M68705EVM

Evaluation Module User's Manual

MOTOROLA

M68705EVM/AD5

November 1990

M68705EVM

EVALUATION MODULE

USER'S MANUAL

Information contained in this document applies to REVision (C) M68705EVM Evaluation Modules.

Fifth Edition

Fourth Edition May 1987

PREFACE

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

PAGE

LIST OF ILLUSTRATIONS

PAGE

FIGURE

LIST OF TABLES

PAGE

TABLE

CHAPTER 1

GENERAL INFORMATION

1

1.1 INTRODUCTION

This manual provides general information, hardware preparation, installation instructions, operating instructions, functional description, and support information for the M68705EVM 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 M68705 HMOS MCU family devices.

Monitor/debugger firmware

One-line assembler/disassembler

Host computer download capability

Dual 4K-byte memory maps:

4K monitor EPROM
4K user pseudo ROM

EPROM MCU programmer

MCU (28-pin and 40-pin) extension I/O port s

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 MHz bus operation (4 MHz, crystal controlled, divided-by-four)
External Clock	4 MHz or smaller, divided-by-four (1 MHz or smaller bus operation)
Memory size:	
RAM	112 bytes
Pseudo ROM	3968 bytes
MCU extension I/O ports:	
Ports A, B, C	HMOS compatible (input/output)
Port D	HMOS compatible (input only and/ or analog-to digital converter)
Terminal/host I/O ports	RS-232 compatible (RxD and TxD only)
Temperature:	
Operating	+25 degrees C
Storage	-40 to +85 degrees C
Relative humidity	0 to 90% (non-condensing)
Power requirements:	
Module	+5 Vdc @ 1.0 A (max) +12 Vdc @ 0.1 A (max) -12 Vdc @ 0.1 A (max)
Programmer (VPP)	+27 Vdc @ 50 mA (max)
Dimensions:	
Width	13.25 in. (33.6 cm)
Length	9.75 in. (24.8 cm)

1.4 GENERAL DESCRIPTION

The EVM provides a tool for designing, debugging, and evaluating MC6805P2/P4/P6, MC6805R2/R3, MC6805U2/U3, MC68705P3/P5, MC68705R3, and MC68705U3/U5 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.

The M6805 Family of HMOS MCU devices are evaluated (emulated) by the EVM resident MC6805R3 MCU. Entering data, program debugging, and EPROM MCU programming is accomplished by the monitor EPROM firmware via an external RS-232C compatible terminal connected to the EVM terminal port connector.

Independent hardware selectable 300-19.2K baud rate selection capabilities are provided for the terminal and host I/O ports. 28-pin and 40-pin MCU I/O port connectors facilitate interconnection of the EVM to the target system for evaluation purposes.

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.

MCU code may be generated using the resident one-line assembler/disassembler, or may be downloaded to the user program RAM 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.

Jumper selectable options such as MC6805R2 or MC68705R3/U3 MCU, 2K or 4K user map, terminal/host baud rate, timer input/prescaler and clock input selection are provided on the EVM, as well as an EPROM MCU programmer. The EPROM MCU programmer, under monitor firmware control, enables the user to program the contents of either 28-pin or 40-pin Dual-In-line Package (DIP) EPROM MCU devices. Switches allow user control of the reset and abort functions, and EPROM 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 MCU I/O port extension cable assembly*

Notes:

(1) * Refer to Chapter 2 for details.

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

CHAPTER 2

HARDWARE PREPARATION AND INSTALLATION INSTRUCTIONS

2.1 INTRODUCTION

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

2

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.

