installed).

CAUTIONS

The resident M68HC05B6 MCU must be inserted right side up in socket U53. Incorrect MCU insertion will cause MCU damage.

The resident M68HC05L6 MCU must be inserted upside down in socket U54. Incorrect MCU insertion will cause MCU damage.

Forcing the resident MCU into the surface mount socket can result in damage to the socket. Surface mount sockets are fragile.

Only one resident MCU may be installed on the EVM for a specific evaluation. Multiple resident MCUs will cause MCU damage and/or erratic EVM operation.

Recommended Resident MCUs for Evaluation

RESIDENT MCU	SOCKET LOCATION	PACKAGE TYPE	MCU EVALUATION
MC68HC05A6	U44	40-PIN DIP	MC68HC05A6
MC68HC05C4 *(MC68HC05C9*)*	U44	40-PIN DIP	MC68HC05C2 MC68HC05C3 MC68HC05C4 MC68HC05C8 MC68HC05P1 MC68HC05P7 MC68HCL05C4 MC68HCL05C8 MC68HSC05C4 MC68HSC05C8
MC68HC05C9*	U44	40-PIN DIP	MC68HC05C9
MC68HC705C8	U44	40-PIN DIP	MC68HC705C8
MC68HC805C4	U44	40-PIN DIP	MC68HC805C4
MC68HC05B4	U53	52-LEAD PLCC	MC68HC05B4
MC68HC05B6	U53	52-LEAD PLCC	MC68HC05B4 MC68HC05B6
MC68HC805B6	U53	52-LEAD PLCC	MC68HC805B6
MC68HC05L6	U54	68-LEAD PLCC	MC68HC05L6

NOTES: (1) * Denotes EVM shipped with MC68HC05C9 device installed in resident MCU socket U44, and the 16K memory map configured via jumper header J25.

(2) * Also denotes MC68HC05C9 resident MCU can evaluate all device types specified for the MC68HC05C4 resident MCU. Jumper must be removed from jumper header J25 to configure 8K memory map operations.

(3) Operational limitations for specific MCUs are provided on the following page.

Resident MC68HC05C9 MCU Operational Limitations

The MC68HC05C9 MCU option register resides internally at location $3FDF. Therefore, when evaluating other HC05xx family devices with option registers residing at $1FDF, all accesses to the resident HC05C9 MCU option register must be made at $3FDF as opposed to $1FDF even though the 8K memory map is selected.

The HC05C9 MCU cannot be used to exactly replicate SPI operations for all HC05Cx MCU devices.

The HC05C9 MCU DDRD has a negative implication affecting serial peripheral interface (SPI) operations. All port D lines used as output lines must have corresponding DDR bits set to a logic 1 (i.e., SCK (master), MOSI (master), MISO (slave)) when evaluating all HC05Cx MCU devices.

The HC05C9 MCU contains a port D data direction register (DDRD) that can be used to emulate the MC68HC05P1 and MC68HC05P7 MCU port D, bit 5 (PD5). The user must be aware that the entire DDRD is implemented and other DDRD bits must not be accidently written to. When evaluating HC05P1 and HC05P7 devices with the resident HC05C9 MCU, the user must write $20 value to DDRD location $07 when configuring PDDR5 as an output line. This step will ensure that other port D pins remain as input lines as required for correct HC05P1 and HC05P7 evaluation.

Resident MC68HC705C8 MCU Operational Limitations

When evaluating with a resident MC68HC705C8 MCU the option register is accessed at location $1FDF. The HC705C8 option registers must be written to if edge-only interrupts are to be used. The HC705C8 option register contains RAM/ROM bits used for dynamic mapping of internal RAM/ROM.

The HC705C8 MCU can be used to exactly replicate SPI operations for all HC05Cx MCU devices.

Resident MC68HC805C4 MCU Operational Limitations

When evaluating with a resident MC68HC805C4 MCU the option register is accessed at location $1FDF. The HC805C4 option register must be written to if edge-only interrupts are to be used. The HC805C4 option register only configures IRQ options.

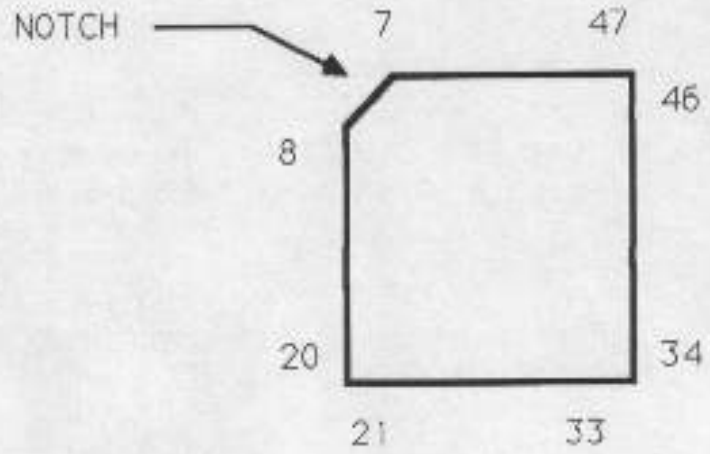
The HC805C4 MCU can be used to exactly replicate SPI operations for all HC05Cx MCU devices.

General Operational Limitations

The EVM cannot emulate the low power capabilities of the MC68HCL05C4/C8 MCU devices.

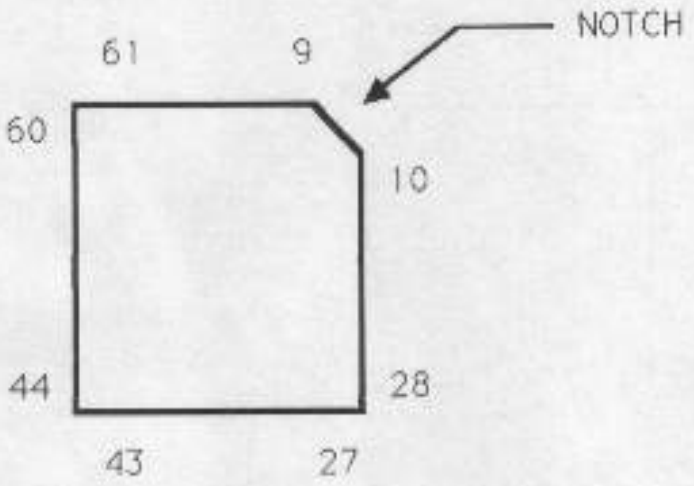
The EVM can emulate the high speed (4 MHz) capabilities of the MC68HSC05C4/C8 MCU devices by replacing the 8 MHz crystal (Y1) with a 16 MHz crystal. Refer to paragraph 2.3.10.

When required, the resident 52-lead PLCC MC68HC05B6 MCU is inserted right side up in resident socket location U53 as shown below. Inspect socket locations U44, U54, Z1, Z2, and Z3. Remove MCUs (if installed).



Resident 52-lead PLCC MC68HC05B6 MCU (U53) Installation

When required, the resident 68-lead PLCC MC68HC05L6 MCU is inserted upside down (horizontal flip) in resident socket location U54 as shown below. Inspect socket locations U44, U53, Z1, Z2, and Z3. Remove MCUs (if installed).



Resident 68-lead PLCC MC68HC05L6 MCU (U54) Installation

2.4.3 MCU Programming Sockets

There are three MCU programming sockets (Z1, Z2, and Z3) available on the EVM. As shown in Figure 2-1, the MC68HC805C4/MC68HC705C8 MCU programming socket Z3 consists of a 40-pin DIP socket, and Z2 consists of a 44-lead PLCC surface mount socket.

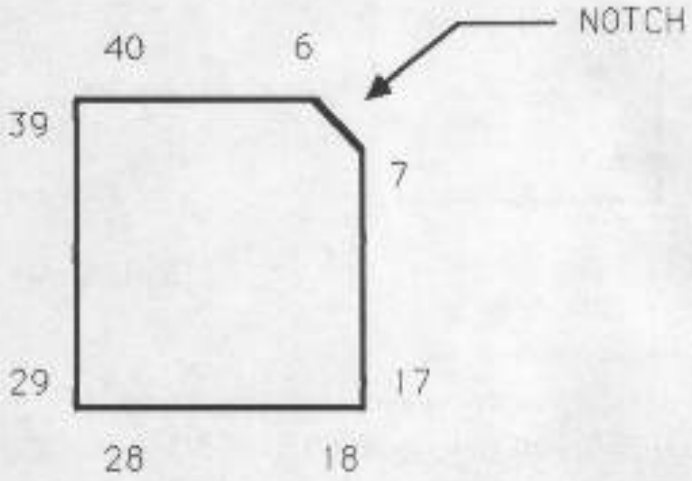
CAUTION

Forcing the OTPROM/EEPROM MCU into the surface mount socket can result in damage to the socket. Surface mount sockets are fragile.

Only one programmable MCU must be installed on the EVM for a specific programming operation. Multiple MCUs will cause MCU damage and/or erratic EVM operation.

When programming the 40-pin DIP MC68HC705C8 OTPROM, MC68HC705C8 EPROM, or MC68HC805C4 EEPROM MCU, the device is inserted into programming socket Z3. Inspect socket locations Z1, Z2, U53, and U54. Remove MCUs (if installed).

When programming the 44-lead PLCC MC68HC705C8 OTPROM or MC68HC805C4 EEPROM MCU, the device must be inserted upside down (horizontal flip) into programming socket Z2 as shown below. Inspect socket locations Z1, Z3, U53, and U54. Remove MCUs (if installed).



44-lead PLCC MC68HC705C8/MC68HC805C4 MCU (Z2) Installation

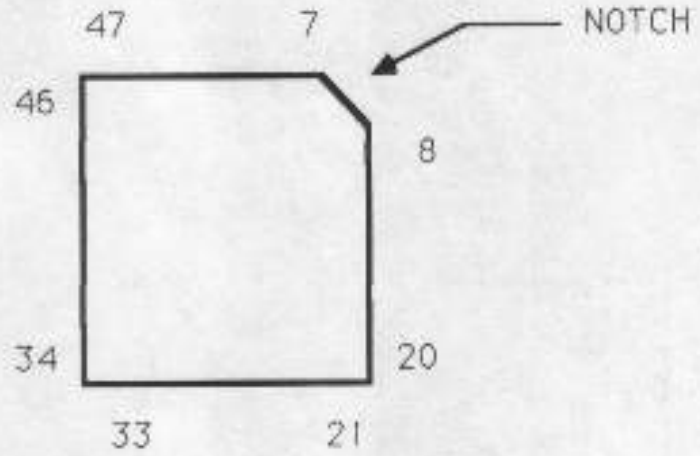
As shown in Figure 2-1, the MC68HC805B6 EEPROM MCU programming socket Z1 consists of a 52-lead PLCC surface mount socket.

CAUTION

Forcing the EEPROM MCU into the surface mount socket can result in damage to the socket. Surface mount sockets are fragile.

Only one MCU must be installed on the EVM for a specific programming operation. Multiple MCUs will cause MCU damage and/or erratic EVM operation.

When programming the 52-lead PLCC MC68HC805B6 EEPROM MCU, the device must be inserted upside down (horizontal flip) in programming socket Z1 as shown below. Inspect socket locations Z2, Z3, U44, and U54. Remove MCUs (if installed).



52-lead PLCC MC68HC805B6 MCU (Z1) Installation

2.5 INSTALLATION INSTRUCTIONS

The EVM is designed for table top operation. A user supplied power supply and RS-232C compatible terminal are required for EVM operation. An RS-232C compatible host computer may be connected to the EVM, but is not required for basic EVM operation.

2.5.1 Power Supply - EVM Interconnection

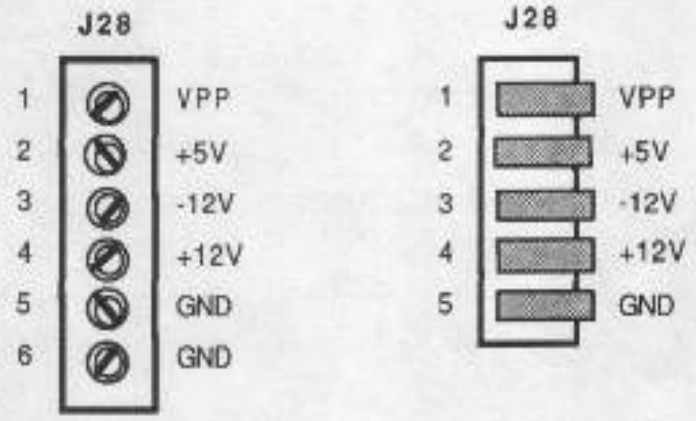
The EVM requires +5 Vdc @ 1.0 A, +12 Vdc @ 0.1 A, and -12 Vdc @ 0.1 A for operation.

Appropriate programming voltage (VPP) is determined by the data sheet of the MCU device to be programmed. Refer to page 3-34 for additional VPP information.

CAUTION

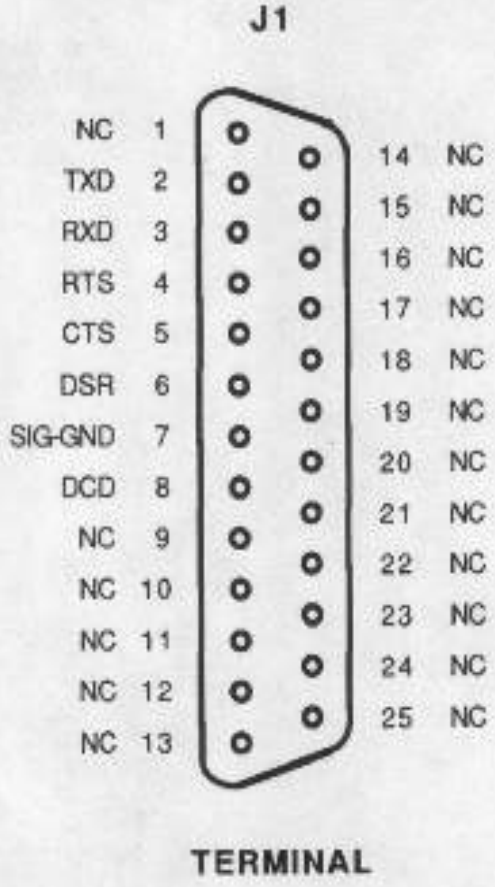
When external VPP power is connected to the EVM power connector J28 pin 1, VPP must be controlled by the EVM programmer VPP switch S5. The user must perform the programming procedures as specified in paragraphs 3.7.1 through 3.7.5. Failure to control VPP power via switch S5 will cause damage to the OTPROM/EPROM MCU device.

The user supplied power supply is connected to connector J28, which is a terminal block designed to accept 14-22 AWG wire. Two types of terminal blocks are used in the manufacturing of the EVM. Interconnection of the power supply wiring to the EVM is accomplished via either connector type shown below.



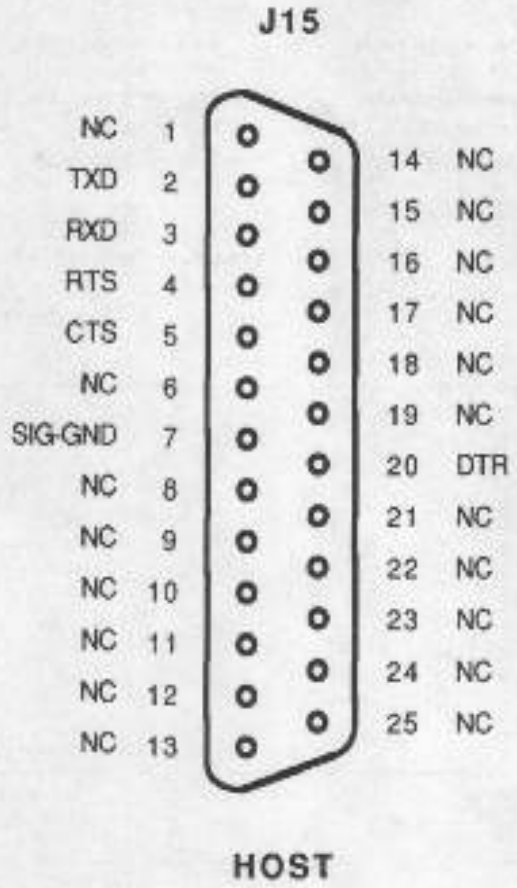
2.5.2 Terminal - EVM Interconnection

Interconnection of an RS-232C compatible terminal to the EVM is accomplished via a user supplied 20 or 25 conductor flat ribbon cable assembly as shown in Figure 2-2. One end of the cable assembly is connected to the EVM connector J1 (shown below) labeled TERMINAL. The other end of the cable assembly is connected to the user supplied terminal. For connector pin assignments and signal descriptions of the EVM terminal port connector J1, refer to Chapter 5.



2.5.3 Host Computer - EVM Interconnection

The EVM can be operated with a host computer directly or a remotely located host computer via a modem. Interconnection of an RS-232C compatible host computer to the EVM is accomplished via a user supplied 20 or 25 conductor flat ribbon cable assembly as shown in Figure 2-2. One end of the cable assembly is connected to the EVM connector J15 (shown below) labeled HOST. The other end of the cable assembly is connected to the user supplied host computer or modem. For connector pin assignments and signal descriptions of the EVM host port connector J15, refer to Chapter 5.



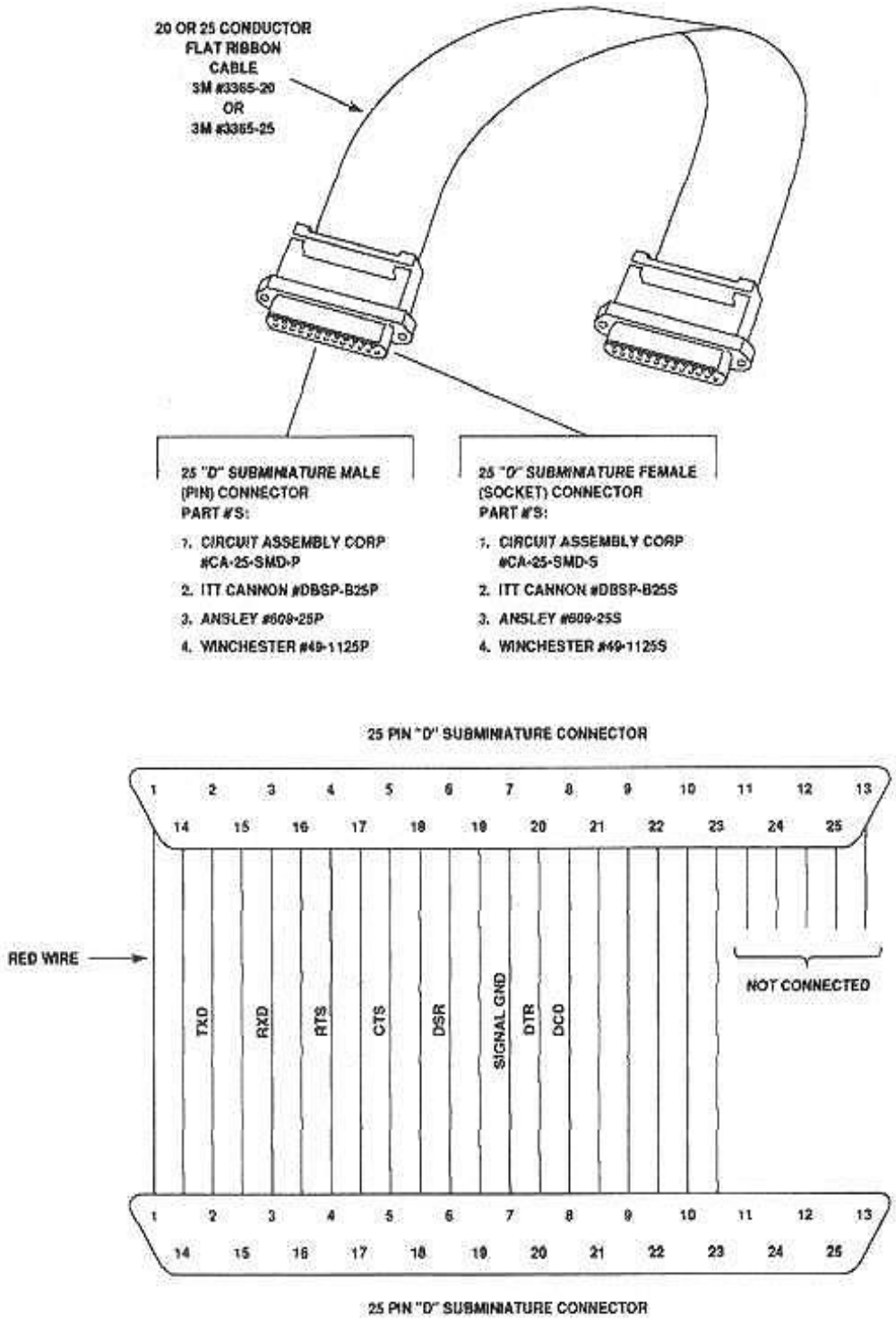
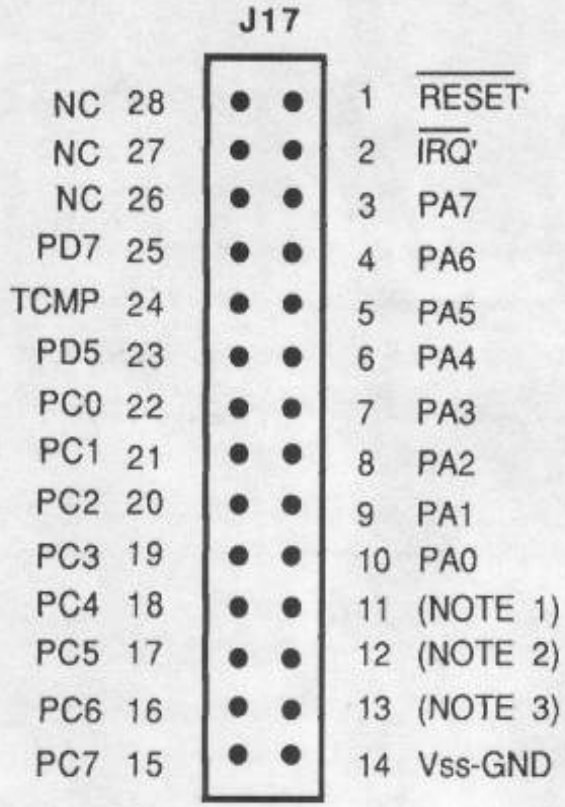


FIGURE 2-2. Terminal or Host Computer Cable Assembly

2.5.4 Target System - EVM 28-Pin Interconnection

Target system to EVM interconnection for the MC68HC05P1/P7 operation is accomplished via EVM connector J17, and a user supplied 28-pin dual-in-line plastic (DIP) cable assembly as shown in Figure 2-3. MCU I/O port connector J17 is a 28-pin header (shown below) that facilitate the interconnection of the cable assembly for evaluation purposes. For connector pin assignments and signal descriptions of the EVM connector J17, refer to Chapter 5.

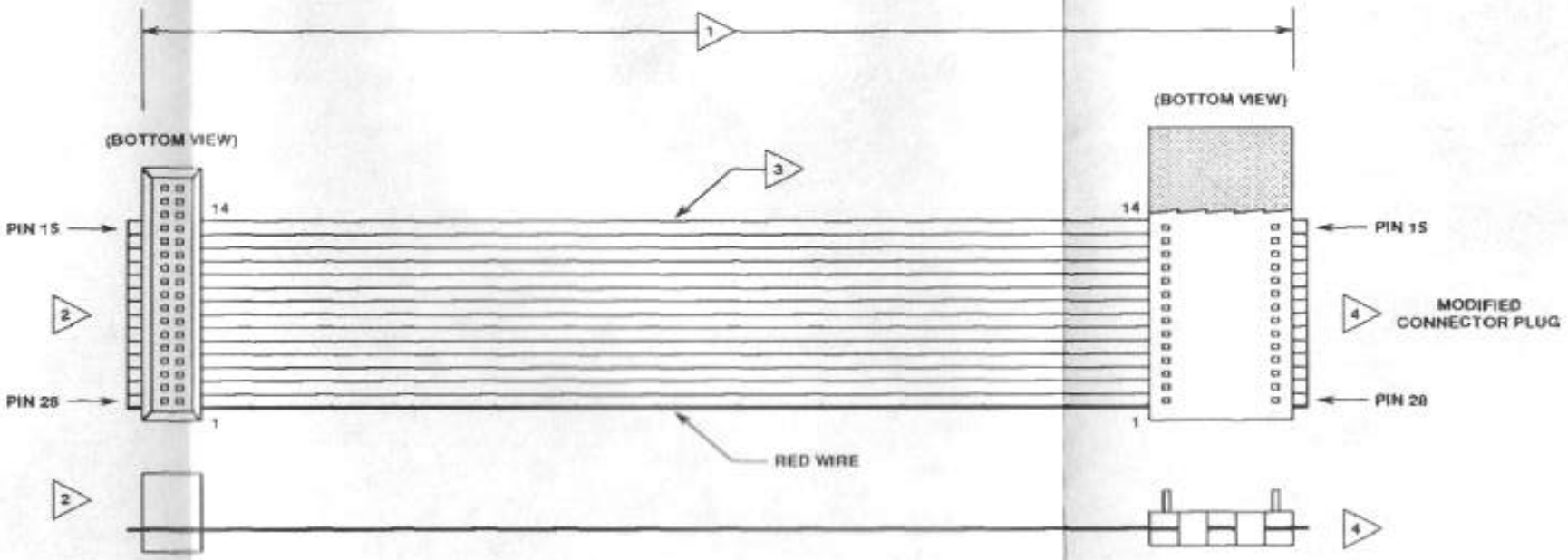


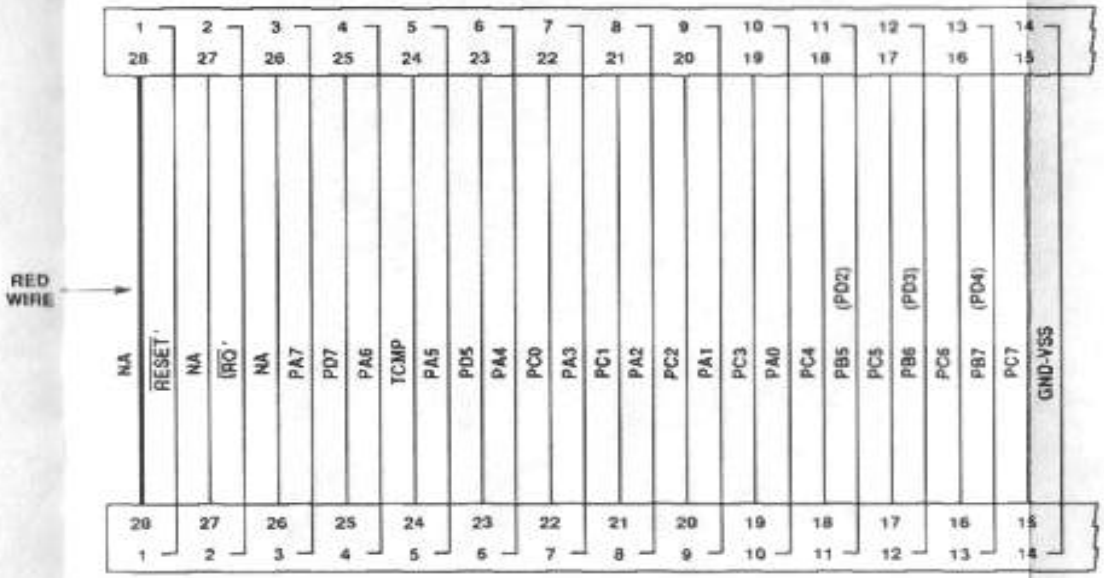
NOTES:

1. HC05P1 = PB5
   HC05P7 = PD2 (SDO)
2. HC05P1 = PB6
   HC05P7 = PD3 (SDI)
3. HC05P1 = PB7
   HC05P7 = PD4 (SCK)

28-Pin MCU I/O Port Connector

(MC68HC05P1/P7)



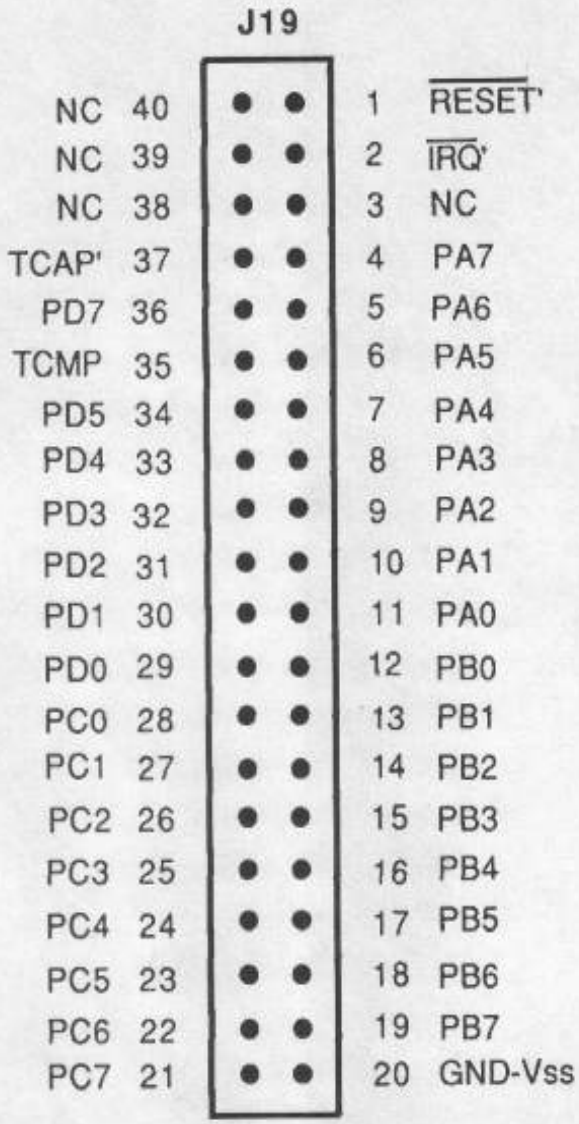


NOTES:

1. CABLE LENGTH NOT TO EXCEED 6 INCHES.
2. 34-PIN EVM CONNECTOR SOCKET. (3M # 3414-7034)
3. 28-CONDUCTOR FLAT RIBBON CABLE.
4. MODIFIED 40-PIN DUAL-IN-LINE PACKAGE (DIP) TARGET SYSTEM CONNECTOR PLUG. (3M # 3506-0000)
5. NA DENOTES NOT APPLICABLE. SIGNAL NOT USED ON EVM.
6. SIGNAL MNEMONIC CONTAINED IN PARENTHESIS DENOTES MC68HC05P7 MCU SIGNAL MNEMONIC.
7. SIGNAL MNEMONIC NOT CONTAINED IN PARENTHESIS DENOTES MC68HC05P1 MCU SIGNAL MNEMONIC.