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

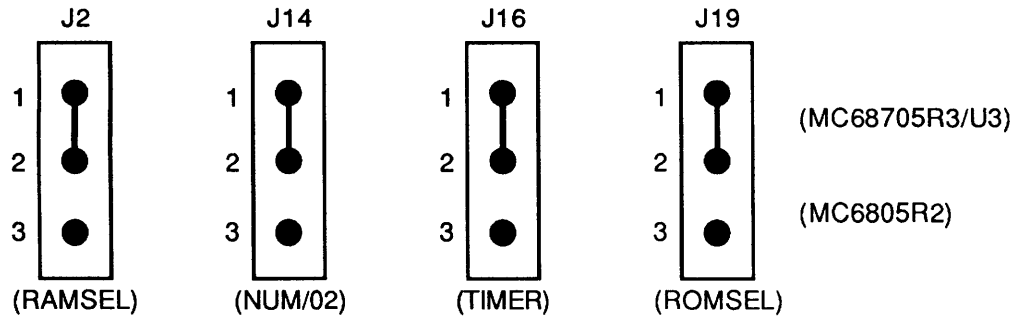
Connectors J1, J3, J4, J9, J18, and P1 facilitate interconnection of external equipment to the EVM. Switches S1 through S5 provide user control of the EVM. Indicators CR1, CR2, CR7, and CR8 provide operational status indications of the EPROM MCU programming circuitry. Refer to Chapter 3 for switch and indicator descriptions.

Reference designations J5 through J7 are not assigned. Jumper header locations J2, J8, J10 through J17, and J19 through J25 provide the following selection capabilities:

a. Resident MCU select (J2, J14, J16, and J19)
b. Host CTS and DCD signal enable (J8 and J11)
c. User map select (J10)
d. Baud rate select (J12 and J13)
e. Timer input select (J15)
f. Clock input select (J17)
g. Timer prescaler select (J20)
h. EPROM MCU programming select (J21 thru J24)
i. Write protect disable (J25)

2.3.1 Resident MCU Select Headers (J2, J14, J16, and J19)

Jumper headers J2, J14, J16, and J19 are used to configure four control lines of the resident MCU socket U21 for either an MC6805R2 or MC68705R3/U3 device. These four control lines affect MCU pins 7, 8, 15, and 16 (NUM/02, TIMER, ROMSEL, and RAMSEL) in the configuration of the EVM resident MCU. The EVM is shipped with an UV erased MC68705R3 MCU installed in socket U21, and jumper headers J2, J14, J16, and J19 are factory-configured as shown below.



If evaluation of the MC68705R2 device is desired, the user must replace the resident MC68705R3 device with the MC6805R2 device and reposition the fabricated jumpers on headers J2, J14, J16, and J19 between pins 2 and 3. Refer to paragraph 2.3.5 and 2.3.7 for additional reconfiguration information.

NOTE

The MC68705R3 device may be used to develop MC6805R2 code. The user must utilize the Timer Control Register (TCR) $009 to set the prescaler values (as specified for the mask ROM device) after power on reset (POR) or normal reset operations. The MC68705R3 device provides additional capabilities with respect to the MC6805R2. These capabilities must be limited by the user.