2.5.5 Target System - EVM 40-Pin Interconnection

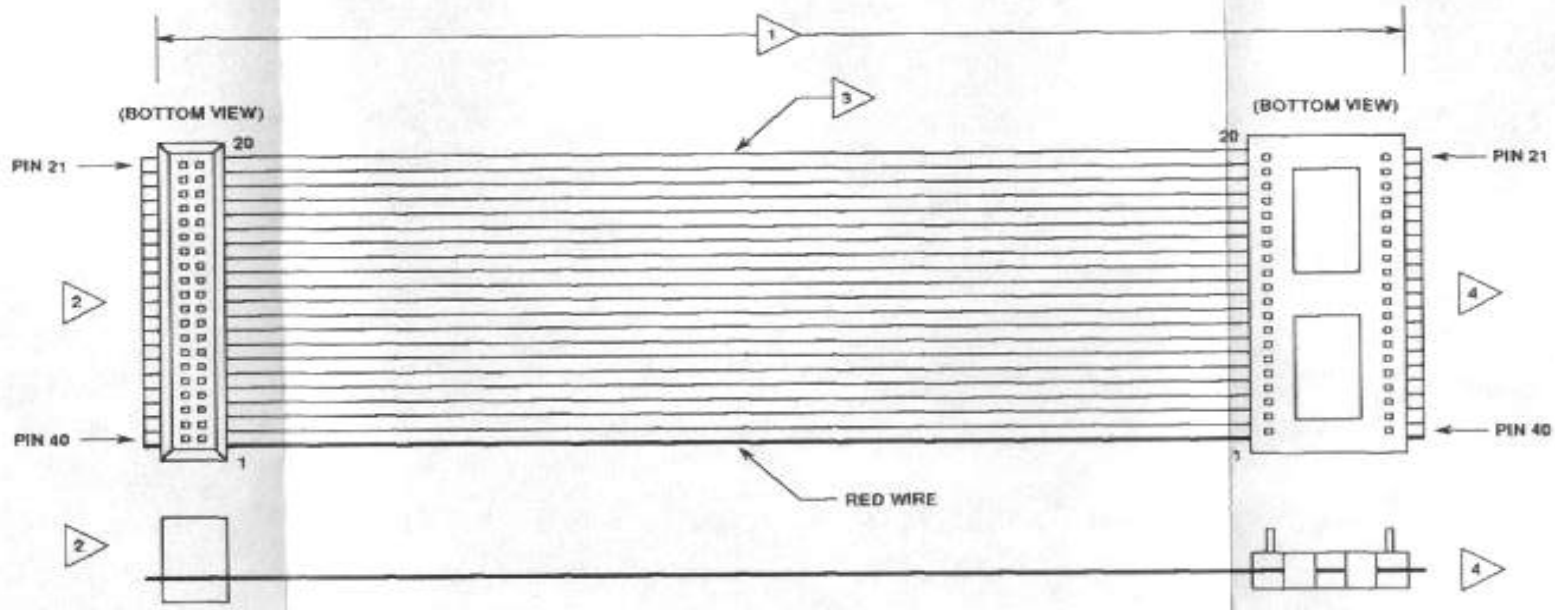
Target system to EVM interconnection for the MC68HC05A6/C2-C4/C8/C9 and MC68HC705C8/805C4 operation is accomplished via EVM connector J19, and a user supplied 40-pin dual-in-line package (DIP) cable assembly as shown in Figure 2-4. Figure 2-5 illustrates the factory supplied 44-lead plastic leaded chip carrier (PLCC) cable assembly. MCU I/O port connector J19 is a 40-pin header (shown below) that facilitate the interconnection of either cable assembly for evaluation purposes. For connector pin assignments and signal descriptions of the EVM connector J19, refer to Chapter 5.

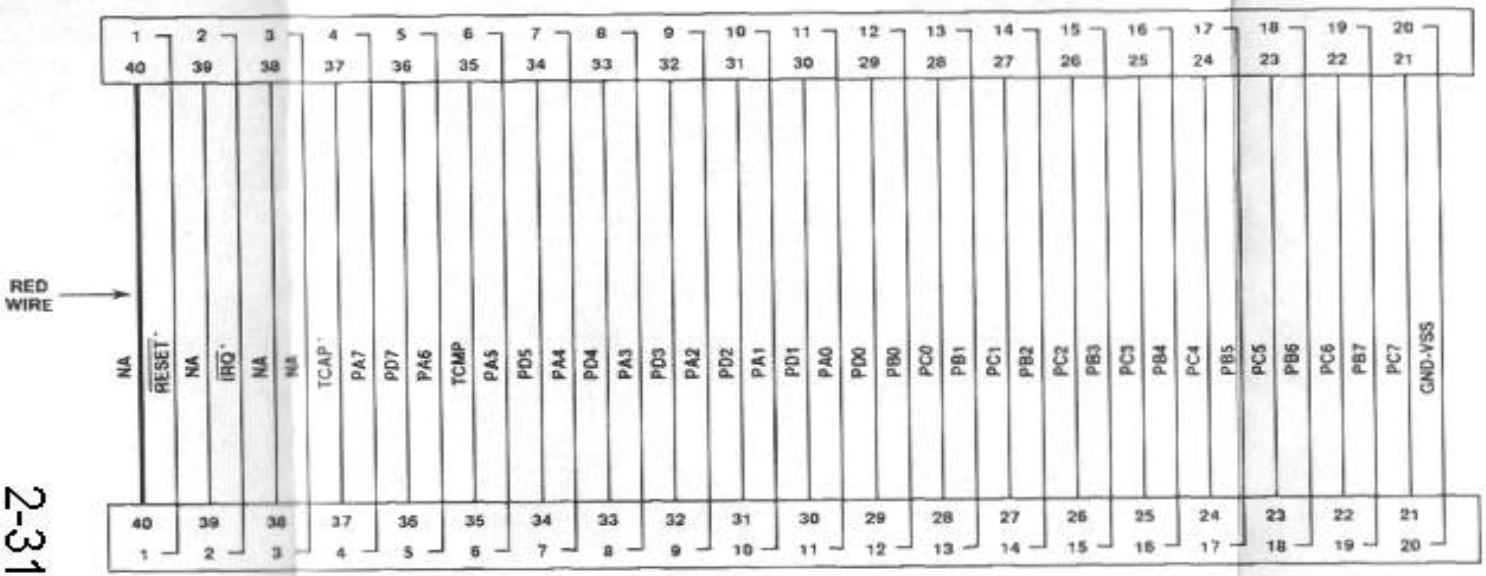


40-Pin MCU I/O Port Connector

(MC68HC05A6/C2-C4/C8/C9)

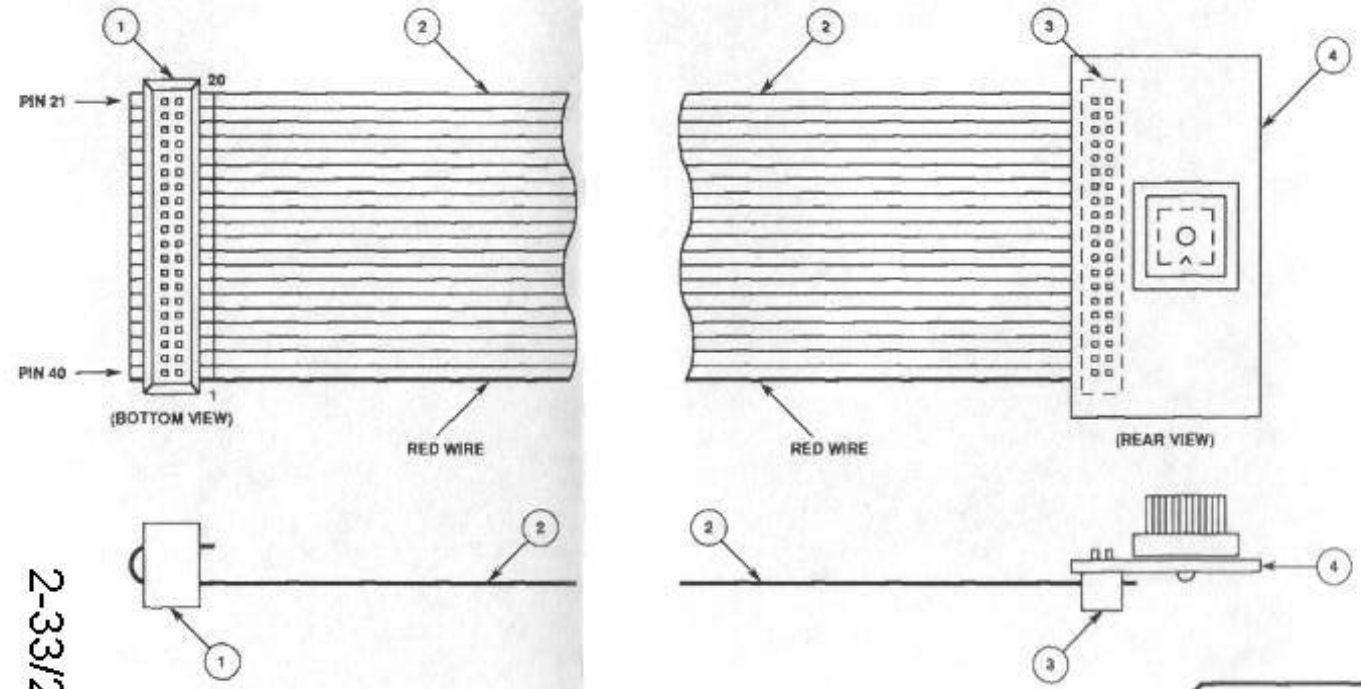
(MC68HC705C8/805C4)





NOTES:

1. CABLE LENGTH NOT TO EXCEED 6 INCHES.
2. 40-PIN EVM CONNECTOR SOCKET. (3M # 3414-7000 OR ANSLEY # 609-4000M)
3. 40-CONDUCTOR FLAT RIBBON CABLE.
4. 40-PIN DUAL-IN-LINE PACKAGE (DIP) TARGET SYSTEM CONNECTOR PLUG. (3M # 3506-0000)
5. NA DENOTES NOT APPLICABLE. SIGNAL NOT USED ON EVM.

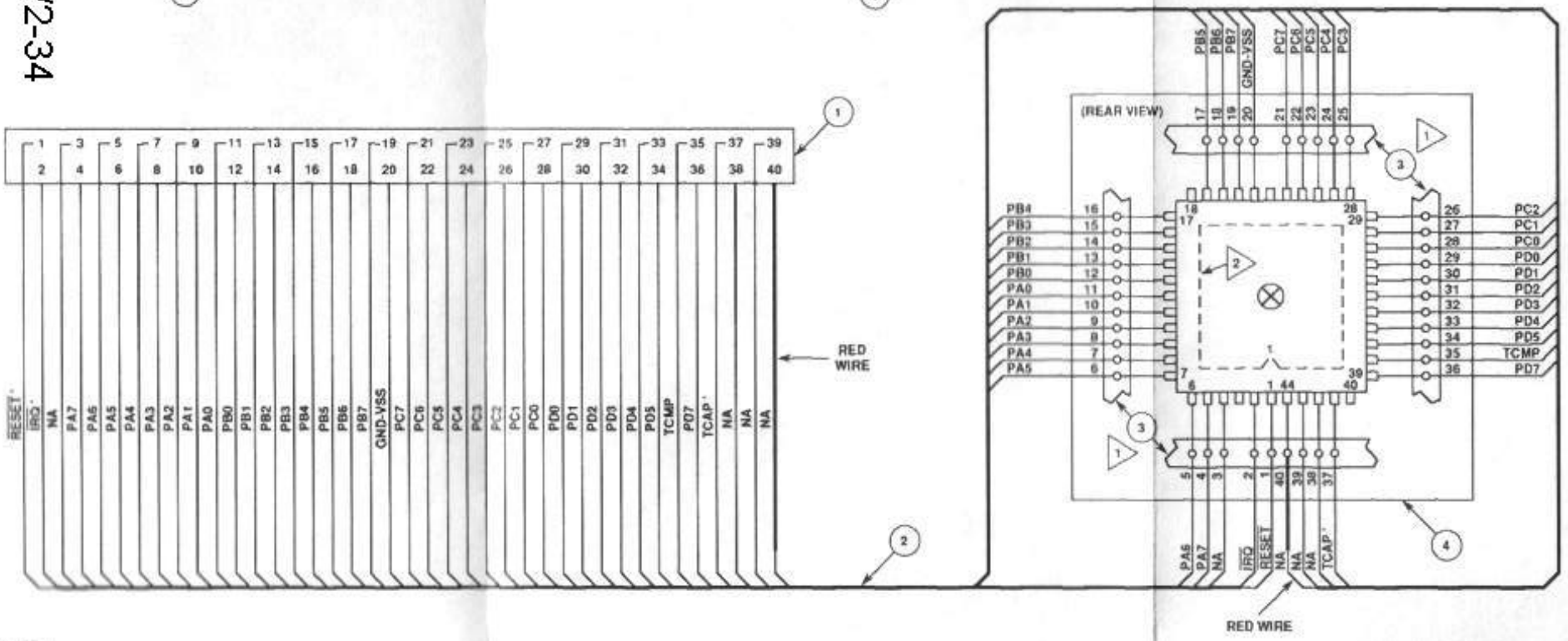


ITEMS:

1. EVM CONNECTOR (40-PIN SOCKET).
2. FLAT RIBBON CABLE.
3. CABLE PLATFORM CONNECTOR.
4. TARGET SYSTEM MCU PLUG PLATFORM (EMULATION CONNECTOR PLUG).

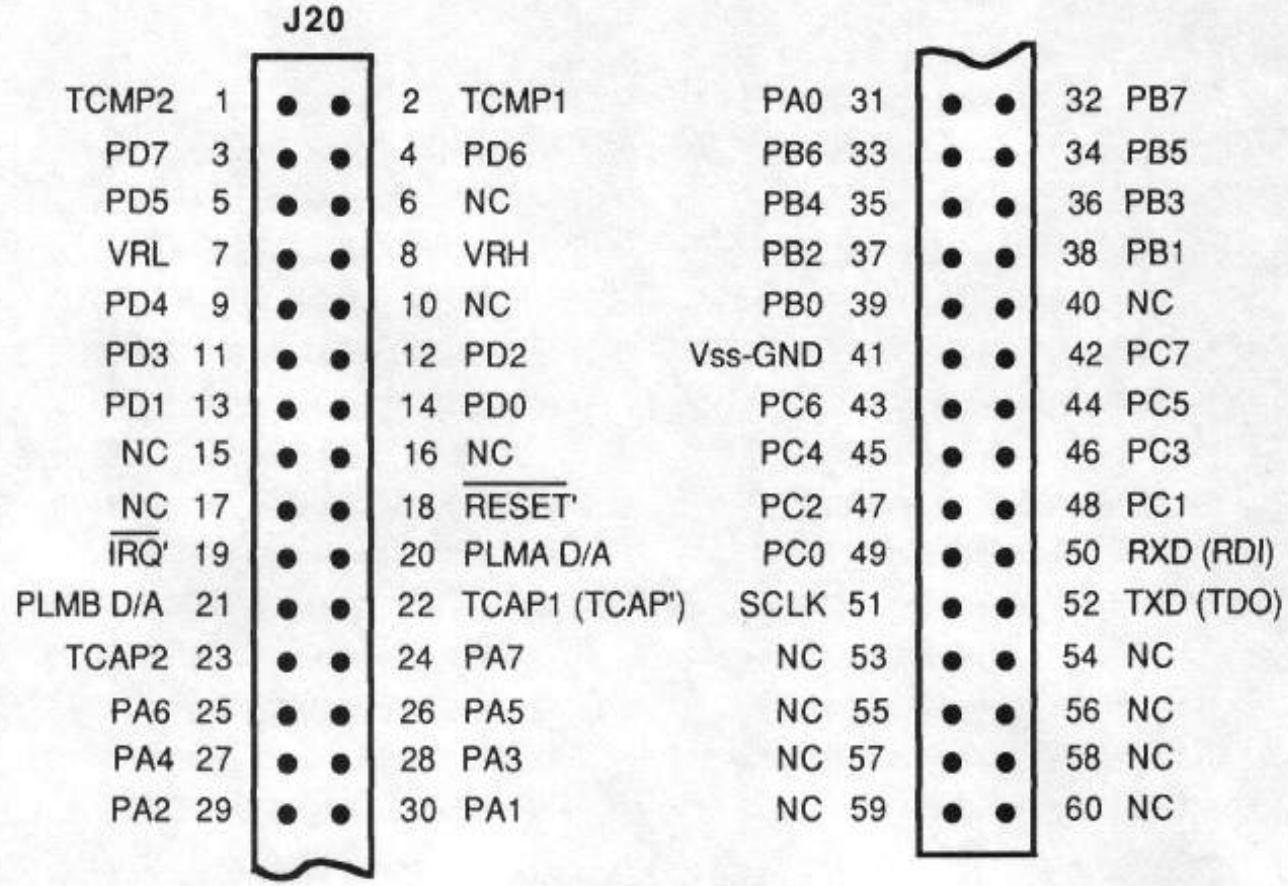
NOTES:

1. CABLE PLATFORM CONNECTOR LAYOUT REDRAWN TO SIMPLIFY SCHEMATIC DIAGRAM.
2. PIN 1 LOCATOR. DOES NOT APPEAR ON REAR OF EMULATION CONNECTOR PLUG. APPEARS ON FRONT (TOP) OF CONNECTOR PLUG PLATFORM.
3. CABLE GROUND (GND) WIRE MUST BE CONNECTED TO TARGET SYSTEM MCU VSS/GND PIN.



2.5.6 Target System - EVM 60-Pin Interconnection

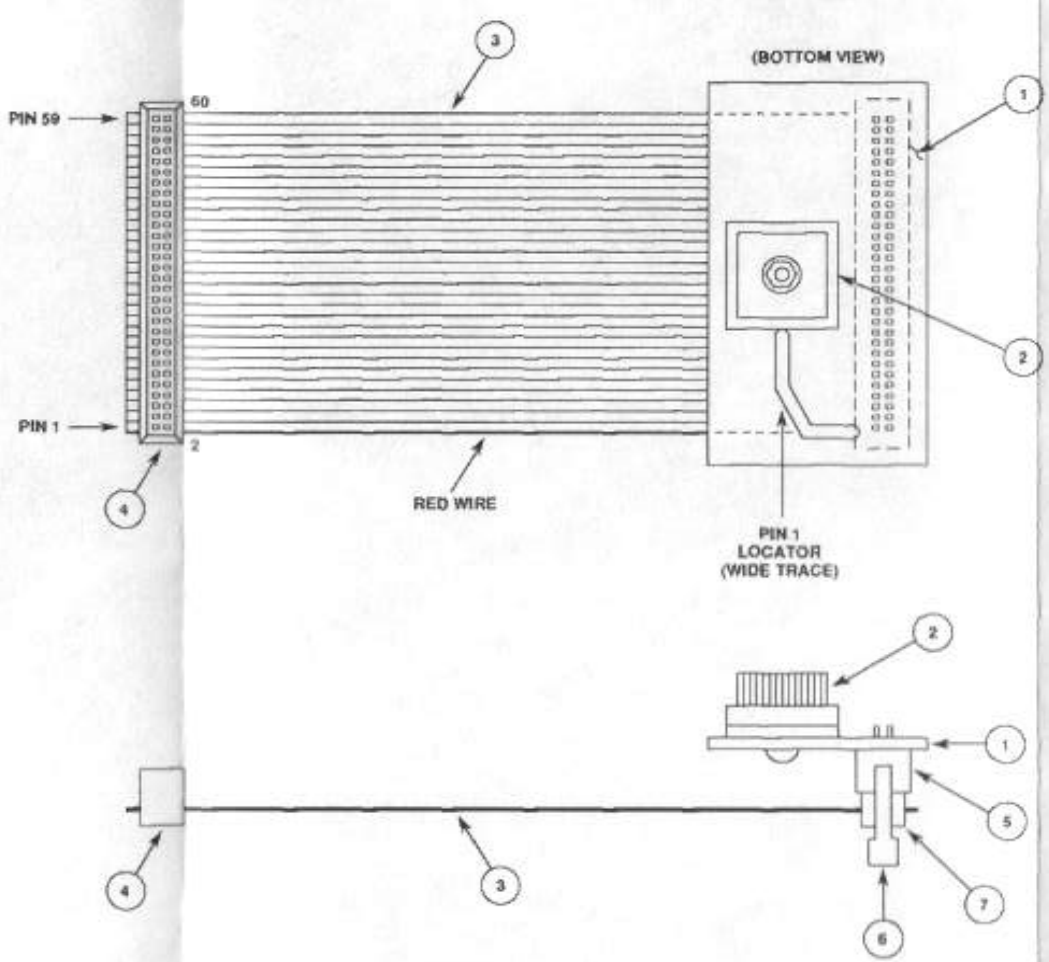
Target system to EVM interconnection for the MC68HC05B4/B6 and MC68HC805B6 operation is accomplished via EVM connector J20, and a factory supplied 52-lead plastic leaded chip carrier (PLCC) cable assembly as shown in Figure 2-6. MCU I/O port connector J20 is a 60-pin header (shown below) that facilitate the interconnection of the cable assembly for evaluation purposes. For connector pin assignments and signal descriptions of the EVM connector J20, refer to Chapter 5.



60-Pin MCU I/O Port Connector

(MC68HC05B4/B6)

(MC68HC805B6)

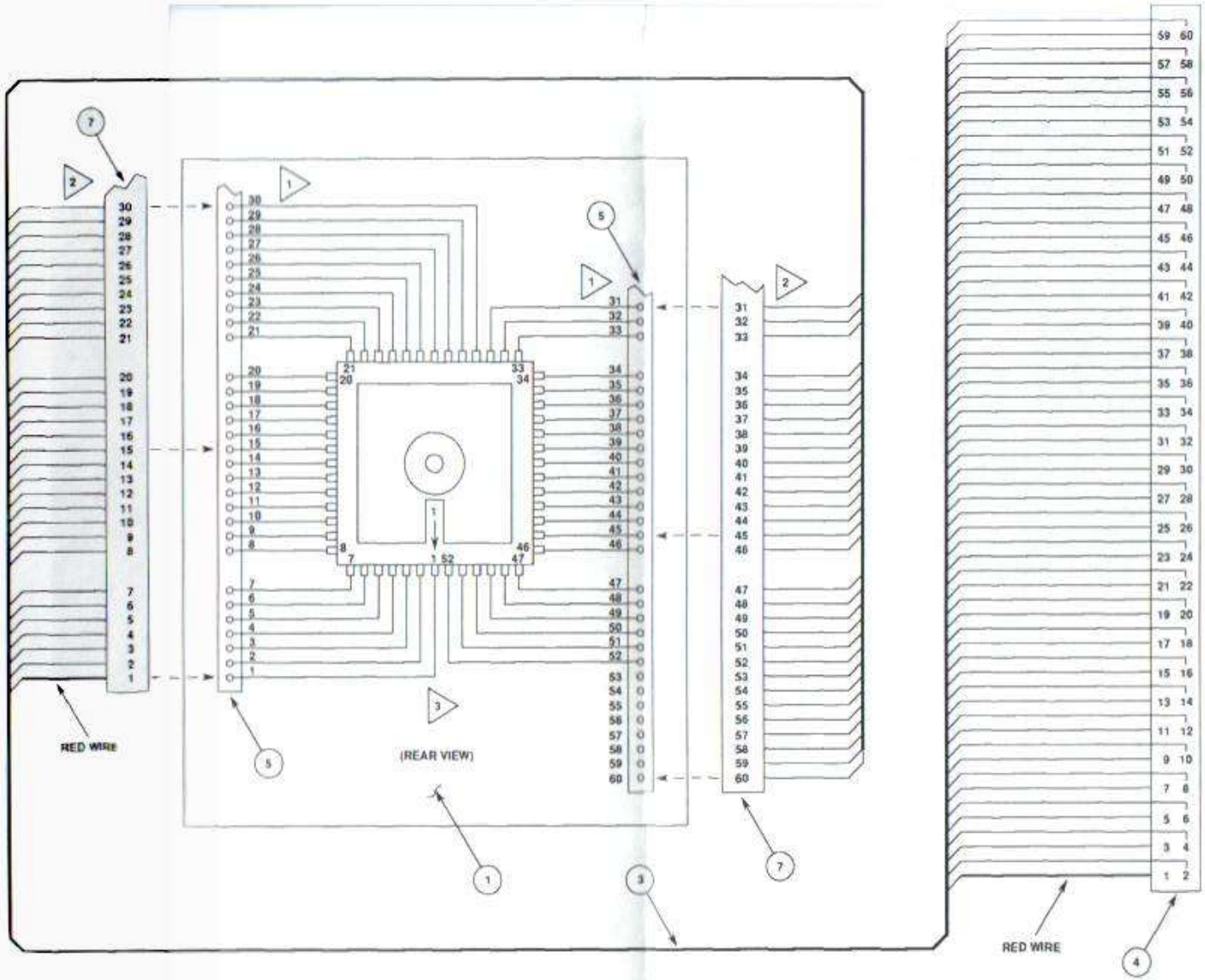


ITEMS:

1. TARGET SYSTEM MCU PLUG PLATFORM. (EMULATION CONNECTOR PLUG)
2. 52-LEAD PLCC PLUG.
3. FLAT RIBBON CABLE.
4. EVM CONNECTOR (60-PIN SOCKET).
5. 60-PIN PLUG, QUICK RELEASE TYPE CONNECTOR.
6. CONNECTOR EJECTOR LATCH.
7. 60-PIN SOCKET.

NOTE:

1. CABLE ASSEMBLY CONFIGURED AT FACTORY FOR M68HC05EVM TARGET SYSTEM OPERATION.



ITEMS:

(1) TARGET SYSTEM MCU PLUG PLATFORM. (EMULATION CONNECTOR PLUG, 52-LEAD PLCC TYPE)

(3) FLAT RIBBON CABLE.

(4) EVM CONNECTOR (60-PIN SOCKET).

(5) 60-PIN PLUG, QUICK RELEASE TYPE CONNECTOR.

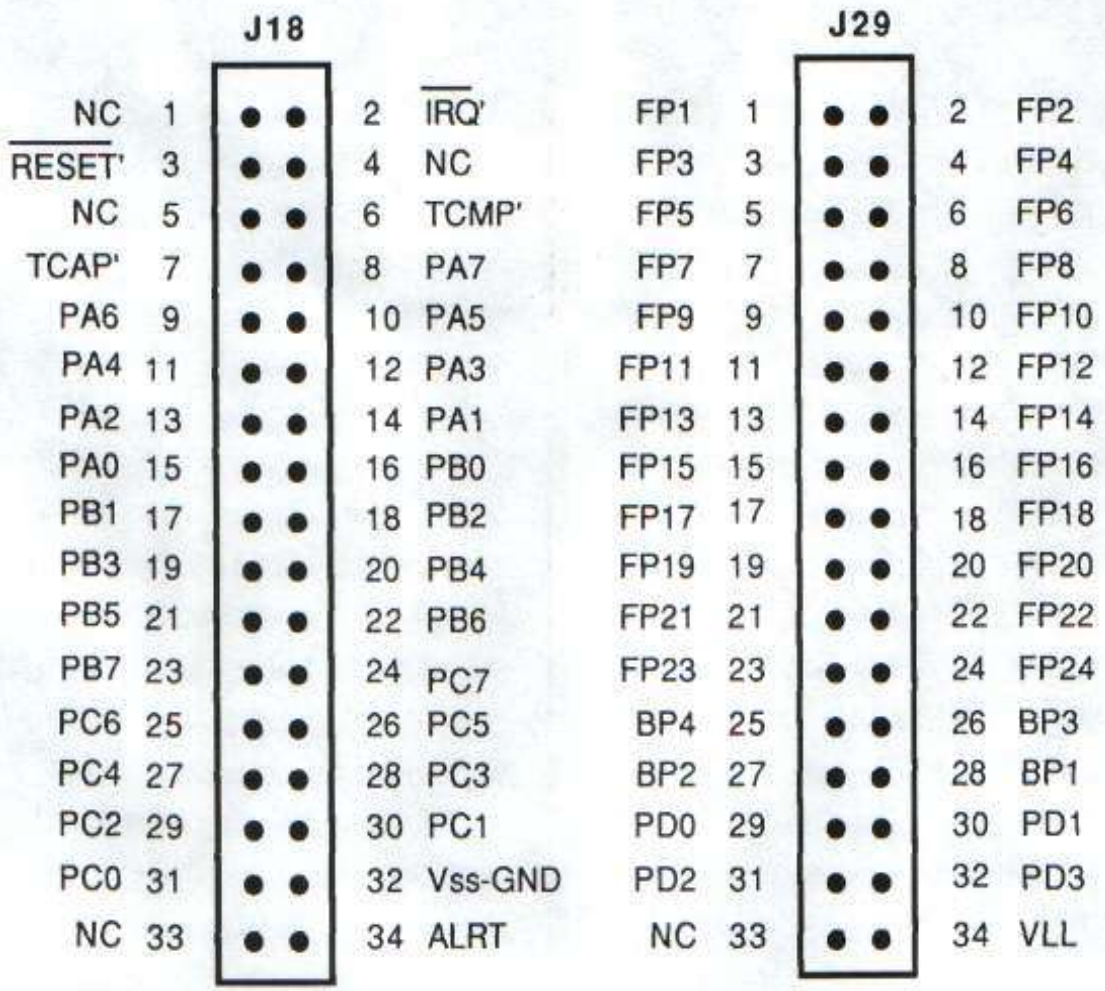
(7) 60-PIN SOCKET.

NOTES:

1. 60-PIN PLUG CONNECTOR (5) MOUNTED ON TOP SIDE OF PLATFORM. CONNECTOR REDRAWN TO SIMPLIFY SCHEMATIC DIAGRAM.
2. 60-PIN CABLE SOCKET (7) CONNECTS TO 60-PIN PLATFORM PLUG CONNECTOR (5) ON TOP SIDE OF PLATFORM.
3. PLATFORM LAYOUT REDRAWN TO SIMPLIFY SCHEMATIC DIAGRAM.
4. REFER TO EVM CONNECTOR J20 ILLUSTRATION FOR SPECIFIC CONNECTOR SIGNAL MNEMONIC.
5. CABLE ASSEMBLY WIRING IS FABRICATED TO A ONE-TO-ONE RELATIONSHIP (E.G., PIN 1 CONNECTED TO PIN 1, PIN 2 CONNECTED TO PIN 2, ETC.).

2.5.7 Target System - EVM 68-Pin Interconnection

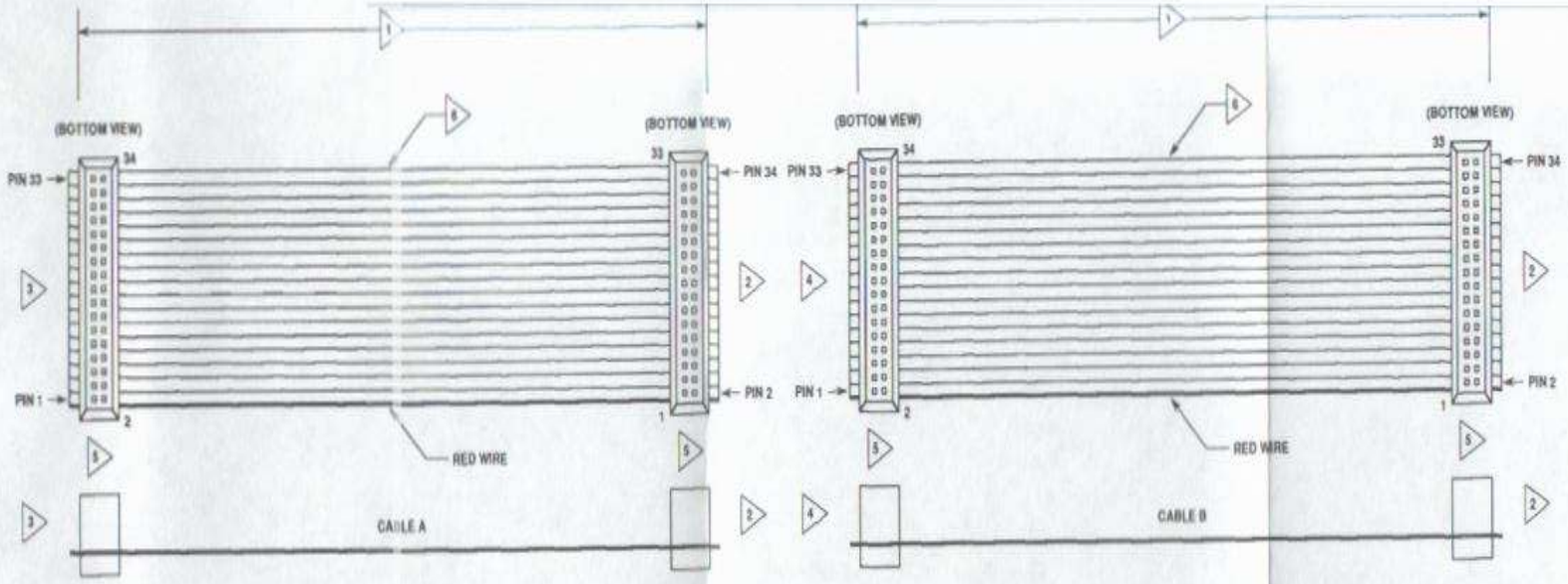
Target system to EVM interconnection for the MC68HC05L6 operation is accomplished via EVM connectors J18 and J29, and user supplied 34-pin header type cable assemblies shown in Figure 2-7. Both MCU I/O port connector A (J18) and connector B (J29) are 34-pin headers (shown below) that facilitate the interconnection of the cable assemblies for evaluation purposes. For connector pin assignments and signal descriptions of EVM connectors J18 and J29, refer to Chapter 5.