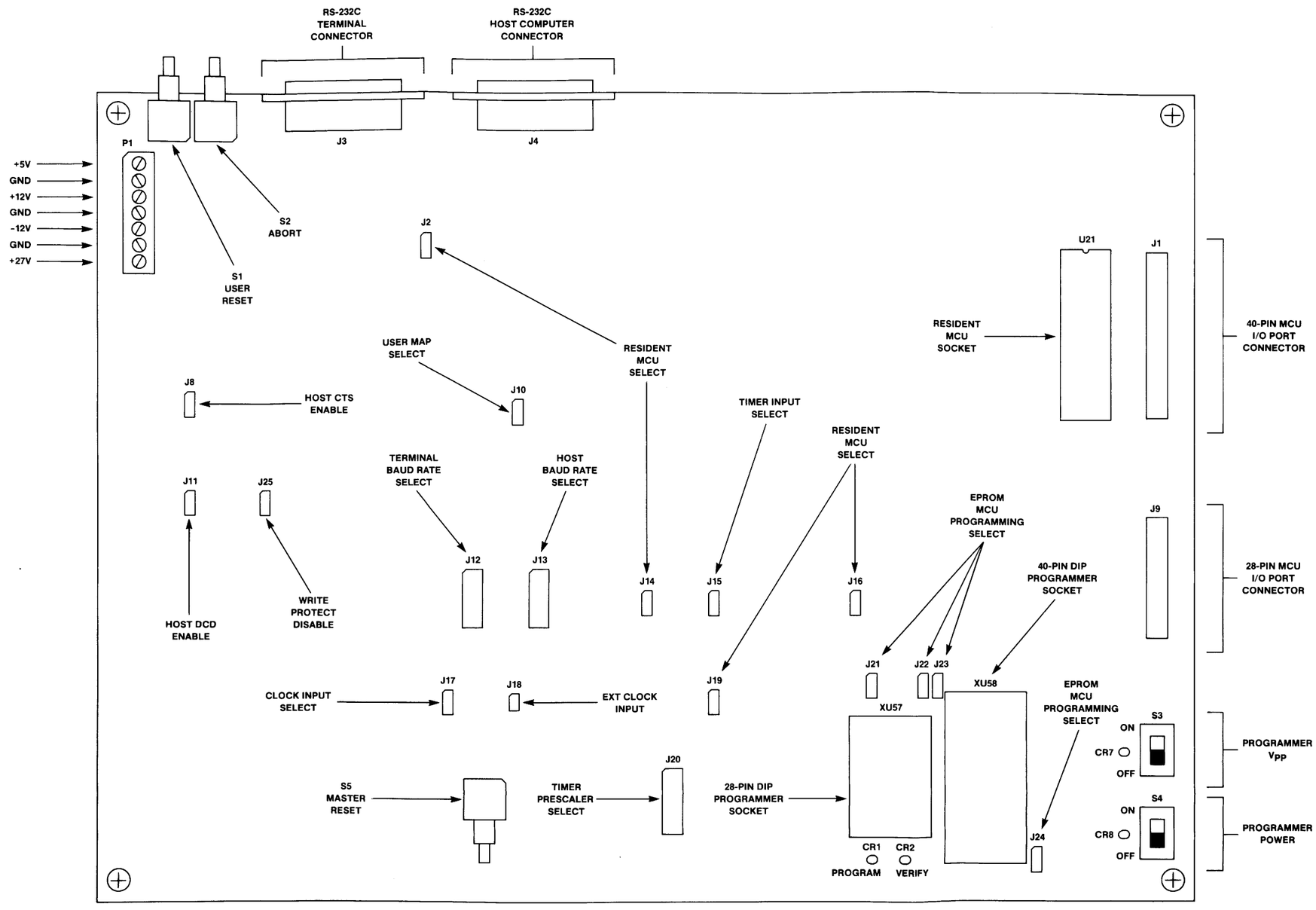
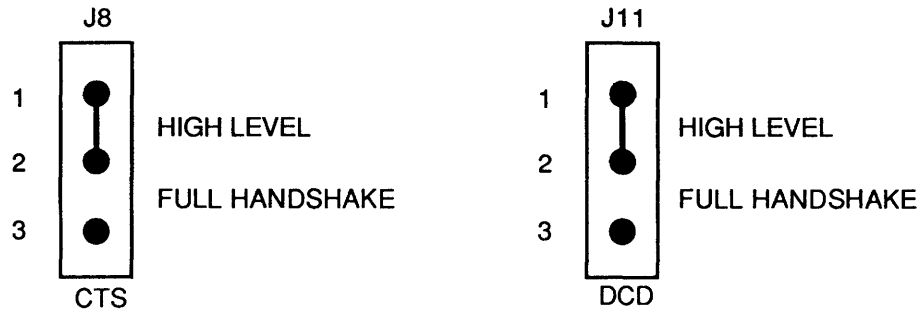


FIGURE 2-1. EVM Connector, Switch, Indicator, and Jumper Header Location Diagram

2.3.2 Host CTS and DCD Signal Enable Headers (J8 and J11)

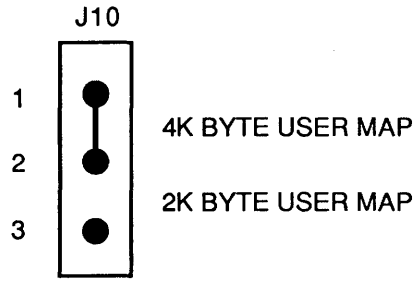
Jumper headers J8 and J11 are used to configure the RS-232C host computer I/O port for semi or full handshaking operation. The EVM is factory-configured for semi handshaking (high level) operation. This semi handshaking configuration (high level) is accomplished by the installation of fabricated jumpers on jumper headers J8 and J11 as shown below.