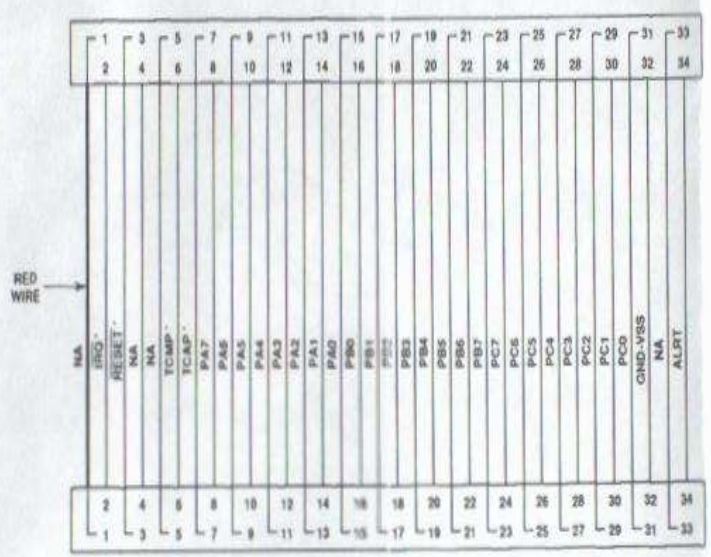


34-Pin MCU I/O Port Connector A

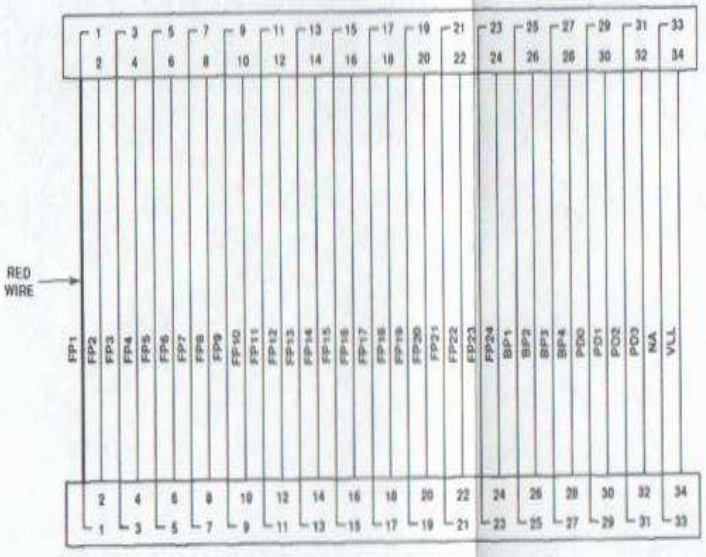
34-Pin MCU I/O Port Connector B

(MC68HC05L6)





CABLE A



CABLE B

NOTES:

1. CABLE LENGTH NOT TO EXCEED 6 INCHES.
2. TARGET SYSTEM EMULATION CONNECTOR SOCKET. MATES WITH HEADER TYPE CONNECTOR PLUG, DOUBLE ROW POST, 34-PIN, AMPTRONICS # 929715-01-17.
3. EVM CONNECTOR SOCKET. MATES WITH EVM HEADER TYPE CONNECTOR PLUG J18.
4. EVM CONNECTOR SOCKET. MATES WITH EVM HEADER TYPE CONNECTOR PLUG J20.
5. 34-PIN CONNECTOR SOCKET. (3M # 3414-7034 OR ANSLEY # 609-3400M)
6. 34-CONDUCTOR FLAT RIBBON CABLE.
7. NA DENOTES NOT APPLICABLE. SIGNAL NOT USED ON EVM.

CHAPTER 3

OPERATING INSTRUCTIONS

3.1 INTRODUCTION

This chapter provides the necessary information to initialize and operate the EVM in a target system environment. One time programmable read only memory (OTPROM), erasable programmable read only memory (EPROM), and electrically erasable programmable read only memory (EEPROM) MCU programming, assembling/disassembling, and downloading procedures are also provided. Information contained in this chapter will be presented in the following order:

a. Control switch descriptions

b. Limitations

c. Operating procedures

d. MCU programming procedures

e. Assembling/disassembling procedures

f. Downloading procedures

The EVM is factory tested for target system evaluation and OTPROM/EPROM/EEPROM MCU programming operations. Refer to paragraph 2.4 COMPONENT IMPLEMENTATION for components required for MC68HC05L6 evaluation, resident MCUs required for specific target system evaluation, and resident MCU and OTPROM/EPROM/EEPROM MCU installation techniques.

The monitor is the resident firmware (EVMbug) for the EVM, which provides a self contained programming and operating environment. The monitor interacts with the user through predefined commands that are entered from a terminal. These commands perform functions such as display or modify memory, display or modify MCU internal registers, program execution under various levels of control, control access to various I/O peripherals connected to the EVM, and control programming and reading of OTPROM/EPROM/EEPROM MCUs.

3.2 CONTROL SWITCHES

The EVM contains five switches that allow the user to control specific functions. Switches S1 through S3 control the reset and abort functions. Switches S4 and S5 control the MCU programming power. Table 3-1 identifies these switches by name, description, and function.

TABLE 3-1. EVM Control Switches

NAME	DESCRIPTION AND FUNCTION
ABORT switch (S1)	Momentary action pushbutton switch - when pressed, returns EVM operation to the monitor map from the user map assuming proper operation of user code. (If MCU gets lost in the user map, the abort is useless and a master reset must be issued.) The abort function has no effect when operating in the monitor map.
USER RESET switch (S2)	Momentary action pushbutton switch - when pressed, resets EVM MCU and user I/O, and enables map switching to the user map. This switch is used as a map switch for execution of user code from the user reset vector.
MASTER RESET switch (S3)	Momentary action pushbutton switch - when pressed, resets EVM MCU and user I/O, places EVM operation in the monitor map, and enables the EVM prompt to be displayed on terminal CRT.
Programmer power switch (S4)	On-off slide switch - applies +5 Vdc and +12 Vdc power to the programmable MCU device installed in sockets Z1, Z2, and Z3. Power is applied when the switch is placed in the ON position.
Programmer Vpp switch (S5)	On-off slide switch - applies programming voltage (Vpp) from connector J28 pin 1 to the programmable MCU device installed in sockets Z1, Z2, and Z3. Vpp is applied when the switch is placed in the ON position.

3.3 LIMITATIONS

The user may not trace an RTI instruction with an interrupt enabled and pending due to MCU interrupt handling. Attempting this will cause an interrupt in the monitor map which will perform a software reset of the EVM. User breakpoints will be left in the user map, but will not be recognized by the monitor. The user stack pointer will also reflect the occurrence of an interrupt.

Mixing interrupt requests (IRQs) and user software interrupts (SWIs) should be avoided whenever possible, due to IRQ-SWI EVM timing. If a concurrent hardware interrupt and SWI should happen, an EVM failure could occur which may stop program execution. Activation of the MASTER RESET switch will restore the EVM to proper operation. These conditions will statistically occur very seldom.

No protection is provided to limit user programs to the exact amount of MCU ROM available. The user must be aware of the memory map of the MCU being simulated and ensure that only valid ROM locations are used.

The baud rates of the terminal and host I/O ports are independent; however, during transparent mode, characters from the host computer to the terminal may occasionally be lost if the terminal baud rate is not higher than the host baud rate. This occurs only during the transparent mode, and does not apply when downloading to memory.

During MC68HC05A6/C2-C4/C8/C9 MCU evaluation, the user should not write to locations $001E and $001F. These locations are Computer Operating Properly (COP) and test register locations.

During MC68HC05B4/B6 MCU evaluation, the user should not write to location $0020. This location is a test register location for factory use only.

Programming and bulk erase operations cannot be performed on MC68HC805C4 MCU A65G mask sets, but blank checking, copying, and verifying can be performed.

The EVM cannot emulate the low power capabilities of the MC68HCL05C4/C8 MCU devices. The EVM can emulate the high speed capabilities of the MC68HSC05C4/C8 MCU devices by the replacement of the 8 MHz crystal (Y1) to a 16 MHz crystal. Refer to paragraph 2.3.10.

(Specific resident MCU operational limitations are provided on the following page.)

Resident MC68HC05C9 MCU Operational Limitations

The MC68HC05C9 MCU option register resides internally at location $3FDF. Therefore, when evaluating other HC05xx family devices with option registers residing at $1FDF, all accesses to the resident HC05C9 MCU option register must be made at $3FDF as opposed to $1FDF even though the 8K memory map is selected.

The HC05C9 MCU cannot be used to exactly replicate SPI operations for all HC05Cx MCU devices.

The HC05C9 MCU DDRD has a negative implication affecting serial peripheral interface (SPI) operations. All port D lines used as output lines must have corresponding DDR bits set to a logic 1 (i.e., SCK (master), MOSI (master), MISO (slave)) when evaluating all HC05Cx MCU devices.

The HC05C9 MCU contains a port D data direction register (DDRD) that can be used to emulate the MC68HC05P1 and MC68HC05P7 MCU port D, bit 5 (PD5). The user must be aware that the entire DDRD is implemented and other DDRD bits must not be accidently written to. When evaluating HC05P1 and HC05P7 devices with the resident HC05C9 MCU, the user must write $20 value to DDRD location $07 when configuring PDDR5 as an output line. This step will ensure that other port D pins remain as input lines as required for correct HC05P1 and HC05P7 evaluation.

Resident MC68HC705C8 MCU Operational Limitations

When evaluating with a resident MC68HC705C8 MCU the option register is accessed at location $1FDF. The HC705C8 option registers must be written to if edge-only interrupts are to be used. The HC705C8 option register contains RAM/ROM bits used for dynamic mapping of internal RAM/ROM.

The HC705C8 MCU can be used to exactly replicate SPI operations for all HC05Cx MCU devices.

Resident MC68HC805C4 MCU Operational Limitations

When evaluating with a resident MC68HC805C4 MCU the option register is accessed at location $1FDF. The HC805C4 option register must be written to if edge-only interrupts are to be used. The HC805C4 option register only configures IRQ options.

The HC805C4 MCU can be used to exactly replicate SPI operations for all HC05Cx MCU devices.

3.4 OPERATING PROCEDURE

Applying power to the EVM causes a Power On Reset (POR) to occur. This POR condition causes the MCU and user I/O port circuitry to be reset, and the monitor invoked. All user registers are in an unknown state during monitor power-up.

A fixed 9600 baud rate is provided for the terminal port, and a software selectable 300-19.2K baud rate selection is provided for the host port. The monitor automatically configures the baud rate setting for the host I/O port. The default baud rate for the host I/O port is 9600 baud. The host I/O port baud rate can be reconfigured via the SPEED command.

The terminal Cathode Ray Tube (CRT) displays the following message:

```
EVMbug-HC05 X.X
S=FF     P=XXXX     A=XX     X=XX     C=E8          111.I...
>
```


where:

X is a revision of the software or an unknown register state.

Condition code register (CCR) HINZC bits are as follows:

1	=	Fixed bit, set to logic 1
1	=	Fixed bit, set to logic 1
1	=	Fixed bit, set to logic 1
H	=	Half carry bit
I	=	Interrupt mask bit
N	=	Negative bit
Z	=	Zero bit
C	=	Carry/borrow bit

Status of the CCR bits are displayed as follows:

When all CCR bits are set (logic 1), bits are displayed as follows:

```
S=XX    P=XXXX    A=XX    X=XX    C=FF        111HINZC
```

When all CCR bits are cleared (logic 0), bits are displayed as follows:

```
S=XX    P=XXXX    A=XX    X=XX    C=E0        111.....
```

When a specific bit is set (I), bits are displayed as follows:

```
S=XX    P=XXXX    A=XX    X=XX    C=E8        111.I...
```


When specific bits are set (H, I, and C), bits are displayed as follows:

```
S=XX    P=XXXX    A=XX    X=XX    C=F9        111HI..C
```

After initialization or return of control to the monitor, the terminal CRT displays the prompt " > " and waits for a response. If an invalid response is entered, the terminal CRT displays "ILLEGAL/INSUFFICIENT ENTRY" followed by the prompt " > ".

The EVM waits for a command line input from the user terminal. When a proper command is entered, the operation continues in one of two basic modes. If the command causes execution of a user program, the monitor may or may not be reentered, depending upon the desire of the user. For the alternate case, the command is executed under the control of the monitor, and the system returns to a waiting condition after the command is completed. During command execution, additional user input may be required, depending on the command function.

The user can use any of the commands supported by the monitor. A standard input routine controls the EVM operation while the user types a command line. Command processing begins only after the command line has been terminated by depressing the keyboard carriage return (RETURN) key.

3.5 COMMAND LINE FORMAT

The command line format is as follows:

><command> [<parameters>](RETURN)

where:

>	EVMbug monitor prompt.
<command>	Command mnemonic.
<parameters>	Expression or address.
(RETURN)	RETURN keyboard key - depressed to enter command.

NOTES:

(1) The command line format is defined using special characters which have the following syntactical meanings:

< >	Enclose syntactical variable
[ ]	Enclose optional fields
[ ]...	Enclose optional fields repeated

These characters are not entered by the user, but are for definition purposes only.

(2) Fields are separated by a single space.

(3) All input numbers are interpreted as hexadecimal. A dollar sign ($) may precede any number input, but is not required.

(4) All input commands can be entered either upper or lower case lettering. All input commands are converted automatically to upper case lettering except for downloading commands sent to the host computer, or when operating in the transparent mode.

(5) A maximum of 30 characters may be entered on a command line. After the 30th character is entered, the monitor automatically terminates the command entry and processes the command line.

(6) Parameters are interpreted to be the last two or three characters in the parameter file. Parameter errors may be corrected by backspacing. This is accomplished via the terminal keyboard (CTRL)H function.

3.6 MONITOR (EVMbug) COMMANDS

The monitor (EVMbug) commands are listed alphabetically by mnemonic in Table 3-2. Each of the commands are described in detail following the tabular command listing.

Additional terminal keyboard functions are as follows:

(BREAK)	Abort command
(CTRL)A	Default transparent mode exit character
(CTRL)H	Backspace
(CTRL)S	Freeze screen
(CTRL)X	Cancel command line
(RETURN)	Enter command

NOTE

When using the control key with a specialized command such as (CTRL)X, the (CTRL) key is depressed and held, then the X key is depressed. Both keys are then released.

During memory display output to terminal, (CTRL)S will delay the output until another character is entered.

Command line input examples in this chapter are amplified with the following:

Underscore entries are user-entered on the terminal keyboard.

Command line input is entered when the keyboard (RETURN) key is depressed.

Typical example of this explanation is as follows:

```
>MD 100 21F
```

TABLE 3-2. Monitor (EVMbug) Commands

COMMAND	DESCRIPTION
ASM <address>	Assembler/disassembler (interactive)
BF <addr1> <addr2> <data>	Block fill memory with data
BR [<address>]...	Breakpoint set
BULK <device>	Bulk erase EEPROM
CHCK [<addr1> <addr2>] <device>	Blank check OTPROM/EPROM/EEPROM
COPY [<addr1> <addr2>] <device>	Copy OTPROM/EPROM/EEPROM
G [<address>]	Go (execute program)
HELP	Help (display commands)
LOAD <port> [=<text>]	Load (S-records*) from I/O port
MD <addr1> [<addr2>]	Memory display
MM <address>	Memory modify (interactive)
NOBR [<address>]...	Remove breakpoint
P [<count>]	Proceed (thru breakpoint)
PROG [<addr1> <addr2>] <device>	Program OTPROM/EPROM/EEPROM
RD	Register display
RM	Register modify (interactive)
SPEED <baud rate>	Host I/O port baud rate select
T [<count>]	Trace
TM [<exit character>]	Transparent mode
VERF [<addr1> <addr2>] <device>	Verify OTPROM/EPROM/EEPROM

NOTE: * Refer to Appendix A for S-record information.

3

3.6.1 Assembler/Disassembler

ASM <address>

where <address> is the starting address for the assembler operation.

The assembler/disassembler is an interactive assembler/editor in which the source program is not saved. Each source line is converted into the proper machine language code and is stored in memory on a line-by-line basis at the time of entry. In order to display an instruction, the machine code is disassembled and the instruction mnemonic and operands are displayed. All valid opcodes are converted to assembly language mnemonic. All invalid opcodes return a Form Constant Byte (FCB) conversion.

The ASM command allows the user to create, modify, and debug MC6805 MCU code. No provision is made for line numbers or labels.

Assembler input must have exactly one space between the mnemonic and the operand. There must be no space between the operand and the index specification (,X) except in the case of indexed no offset. Assembler input must be terminated by a carriage return. No comments, etc., are allowed after the instruction input. Examples are as follows:

a. >LDA 0,X

b. >STA 10,X

c. >ASRA

d. >COMX

e. >CMP 200

After each new assembler input line, the new line is disassembled for the user before stepping to the new instruction. The new line may assemble to a different number of bytes than the previous one.

For Branch if High or Same (BHS)/Branch if Carry Clear (BCC) mnemonics, disassembly displays the BCC mnemonic. For Branch if Lower (BLO)/ Branch if Carry Set (BCS) mnemonics, disassembly displays the BCS mnemonic.

Branch address offsets are automatically calculated by the assembler, thus the address is inputted as the operand rather than an offset value.

The assembler is terminated by entering a period (.) followed by a carriage return as the only entry on the command input line. Entering a carriage return alone on an input line steps to the next instruction.

Entering (CTRL)X cancels an input line. The monitor remains in the assembler mode. If an invalid address is specified, the invalid address and the message "BAD MEMORY" is displayed on the terminal CRT.

EXAMPLES **DESCRIPTION**

```
>ASM 100
100 9D          NOP          >LDA #$55      Immediate mode addressing,
100 A6 55       LDA  #$55                   requires # before operand.

102 C1 00 9D    CMP  $009D   >STA $60       Direct mode addressing, may
102 B7 60       STA  $60                    have $ but not necessary.

104 9D          NOP          >LDA 0,X       Index mode, if not offset (,X)
104 E6 00       LDA  $00,X                  will be accepted.

106 FB          ADD  ,X      >BRA $100      Branch offsets calculated
106 20 F8       BRA  $0100                  automatically, address required
                                            as conditional branch operand.

108 FF          STX  ,X       >.            Assembler operation terminated.
>
```

Refer to the end of this chapter for additional operational information pertaining to the use of the assembler/disassembler.

3.6.2 Block Fill

BF <address1> <address2> <data>

where:

<address1>	Lower limit for fill operation.
<address2>	Upper limit for fill operation.
<data>	Fill pattern hexadecimal value.

The BF command allows the user to repeat a specific pattern throughout a determined user memory range.

Caution should be used when modifying locations which are internal to the MC68HC05 MCU device (i.e., port addresses, timer registers, etc.).

EXAMPLES	DESCRIPTION
>BF 80 100 FF	Fill each byte of memory from 80 through 100 with data pattern FF.
>BF 200 200 0	Set location 200 to 0.

3.6.3 Breakpoint Set

BR [<address>]...

The BR command sets the address into the breakpoint address table. During execution of the user program, a debug halt occurs immediately preceding the execution of any instruction address in the breakpoint table.

The user should not place a breakpoint on a software interrupt SWI instruction because this is the instruction that the monitor uses to breakpoint/single step a user program. However, the user may use the SWI instruction in the user program.

A maximum of five breakpoints may be set. After setting the breakpoint, the current breakpoint addresses, if any, are displayed. All multiple breakpoints are entered on the same line.

COMMAND FORMATS	DESCRIPTION
BR	Display all current breakpoints.
BR <address>	Set breakpoint.

EXAMPLES	DESCRIPTION
>BR 324 Brkpts=0324 >	Set breakpoint at address location 324.
>BR 324 212 100 Brkpts=0324 0212 0100 >	Sets three breakpoints. Breakpoints at same address will result in only one breakpoint being set.
>BR Brkpts=0324 0212 0100 >	Display all current breakpoints.

3.6.4 Bulk Erase

BULK <device>

where: <device> C4 for the MC68HC805C4 EEPROM MCU
B6 for the MC68HC805B6 EEPROM MCU

The BULK command is invoked without an address, and allows the user to bulk erase all MCU internal EEPROM installed in programming sockets Z2 or Z3. Prior to entering this command, the user must follow the EEPROM erasing procedure as described in paragraph 3.7.2.

EXAMPLES	DESCRIPTION
>BULK C4	Erase MC68HC805C4 EEPROM.
BLANK CHECKING	Blank checking in process message.
>	Prompt indicates blank EEPROM.
>BULK C4	Erase MC68HC805C4 EEPROM.
BLANK CHECKING	Blank checking in process message.
PART NOT BLANK	Message indicates EEPROM not blank.
>	Prompt.

NOTE

MC68HC805B6 EEPROM checking takes approx. 30 seconds (E = 2 MHz).

>BULK B6	Erase MC68HC805B6 EEPROM.
BLANK CHECKING	Blank checking in process message.
>	Prompt indicates blank EEPROM.
>BULK B6	Erase MC68HC805B6 EEPROM.
BLANK CHECKING	Blank checking in process message.
PART NOT BLANK	Message indicates EEPROM not blank.
>	Prompt.

NOTE

MC68HC805C4 and MC68HC805B6 EEPROM byte erasing cannot be performed on the EVM.

Individual byte locations are changed by copying the MCU EEPROM contents into the EVM user map pseudo ROM, bulk erasing the MCU internal EEPROM, memory modifying the specific byte location, then reprogramming the MCU internal EEPROM. Individual byte locations are verified by copying the MCU EEPROM contents into the EVM user map pseudo ROM, and displaying the specific memory location. The following example shows how to change and verify an individual EEPROM byte location.

EXAMPLE	DESCRIPTION
>COPY B6	Copy MC68HC805B6 EEPROM contents.
>	Prompt indicates copy sequence completed.
>BULK B6	Erase MC68HC805B6 EEPROM.
BLANK CHECKING	Blank checking in process message.
>	Prompt indicates blank EEPROM completed.
>MM 103	Display memory location $103.
0103=55>77.	Change data at $103 and terminate MM operation.
>	Prompt indicates memory modify completed.
>PROG B6	Program MC68HC805B6 EEPROM contents.
BLANK CHECKING	Blank checking in process message.
XXXX	EEPROM address location XXXX updated.
VERIFYING	Program verification in process.
>	Prompt indicates program sequence completed.
>COPY B6	Copy MC68HC805B6 EEPROM contents.
>	Prompt indicates copy sequence completed.
>MD 100	Display memory locations starting at $100.

```
0100    55 55 55 77 55 55 55 55 55 55 55 55 55 55 55 55   ...............
>
```

3.6.5 Check

CHCK [<address1> <address2>] <device>

where:	<address1>	Beginning address of the memory to be checked.
	<address2>	Ending address of the memory to be checked.
	<device>	C4 for the MC68HC805C4 EEPROM MCU C8 for the MC68HC705C8 OTPROM/EPROM MCU B6 for the MC68HC805B6 EEPROM MCU

The CHCK command allows the user to be sure that the MCU internal OTPROM/EPROM/EEPROM (installed in programming sockets Z1, Z2, or Z3) is properly erased. Prior to entering this command, the user must follow the OTPROM/EPROM/EEPROM content checking procedure as described in paragraph 3.7.1. The CHCK command is now entered via the terminal keyboard to check the contents of the MCU internal OTPROM/EPROM/EEPROM. The results (blank (>) or not blank) are displayed on the terminal CRT. The user must be aware of memory map boundaries of the applicable MCU device when specifying start and end address locations.

EXAMPLES	DESCRIPTION
>CHCK C4	Check MC68HC805C4 EEPROM.
BLANK CHECKING	Blank checking all EEPROM locations.
>	Prompt indicates blank EEPROM.

NOTE

MC68HC705C8 EPROM UV window must be covered during checking.

>CHCK 100 3F5 C8	Check MC68HC705C8 OTPROM/EPROM.
BLANK CHECKING	Blank checking locations $0100 thru $03F5.
PART NOT BLANK	Message indicates OTPROM/EPROM not blank.
>	Prompt.

NOTE

MC68HC805B6 EEPROM checking takes approx. 30 seconds (E = 2 MHz).

>CHCK 800 9FF B6	Check MC68HC805B6 EEPROM.
BLANK CHECKING	Blank checking locations $0800 thru $09FF.
PART NOT BLANK	Message indicates EEPROM not blank.
>	Prompt.

3.6.6 Copy

COPY [<address1> <address2>] <device>

where:	<address1>	Beginning address of the memory to be copied.
	<address2>	Ending address of the memory to be copied.
	<device>	C4 for the MC68HC805C4 EEPROM MCU
		C8 for the MC68HC705C8 OTPROM/EPROM MCU
		B6 for the MC68HC805B6 EEPROM MCU

The COPY command allows the user to copy the contents of the programmed MCU internal OTPROM/EPROM/EEPROM (installed in programming sockets Z1, Z2, or Z3) into the EVM user map pseudo ROM. Prior to entering this command, the user must follow the OTPROM/EPROM/EEPROM copying procedure as described in paragraph 3.7.4. The COPY command is now entered via the terminal keyboard to enable the MCU OTPROM/EPROM/EEPROM contents to be copied into the EVM user map. The EVMbug prompt is displayed on the terminal CRT upon completion of the copying operation. The user must be aware of memory map boundaries of the applicable MCU device when specifying start and end address locations.

EXAMPLES **DESCRIPTION**

```
>COPY 20 4F C4                Copy MC68HC805C4 EEPROM contents.
>                             Prompt indicates copy sequence completed.
                              EEPROM locations $0020 thru $004F copied.
```

NOTE

MC68HC705C8 EPROM UV window must be covered during copying.

```
>COPY C8                      Copy MC68HC705C8 OTPROM/EPROM contents.
>                             Prompt indicates copy sequence completed.
                              All OTPROM/EPROM locations copied.
```

NOTE

MC68HC805B6 EEPROM copying takes approx. 30 seconds (E = 2 MHz).

```
>COPY 100 1EFF B6             Copy MC68HC805B6 EEPROM contents.
>                             Prompt indicates copy sequence completed.
                              EEPROM locations $0100 thru $1EFF copied.
```

3.6.7 Go

G [<address>]

where <address> is the starting address where program execution begins.

The G command allows the user to initiate user program execution (free run in real time). The user may optionally specify a starting address where execution is to begin. Execution starts at the current Program Counter (PC) address location, unless a starting address is specified. Program execution continues until a breakpoint is encountered, or the EVM ABORT switch S1 is activated (pressed), or the MASTER RESET switch S3 is pressed.

EXAMPLES	DESCRIPTION
>G	Go to user map and begin program execution at current PC address location.
>G 100	Go to user map and begin program execution at PC address location $100.
>G 100 Abort S=FF P=0104 A=55 X=FF C=E8 111.I...	Transfer of EVMbug monitor control. ABORT switch S1 is used to restore EVMbug monitor control if no breakpoints were preset prior to the G command entry.

3.6.8 Help

HELP

The HELP command enables the user available EVM command information to be displayed on the terminal CRT for quick reference purposes.

EXAMPLE

```
>HELP

BREAK = Abort command, CTRL-A = Exit transparent mode, CTRL-H = Backspace,
CTRL-S = Freeze screen, CTRL-X = Cancel command line
ASM <START ADDR>- Single line assembler/disassembler
BF <START ADDR> <END ADDR> <DATA>- Block fill memory
BR [<ADDR1 - ADDR5>]- Set 1 to 5 breakpoints
BULK <DEVICE>- Bulk erase MCU EEPROM
CHCK [<START ADDR> <END ADDR>] <device>- Blank check MCU OT/E/EEPROM
COPY [<START ADDR> <END ADDR>] <device>- Copy MCU OT/E/EEPROM to memory
G [<START ADDR>]- Execute user program
LOAD <PORT> [=<TEXT>]- Download (H)ost or (T)erminal port to memory
MD <START ADDR> [<END ADDR>]- Display memory
MM <ADDRESS>- Modify memory
NOBR [<ADDR1 - ADDR5>]- Remove breakpoints
P [<COUNT>]- Proceed 1-FF times through a breakpoint
PROG [<START ADDR> <END ADDR>] <device>- Program MCU OT/E/EEPROM from memory
RD- Register display
RM- Register modify
SPEED [<BAUD RATE>]- Display/select host baud rate
T [<COUNT>]- Trace 1-FF instructions
TM [<EXIT CHARACTER>]- Enter transparent mode
VERF [<START ADDR> <END ADDR>] <device>- Verify MCU OT/E/EEPROM to memory
>
```

3.6.9 Load

LOAD <port> [=<text>]

where:

<port> H for host port or T for terminal port.

<text> Text following the = sign is the host command sent to the port, which instructs the external host computer to download S-records to the EVM.

3

The LOAD command moves (downloads) object data in S-record format (see Appendix A) from an external host computer to EVM user pseudo ROM.

When downloading data from a host computer file, the data received by the EVM monitor is echoed to the terminal. If the terminal is running at a baud rate less than the host computer, the S-record echo may be scrambled but the data is entered correctly to EVM user pseudo ROM.

As the EVM monitor processes only valid S-record data, it is possible for the monitor to hang up during a load operation (there is no timeout because the terminal and host computer may be running at different baud rates).

If an S-record starting address points to an invalid memory location, the invalid address and the message "BAD MEMORY" is displayed on the terminal CRT.

EXAMPLES	DESCRIPTION
>LOAD T	LOAD command entered to download data from host computer (e.g., IBM-PC) to EVM via terminal port.
>LOAD H=COPY EXORT.LX,#CN	LOAD command entered to download data from EXORciser to EVM through host port. EXORT file with copy to terminal implemented.

Refer to the downloading procedures at the end of this chapter for additional information pertaining to the use of the LOAD command.

3.6.10 Memory Display

MD <address1> [<address2>]

where:

<address1>	Beginning address of the memory to be displayed.
<address2>	Ending address of the memory to be displayed.

The MD command is used to display a section of user memory beginning at address1 and continuing to address2. If address2 is not entered, 16 bytes are displayed beginning at address1. If address1 is greater than address2, the monitor prompt is displayed.

EXAMPLES

```
>MD 80 A0
0080    AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA   ................
0090    AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA   ................
00A0    AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA   ................
>


>MD 122
0122    AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA   ................
>
```

3.6.11 Memory Modify

MM <address>

where <address> is the memory location at which to start display/modify.

The MM command allows the user to examine/modify contents in user memory at specified locations in an interactive manner. Once entered, the MM command has several submodes of operation that allow modification and verification of data. The following terminators are recognized.

[<data>](RETURN)	Update location and sequence forward.
[<data>]^(RETURN)	Update location and sequence backward.
[<data>]=(RETURN)	Update location and reopen same location.
[<data>].(RETURN)	Update location and terminate.

An entry of only ".(RETURN)" terminates the memory modify interactive operation. (CTRL)X may be used to cancel any input line; the monitor remains in this command.

If an invalid address is specified, the invalid address and the message "BAD MEMORY" is displayed on the terminal CRT.

EXAMPLES	DESCRIPTION
>MM 100	Display memory location 100.
0100=00>AA=	Change data at 100 and reexamine location.
0100=AA>	
0101=11>44^	Change data at 101 and backup one location.
0100=AA>	
0101=44>33.	Change data at 101 and terminate MM operation.
>	
>MM 102	Display memory location 102.
0102=22>55	Do not change data at 103.
0103=AA>.	Terminate operation.
>	

3.6.12 Remove Breakpoint

NOBR [<address>]...

The NOBR command is used to remove one or more breakpoints from the internal breakpoint table. This command functions oppositely of the BR command. After removing the breakpoint, the current breakpoint address, if any, are displayed.

COMMAND FORMATS	DESCRIPTION
NOBR	Removes all current breakpoints.
NOBR <address>	Removes breakpoint.

EXAMPLES **DESCRIPTION**

```
>NOBR 200                          Removes breakpoint located at 200.
Brkpts=0321    0080    0420
>

>NOBR 321 80                       Removes breakpoints located at 321
Brkpts=0420                        and 80.
>

>NOBR                              Removes all breakpoints.
Brkpts=
>
```

3.6.13 Proceed

P [<count>]

where <count> is the number (in hexadecimal, $FF max.) of times the current breakpoint location is to be passed before the breakpoint returns control to the monitor. All other breakpoints are ignored during this command.

This command is ideal for applications where registers must be examined after a given number of passes within a software loop.

EXAMPLE	DESCRIPTION
>P 5	Current breakpoint location is passed five times before breakpoint returns control to the monitor.

3.6.14 Program

PROG [<address1> <address2>] <device>

where:

<address1>	Beginning address of the memory to be checked.
<address2>	Ending address of the memory to be checked.
<device>	<C4> for MC68HC805C4 EEPROM MCU <1C8> for MC68HC705C8 OTPROM/EPROM MCU (Mask set 0B67H.) <C8> for MC68HC705C8 OTPROM/EPROM MCU (Mask set 1B67H, B44S, and future mask sets.) <B6> for MC68HC805B6 EEPROM MCU

The PROG command is used to program MCU internal OTPROM/EPROM/EEPROM installed in programming sockets Z1, Z2, or Z3.

CAUTION

Do not attempt to program a mask programmed MCU device (e.g., MC68HC05C4 or MC68HC05C8). Damage will occur to the device.

Prior to entering this command, the user must follow the OTPROM/EPROM/EEPROM programming procedure as described in paragraph 3.7.3.

The PROG command is now entered via the terminal keyboard to program the MCU internal OTPROM/EPROM/EEPROM. The EVM checks the OTPROM/EPROM/EEPROM contents. If any OTPROM/EPROM/EEPROM locations are not empty, the EVM monitor will prompt the user with a message to either continue the programming sequence or exit the PROG command. A (RETURN) entry will continue with the programming operation, while any other character will exit the PROG command. This allows changes to be made to already programmed devices. During the programming operation, the terminal CRT display is updated with each address being programmed. The user must be aware of memory map boundaries of the applicable MCU device when specifying start and end address locations.

NOTE

Excessive programming the same location will cause data degradation if the location is not erased first.

EXAMPLES	DESCRIPTION
>PROG 20 7FF C4	Program MC68HC805C4 EEPROM.
BLANK CHECKING	Blank checking locations $0020 thru $07FF.
XXXX	Programming locations $0020 thru $07FF.
VERIFYING	Program verification in process.
>	Prompt indicates program sequence completed.
>PROG 20 7FF C4	Program MC68HC805C4 EEPROM.
BLANK CHECKING	Programming locations $0020 thru $07FF.
XXXX	EEPROM address location XXXX updated.
VERIFYING	Program verification in process.
DOES NOT VERIFY	Program sequence did not verify correctly.
>	Prompt indicates program sequence completed.

3

NOTE

MC68HC705C8 EPROM UV window must be covered during programming.

>PROG 200 3FF C8	Program MC68HC705C8 OTPROM/EPROM.
BLANK CHECKING	Blank checking locations $0200 thru $03FF.
XXXX	Programming locations $0200 thru $03FF.
XXXX	Reprogramming locations $0200 thru $03FF.
VERIFYING	Program verification in process message.
>	Prompt indicates program sequence completed.
>PROG 200 3FF C8	Program MC68HC705C8 OTPROM/EPROM.
BLANK CHECKING	Blank checking locations $0200 thru $03FF.
PART NOT BLANK	Address locations not blank.
HIT RETURN TO PROGRAM	Initiate programming sequence.
XXXX	OTPROM/EPROM address location XXXX updated.
XXXX	Reprogramming locations $0200 thru $03FF.
VERIFYING	Program verification in process message.
>	Prompt indicates program sequence completed.

NOTE

MC68HC805B6 EEPROM checking takes approx. 30 seconds (E = 2 MHz).
MC68HC805B6 EEPROM verifying takes approx. 30 seconds (E = 2 MHz).

>PROG 900 A76 B6	Program MC68HC805B6 EEPROM.
BLANK CHECKING	Blank checking locations $0900 thru $0A76.
XXXX	Programming locations $0900 thru $0A76.
VERIFYING	Program verification in process.
>	Prompt indicates program sequence completed.
>PROG 900 A76 B6	Program MC68HC805B6 EEPROM.
BLANK CHECKING	Programming locations $0900 thru $0A76.
XXXX	EEPROM address location XXXX updated.
VERIFYING	Program verification in process.
DOES NOT VERIFY	Program sequence did not verify correctly.
>	Prompt indicates program sequence completed.

NOTE

MC68HC705C8 MCU secure EPROM feature is performed after a successful EPROM program and verify operation.

The secure EPROM feature is performed by programming the MC68HC705C8 MCU option register ($1FDF) security (SEC) bit. This is accomplished by memory modifying pseudo ROM address location $1FDF (bit 3 = logic 1), reprogramming MCU option register $1FDF, and verifying contents.

The following example shows how to program the MC68HC705C8 option register security bit.

NOTES

<device> = <1C8> for MC68HC705C8 mask set 0B67H.

<device> = <C8> for MC68HC705C8 mask set 1B67H, B44S, and future mask sets.

MC68HC705C8 EPROM UV window must be covered during all operations.

EXAMPLES	DESCRIPTION
>MM 1FDF	Memory modify pseudo ROM location $1FDF.
1FDF=XX>08.	Set bit 3 to logic one (1).
>PROG 1FDF 1FDF C8 or 1C8	Program MC68HC705C8 option register $1FDF.
BLANK CHECKING	Blank checking location $1FDF.
1FDF	Programming location $1FDF.
VERIFYING	Program verification in process message.
	Program verification halts.

(PRESS MASTER RESET SWITCH S3 TO RELEASE FROM VERIFICATION MODE.)

>	Prompt returns after pressing master reset.

3.6.15 Register Display

RD

The RD command displays the MCU register contents.

COMMAND FORMAT

>RD

DESCRIPTION

Contents of the following registers are displayed:

S	=	Stack pointer
P	=	Program counter
A	=	Accumulator A
X	=	Index register
C	=	Condition codes

EXAMPLE

```
>RD
Regs
S=FF     P=0102     A=31     X=FF     C=F9          111HI..C
>
```

DESCRIPTION

Condition code register (CCR) 111HINZC bits are displayed as follows:

All CCR bits set (logic 1)	C=FF	111HINZC
All CCR bits cleared (logic 0)	C=00	111.....
Specific CCR bit set (I)	C=10	111.I...
Specific CCR bits set (H, I, and Z)	C=34	111HI.Z.

3.6.16 Register Modify

RM

The RM command is used to modify the MCU registers contents. The RM command takes no parameters and starts by displaying the S (stack pointer) register contents and allowing changes to be made. The order of the registers displayed are as follows:

S	(stack pointer)
P	(program counter)
A	(accumulator A)
X	(index register)
C	(condition code)

Once entered, the RM command has several submodes of operation that allow modification and verification of data. The following terminators are recognized.

[<data>](RETURN)	Update register and sequence forward.
[<data>]^(RETURN)	Update register and sequence backward.
[<data>]=(RETURN)	Update register and reopen same location.
[<data>].(RETURN)	Update register and terminate.

An entry of only ".(RETURN)" terminates the register modify interactive mode. (CTRL)X may be used to cancel any input line; the monitor remains in this command. The stack pointer is not user modifiable.

EXAMPLES	DESCRIPTION
>RM	Command entered.
S=FF	
P=0102>100	Change P register and go to A register.
A=31>AA	Change A register.
X=FF>	Examine X register.
C=F9>.	Examine C register and terminate command.
>	

3.6.17 Speed

SPEED <baud rate>

where <baud rate> is the number 300, 600, 1200, 2400, 4800, 9600, or 19.2K.

The monitor configures the baud rate of the host I/O port to 9600 baud upon reset. The SPEED command only allows the user to reassign or modify the baud rate operating speed of the EVM host I/O port for specific applications.

NOTE

When entering the 19.2K baud rate, the decimal point is not required for the command entry. This baud rate command is entered as SPEED 192.

EXAMPLES	DESCRIPTION
>SPEED 9600 >	Displays current host I/O port baud rate.
>SPEED 192 >	Reassign or modify EVM host I/O port to operate at the baud rate speed of 19,200 (19.2K) baud. Note that decimal point is not entered during the command entry.

3.6.18 Trace

T [<count>]

where <count> is the number (in hexadecimal, $FF max.) of instructions to execute.

The T command allows the user to monitor program execution on an instruction-by-instruction basis. The user may optionally execute several instructions at a time by entering a count value (up to $FF). Execution starts at the current PC. The PC displayed with the event message is of the next instruction to be executed. During the tracing operation, breakpoints are active and the user program execution stops upon the PC encountering a breakpoint address.

The user should not try to trace an instruction that branches to itself (e.g., BRA). Because the monitor places an SWI instruction on the object of the branch, the instruction would never be executed. However, it would look to the user as if the instruction executed. The user may enter a G command while the PC points to this type of instruction as long as the instruction is not a breakpoint address.

The monitor issues an "ILLEGAL/INSUFFICIENT ENTRY" message if the user attempts to trace at an address that contains an invalid opcode.

SINGLE TRACE EXAMPLE

```
>T
0101  4C         INCA
S=FF    P=0101    A=00    X=FF    C=EA         111.I.Z.
>
```

MULTIPLE TRACE EXAMPLE

```
>T 2
0102  9D         NOP
S=FF    P=0102    A=01    X=FF    C=E8         111.I...
0103  20 FC      BRA   $0101
S=FF    P=0103    A=01    X=FF    C=E8         111.I...
>
```

3.6.19 Transparent Mode

TM [<exit character>]

where <exit character> is the user entry to terminate the transparent mode.

The TM command connects the EVM host port to the terminal port, which allows direct communication between the terminal and the host computer. All I/O between the ports are ignored by the EVM until the exit character is entered from the terminal.

3

When the TM command is entered, an exit character is entered following the TM command (e.g., >TM X). The exit character (X) can be any keyboard character (printable or non-printable). The default exit command is (CTRL)A. When the user task is completed, the transparent mode is terminated by entering the same exit character.

EXAMPLES	DESCRIPTION
>TM (RETURN) MDOS =DIR . . . =(CTRL)A >	Command followed by two carriage returns. (CTRL)A is default exit command. MDOS program entered. Task complete. Enter exit command (CTRL)A. TM command terminated.
>TM X (RETURN) MDOS =DIR . . . =X >	Command followed by exit character (X). MDOS program entered. Directory called up. Task complete. Enter exit character (X). TM command terminated.

Refer to the downloading procedures at the end of this chapter for additional information pertaining to the use of the TM command.

3.6.20 Verify

VERF [<address1> <address2>] <device>

where:	<address1>	Beginning address of the memory to be checked.
	<address2>	Ending address of the memory to be checked.
	<device>	C4 for the MC68HC805C4 EEPROM MCU C8 for the MC68HC705C8 OTPROM/EPROM MCU B6 for the MC68HC805B6 EEPROM MCU

The VERF command allows the user to verify the contents of the programmed MCU internal OTPROM/EPROM/EEPROM (installed in programming sockets Z1, Z2, or Z3) against the EVM user pseudo ROM. Prior to entering this command, the user must follow the OTPROM/EPROM/EEPROM verifying procedure as described in paragraph 3.7.5. The VERF command is now entered via the terminal keyboard to enable the MCU OTPROM/EPROM/EEPROM contents to be verified with the contents of the user pseudo ROM. If any differences between the MCU OTPROM/EPROM/EEPROM and the user pseudo ROM exist, an error message will be displayed on the terminal CRT. The user must be aware of memory map boundaries of the applicable MCU device when specifying start and end address locations.

3

EXAMPLES **DESCRIPTION**

```
>VERF C4                    Verify MC68HC805C4 EEPROM.
VERIFYING                   Verifying all EEPROM locations.
>                           Prompt indicates verified EEPROM.
```

NOTE

MC68HC705C8 EPROM UV window must be covered during verifying.

```
>VERF 500 1FFF C8           Verify MC68HC705C8 EEPROM.
VERIFYING                   Verifying locations $0500 thru $1FFF.
>                           Prompt indicates verified OTPROM/EPROM.
```

NOTE

MC68HC805B6 EEPROM verifying takes approx. 30 seconds (E = 2 MHz).

```
>VERF 945 A23 B6            Verify MC68HC805B6 EEPROM.
VERIFYING                   Verifying locations $945 thru $A23.
DOES NOT VERIFY             Verify sequence did not verify correctly.
>                           Prompt indicates program sequence completed.
```

3.7 MCU PROGRAMMING PROCEDURES

The EVM contains programming circuitry that enable the user to check, bulk erase, program, copy, and verify MC68HC805C4/MC68HC805B6 MCU internal EEPROM. The EVM can also check, program, copy, and verify MC68HC705C8 MCU internal OTPROM/EPROM. The EVM is factory-configured and tested for MC68HC705C8, MC68HC805C4, and MC68HC805B6 MCU programming operations.

MC68HC705C8 OTPROM MCU devices are shipped in an erased state and cannot be erased. Electrical erasing operations cannot be performed on OTPROM MCU devices.

MC68HC705C8 EPROM MCU devices are erased by the exposure of a high-intensity ultraviolet (UV) light with a wavelength of 2537 Angstrom (A). The recommended dose (UV intensity x exposure time) is 15 Ws/cm2. UV lamps should be used without shortwave filters, and the EPROM MCU device positioned about one inch from the UV lamps.

NOTE

MC68HC705C8 EPROM MCU UV erase window must be covered at all times (except during erasing) to avoid erratic MCU operation.

When performing programming procedures the user must be aware of the following general warnings/limitations (notes) and cautions:

NOTES

The EVM must be disconnected from target system equipment when performing programming procedures.

Excessive programming in the same location will cause data degradation if the location is not erased first.

CAUTIONS

PLCC type MCU devices must be inserted upside down in surface mount sockets. Incorrect MCU insertion will cause MCU damage.

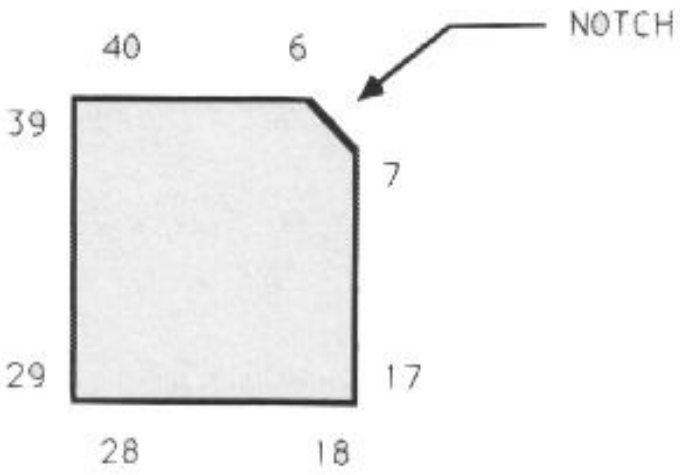
Forcing the MCU into the surface mount socket can result in damage to the socket. Surface mount sockets are fragile.

Only one MCU must be installed on the EVM for a specific programming operation. Multiple MCUs will cause MCU damage and/or erratic EVM operation.

When programming a specific MCU (e.g., MC68HC805C4), the applicable resident MCU (e.g., MC68HC05C4) must be installed. (When programming a MC68HC805B6 MCU, the applicable resident MC68HC05B6 MCU must be installed.) Programming socket locations are identified for the specific devices, packages, and memory types as follows:

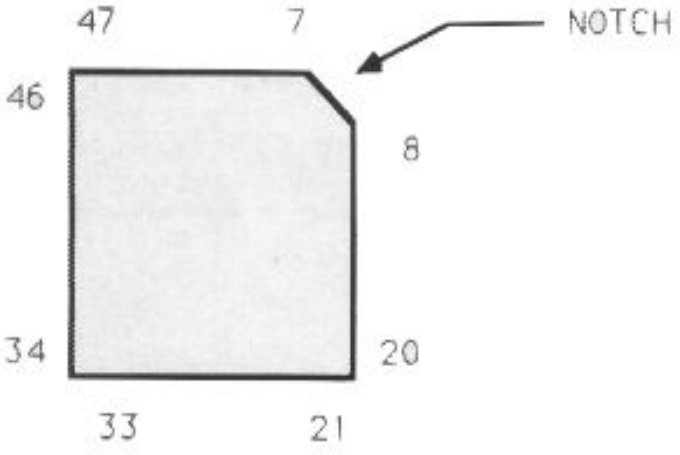
DEVICE TYPE	DEVICE PACKAGE	MEMORY TYPE	PROGRAMMING SOCKET
MC68HC705C8	40-pin DIL	OTPROM/ EPROM	Z3
MC68HC705C8	44-lead PLCC	OTPROM	Z2
MC68HC805B6	52-lead PLCC	EEPROM	Z1
MC68HC805C4	40-pin DIL	EEPROM	Z3
MC68HC805C4	44-lead PLCC	EEPROM	Z2

When programming the 44-lead PLCC MC68HC705C8 OTPROM or MC68HC805C4 EEPROM MCU, the device must be inserted upside down (horizontal flip) into programming socket Z2 as shown below. Inspect socket locations Z1, Z3, U53, and U54. Remove MCUs (if installed).



44-lead PLCC MC68HC705C8/MC68HC805C4 MCU (Z2) Installation

When programming the 52-lead PLCC MC68HC805B6 EEPROM MCU, the device must be inserted upside down (horizontal flip) in programming socket Z1 as shown below. Inspect socket locations Z2, Z3, U44, and U54. Remove MCUs (if installed).



52-lead PLCC MC68HC805B6 MCU (Z1) Installation

The MC68HC705C8 OTPROM/EPROM MCU does not contain an internal charge pump. When performing the following programming procedures, an external VPP power source (shown below) is connected to the EVM power connector J28 pin 1 (labeled VPP).

MC68HC705C8	(Mask 0B67H)	+14.00 Vdc +/- 0.10 Vdc @ 10 mA
	(Mask 1B67H)	+14.75 Vdc +/- 0.25 Vdc @ 10 mA
	(Mask B44S)	+14.75 Vdc +/- 0.25 Vdc @ 10 mA
	(Future Masks)	+14.75 Vdc +/- 0.25 Vdc @ 10 mA

CAUTION

When external VPP power is connected to the EVM power connector J28 pin 1, VPP must be controlled by the EVM programmer VPP switch S5.

The user must perform the programming procedures as specified in paragraphs 3.7.1 through 3.7.5. Failure to control VPP power via switch S5 will cause damage to the OTPROM/EPROM MCU device.

NOTE

Refer to paragraphs 2.3.3 and 2.3.4 for J16 and J21 jumper header reconfiguration information when configuring the EVM for the program mode of operation.

When programming the MC68HC705C8 MCU, reinstall J16 fabricated jumper between pins 2 and 3, J22 fabricated jumper between pins 1 and 2, and J24 fabricated jumper between pins 3 and 5.

The MC68HC805C4 EEPROM MCU internal charge pump is not used when performing the following programming procedures. An external VPP power source (+19.0 Vdc @ 100 uA) is connected to the EVM power connector J28 pin 1 (labeled VPP).

The MC68HC805B6 EEPROM MCU internal charge pump is used when programming the 256 bytes of byte erasable EEPROM (EEPROM1). The MC68HC805B6 EEPROM MCU internal charge pump is not used when programming the 5,952 bytes of bulk erasable EEPROM (EEPROM6). An external VPP power source (shown below) is connected to the EVM power connector J28 pin 1 (labeled VPP).

MC68HC805B6	(BULK erase)	+19 Vdc @ 2 mA
	(Program)	+19 Vdc @ 100 uA

The following paragraphs describe the MCU programming procedures. Once any of the commands (BULK, CHCK, COPY, PROG, or VERF) are initiated, the user need not repeat the initial procedural steps involving switches S3, S4, and S5 manipulation. For example, upon completion of steps (a) through (f) of the erasing procedure, programming may begin by starting with step (g) of the programming procedure.

3.7.1 Checking *(MC68HC705C8/MC68HC805C4/MC68HC805B6)*

OTPROM/EPROM/EEPROM content checking is accomplished by the use of the CHCK command. This command allows the user to be sure that the MCU internal OTPROM/EPROM/EEPROM is properly erased. To perform the content checking procedure, perform the following:

a. Place programmer VPP switch S5 to the OFF position.

b. Place programmer power switch S4 to the OFF position.

c. Press MASTER RESET switch S3.

CAUTION

PLCC type MCU devices must be inserted upside down in surface mount sockets. Incorrect MCU insertion will cause MCU damage.

NOTE

MC68HC705C8 EPROM MCU UV erase window must be covered at all times (except during erasing) to avoid erratic MCU operation.

d. Insert MCU device into applicable programming socket Z1, Z2, or Z3.

e. Place programmer power switch S4 to the ON position.

f. Place programmer VPP switch S5 to the ON position.

g. Enter CHCK command via terminal keyboard to check contents of the MCU internal OTPROM/EPROM/EEPROM. OTPROM/EPROM/EEPROM contents are checked and the results (blank or not blank) are displayed on the terminal CRT.

Wait until EVMbug prompt is displayed on the terminal CRT.

h. Press MASTER RESET switch S3.

i. Place programmer VPP switch S5 to the OFF position.

j. Place programmer power switch S4 to the OFF position.

k. Remove MCU device from programming socket.

3.7.2 Erasing *(MC68HC805C4/MC68HC805B6 ONLY)*

EEPROM erasing is accomplished by the use of the BULK command. This command allows the user to erase all memory array locations in the MCU internal EEPROM. To perform the EEPROM erasing procedure, perform the following:

a. Place programmer VPP switch S5 to the OFF position.

b. Place programmer power switch S4 to the OFF position.

c. Press MASTER RESET switch S3.

CAUTION

PLCC type MCU devices must be inserted upside down in surface mount sockets. Incorrect MCU insertion will cause MCU damage.

d. Insert EEPROM MCU device into applicable programming socket Z1, Z2, or Z3.

e. Place programmer power switch S4 to the ON position.

f. Place programmer VPP switch S5 to the ON position.

g. Enter BULK erase command via terminal keyboard to erase all EEPROM contents.

Wait until EVMbug prompt is displayed on the terminal CRT.

h. Press MASTER RESET switch S3.

i. Place programmer VPP switch S5 to the OFF position.

j. Place programmer power switch S4 to the OFF position.

k. Remove MCU device from programming socket.

3.7.3 Programming *(MC68HC705C8/MC68HC805C4/MC68HC805B6)*

OTPROM/EPROM/EEPROM programming is accomplished by the use of the PROG command. This command allows the user to program the MCU internal OTPROM/EPROM/EEPROM. To perform the OTPROM/EPROM/EEPROM programming procedure, perform the following:

a. Place programmer VPP switch S5 to the OFF position.

b. Place programmer power switch S4 to the OFF position.

c. Press MASTER RESET switch S3.

CAUTION

PLCC type MCU devices must be inserted upside down in surface mount sockets. Incorrect MCU insertion will cause MCU damage.

NOTE

MC68HC705C8 EPROM MCU UV erase window must be covered at all times (except during erasing) to avoid erratic MCU operation.

d. Insert OTPROM/EPROM/EEPROM MCU device into applicable programming socket Z1, Z2, or Z3.

e. Place programmer power switch S4 to the ON position.

f. Place programmer VPP switch S5 to the ON position.

g. Enter PROG command via terminal keyboard. If any OTPROM/EPROM/EEPROM locations are not empty, EVM monitor will prompt the user with a message to either continue the programming sequence or exit the PROG command.

A (RETURN) entry will continue with the programming operation, while any other character will exit the PROG command. This allows changes to be made to already programmed devices. During the programming operation, the terminal CRT display is updated with each address being programmed. Upon completion of the programming sequence, the sequence is automatically verified and the status is displayed on the terminal CRT.

Wait until EVMbug prompt is displayed on the terminal CRT.

h. Press MASTER RESET switch S3.

i. Place programmer VPP switch S5 to the OFF position.

j. Place programmer power switch S4 to the OFF position.

k. Remove MCU device from programming socket.

3.7.4 Copying *(MC68HC705C8/MC68HC805C4/MC68HC805B6)*

OTPROM/EPROM/EEPROM content copying is accomplished by the use of the COPY command. This command allows the user to copy the contents of the programmed MCU internal OTPROM/EPROM/EEPROM into the EVM user pseudo ROM. To perform the OTPROM/EPROM/EEPROM copying procedure, perform the following:

a. Place programmer VPP switch S5 to the OFF position.

b. Place programmer power switch S4 to the OFF position.

c. Press MASTER RESET switch S3.

CAUTION

PLCC type MCU devices must be inserted upside down in surface mount sockets. Incorrect MCU insertion will cause MCU damage.

NOTE

MC68HC705C8 EPROM MCU UV erase window must be covered at all times (except during erasing) to avoid erratic MCU operation.

d. Insert OTPROM/EPROM/EEPROM MCU device into applicable programming socket Z1, Z2, or Z3.

e. Place programmer power switch S4 to the ON position.

f. Place programmer VPP switch S5 to the ON position.

g. Enter COPY command via terminal keyboard to copy the contents of the MCU OTPROM/EPROM/EEPROM into EVM user pseudo ROM.

Wait until EVMbug prompt is displayed on the terminal CRT.

h. Press MASTER RESET switch S3.

i. Place programmer VPP switch S5 to the OFF position.

j. Place programmer power switch S4 to the OFF position.

k. Remove MCU device from programming socket.

3.7.5 Verifying *(MC68HC705C8/MC68HC805C4/MC68HC805B6)*

OTPROM/EPROM/EEPROM content verification is accomplished by the use of the VERF command. This command allows the user to verify the contents of the programmed MCU internal OTPROM/EPROM/EEPROM against the EVM user pseudo ROM. To perform the OTPROM/EPROM/EEPROM verification procedure, perform the following:

a. Place programmer VPP switch S5 to the OFF position.

b. Place programmer power switch S4 to the OFF position.

c. Press MASTER RESET switch S3.

CAUTION

PLCC type MCU devices must be inserted upside down in surface mount sockets. Incorrect MCU insertion will cause MCU damage.

NOTE

MC68HC705C8 EPROM MCU UV erase window must be covered at all times (except during erasing) to avoid erratic MCU operation.

d. Insert OTPROM/EPROM/EEPROM MCU device into applicable programming socket Z1, Z2, or Z3.

e. Place programmer power switch S4 to the ON position.

f. Place programmer VPP switch S5 to the ON position.

g. Enter VERF command via terminal keyboard to verify the contents of the MCU OTPROM/EPROM/EEPROM to the EVM user pseudo ROM. If any differences between the MCU OTPROM/EPROM/EEPROM and the user pseudo ROM exist, an error message will be displayed on the terminal CRT.

Wait until EVMbug prompt is displayed on the terminal CRT.

h. Press MASTER RESET switch S3.

i. Place programmer VPP switch S5 to the OFF position.

j. Place programmer power switch S4 to the OFF position.

k. Remove MCU device from programming socket.

3.8 ASSEMBLING/DISASSEMBLING PROCEDURES

The assembler/disassembler is an interactive assembler/editor in which the source program is not saved. Each source line is converted into the proper machine language code and is stored in memory on a line-by-line basis at the time of entry. In order to display an instruction, the machine code is disassembled and the instruction mnemonic and operands are displayed. All valid opcodes are converted to assembly language mnemonic. All invalid opcodes return a Form Constant Byte (FCB) conversion.

The ASM command allows the user to create, modify, and debug MC6805 MCU code. Assembler input must have exactly one space between the mnemonic and the operand. There must be no space between the operand and the index specification (,X) except in the case of indexed no offset. Assembler input must be terminated by a carriage return. No comments, etc., are allowed after the instruction input.

After each new assembler input line, the new line is disassembled for the user before stepping to the new instruction. The new line may assemble to a different number of bytes than the previous one.

For Branch if High or Same (BHS)/Branch if Carry Clear (BCC) mnemonics, disassembly displays the BCC mnemonic. For Branch if Lower (BLO)/Branch if Carry Set (BCS) mnemonics, disassembly displays the BCS mnemonic.

Branch address offsets are automatically calculated by the assembler, thus the address is inputted as the operand rather than an offset value.

The assembler is terminated by entering a period (.) followed by a carriage return as the only entry on the command input line. Entering a carriage return alone on an input line steps to the next instruction.

Entering (CTRL)X cancels an input line. The monitor remains in the assembler mode. If an invalid address is specified, the invalid address and the message "BAD MEMORY" is displayed on the terminal CRT.

The following pages describe how to operate the assembler/disassembler by creating a typical program loop, and debugging the program by the use of the EVM monitor commands. A typical program loop is first assembled. Routine examples are then provided to illustrate how to perform breakpoint setting, proceeding from breakpoint, register display and modification, and initiation of user program execution.

A program loop is assembled as follows:

EXAMPLE PROGRAM						PROGRAM DESCRIPTION
>ASM 100						*Enter assembler mode.*
0100	B7 B4		STA	$B4	>CLRA	*Clear inner-loop counter.*
0100	4F		CLRA			
0101	B4 24		AND	$24	>CLRX	*Clear outer-loop counter.*
0101	5F		CLRX			
0102	24 97		BCC	$009B	>INCA	*Increment inner-loop counter.*
0102	4C		INCA			
0103	97		TAX		>BNE 102	*Wait for counter overflow.*
0103	26 FD		BNE	$0102		
0105	9F		TXA		>INCX	*Increment outer-loop counter.*
0105	5C		INCX			
0106	AF		FCB	$AF	>BNE 102	*Wait for counter overflow.*
0106	26 FA		BNE	$0102		
0108	DF 00	80	STX	$0080,X	>BRA 100	*Go to program start.*
0108	20 F6		BRA	$0100		
010A	80		RTI		>.	*Exit assembler mode.*

The following routines are performed on the program loop just assembled:

TERMINAL CRT/KEYBOARD	ROUTINE DESCRIPTION
>RD	*Register display user machine state.*
S=FF P=0FF8 A=60 X=FF C=E8 111.I...	
>	
>RM	*Modify program counter register.*
S=FF	
P=0FF8>100=	
P=0100>(RETURN)	
A=60>_	
>BR 103 106	*Set breakpoints.*
Brkpts=0103 0106	
>G	*Begin execution of program.*
Brkpt	
S=FF P=0103 A=01 X=00 C=E8 111.I...	
>	
>P 45	*Proceed 45 times within loop.*
Brkpt	
S=FF P=0103 A=46 X=00 C=E8 111.I...	
>	
>NOBR 103	*Remove breakpoint.*
Brkpts=0106	
>G	*Continue program execution.*
Brkpts	
S=FF P=0106 A=00 X=01 C=E8 111.I...	
>	
>T 2	*Monitor program execution.*
0102 4C INCA	
S=FF P=0102 A=00 X=01 C=E8 111.I...	
0103 26 FD BNE $0102	
S=FF P=0103 A=01 X=01 C=E8 111.I...	
>	

3.9 DOWNLOADING PROCEDURES

This portion of text describes the EVM downloading procedures. The downloading operation enables the user to transfer information from a host computer to the EVM using the LOAD command.

Specific downloading procedures are described enabling the user to perform downloading operations with an EXORciser and IBM Personal Computer (PC) host computer systems. EXORciser downloading operation is accomplished utilizing the TM and LOAD commands. The TM (Transparent Mode) command connects the EVM host port to the terminal port, which allows direct communication between the terminal and host computer. All I/O between the ports are ignored by the EVM until the exit command (CTRL)A is entered from the terminal. The LOAD command moves data information in S-record format (see Appendix A) from an external host computer to the EVM user pseudo ROM. When moving data to the EVM, the same data transmitted through the host port is also echoed to the terminal port.

Stopping a host I/O port downloading operation already in progress is accomplished by depressing any alphanumeric key on the terminal keyboard. If an incorrect keyboard entry is made during a downloading procedure which causes a terminal lockup condition, this lockup condition is removed by depressing any alphanumeric keyboard key.

The transparent mode of operation is not applicable to the IBM-PC to EVM operation. Therefore the TM command is not utilized in the IBM-PC downloading procedure.

The following pages provide examples and descriptions of how to perform EVM downloading operations in conjunction with an EXORciser and IBM-PC host computer systems.

NOTE

For MC68HC705C8 MCU downloading operations only, reinstall J16 fabricated jumper between pins 2 and 3, J22 fabricated jumper between pins 3 and 5, and J24 fabricated jumper between pins 1 and 2.

3.9.1 EXORciser to EVM

To perform the EXORciser to EVM downloading procedure, perform/observe the following:

EXAMPLES	DESCRIPTION
>TM (RETURN) EXBUG09 2.1 *E MDOS MDOS09 3.05 =(CTRL)A >LOAD H=COPY EXORT.LX,#CN	*EXORciser initialized into MDOS via TM command to download S-records.* *Exit transparent mode.* *LOAD command entered to download data to EVM through host port.* *EXORT file with copy to terminal implemented.*

```
>LOAD H=COPY EXORT.LX,#CN
COPY EXORT.LX,#CN
S0030000FC
S110001F424547204C4F414420484552457E
S11101002431303020574494C4C204C4F4144A0
S11102002432303020574494C4C204C4F41449E
S11103002433303020574494C4C204C4F41449C
S11104002434303020574494C4C204C4F41449A
S11105002435303020574494C4C204C4F414498
S11106002435303020574494C4C204C4F414496
S11107002435303020574494C4C204C4F414494
S9030000FC
>
```

LOAD command entered to download data. Host port will display data as transferred.

3.9.2 IBM-PC to EVM

Prior to performing any IBM-PC operation, ensure that both IBM-PC and EVM baud rates are identical, and that the IBM-PC asynchronous port is configured for terminal mode of operation. If the asynchronous port is hard wired for host mode of operation and cannot be reconfigured for a terminal mode of operation, the use a null modem (transmit (TxD) and receive (RxD) and associated handshake lines are cross coupled) is required.

NOTE

IBM-PC to EVM interconnection is accomplished by a single RS-232C cable assembly. This cable is connected to the EVM terminal I/O port connector J1 for downloading operations.

To perform the IBM-PC to EVM downloading procedure, perform/observe the following:

EXAMPLE	DESCRIPTION
C>KERMIT	*IBM-PC prompt. Enter Kermit program.*
IBM-PC Kermit-MS VX.XX	
Type ? for help	
Kermit-MS>SET BAUD 9600	*Set IBM-PC baud rate.*
Kermit-MS>CONNECT	*Connect IBM-PC to EVM.*
[Connecting to host, type Control-] C to return to PC]	
(RETURN)	
>LOAD T	*EVM download command (via terminal port) entered.*
(CTRL)]C	
Kermit-MS>PUSH	
The IBM Personal Computer DOS	
Version X.XX (C)Copyright IBM Corp 1981, 1982, 1983	
C>TYPE (File Name) > COM1	*Motorola S-record file name.*
C>EXIT	*S-record downloading completed.*
Kermit-MS>CONNECT	*Return to EVM monitor program.*
>(CTRL)]C	
Kermit-MS>EXIT	*Exit Kermit program.*

CHAPTER 4

FUNCTIONAL DESCRIPTION

4.1 INTRODUCTION

This chapter provides an overall description of the EVM. This description is supported by a block diagram (Figure 4-1) that illustrates the interconnection of the EVM circuits and I/O ports, and a memory map diagram (Figure 4-2).

4.2 EVM DESCRIPTION

The EVM is designed to evaluate an MC68HC05 MCU based target system via the resident MC68HC05XX MCU. The EVM contains two memory maps (monitor or user map) that are switchable (transparent to the user) to allow modification of user memory and execution of user programs. Data transfer within the EVM is controlled by the monitor ROM firmware. This firmware is controlled from an external RS-232C compatible terminal.

Figure 4-1 illustrates the EVM block diagram. Basically, the EVM consists of the following functional circuits:

a. MCU and control

b. Monitor and user memory

c. Terminal and host computer I/O ports

d. MCU extension I/O ports

e. OTPROM/EPROM/EEPROM MCU programmer

4.2.1 MCU and Control Circuits

The EVM simulates the single chip mode of operation. The M68HC05 Family of HCMOS MCU devices are evaluated (emulated) by the EVM resident MC68HC05xx MCU. Several types of resident MCUs can be used. The EVM is shipped with an MC68HC05C9 resident MCU device. This device will evaluate MC68HC05C2/C3/C4/C8/C9, HC805C4, HCL05C4/C8, HSC05C4/C8, and HC05P1/P7 MCUs.

For other MC68HC05 MCU devices (e.g., MC68HC05A6, HC705C8, HC05B4/B6, HC805B6, and HC05L6), the resident MC68HC05C9 MCU is replaced by the specific MCU device required for evaluation. Refer to paragraph 2.4.2.

The EVM cannot emulate the low power capabilities of the MC68HCL05C4/C8. The EVM can emulate the high speed capabilities of the MC68HSC05C4/C8 by the replacement of the 8 MHz crystal (Y1) to a 16 MHz crystal. Refer to paragraph 2.3.10.

The EVM contains an MC68HC05XX MCU and associated control circuits that provide the basic evaluation capabilities for target system use. As shown in Figure 4-1, specific control circuits are implemented into the EVM, and are as follows:

a. Map switching

b. Abort

c. Address decoding

4.2.1.1 Map Switching. The EVM operates in either one of two memory maps (monitor or user map) as illustrated in Figure 4-2. Two types of memory map switching (temporary or permanent) can be performed. Temporary map switching allows modification of user memory, and permanent map switching allows execution of user programs.

Temporary map switching is used to modify user pseudo ROM (RAM). The opcode and operand are fetched from the monitor map, memory maps are switched for one cycle, and the read or write cycle is executed in user space. Memory maps are then automatically switched back to the monitor map on the next cycle.

Permanent map switching is initiated by a command from the monitor, or by the USER RESET switch S2. The return from interrupt (RTI) opcode is fetched from the monitor map, memory maps are switched, and the user register contents are fetched from the user stack. The Program Counter (PC) content is fetched from the user stack and execution proceeds from the current PC value.

When the USER RESET switch S2 is activated (pressed), the MCU and user I/O ports are reset, memory maps are switched to the user map, and the reset vector is fetched from the user map by the MCU. Breakpoints are ignored during this operation.

Execution of user code continues until a software interrupt (SWI) is decoded on the data bus during a Load Instruction Register (LIR) cycle. SWI occurs when either a breakpoint is detected, or the ABORT switch S1 is activated. The memory map is switched back to the monitor map after the user register contents are saved on the user stack. An SWI which occurs when no breakpoint is set -- or when the ABORT switch is not activated -- does not cause memory map switching to take place. This allows user SWIs to be executed in real time.

The ABORT switch, when activated, forces an SWI on the data bus during the LIR cycle. The user register contents are saved on the user stack, and memory maps are switched to the monitor map. The ABORT switch has no effect on the monitor map. When the ABORT switch is activated while in the user map, the monitor map is re-entered (assuming the MCU is operating properly in the user map).

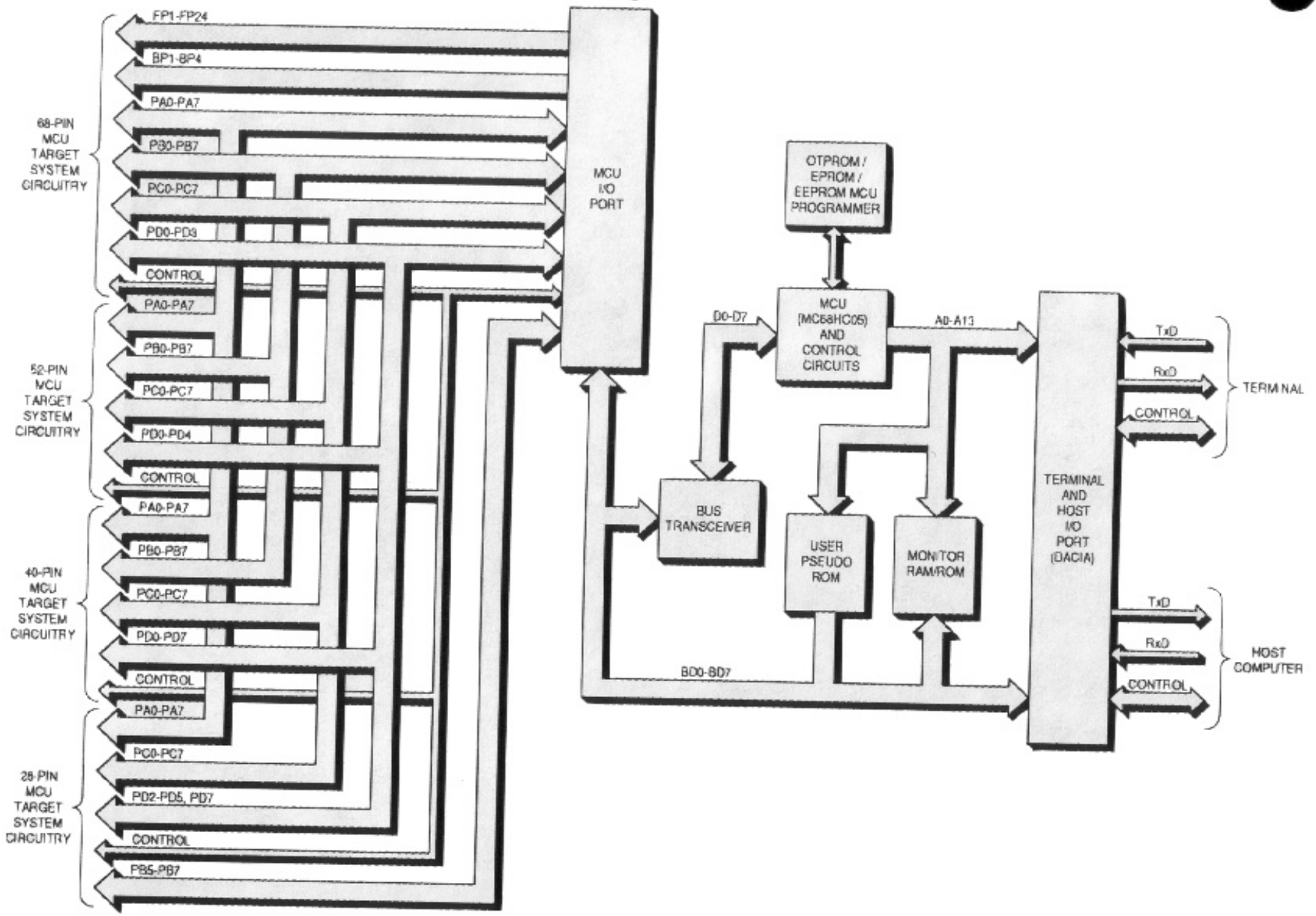


FIGURE 4-1. EVM Block Diagram

4.2.1.2 Abort. The abort circuitry generates internal abort signals upon the activation of the ABORT switch S1. A software interrupt opcode is placed on the MCU data bus synchronously with the LIR* signal, and the memory map is switched to the monitor map.

NOTE

This memory map switching operation assumes proper operation in the user map. If the MCU is not operating properly in the user map (i.e., an illegal opcode or stop instruction executed) the ABORT switch may not cause a map switch.

4.2.1.3 Address Decoding. Address decoding is accomplished via a 82S100 Field Programmable Logic Array (FPLA) device that provides the required chip select signals for memory and peripheral device circuitry that are memory mapped in the EVM.

4.2.2 Monitor and User Memory

The EVM operates in either one of two memory maps (monitor or user map) as shown in Figures 4-2 through 4-6. Both memory maps can be configured for either 8K or 16K bytes, and are decoded via the FPLA.

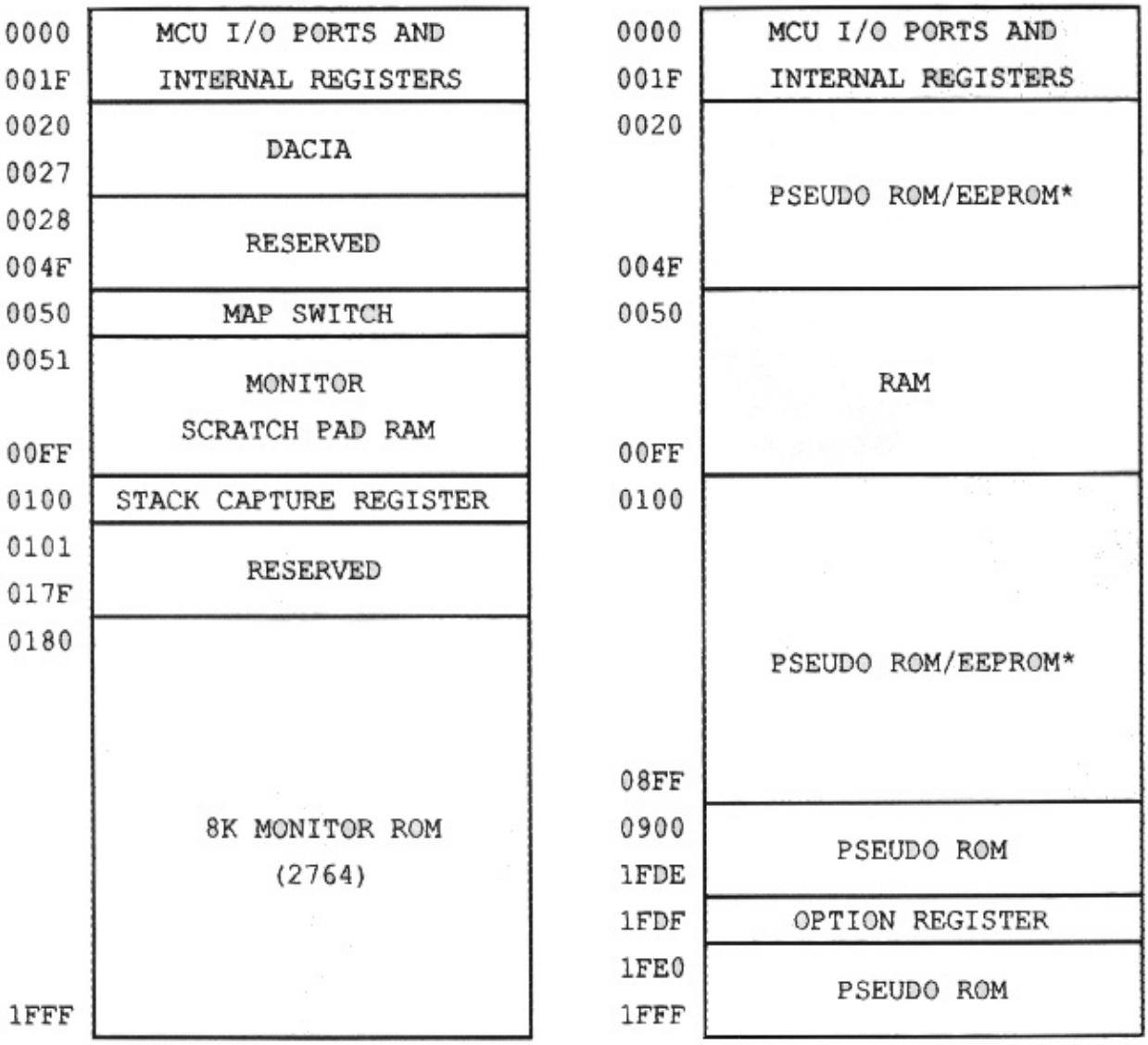
Jumper header J25 is used to select either the 16K-byte memory map which is required for MC68HC05C9 operations, or the 8K-byte memory map which is applicable for all other operations. Refer to paragraph 2.3.14 for jumper header configuration information.

NOTE

The entire 8K-/16K-byte user map (pseudo ROM/EEPROM) is available to the user, although all M68HC05 MCU family devices do not have user ROM throughout the entire 8K-/16K-byte memory map. Refer to specific device data sheet for valid program space locations.

The EVM memory maps are presented as follows:

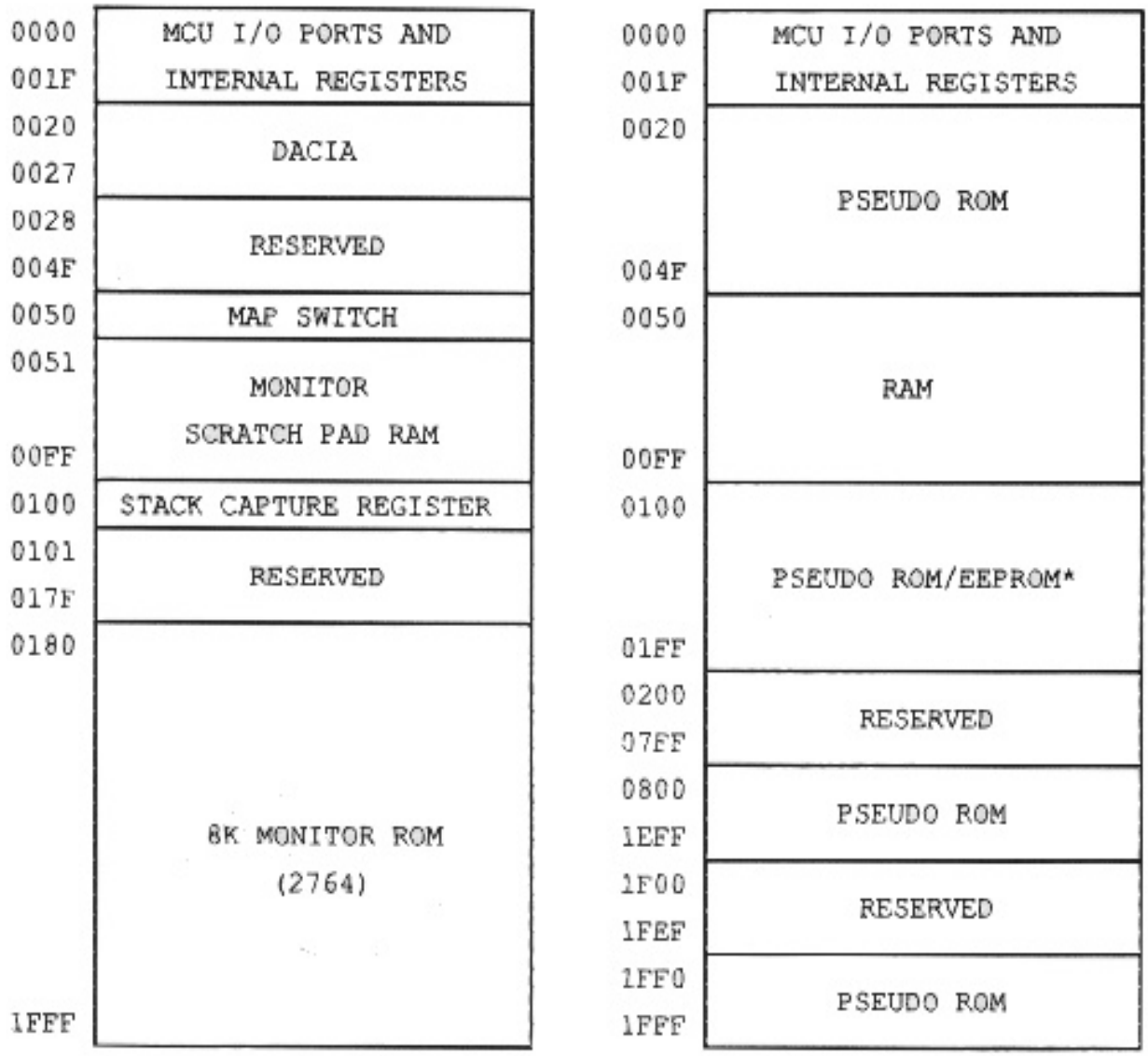
Figure 4-2 MC68HC05A6 memory map
Figure 4-3 MC68HC05B4/B6 and 805B6 memory map
Figure 4-4 MC68HC05C2/C3/C4/C8, 705C8 and 805C4 memory map
Figure 4-5 MC68HC05C9 memory map
Figure 4-6 MC68HC05P1/P7 memory map



Notes:

1. Jumper header J25, pins 1 and 2 not connected for 8K-byte memory map.
2. * Denotes for pseudo ROM, reconfigure jumper header J22, pins 1 to 2. For EEPROM, reconfigure jumper header J22, pins 5 to 6.

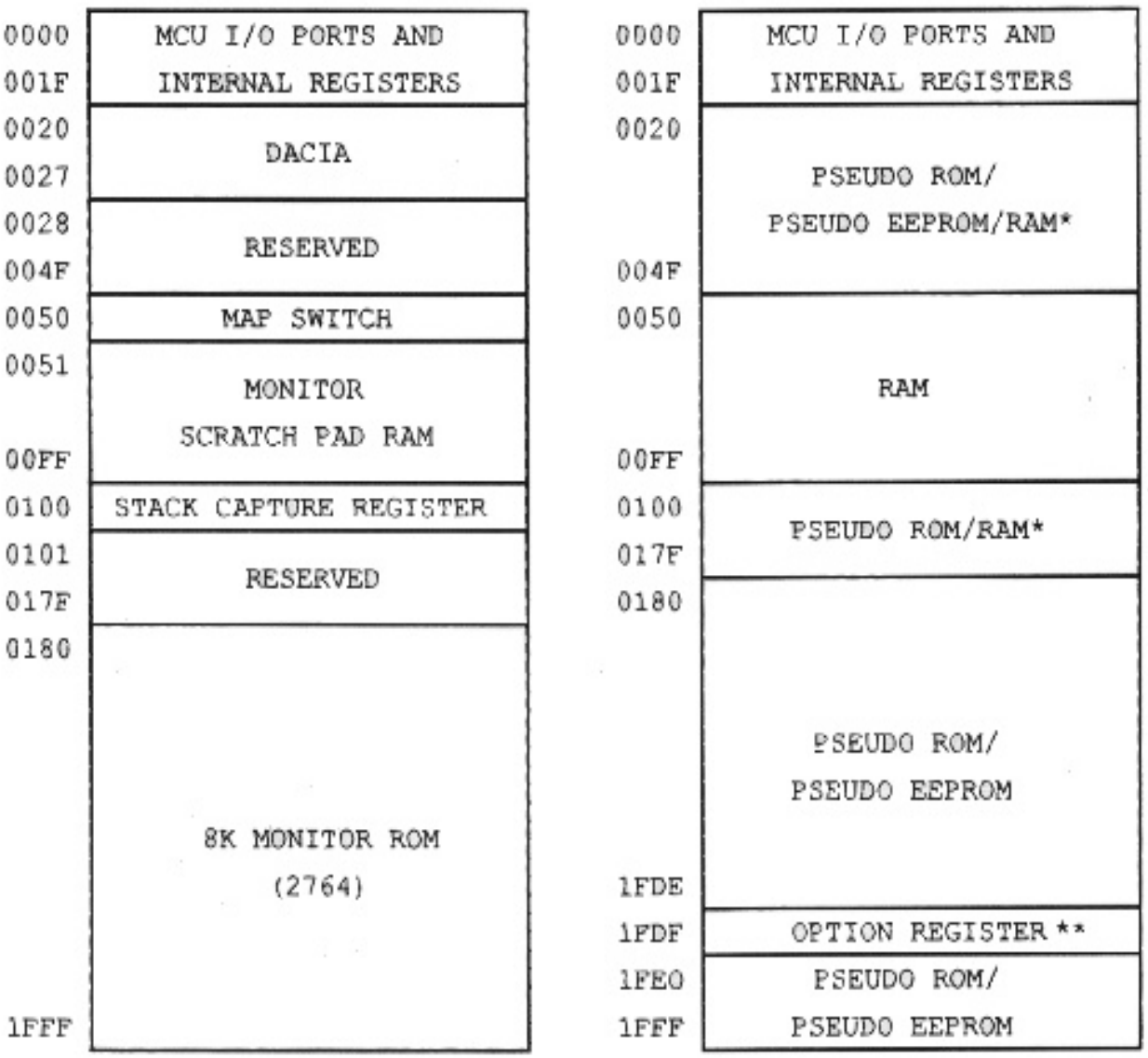
FIGURE 4-2. EVM (A6) Memory Map



Notes:

1. Jumper header J25, pins 1 and 2 not connected for 8K-byte memory map.
2. * Denotes for B6/805B6 operation, reconfigure jumper header J22, pins 2 to 4.

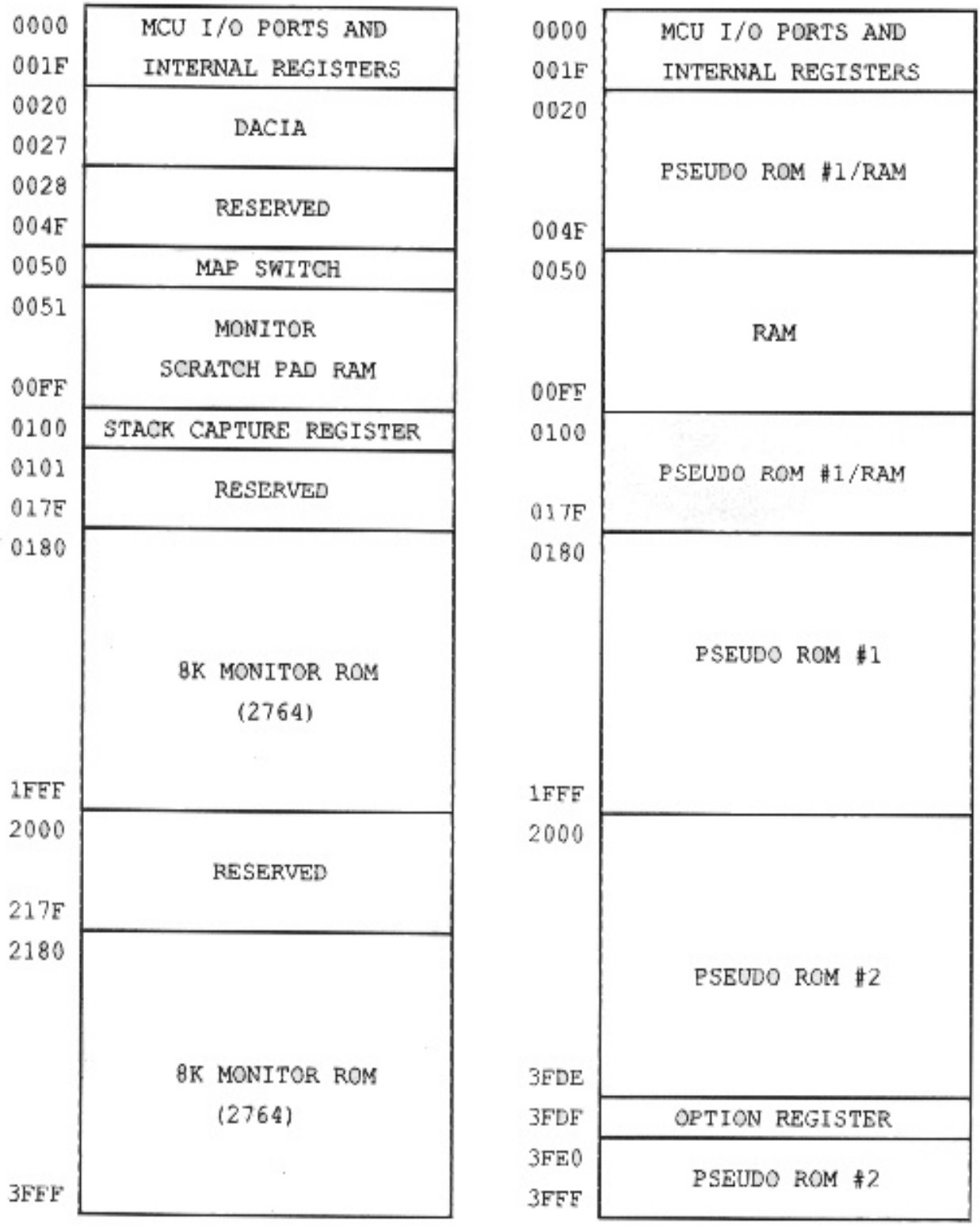
FIGURE 4-3. EVM (B4/B6/805B6) Memory Map



Notes:

1. Jumper header J25, pins 1 and 2 not connected for 8K-byte memory map.
2. * Denotes RAM available only when MC68HC705C8 is the resident MCU.
3. ** Denotes available only when MC68HC705C8 is the resident MCU.
4. MC68HC705C8 MCU memory map differs from the actual EVM memory map by $32 bytes (i.e., RAM locations $0100-$017F instead of $0100-$015F). User code must start at $0180 instead of $0160 to avoid conflict with remapping RAM.

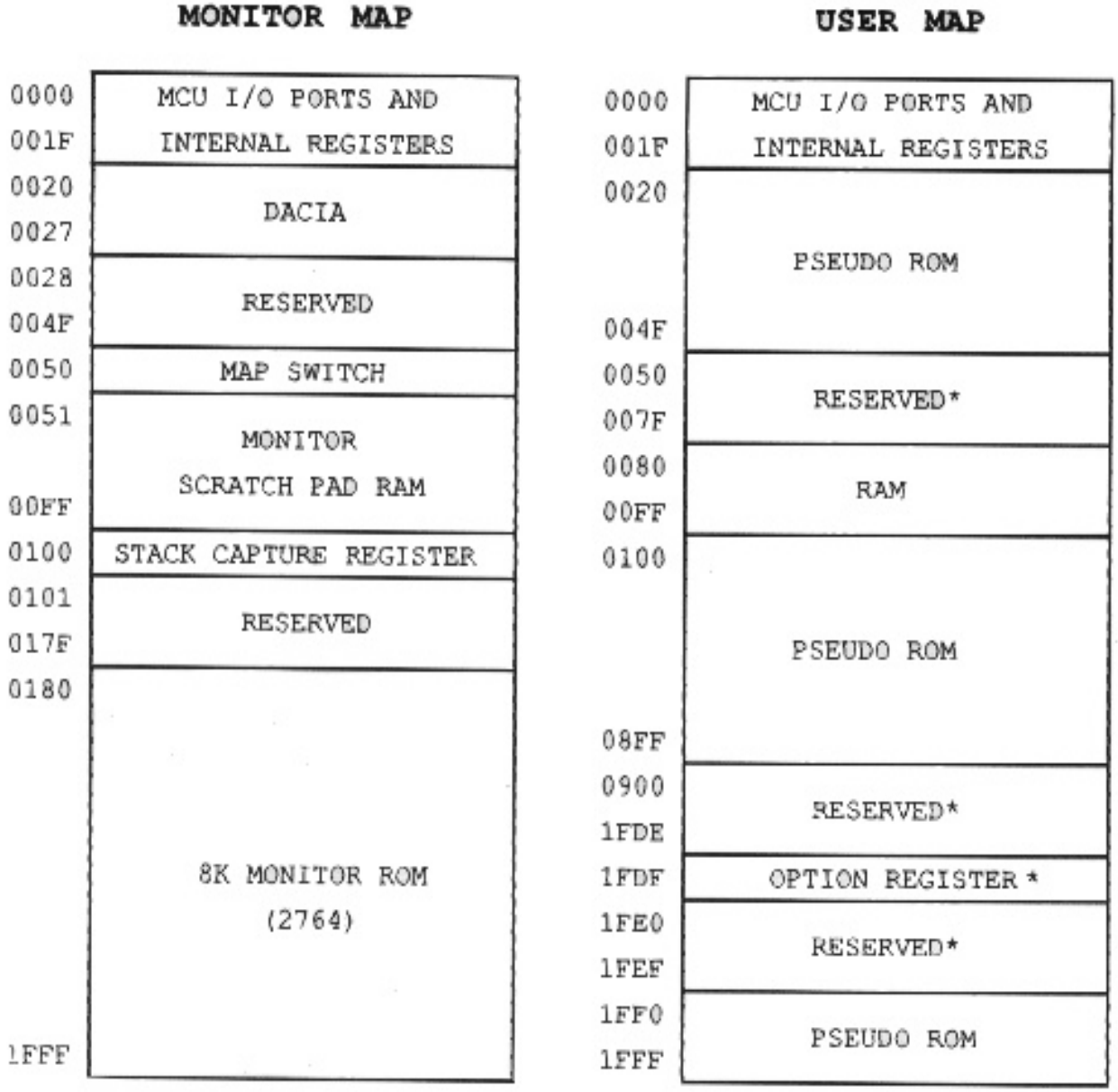
FIGURE 4-4. EVM (C2/C3/C4/C8/705C8/805C4) Memory Map



Note:

1. Jumper header J25, pins 1 and 2 connected for 16K-byte memory map.

FIGURE 4-5. EVM (C9) Memory Map



Notes:

1. Jumper header J25, pins 1 and 2 not connected for 8K-byte memory map.
2. * Denotes available only when MC68HC705C8 or 805C4 is the resident MCU.

FIGURE 4-6. EVM (P1/P7) Memory Map

4.2.2.1 Monitor Map Area. As shown in Figures 4-2 through 4-6, the 16K monitor map area contains the MCU I/O ports and internal registers, DACIA (terminal/host), map switch register, monitor scratch pad RAM, monitor stack capture register, and 8K bytes monitor ROM. The EVMbug monitor EPROM contents are not available to the user.

All monitor operations are controlled via the monitor I/O (terminal and host DACIA). Both terminal and host computer I/O ports are also controlled by the DACIA. The DACIA is available only in the monitor map. User programs in the user map cannot access these peripheral ports.

The monitor scratch pad RAM/stack capture register is used by the monitor for general monitor operations such as temporary data storage, command entries, downloading, etc..

The monitor map switch register is located at $0050, and is used for temporary and permanent map switching operations, user memory protection when in the monitor map, and breakpoint/trace/abort monitor flagging. The monitor map switch register is only controlled by the monitor.

4.2.2.2 User Map Area. As shown in Figures 4-2 through 4-6, the user map area consists of the MCU I/O ports and internal registers, xx bytes of page zero pseudo ROM, xxx bytes of user RAM, and xxxx bytes of user pseudo ROM.

User program space (user pseudo ROM) is RAM. This RAM is write protected via jumper header J3 during user program execution. Refer to paragraph 2.3.15 for additional information pertaining to the write protect header J3. This feature requires all programs to be ROMable and protects against program errors which would otherwise overwrite the program space.

NOTE

The entire 8K/16K-byte memory map (pseudo ROM/EEPROM) is available to the user, although all M68HC05 MCU family devices do not have user ROM throughout the entire 8K/16K-byte map. Refer to specific device data sheet for valid program space locations.

4.2.3 Terminal and Host Computer I/O Ports

Both the terminal and host computer I/O ports share a common R65C52 Dual Asynchronous Communications Interface Adapter (DACIA) high speed device. This device contains two independent full duplex channels with buffered receivers and transmitters. In addition, internal programmable baud rate generation, dual sets of registers for independent channel control and monitoring are also provided on the DACIA.

4.2.3.1 Terminal I/O Port Circuitry. The EVM terminal I/O port communicates with an RS-232C compatible terminal via the TERMINAL connector J1 and a user supplied cable assembly. The first half of the R65C52 DACIA - based terminal interface circuitry provides communication and data transfer operations for the EVM and user terminal. Because of the lack of internal timing logic on the DACIA, auto baud rate capabilities could not be implemented on the EVM. Therefore a fixed 9600 baud rate was implemented for the terminal port only. RS-232C drivers/receivers are also implemented for this port. For connector J1 pin assignments and signal descriptions, refer to Chapter 5.

A software transparent mode allows direct communications between the terminal and host computer I/O ports. Also, files may be downloaded through the terminal I/O port using the LOAD command.

4.2.3.2 Host Computer I/O Port Circuitry. The EVM host computer I/O port communicates with an RS-232C compatible host computer directly or by modem via the HOST connector J15 and a user supplied cable assembly. The second half of the R65C52 DACIA - based host interface circuitry provides communications and data transfer operations for the EVM and user host computer. Software selectable (via the SPEED command) 300-19.2k baud rate generation capabilities, and RS-232C drivers/receivers are also implemented for this port. For connector J15 pin assignments and signal descriptions, refer to Chapter 5.

4.2.4 MCU Extension I/O Ports

The EVM provides four user HCMOS compatible MCU extension I/O ports for target system evaluation of 28- pin, 40-pin, 44-lead, 52-lead, and 68-lead M6805 HCMOS family of MCU devices. These device types include MC68HC05A6, MC68HC05C2/C3/C4/C8/C9, MC68HC05B4/B6, MC68HC05L6, MC68HC05P1/P7, MC68HC705C8, MC68HC805B6, and MC68HC805C4.

NOTE

The EVM cannot emulate the low power capabilities of the MC68HCL05C4/C8 MCU devices. The EVM can emulate the high speed (4 MHz) capabilities of the MC68HSC05C4/C8 MCU devices by the replacement of the 8 MHz crystal (Y1) to a 16 MHz crystal. Refer to paragraph 2.3.10.

4.2.4.1 28-Pin I/O Port. Target system to EVM interconnection for the MC68HC05P1/P7 operation is accomplished via EVM connector J17, and a user supplied 28-pin dual-in-line plastic (DIP) cable assembly. MCU I/O port connector J17 is a 28-pin header that facilitate the interconnection of the cable assembly for evaluation purposes. Refer to paragraphs 2.3.6 and 2.3.7 for P1/P7 port configuration information. For connector J17 pin assignments and signal descriptions, refer to Chapter 5.

4

4.2.4.2 40-Pin I/O Port. Target system to EVM interconnection for the MC68HC05A6, MC68HC05C2/C3/C4/C8/C9, MC68HC705C8, and MC68HC805C4 operation is accomplished via EVM connector J19, and a user supplied 40-pin dual-in-line plastic (DIP) cable assembly. A factory supplied 44-lead plastic leaded chip carrier (PLCC) cable assembly is also available. MCU I/O port connector J19 is a 40-pin header that facilitate the interconnection of either cable assembly for evaluation purposes. For connector J19 pin assignments and signal descriptions, refer to Chapter 5.

4.2.4.3 60-Pin I/O Port. Target system to EVM interconnection for the MC68HC05B4/B6 and MC68HC805B6 operation is accomplished via EVM connector J20, and a factory supplied 52-lead plastic leaded chip carrier (PLCC) cable assembly. MCU I/O port connector J20 is a 60-pin header that facilitate the interconnection of the cable assembly for evaluation purposes. For connector J20 connector pin assignments and signal descriptions, refer to Chapter 5.

4.2.4.4 34-Pin I/O Ports. Target system to EVM interconnection for the MC68HC05L6 operation is accomplished via EVM connectors J18 and J29, and two user supplied 34-pin header type cable assemblies. (EVM connectors J18 and J29 are also user supplied and installed. Refer to paragraph 2.4.1.) MCU I/O port connector J18 is a 34-pin header that facilitate the interconnection of cable assembly A for evaluation purposes. MCU I/O port connector J29 is a 34-pin header that facilitate the interconnection of cable assembly B for evaluation purposes. For connectors J18 and J29 pin assignments and signal descriptions, refer to Chapter 5.

4.2.5 OTPROM/EPROM/EEPROM MCU Programmer

The MCU programmer accommodates and programs either the MC68HC705C8 OTPROM/EPROM, MC68HC805C4 EEPROM, or the MC68HC805B6 EEPROM MCU device. The device packages are 40-pin Dual-In-line Packages (DIPs) and 44-lead/52-lead Plastic Leaded Chip Carriers (PLCCs). Programming socket located at Z3 is used for the 40-pin DIP package, and sockets located at Z1 and Z2 are used for the 52-lead and 44-lead PLCC packages, respectively.

The MC68HC705C8 OTPROM/EPROM and MC68HC805C4 EEPROM MCU programming sockets (Z2 and Z3) are factory supplied and installed. Z3 consists of a 40-pin DIP socket and Z2 consists of a 44-lead PLCC surface mount socket. The MC68HC805B6 EEPROM MCU programming socket Z1 (52-lead PLCC surface mount) is also factory supplied and installed.

4.2.5.1 MC68HC805C4 EEPROM MCU Programming. EEPROM MCU programming is controlled by the MASTER RESET switch S3, programmer power switch S4, and programmer VPP switch S5. Programming VPP and power are removed and a master reset is issued. The EEPROM MCU device to be programmed (slave) is inserted in the applicable programming socket Z2 or Z3. Programming power and VPP are now applied. Programming commands (BULK, CHCK, COPY, PROG, or VERF) are now entered via the terminal keyboard. When the PROG command is executed, programming data residing in the user map is transferred via the EVM resident MCU (master) to the slave MCU. Upon completion of any MCU EEPROM programming operation, a master reset is issued and programming VPP and power are removed. The slave MCU is now removed from the programming socket.

In a similar method, by executing a COPY command, the contents of the slave MCU, installed in programming socket Z2 or Z3 is copied to the user map pseudo EEPROM/ROM. Additional functions for checking, verifying, and erasing MCU EEPROM contents are also provided via the CHCK, VERF, and BULK commands, respectively. For additional information pertaining to the programming procedures and applicable terminal keyboard commands, refer to Chapter 3.

The EVMbug monitor EPROM contains the programming algorithm (instructions). This program is downloaded into the slave MCU (via slave bootstrap loader). The slave MCU, while operating in the bootstrap mode, obtains the program from the EVM resident MCU (master) via the Serial Communications Interface (SCI). Once the slave MCU receives and places the program into internal RAM, program control is passed to the first internal RAM location and program execution begins.

The slave program monitors the SCI receive line for a 5-byte data block. This data block contains a 2-byte start address, 2-byte end address, and a 1-byte command (BULK or PROG). Once the command is received by the slave MCU, (depending upon the command) the program is executed. On completion of the program, the slave MCU returns to the bootstrap loader program. CHCK, COPY, and VERF commands utilize the dump EEPROM mode on the slave MCU.

4.2.5.2 MC68HC805B6 EEPROM MCU Programming. EEPROM MCU programming is controlled by the MASTER RESET switch S3, programmer power switch S4, and programmer VPP switch S5. Programming VPP and power are removed and a master reset is issued. The EEPROM MCU device to be programmed (slave) is inserted in the programming socket Z1. Programming power and VPP are now applied. Programming commands (BULK, CHCK, COPY, PROG, or VERF) are now entered via the terminal keyboard.

EEPROM MCU programming operations (bulk erase, check, copy, program, or verify) are controlled in a different manner as described in paragraph 4.2.5.1. The slave MCU is initialized after reset and monitors the resident MCU SCI for bursts of data in 3-byte lengths. The first byte is the most-significant-address byte, the second byte is the least-significant-address byte, and the third byte is the data byte. Program data is contained in the data byte. The data byte will contain $FF for the copy, check, and verify operations.

Upon receipt of the 3-byte burst, the slave MCU transmits the specified address (1 byte) back to the resident MCU. For the erase operation, the slave MCU is placed in the erase mode of operation, taken out of reset, and delayed for 100 mS. The slave MCU is then invoked to perform the erase operation.

4.2.5.3 MC68HC705C8 OTPROM/EPROM MCU Programming. OTPROM/EPROM MCU programming operations (check, copy, program, or verify) are controlled in an identical manner as described in paragraph 4.2.5.1. The only exception is that the bootstrap loader program will not perform the erase operation.

MC68HC705C8 One Time Programmable ROM (OTPROM) MCU devices are shipped in an erased state and cannot be erased. Electrical erasing operations cannot be performed on OTPROM MCU devices.

MC68HC705C8 EPROM MCU devices are erased by the exposure of a high-intensity ultraviolet (UV) light with a wavelength of 2537 Angstrom (A). The recommended dose (UV intensity x exposure time) is 15 Ws/cm2. UV lamps should be used without shortwave filters, and the EPROM MCU device positioned about one inch from the UV lamps.

CHAPTER 5

SUPPORT INFORMATION

5.1 INTRODUCTION

This chapter provides the connector signal descriptions, parts list with associated parts location diagram, and schematic diagrams for the EVM.

5.2 CONNECTOR SIGNAL DESCRIPTIONS

The EVM provides two RS-232C I/O port connectors J1 and J15 which are used to interconnect the EVM to an RS-232C compatible terminal and host computer, respectively.

Five MCU I/O port connectors J17 through J20, and J29 are used to interconnect the EVM to the target system equipment. Connector J19 is used for 40-pin MCU device evaluation (e.g., MC68HC05A6/C2-C4/C8/C9 and MC68HC705C8/805C4). Connector J17 is used for 28-pin MCU device evaluation (e.g., MC68HC05P1/P7). Connector J20 is used for 52-pin MCU device evaluation (e.g., MC68HC05B4/B6 and MC68HC805B6). Connectors J18 and J29 are used for 68-pin MCU device evaluation (e.g., MC68HC05L6).

Connector J28 interconnects an external power supply to the EVM.

Pin assignments for the above connectors are identified in Tables 5-1 through 5-8 as follows:

Table 5-1. RS-232C Terminal Port Connector J1 Pin Assignments

Table 5-2. RS-232C Host Port Connector J15 Pin Assignments

Table 5-3. 28-Pin MCU I/O Port Connector J17 Pin Assignments

Table 5-4. 40-Pin MCU I/O Port Connector J19 Pin Assignments

Table 5-5. 60-Pin MCU I/O Port Connector J20 Pin Assignments

Table 5-6. 34-Pin MCU I/O Port Connector (A) J18 Pin Assignments

Table 5-7. 34-Pin MCU I/O Port Connector (B) J29 Pin Assignments

Table 5-8. Input Power Connector J28 Pin Assignments

Connector signals are identified by pin number, signal mnemonic, and signal name and description.

TABLE 5-1. RS-232C Terminal Port Connector J1 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	------	Not connected.
2	TXD	TRANSMIT DATA - Serial data input line.
3	RXD	RECEIVE DATA - Serial data output line.
4	RTS	REQUEST TO SEND - An input signal used to request permission to transfer data. (Refer to paragraph 2.3.11.)
5	CTS	CLEAR TO SEND - An output signal used to indicate a ready-to-transfer data status.
6	DSR	DATA SET READY - An output signal (held high) used to indicate an on-line/in-service/active status.
7	SIG-GND	SIGNAL GROUND - This line provides signal ground or common return connection between the EVM and RS-232C compatible terminal. This line establishes the common ground reference potential between the EVM and RS-232C compatible terminal circuitry.
8	DCD	DATA CARRIER DETECT - An output signal (held high) used to indicate an acceptable carrier signal has been detected.
9-25	------	Not connected.

5

TABLE 5-2. RS-232C Host Port Connector J15 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	-----	Not connected.
2	TXD	TRANSMIT DATA - Serial data output line.
3	RXD	RECEIVE DATA - Serial data input line.
4	RTS	REQUEST TO SEND - An output signal used to request permission to transfer data.
5	CTS	CLEAR TO SEND - An input signal used to indicate ready-to-transfer data status. (Refer to paragraph 2.3.9.)
6	-----	Not connected.
7	SIG-GND	SIGNAL GROUND - This line provides signal ground or common return connection between the EVM and RS-232C compatible host computer. This line establishes the common ground reference potential between the EVM and RS-232C compatible host computer circuitry.
8-19	-----	Not connected.
20	DTR	DATA TERMINAL READY - An output line (held high) used to indicate an on-line/in-service/active status.
21-25	-----	Not connected.

TABLE 5-3. 28-Pin MCU I/O Port Connector J17 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	$\overline{\text{RESET}}$*	RESET - An active low output signal that applies an MCU reset to the target system.
2	$\overline{\text{IRQ}}$*	INTERRUPT REQUEST - An active low input signal that asynchronously applies an MCU interrupt.
3-10	PA7-PA0	PORT A (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
11-13	PB5-PB7 PD2-PD4	PORT B (bits 5-7) [MC68HC05P1] PORT B (bits 5-7) [MC68HC05P7] - General purpose I/O lines controlled by software via data direction and data registers. PD2, PD3, PD4 lines are shared with the serial I/O port lines SDO, SDI, and SCK; respectively.
14	GND-VSS	GROUND
15-22	PC7-PC0	PORT C (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
23	PD5	PORT D (bit 5) - Special function port I/O line used by the corresponding DDR bit at address location $0007.
24	PD6 (TCMP)	PORT D (bit 6) - Special function port output line used by the output compare feature of the MCU programmable timer system.
25	PD7 (TCAP')	PORT D (bit 7) - Special function port input line used by the input capture feature of the MCU programmable timer system.
26-28	NC	Not connected.

TABLE 5-4. 40-Pin MCU I/O Port Connector J19 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	$\overline{\text{RESET}}$	RESET - An active low output signal that applies an MCU reset to the target system.
2	$\overline{\text{IRQ}}$	INTERRUPT REQUEST - An active low input signal that asynchronously applies an MCU interrupt.
3	NC	Not connected.
4-11	PA7-PA0	PORT A (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
12-19	PB0-PB7	PORT B (bits 0-7) - General purpose I/O lines controlled by software via data direction and data registers.
20	GND-VSS	GROUND
21-28	PC7-PC0	PORT C (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
29-34 36	PD0-PD5 PD7	PORT D (bits 0-5 and 7) - General purpose input or I/O lines used bythe MCU serial communications communications interface (SCI) and serial peripheral interface (SPI) circuitry.
35	TCMP	TIMER COMPARE - An output signal used by the output compare feature of the MCU programmable timer system.
37	TCAP	TIMER CAPTURE - An input signal used by the input capture feature of the MCU programmable timer system.
38-40	NC	Not connected.

5

TABLE 5-5. 60-Pin MCU I/O Port Connector J20 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	TCMP2	TIMER COMPARE 2 - An output signal used by the output compare feature of the MCU programmable timer system.
2	TCMP1	TIMER COMPARE 1 - An output signal used by the output compare feature of the MCU programmable timer system.
3 4 5 9 11 12 13 14	PD7/AN7 PD6/AN6 PD5/AN5 PD4/AN4 PD3/AN3 PD2/AN2 PD1/AN1 PD0/AN0	PORT D/ANALOG CHANNEL (bits 0-7) - General purpose input or analog-to-digital (A/D) channel input lines.
6	NC	Not connected.
7	VRL	VOLTAGE REFERENCE LOW - Input reference supply voltage (low) line for the MCU A/D converter.
8	VRH	VOLTAGE REFERENCE HIGH - Input reference supply voltage (high) line for the MCU A/D converter.
10, 15-17	NC	Not connected.
18	$\overline{\text{RESET}}$	RESET - An active low output signal that applies an MCU reset to the target system.
19	$\overline{\text{IRQ}}$	INTERRUPT REQUEST - An active low input signal that asynchronously applies an MCU interrupt.
20	PLMA D/A	PULSE LENGTH MODULATION A OUTPUT DIGITAL/ANALOG - An output line used by the MCU D/A circuitry.

5

TABLE 5-5. 60-Pin MCU I/O Port Connector J20 Pin Assignments (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
21	PLMB D/A	PULSE LENGTH MODULATION B OUTPUT DIGITAL/ANALOG - An output line used by the MCU D/A circuitry.
22	TCAP1 (TCAP')	TIMER CAPTURE 1 - An input signal used by the input capture feature of the MCU programmable timer system.
23	TCAP2	TIMER CAPTURE 2 - An input signal used by the input capture feature of the MCU programmable timer system.
24-31	PA7-PA0	PORT A (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
32-39	PB7-PB0	PORT B (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
40	NC	Not connected.
41	GND-VSS	GROUND
42-49	PC7-PC0	PORT C (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
50	RXD (RDI)	RECEIVE DATA (RECEIVE DATA IN) - Serial data input line used by the MCU SCI circuitry.
51	SCLK	SERIAL CLOCK - Serial transmitter clock output line used by the MCU SCI circuitry.
52	TXD (TDO)	TRANSMIT DATA (TRANSMIT DATA OUT) - Serial data output line used by the MCU SCI circuitry.
53-60	NC	Not connected.

5

TABLE 5-6. 34-Pin (MCU I/O Port Connector (A) J18 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	NC	Not connected.
2	$\overline{IRQ}$*	INTERRUPT REQUEST - An active low input signal that asynchronously applies an MCU interrupt.
3	$\overline{RESET}$*	RESET - An active low output signal that applies an MCU reset to the target system.
4,5	NC	Not connected.
6	TCAP*	TIMER CAPTURE - An input signal used by the input capture feature of the MCU programmable timer system.
7	TCMP*	TIMER COMPARE - An output signal used by the output compare feature of the MCU programmable timer system.
8-15	PA7-PA0	PORT A (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
16-23	PB0-PB7	PORT B (bits 0-7) - General purpose I/O lines controlled by software via data direction and data registers.
24-31	PC7-PC0	PORT C (bits 7-0) - General purpose I/O lines controlled by software via data direction and data registers.
32	GND-VSS	GROUND
33	NC	Not connected.
34	ALRT	ALERT - An audio frequency tone generator square wave output signal.

5

TABLE 5-7. 34-Pin MCU I/O Port Connector (B) J29 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1-24	FP1-FP24	FRONTPLANE (lines 1-24) - Twenty four output lines that provide frontplane drive signals to an external liquid crystal display (LCD) device.
25-28	BP1-BP4	BACKPLANE (lines 1-4) - Four output lines that provide backplane drive signals to an external LCD device.
29-32	PD0-PD3	PORT D (bits 0-3) - General purpose input or I/O lines used by the MCU serial peripheral interface (SPI) circuitry.
33	NC	Not connected.
34	VLL	LCD POWER - +3.0 to +6.0 Vdc input power line used to supply the operating voltage for the MCU LCD driver circuitry. Refer to MC68HC05L6 device data sheet for specific voltage levels.

5

TABLE 5-8. Input Power Connector J28 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	VPP	Programming Voltage - Input voltage (+19.0 Vdc @ 100 uA) used by the EVM programming circuits. Refer to device data sheet for specific programming voltage.
2	+5V	+5 Vdc Power - Input voltage (+5 Vdc @ 1.0 A) used by the EVM logic circuits.
3	-12V	-12 Vdc Power - Input voltage (-12 Vdc @ 0.1 A) used by the EVM logic circuits.
4	+12V	+12 Vdc Power - Input voltage (+12 Vdc @ 0.1 A) used by the EVM logic circuits.
5, 6	GND	GROUND

5

5.3 PARTS LIST

Table 5-9 lists the components of the EVM by reference designation order. The reference designation is used to identify the particular part on the parts location diagram (Figure 5-1) that is associated with the parts list table. This parts list reflects the latest issue of hardware at the time of printing.

TABLE 5-9. EVM Parts List

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
	Printed Wiring Board (PWB), M68HC05EVM
C1	Capacitor, electrolytic, 2.2 uF, +/- 20%, @ 16 Vdc
C2-C19, C21-C26, C29-C38, C42, C48-C66	Capacitor, ceramic, 0.1 uF, +/- 20%, @ 50 Vdc
C20, C27, C28, C39	Capacitor, ceramic, 0.01 uF, +/- 20%, @ 50 Vdc
C40, C41	Capacitor, mica, 25 pF, +/- 10%, @ 50 Vdc
C43-C45	Capacitor, electrolytic, 47 uF, +/- 20%, @ 25 Vdc
C46, C47	Capacitor, mica, 15 pF, +/- 10%, @ 50 Vdc
D1, D3, D4	Diode, 1N914
D2	Diode, 1N5817
D5, D6, D7	Diode, 1N4001
D8	Diode, 1N4735
F1	Fuse, Bussman # GMA 5mm x 20mm, 250 V, 1.5 A
J1, J15	Connector, DB25S, Dupont # 86858-32S
J2-J5, J7, J16, J21, J24, J27	Header, single row post, 3 pin, Aptronics # 929705-01-03

TABLE 5-9. EVM Parts List (cont'd)

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
J6, J22	Header, double row post, 3 pin, Aptronics # 929715-01-03
J8-J14, J23, J25, J26	Header, single row post, 2 pin, Aptronics # 929705-01-02
J17	Header, double row post, 14 pin, Aptronics # 929715-01-14
J18, J29	Header, double row post, 17 pin, Aptronics # 929715-01-17
J19	Header, double row post, 20 pin, Aptronics # 929715-01-20
J20	Header, double row post, 30 pin, Aptronics # 929715-01-30
J28	Terminal block, 6 position, Electrovert # 25.104.0653, or 5 position, 2S series, Augat RDI # 2SV-05
Q1, Q2	Transistor, 2N4400
R1, R8, R21, R24-R27, R30, R32	Resistor, 4.7k ohm, 5%, 1/4 W
R2-R5, R12	Resistor, 470 ohm, 5%, 1/4 W
R6, R18, R20, R23, R31, R34	Resistor, 10k ohm, 5%, 1/4 W
R7, R10	Resistor, 330 ohm, 5%, 1/4 W
R9	Resistor, 2k ohm, 5%, 1/4 W
R11	Resistor, 100k ohm, 5%, 1/4 W
R13-R16	Resistor, 39k ohm, 5%, 1/4 W

TABLE 5-9. EVM Parts List (cont'd)

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
R17, R28, R29	Resistor, 1k ohm, 5%, 1/4 W
R19, R22	Resistor, 2.7k ohm, 5%, 1/4 W
R33, R35	Resistor, 10M ohm, 5%, 1/4 W
RN1	Resistor network, nine 10k ohm, Allen-Bradley # 710A103
S1-S3	Switch, pushbutton, C&K # 8125-SD9R2BE
S4	Switch, slide, DPDT, Stackpole # S-9022CD01-0, or Switchcraft # C56206L2
S5	Switch, slide, SPDT, C&K # 1101M2CQE
U1	I.C. 74F32
U2	I.C. 74F00
U3	I.C. MC1488, RS-232C driver
U4, U10, U12, U21, U26, U29, U30, U33	I.C. 74HC74
U5	I.C. 7407
U6, U22, U24	I.C. 74HC08
U7	I.C. MC1489A, RS-232C receiver
U8	I.C. 74HC14
U9, U27	I.C. 74HC02
U11	I.C. 74HC04
U13	I.C. 74F74
U14, U23, U38	I.C. 74HC00

TABLE 5-9. EVM Parts List (cont'd)

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
U15, U35	I.C. 74HC11
U16	I.C. 74HC4066
U17, U37	I.C. 74HC32
U18, U19, U28, U36	I.C. 74HC10
U20	I.C. 74HC393
U25	I.C. XC68HC26, PRU
U31	I.C. 74LS260
U32	I.C. R65C52P2, DACIA
U34, U42	I.C. 74HCU04
U39, U40	I.C. PLS100, FPLA, 50 ns
U41, U43, U51	I.C. 74HC373
U44	I.C. MC68HC05C9, resident MCU
U45, U47, U48	I.C. 6264 (MCM 6064P) 8Kx8 RAM, 150 ns
U46	I.C. 2764, EVMbug monitor EPROM, 150 ns, programmed (see note)
U49	I.C. 74HC374
U50	I.C. 74HC646
U52	I.C. 74HC244
U53	I.C. MC68HC05B6, resident MCU, (not supplied)
U54	I.C. MC68HC05L6, resident MCU, (not supplied)
VR1	Voltage detector, 3.80-4.20 Vdc, Motorola # MC34064P or Seiko # S-8054HN

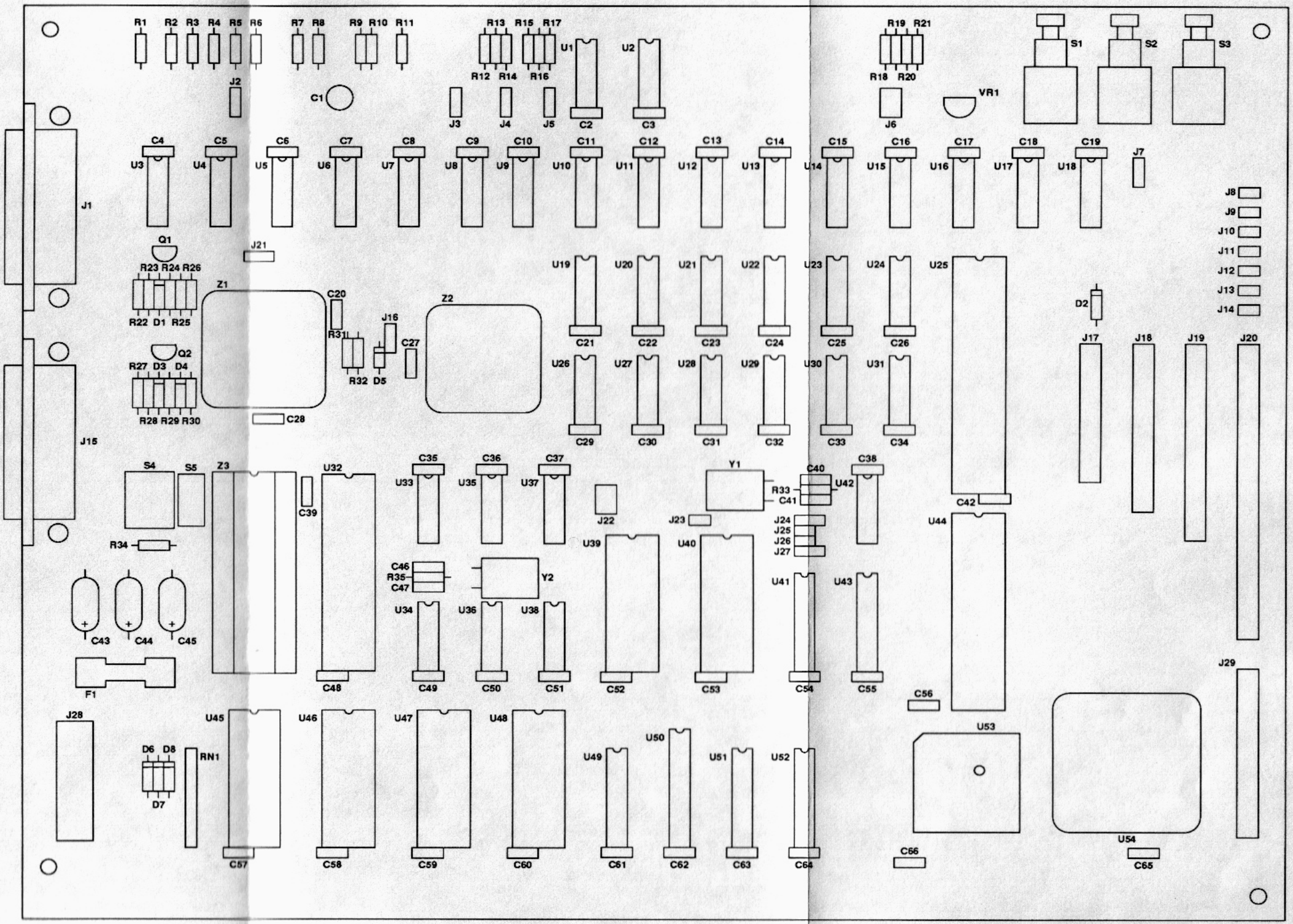
TABLE 5-9. EVM Parts List (cont'd)

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
Y1	Crystal, 8.0 MHz, ECS # 80, or Nymph # NYP080-18
Y2	Crystal, 3.6864 MHz, ECS # ECS3686.4, or Nymph # NYP037
Z1	Socket, surface mount, 52-lead, PLCC, Plastronics # P2052SP, MC68HC805B6 MCU programming
Z2	Socket, surface mount, 44-lead, PLCC, Plastronics # P2044SP, MC68HC705C8/805C4 MCU programming
Z3	Socket, low insertion force, 40-pin, DIP, Wells # 613-7400116, MC68HC705C8/805C4 MCU programming
	Fabricated jumper, Aptronics # 929955-00 (use with jumper headers)
	I.C. socket, 14-pin DIP, low profile, Robinson Nugent # ICL-143-S6-TG (for U10 and U52)
	I.C. socket, 28-pin DIP, low profile, Robinson Nugent # ICL-286-S7-TG (for U26 thru U31)
	I.C. socket, 40-pin DIP, low profile, Robinson Nugent # ICL-406-S7-TG (for U32 and U34)
	I.C. socket, 48-pin DIP, low profile, Augat # 248-A629D (for U20)
	I.C. socket, 52-pin, PC mount, PLCC, AMP # 821551-1 (for U41)
	I.C. socket, surface mount, 68-lead PLCC, Plastronics # P2068SP (for U54), (not supplied)

Note: When ordering, use number labeled on part.

5.4 DIAGRAMS

Figure 5-2 is the EVM schematic diagram.

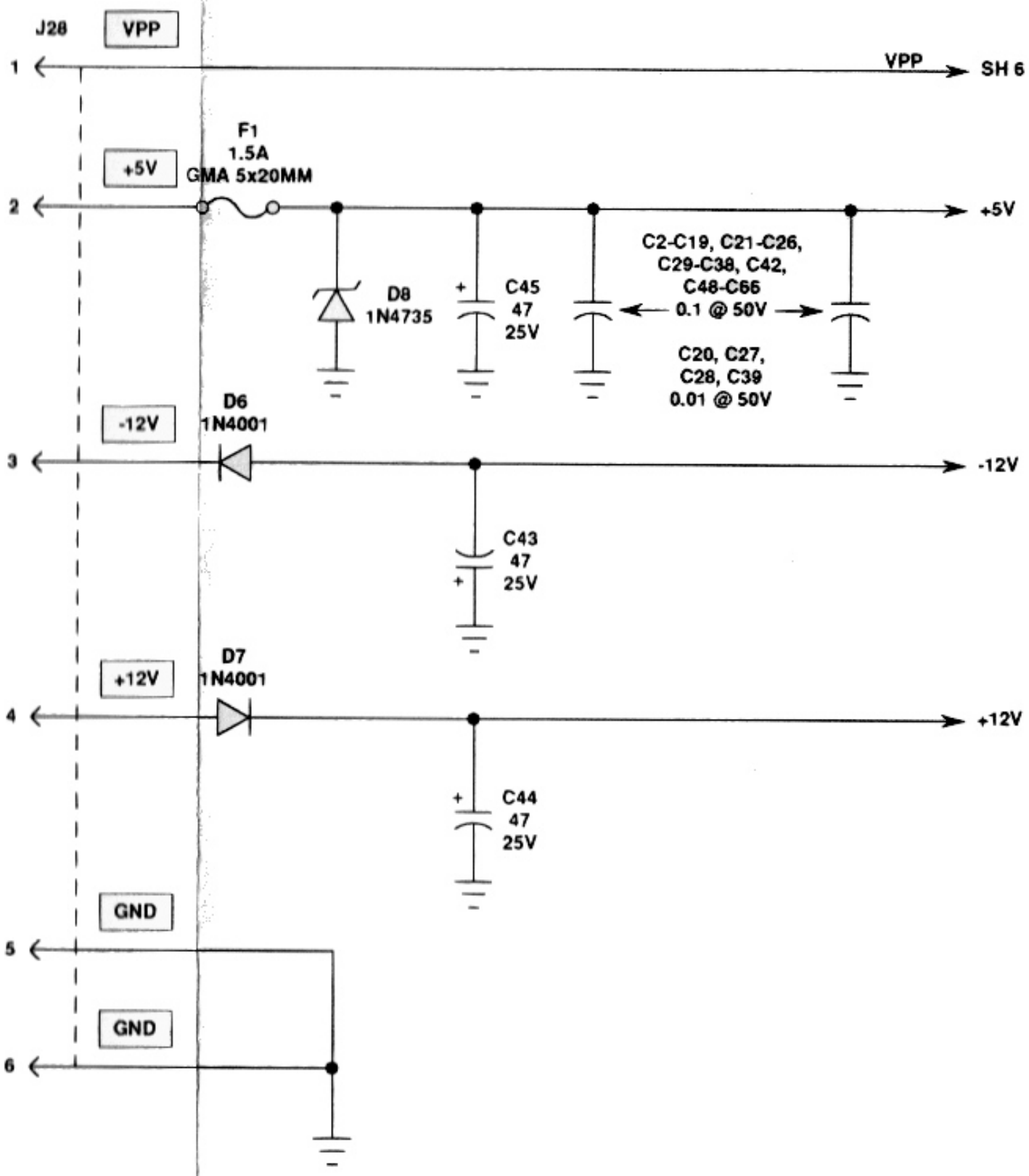

R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11
R12 R13 R14 R15 R16 R17
R18 R19 R20 R21
J1 J2 J3 J4 J5 J6 J7 J8 J9 J10 J11 J12 J13 J14 J15 J16 J17 J18 J19 J20 J21 J22 J23 J24 J25 J26 J27 J28 J29
C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15 C16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26 C27 C28 C29 C30 C31 C32 C33 C34 C35 C36 C37 C38 C39 C40 C41 C42 C43 C44 C45 C46 C47 C48 C49 C50 C51 C52 C53 C54 C55 C56 C57 C58 C59 C60 C61 C62 C63 C64 C65 C66
U1 U2 U3 U4 U5 U6 U7 U8 U9 U10 U11 U12 U13 U14 U15 U16 U17 U18 U19 U20 U21 U22 U23 U24 U25 U26 U27 U28 U29 U30 U31 U32 U33 U34 U35 U36 U37 U38 U39 U40 U41 U42 U43 U44 U45 U46 U47 U48 U49 U50 U51 U52 U53 U54
Q1 Q2
R22 D1 R23 R24 R25 R26 R27 D3 D4 R28 R29 R30 R31 R32 D5 D2 D6 D8 D7 RN1 R33 R34 R35
S1 S2 S3 S4 S5
VR1
Z1 Z2 Z3
Y1 Y2
F1

REV D

NOTES:

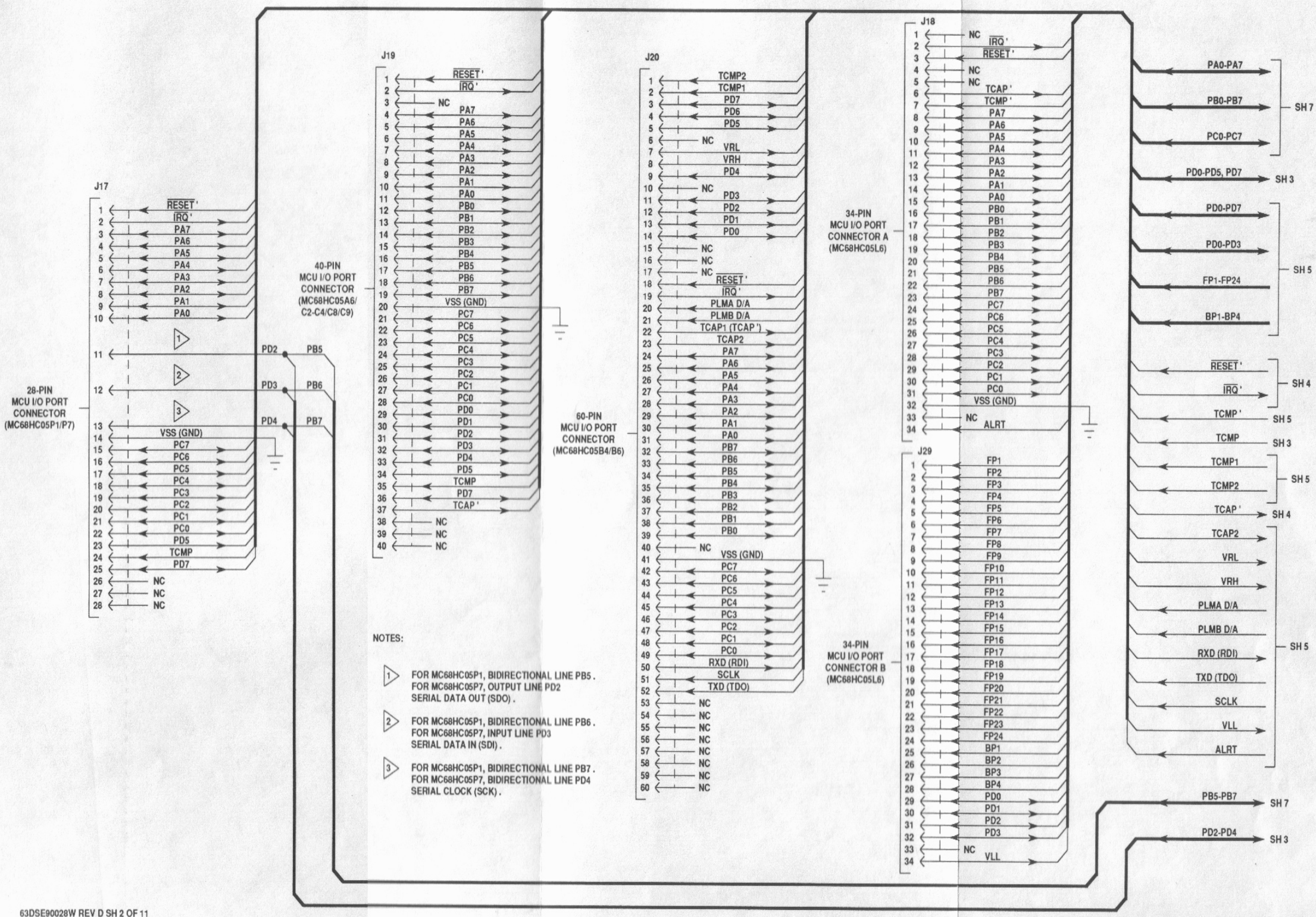
1. UNLESS OTHERWISE SPECIFIED:
   ALL RESISTORS ARE IN OHMS, ±5%, 1/4 W .
   ALL CAPACITORS ARE IN μF .
   ALL VOLTAGES ARE DC .

2. DEVICE TYPE NUMBER LISTED BELOW IS FOR REFERENCE ONLY .
   THE NUMBER VARIES WITH THE MANUFACTURER .



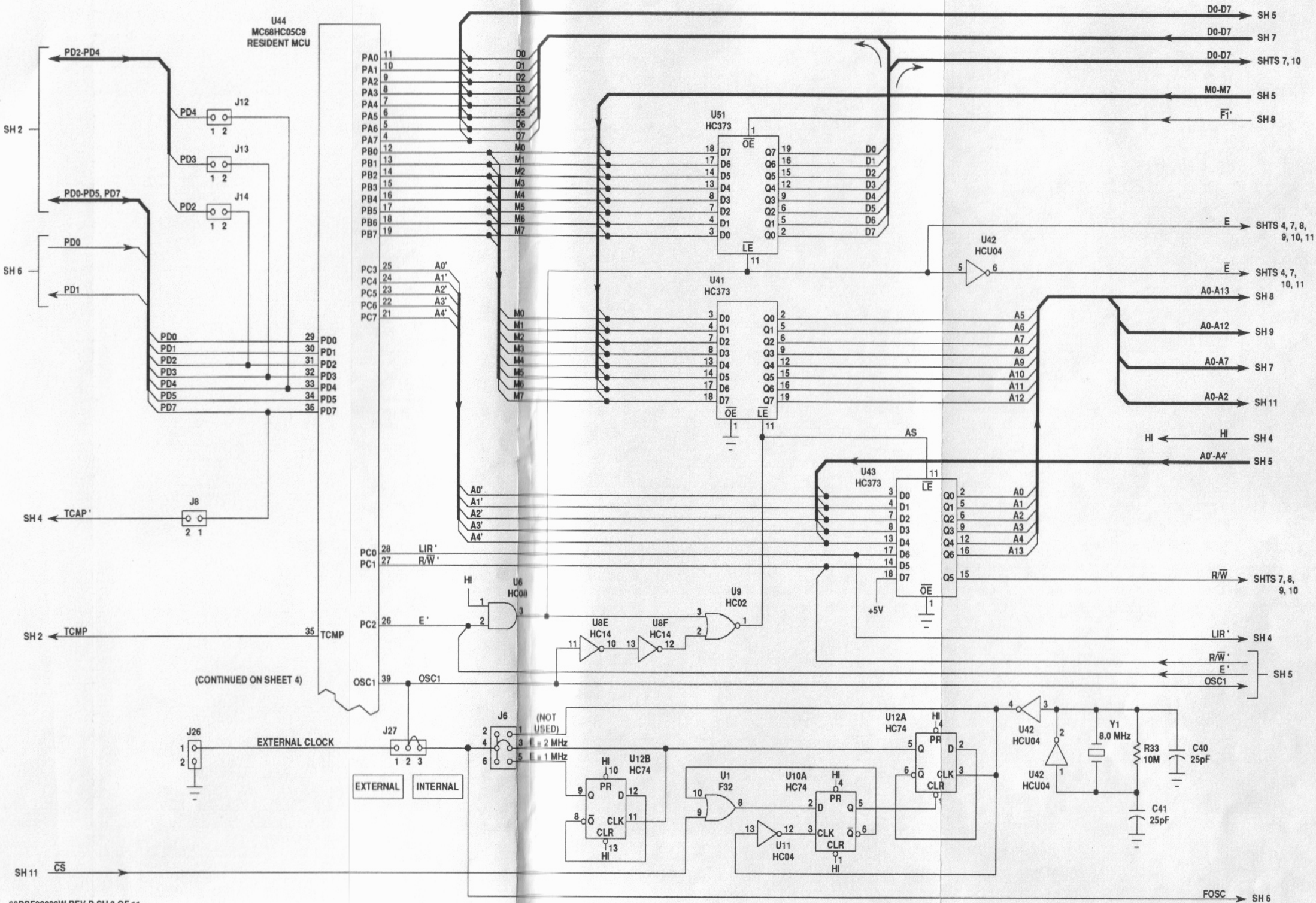
REF DES	DEVICE TYPE	NOTES	GND	+5V	+12V	-12V
U1	74F32		7	14		
U2	74F00		7	14		
U3	MC1488	RS-232C DRIVER	7		14	1
U4	74HC74		7	14		
U5	7407		7	14		
U6	74HC08		7	14		
U7	MC1489A	RS-232C RECEIVER	7	14		
U8	74HC14		7	14		
U9	74HC02		7	14		
U10	74HC74		7	14		
U11	74HC04		7	14		
U12	74HC74		7	14		
U13	74F74		7	14		
U14	74HC00		7	14		
U15	74HC11		7	14		
U16	74HC4066		7		14	
U17	74HC32		7	14		
U18	74HC10		7	14		
U19	74HC10		7	14		
U20	74HC393		7	14		
U21	74HC74		7	14		
U22	74HC08		7	14		
U23	74HC00		7	14		
U24	74HC08		7	14		
U25	XC68HC26	PRU	17, 41	8, 32		
U26	74HC74		7	14		
U27	74HC02		7	14		
U28	74HC10		7	14		
U29	74HC74		7	14		

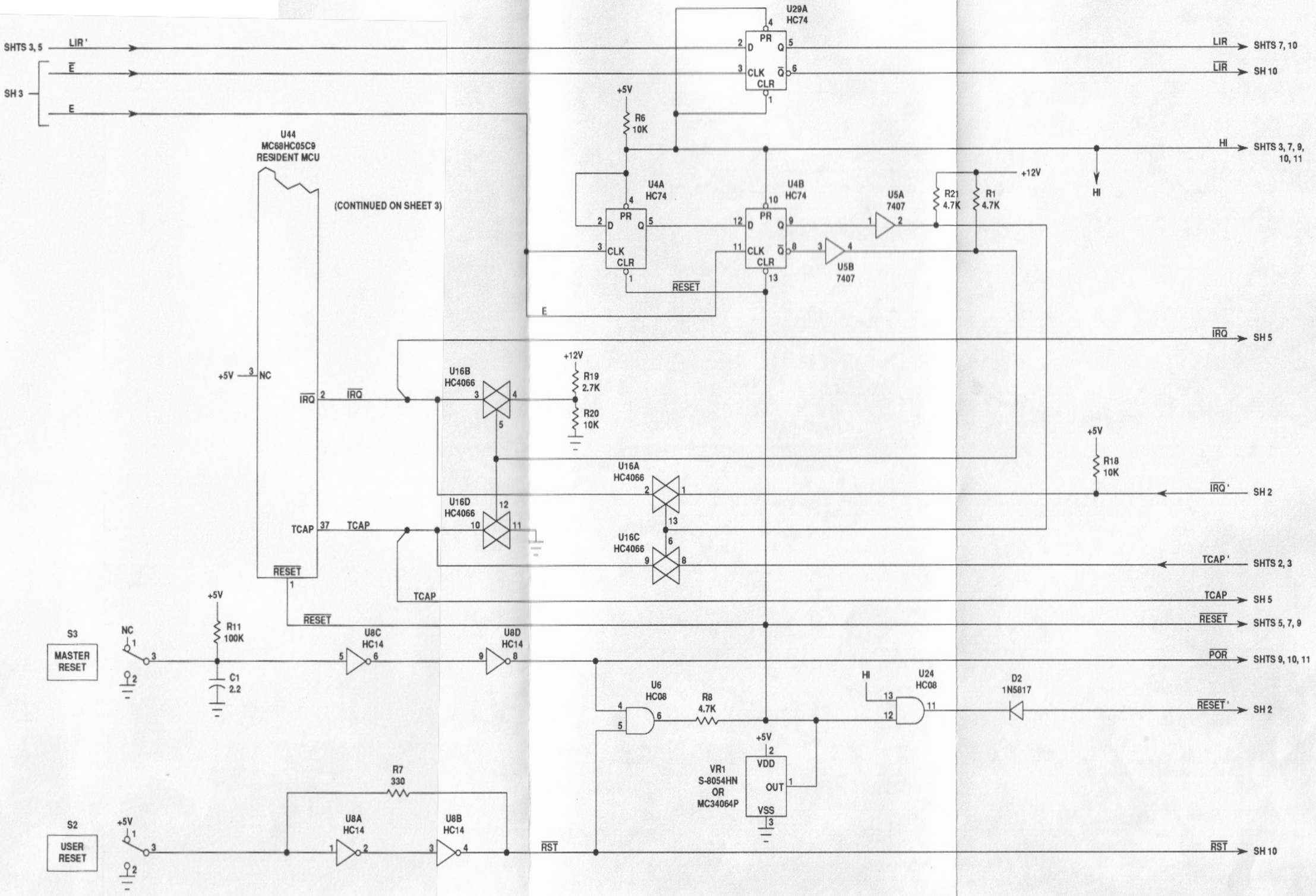
REF DES	DEVICE TYPE	NOTES	GND	+5V	+12V	-12V
U30	74HC74		7	14		
U31	74LS260		7	14		
U32	R65C52P2	DACIA	20	40		
U33	74HC74		7	14		
U34	74HCU04		7	14		
U35	74HC11		7	14		
U36	74HC10		7	14		
U37	74HC32		7	14		
U38	74HC00		7	14		
U39	PLS100	FPLA	14	28		
U40	PLS100	FPLA	14	28		
U41	74HC373		10	20		
U42	74HCU04		7	14		
U43	74HC373		10	20		
U44	MC68HC05C9	RESIDENT MCU	20	3, 40		
U45	6264	8Kx8 EPROM (150ns)	14	28		
U46	2764	8Kx8 EPROM (150ns)	14	28		
U47	6264	8Kx8 EPROM (150ns)	14	28		
U48	6264	8Kx8 EPROM (150ns)	14	28		
U49	74HC374		10	20		
U50	74HC646	BUS TRANSCEIVER	12	24		
U51	74HC373		10	20		
U52	74HC244		10	20		
U53	MC68HC05B6	RESIDENT MCU SOCKET	41	10, 15		
U54	MC68HC05L6	RESIDENT MCU SOCKET	32	1		
Z1	MC68HC805B6	PROG SOCKET (PLCC)	41	10		
Z2	MC68HC705C8/ MC68HC805C4	PROG SOCKET (PLCC)	20	40		
Z3	MC68HC705C8/ MC68HC805C4	PROG SOCKET (DIP)	22	44		

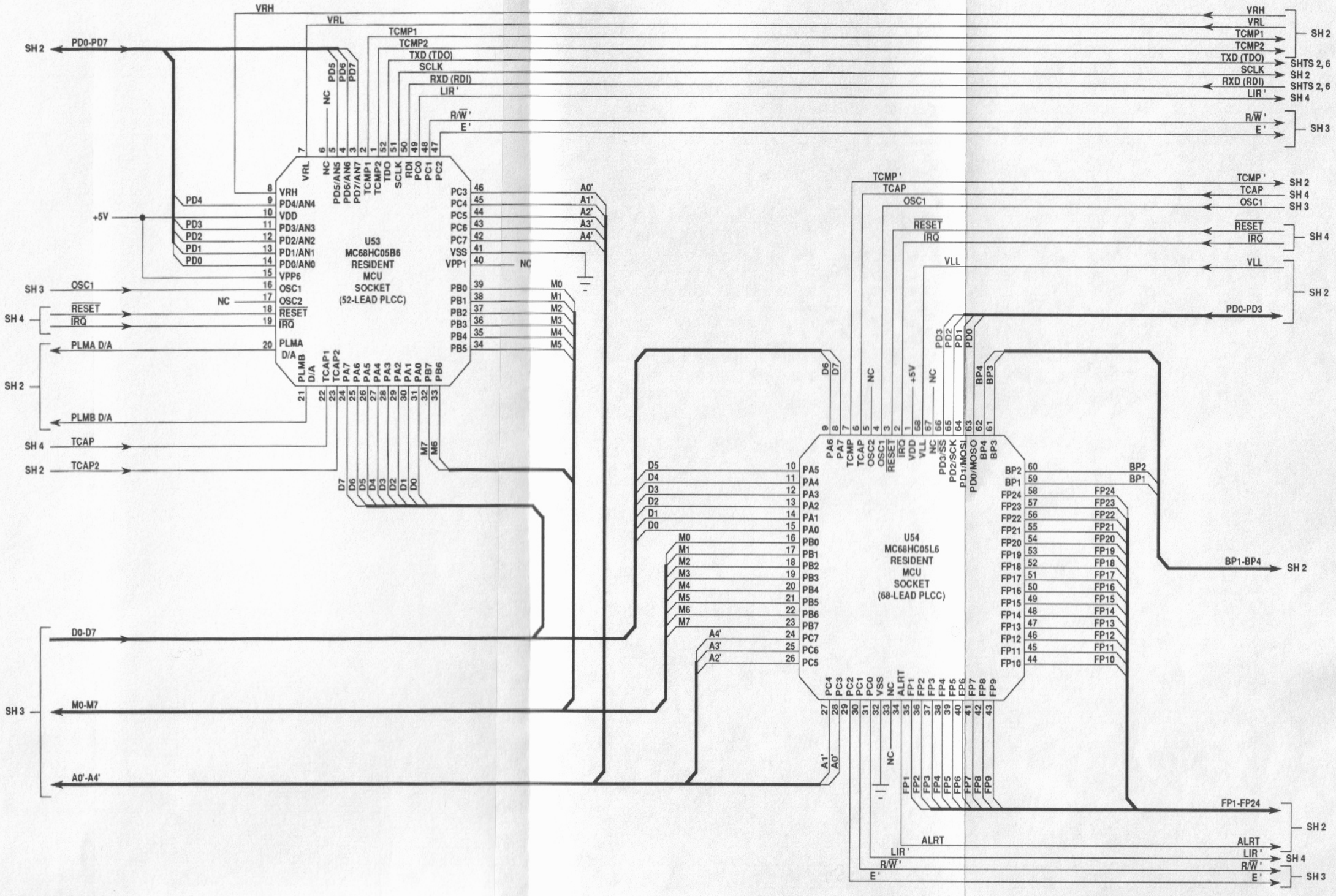


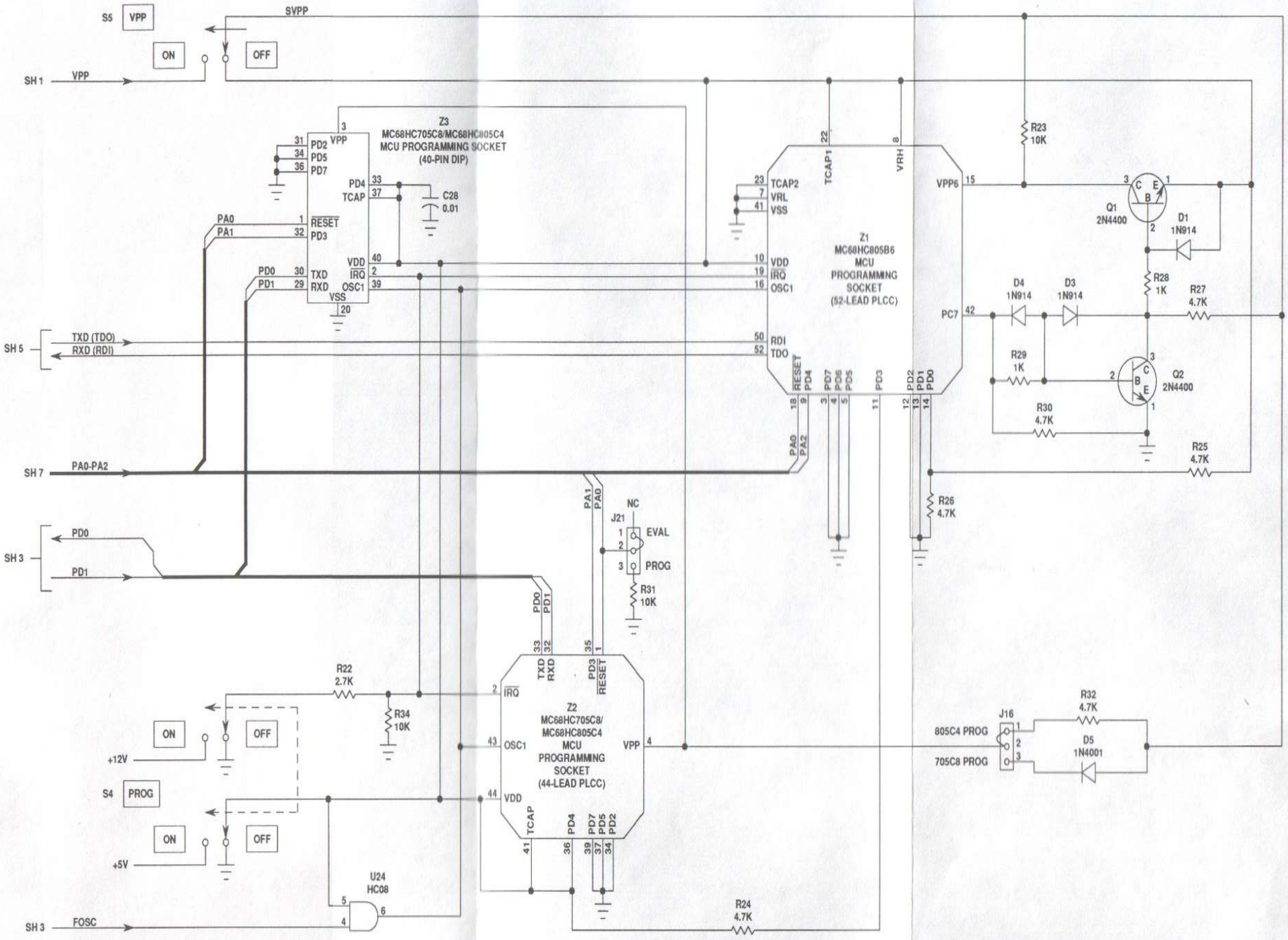
NOTES:

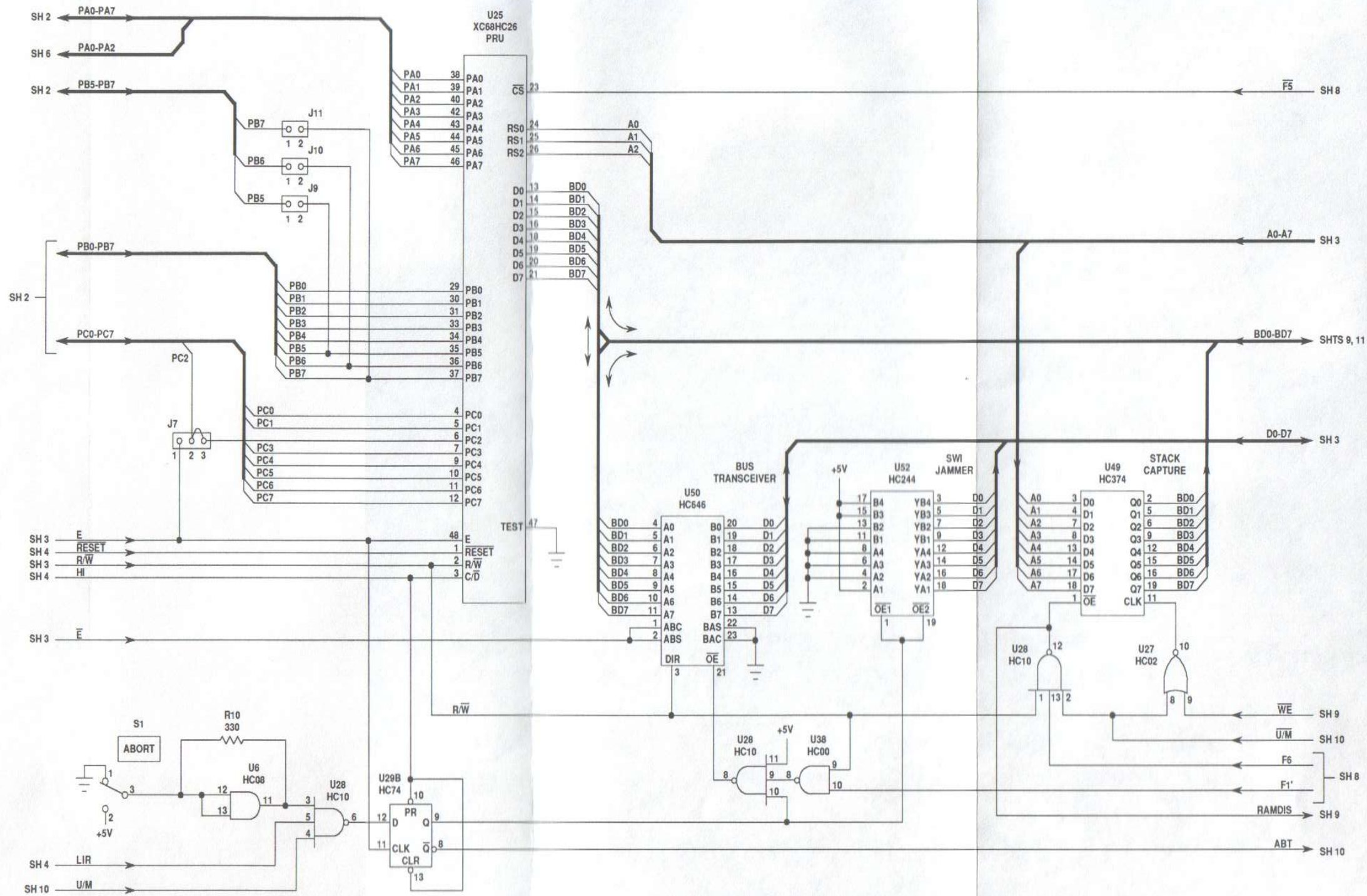
1. FOR MC68HC05P1, BIDIRECTIONAL LINE PB5.
   FOR MC68HC05P7, OUTPUT LINE PD2
   SERIAL DATA OUT (SDO).
2. FOR MC68HC05P1, BIDIRECTIONAL LINE PB6.
   FOR MC68HC05P7, INPUT LINE PD3
   SERIAL DATA IN (SDI).
3. FOR MC68HC05P1, BIDIRECTIONAL LINE PB7.
   FOR MC68HC05P7, BIDIRECTIONAL LINE PD4
   SERIAL CLOCK (SCK).

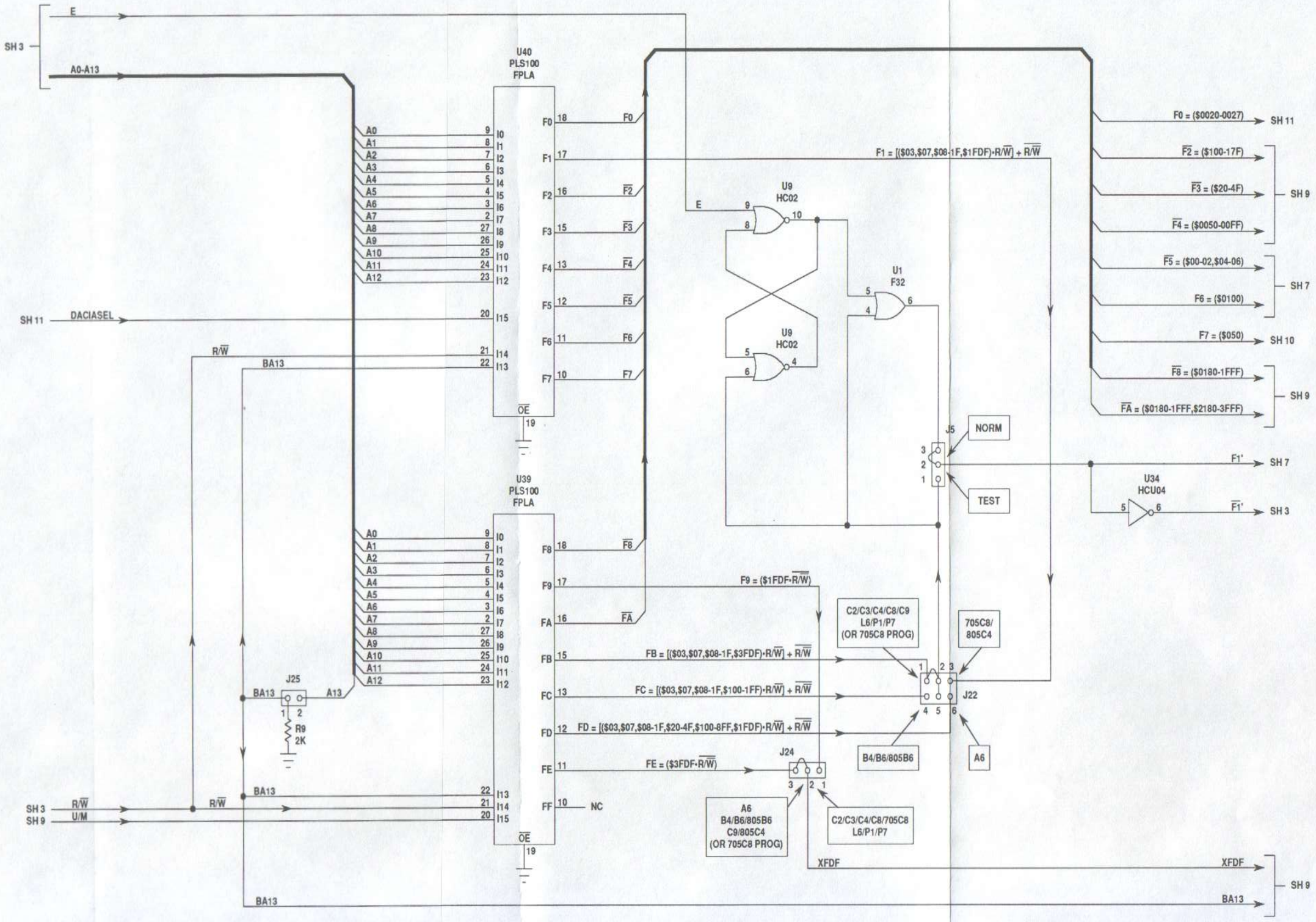

U44
MC68HC05C9
RESIDENT MCU
PD2-PD4
PD0-PD5, PD7
PD0
PD1
J12
J13
J14
SH 2
SH 6
SH 4
SH 2
TCAP '
TCMP
J8
(CONTINUED ON SHEET 4)
U51
HC373
U41
HC373
U43
HC373
U42
HCU04
U6
HC08
U9
HC02
U8E
HC14
U8F
HC14
D0-D7
M0-M7
F1'
E
E
AS
A0-A13
A0-A12
A0-A7
A0-A2
HI
A0'-A4'
R/W
LIR '
R/W '
E '
OSC1
SHTS 4, 7, 8, 9, 10, 11
SHTS 4, 7, 10, 11
SHTS 7, 8, 9, 10
SH 4
SH 5
SH 7
SH 8
SH 9
SH 11
SHTS 7, 10
+5V
J26
J27
J6
EXTERNAL CLOCK
EXTERNAL
INTERNAL
(NOT USED)
E = 2 MHz
E = 1 MHz
U12B
HC74
U12A
HC74
U10A
HC74
U1
F32
U11
HC04
U42
HCU04
Y1
8.0 MHz
R33
10M
C40
25pF
C41
25pF
SH 11
CS
FOSC
SH 6
63DSE90028W REV D SH 3 OF 11

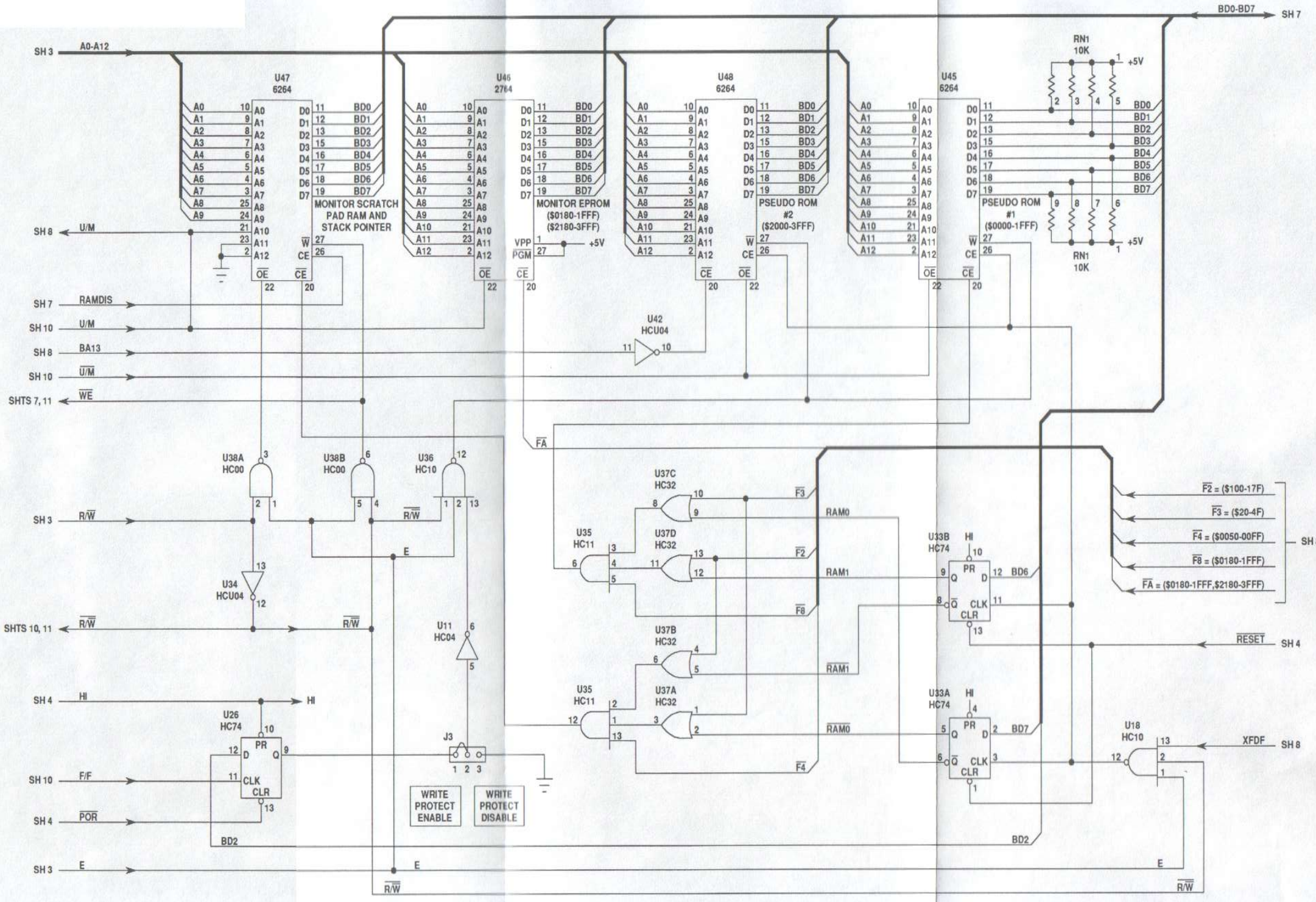

SHTS 3, 5
LIR'
E
SH 3
E
U29A
HC74
PR
D
CLK
CLR
Q
LIR
SHTS 7, 10
LIR
SH 10
+5V
R6
10K
HI
SHTS 3, 7, 9, 10, 11
U44
MC68HC05C9
RESIDENT MCU
(CONTINUED ON SHEET 3)
U4A
HC74
U4B
HC74
U5A
7407
U5B
7407
R21
4.7K
R1
4.7K
+12V
RESET
E
IRQ
SH 5
+5V
NC
IRQ
U16B
HC4066
R19
2.7K
R20
10K
+5V
R18
10K
U16A
HC4066
IRQ'
SH 2
TCAP
U16D
HC4066
U16C
HC4066
TCAP'
SHTS 2, 3
TCAP
TCAP
SH 5
RESET
RESET
SHTS 5, 7, 9
S3
MASTER RESET
NC
R11
100K
C1
2.2
U8C
HC14
U8D
HC14
POR
SHTS 9, 10, 11
U6
HC08
R8
4.7K
U24
HC08
D2
1N5817
RESET'
SH 2
VR1
S-8054HN
OR
MC34064P
VDD
OUT
VSS
R7
330
S2
USER RESET
U8A
HC14
U8B
HC14
RST
RST
SH 10

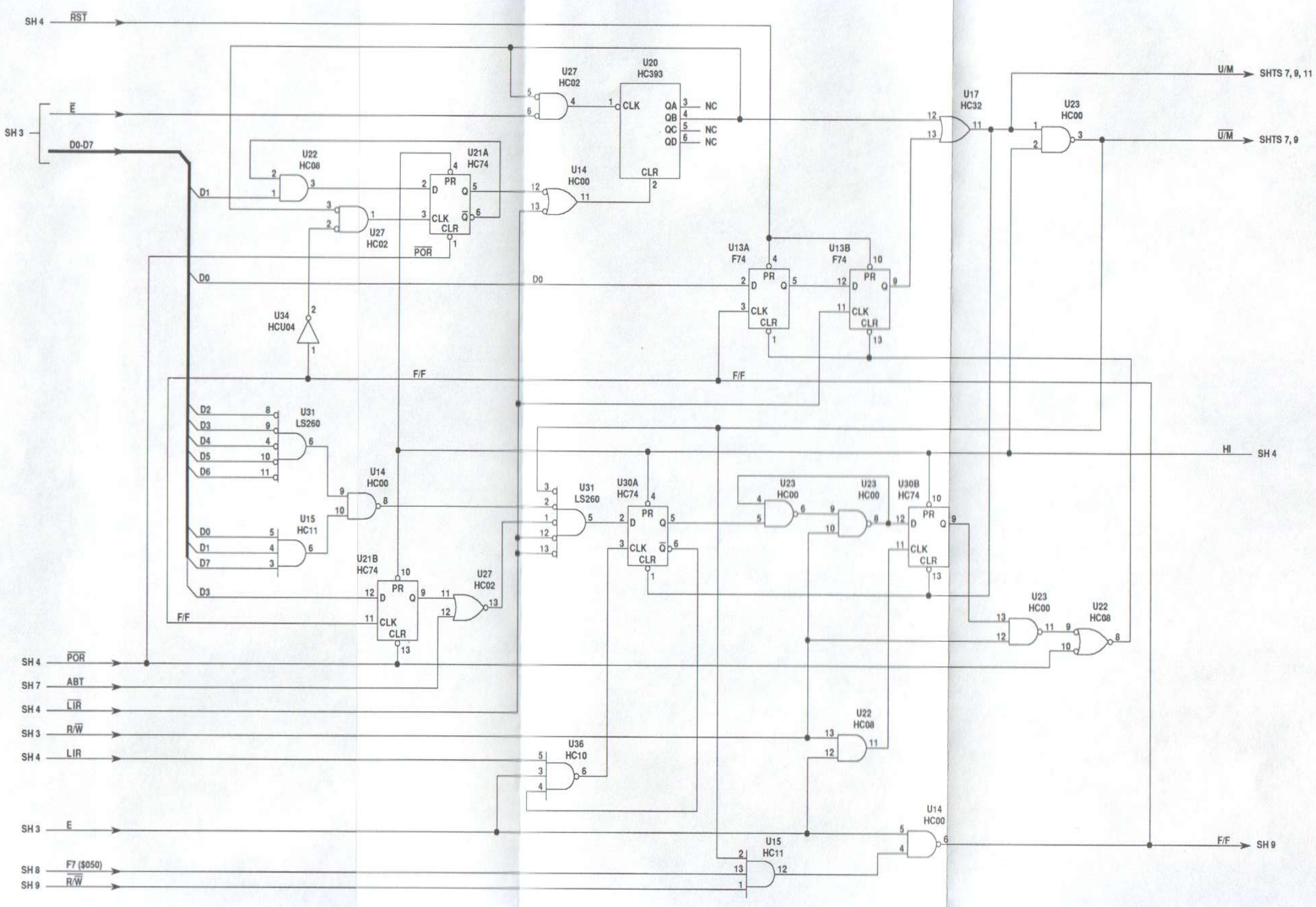

U53
MC68HC05B6
RESIDENT
MCU
SOCKET
(52-LEAD PLCC)
U54
MC68HC05L6
RESIDENT
MCU
SOCKET
(68-LEAD PLCC)
VRH
VRL
TCMP1
TCMP2
TXD (TDO)
SCLK
RXD (RDI)
LIR '
R/W '
E '
PD0-PD7
+5V
OSC1
RESET
IRQ
PLMA D/A
PLMB D/A
TCAP
TCAP2
M0-M7
A0'-A4'
D0-D7
TCMP '
TCAP
OSC1
VLL
PD0-PD3
BP1-BP4
FP1-FP24
ALRT
SH 2
SH 3
SH 4
SHTS 2, 6

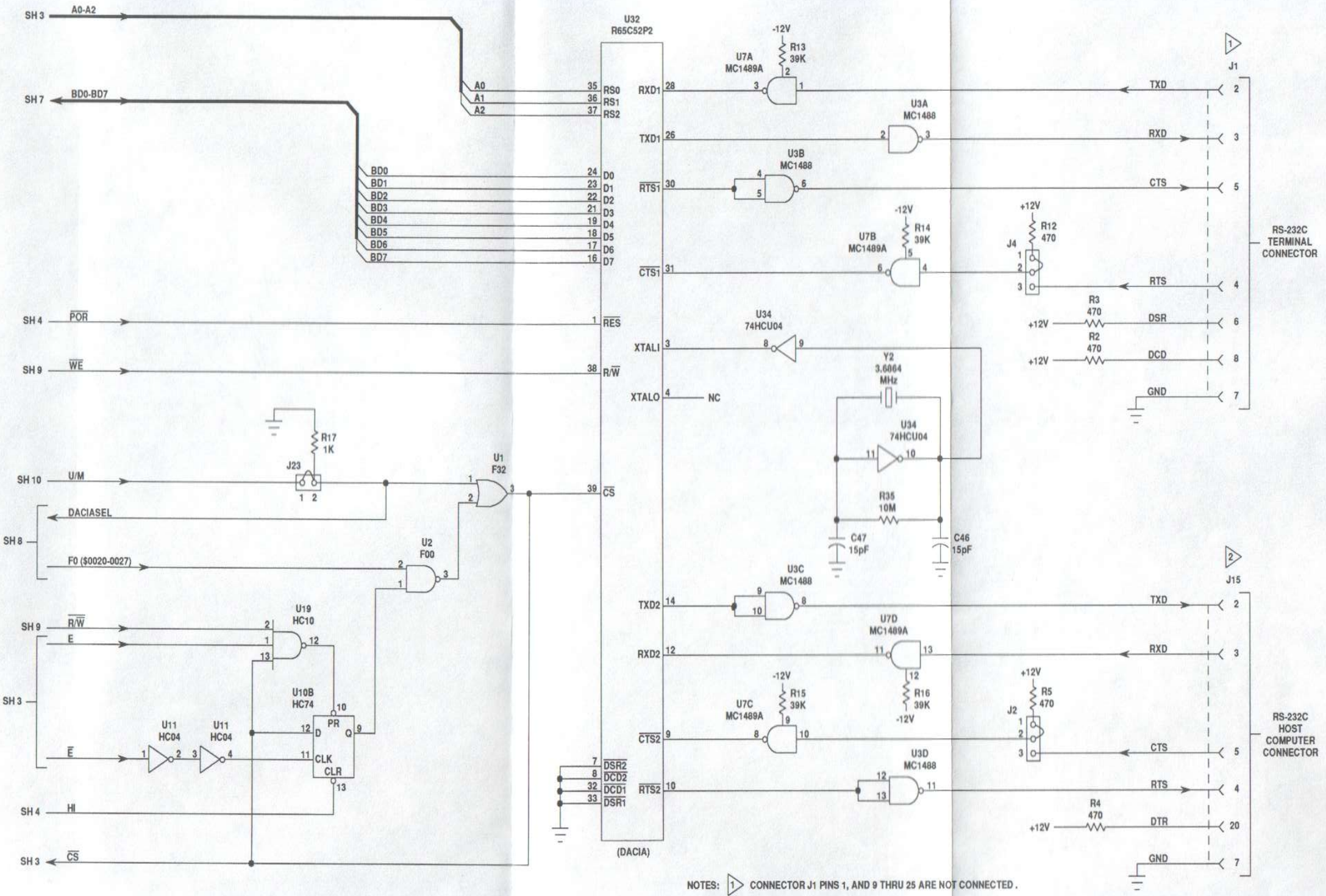

S5
VPP
ON
OFF
SVPP
SH 1
VPP
Z3
MC68HC705C8/MC68HC805C4
MCU PROGRAMMING SOCKET
(40-PIN DIP)
PD2
PD5
PD7
PD4
TCAP
C28
0.01
PA0
PA1
RESET
PD3
PD0
PD1
TXD
RXD
OSC1
VDD
IRQ
VSS
SH 5
TXD (TDO)
RXD (RDI)
SH 7
PA0-PA2
SH 3
PD0
PD1
Z1
MC68HC805B6
MCU
PROGRAMMING
SOCKET
(52-LEAD PLCC)
TCAP2
VRL
VSS
TCAP1
VRH
VDD
IRQ
OSC1
RDI
TDO
RESET
PD4
PD7
PD6
PD5
PD3
PD2
PD1
PD0
PA0
PA2
R23
10K
VPP6
PC7
Q1
2N4400
D1
1N914
D4
1N914
D3
1N914
R28
1K
R27
4.7K
R29
1K
Q2
2N4400
R30
4.7K
R25
4.7K
R26
4.7K
J21
NC
EVAL
PROG
R31
10K
PA1
PA0
PD0
PD1
R22
2.7K
R34
10K
Z2
MC68HC705C8/
MC68HC805C4
MCU
PROGRAMMING
SOCKET
(44-LEAD PLCC)
IRQ
OSC1
VDD
VPP
TXD
RXD
PD3
RESET
TCAP
PD4
PD7
PD5
PD2
+12V
+5V
S4
PROG
U24
HC08
FOSC
R24
4.7K
J16
805C4 PROG
705C8 PROG
R32
4.7K
D5
1N4001


SH 2
PA0-PA7
SH 6
PA0-PA2
SH 2
PB5-PB7
PB7
J11
PB6
J10
PB5
J9
SH 2
PB0-PB7
PC0-PC7
PC2
J7
U25
XC68HC26
PRU
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
TEST
CS
RS0
RS1
RS2
D0
D1
D2
D3
D4
D5
D6
D7
F5
SH 8
A0
A1
A2
A0-A7
SH 3
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
BD0-BD7
SHTS 9, 11
D0-D7
SH 3
BUS
TRANSCEIVER
U50
HC646
+5V
U52
HC244
SWI
JAMMER
U49
HC374
STACK
CAPTURE
SH 3
E
SH 4
RESET
SH 3
R/W
SH 4
HI
SH 3
E
R/W
S1
ABORT
R10
330
+5V
U6
HC08
U28
HC10
U29B
HC74
PR
CLK
CLR
SH 4
LIR
SH 10
U/M
U28
HC10
+5V
U38
HC00
U28
HC10
U27
HC02
WE
SH 9
U/M
SH 10
F6
F1'
SH 8
RAMDIS
SH 9
ABT
SH 10


SH 3
E
A0-A13
U40
PLS100
FPLA
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
I0
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
I11
I12
F0
F1
F2
F3
F4
F5
F6
F7
OE
SH 11
DACIASEL
I15
R/W
BA13
I14
I13
U39
PLS100
FPLA
F8
F9
FA
FB
FC
FD
FE
FF
NC
J25
R9
2K
A13
SH 3
SH 9
U/M
U9
HC02
U1
F32
J5
NORM
TEST
U34
HCU04
F1'
J22
J24
C2/C3/C4/C8/C9
L6/P1/P7
(OR 705C8 PROG)
705C8/
805C4
B4/B6/805B6
A6
C9/805C4
C2/C3/C4/C8/705C8
XFDF
F0 = ($0020-0027)
F1 = [($03,$07,$08-1F,$1FDF)·R/W] + R/W
F2 = ($100-17F)
F3 = ($20-4F)
F4 = ($0050-00FF)
F5 = ($00-02,$04-06)
F6 = ($0100)
F7 = ($050)
F8 = ($0180-1FFF)
FA = ($0180-1FFF,$2180-3FFF)
F9 = ($1FDF·R/W)
FB = [($03,$07,$08-1F,$3FDF)·R/W] + R/W
FC = [($03,$07,$08-1F,$100-1FF)·R/W] + R/W
FD = [($03,$07,$08-1F,$20-4F,$100-8FF,$1FDF)·R/W] + R/W
FE = ($3FDF·R/W)
SH 11
SH 9
SH 7
SH 10
SH 3


BD0-BD7
SH 7
SH 3
A0-A12
U47
6264
MONITOR SCRATCH PAD RAM AND STACK POINTER
U46
2764
MONITOR EPROM ($0180-1FFF) ($2180-3FFF)
U48
6264
PSEUDO ROM #2 ($2000-3FFF)
U45
6264
PSEUDO ROM #1 ($0000-1FFF)
RN1
10K
+5V
SH 8
U/M
SH 7
RAMDIS
SH 10
U/M
SH 8
BA13
SH 10
U/M
SHTS 7, 11
WE
U42
HCU04
FA
U38A
HC00
U38B
HC00
U36
HC10
SH 3
R/W
E
U34
HCU04
SHTS 10, 11
R/W
U11
HC04
U37C
HC32
U37D
HC32
U37B
HC32
U37A
HC32
U35
HC11
U35
HC11
F3
F2
F8
F4
RAM0
RAM1
RAM1
RAM0
U33B
HC74
U33A
HC74
F2 = ($100-17F)
F3 = ($20-4F)
F4 = ($0050-00FF)
F8 = ($0180-1FFF)
FA = ($0180-1FFF,$2180-3FFF)
SH 8
RESET
SH 4
SH 4
HI
HI
U26
HC74
J3
WRITE PROTECT ENABLE
WRITE PROTECT DISABLE
SH 10
F/F
SH 4
POR
BD2
SH 3
E
R/W
U18
HC10
XFDF
SH 8


SH 4
RST
SH 3
E
D0-D7
U27
HC02
U20
HC393
CLK
QA
QB
QC
QD
NC
CLR
U17
HC32
U/M
SHTS 7, 9, 11
U23
HC00
U/M
SHTS 7, 9
U22
HC08
U21A
HC74
PR
D
Q
CLK
CLR
U14
HC00
D1
U27
HC02
POR
D0
U13A
F74
U13B
F74
U34
HCU04
F/F
D2
D3
D4
D5
D6
U31
LS260
U14
HC00
HI
SH 4
D0
D1
D7
U15
HC11
U21B
HC74
U27
HC02
D3
F/F
U31
LS260
U30A
HC74
U23
HC00
U23
HC00
U30B
HC74
U23
HC00
U22
HC08
SH 4
POR
SH 7
ABT
SH 4
LIR
SH 3
R/W
SH 4
LIR
U36
HC10
U22
HC08
SH 3
E
U14
HC00
F/F
SH 9
SH 8
F7 ($050)
SH 9
R/W
U15
HC11



NOTES: 1 CONNECTOR J1 PINS 1, AND 9 THRU 25 ARE NOT CONNECTED.

2 CONNECTOR J15 PINS 1, 6, 8 THRU 19, 21, AND 21 THRU 25 ARE NOT CONNECTED.

APPENDIX A

S-RECORD INFORMATION

INTRODUCTION

The S-record format for output modules was devised for the purpose of encoding programs or data files in a printable format for transportation between computer systems. The transportation process can thus be visually monitored and the S-records can be more easily edited.

S-RECORD CONTENT

When viewed by the user, S-records are essentially character strings made of several fields which identify the record type, record length, memory address, code/data, and checksum. Each byte of binary data is encoded as a 2-character hexadecimal number: the first character representing the high-order 4 bits, and the second the low-order 4 bits of the byte.

The 5 fields which comprise an S-record are shown below:

where the fields are composed as follows:

FIELD	PRINTABLE CHARACTERS	CONTENTS
Type	2	S-record type - S0, S1, etc.
Record length	2	The count of the character pairs in the record, excluding the type and record length.
Address	4, 6, or 8	The 2-, 3-, or 4-byte address at which the data field is to be loaded into memory.
Code/data	0-2n	From 0 to n bytes of executable code, memory loadable data, or descriptive information. For compatibility with teletypewriters, some programs may limit the number of bytes to as few as 28 (56 printable characters in the S-record).
Checksum	2	The least significant byte of the one's complement of the sum of the values represented by the pairs of characters making up the record length, address, and the code/data fields.

A

Each record may be terminated with a CR/LF/NULL. Additionally, an S-record may have an initial field to accommodate other data such as line numbers generated by some time-sharing systems.

Accuracy of transmission is ensured by the record length (byte count) and checksum fields.

S-RECORD TYPES

Eight types of S-records have been defined to accommodate the several needs of the encoding, transportation, and decoding functions. The various Motorola upload, download, and other record transportation control programs, as well as cross assemblers, linkers, and other file-creating or debugging programs, utilize only those S-records which serve the purpose of the program. For specific information on which S-records are supported by a particular program, the user manual for that program must be consulted.

NOTE

The EVM monitor supports only the S1 and S9 records. All data before the first S1 record is ignored. Thereafter, all records must be S1 type until the S9 record terminates data transfer.

An S-record format module may contain S-records of the following types:

S0	The header record for each block of S-records. The code/data field may contain any descriptive information identifying the following block of S-records. The address field is normally zeroes.
S1	A record containing code/data and the 2-byte address at which the code/data is to reside.
S2-S8	Not applicable to EVM.
S9	A termination record for a block of S1 records. The address field may optionally contain the 2-byte address of the instruction to which control is to be passed. If not specified, the first entry point specification encountered in the object module input will be used. There is no code/data field.

A

Only one termination record is used for each block of S-records. Normally, only one header record is used, although it is possible for multiple header records to occur.

S-RECORD CREATION

S-record format programs may be produced by several dump utilities, debuggers, or several cross assemblers or cross linkers. Several programs are available for downloading a file in S-record format from a host system to an 8-bit or 16-bit microprocessor-based system.

S-RECORD EXAMPLE

Shown below is a typical S-record format module, as printed or displayed:

```
S006000048444521B
S11300002B5F245F2212226A000424290008237C2A
S113001000020008000826290018538123410018l3
S11300204lE900084E42234300182342000824A952
S107003000144ED492
S9030000FC
```

The above module consists of an S0 header record, four S1 code/data records, and an S9 termination record.

The S0 header record is comprised of the following character pairs:

S0	S-record type S0, indicating a header record.
06	Hexadecimal 06 (decimal 6), indicating six character pairs (or ASCII bytes) follow.
00 00	Four-character 2-byte address field, zeroes.
48 44 52	ASCII H, D, and R - "HDR".
1B	Checksum of S0 record.

The first S1 code/data record is explained as follows:

S1	S-record type S1, indicating a code/data record to be loaded/verified at a 2-byte address.
13	Hexadecimal 13 (decimal 19), indicating 19 character pairs, representing 19 bytes of binary data, follow.
00	Four-character 2-byte address field; hexadecimal address 0000, indicates location where the following data is to be loaded.

A

The next 16 character pairs are the ASCII bytes of the actual program code/data. In this assembly language example, the hexadecimal opcodes of the program are written in sequence in the code/data fields of the S1 records:

OPCODE	INSTRUCTION	
28 5F	BHCC	$0161
24 5F	BCC	$0163
22 12	BHI	$0118
22 6A	BHI	$0172
00 04 24	BRSET	0,$04,$012F
29 00	BHCS	$010D
08 23 7C	BRSET	4,$23,$018C

. (Balance of this code is continued in the code/data fields of the
. remaining S1 records, and stored in memory location 0010, etc..)
.

2A Checksum of the first S1 record.

The second and third S1 code/data records each also contain $13 (19) character pairs and are ended with checksums 13 and 52, respectively. The fourth S1 code/data record contains 07 character pairs and has a checksum of 92.

The S9 termination record is explained as follows:

S9 S-record type S9, indicating a termination record.

03 Hexadecimal 03, indicating three character pairs (3 bytes) follow.

00 Four-character 2-byte address field, zeroes.
00

FC Checksum of S9 record.

Each printable character in an S-record is encoded in hexadecimal (ASCII in this example) representation of the binary bits which are actually transmitted. For example, the first S1 record above is sent as shown below.

TYPE				LENGTH				ADDRESS								CODE/DATA										CHECKSUM			
S		1		1		3		0		0		0		0		2		0		6		F		...		2		A	
5	3	3	1	3	1	3	3	3	0	3	0	3	0	3	0	3	2	3	8	3	5	4	8	...		3	2	4	1
0101	0011	0011	0001	0011	0001	0011	0011	0011	0000	0011	0000	0011	0000	0011	0000	0011	0010	0011	1000	0011	0101	0100	0110	...		0011	0010	0100	0001

A