Should the host computer or modem require full handshake capability, jumpers are to be repositioned between pins 2 and 3. Refer to the schematic diagrams located in Chapter 5 for CTS and DCD signal wiring information.

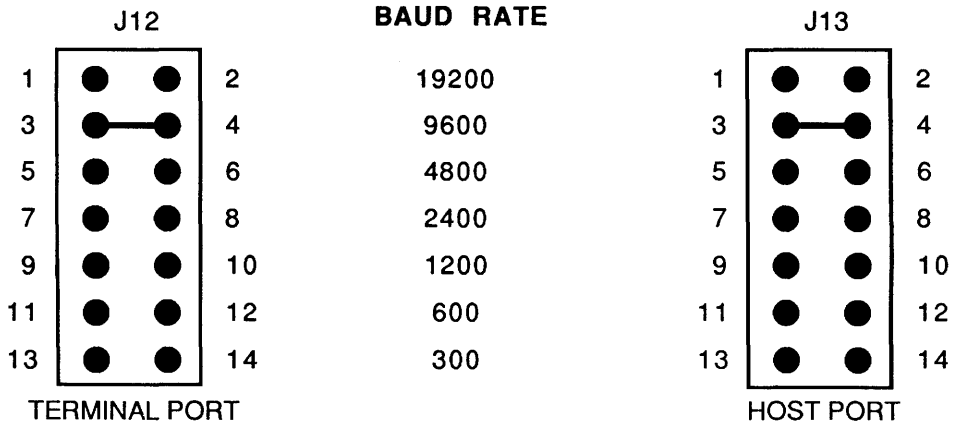
2.3.3 User Map Select Header (J10)

Jumper header J10 is used to select either a 2K-byte user map for MC68705P3 evaluation or a 4K-byte user map for MC68705R3/U3 evaluation. The EVM is factory-configured and shipped with the 4K-byte user map selected. This is accomplished by the installation of a fabricated jumper on pins 1 and 2 as shown below.



2.3.4 Baud Rate Select Headers (J12 and J13)

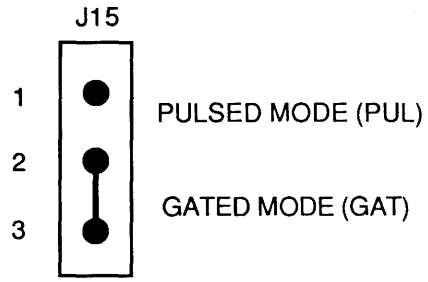
Jumper headers J12 and J13 are used to select the baud rate for the RS-232C terminal and host computer I/O ports. The EVM is factory-configured and shipped with the jumpers installed for 9600 baud rate operation as shown below.



In the transparent mode, which allows direct communications between the terminal and the host, the host computer baud rate should be set one increment lower than the terminal (e.g., terminal = 2400 baud, host = 1200 baud). If both baud rates are the same, occasionally a character may not be received by the terminal. Download operation is not affected by equal baud rates.

2.3.5 Timer Input Select Header (J15)

Jumper header J15 is used to select either a pulsed (PUL) or gated (GAT) mode of operation for the MC6805R2 MCU, if installed in socket U21. The EVM is factory-configured and shipped with the gated mode selected. This is accomplished by the installation of a fabricated jumper on pins 2 and 3 as shown below.

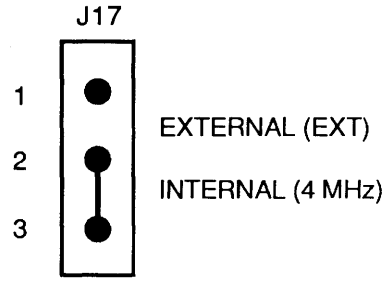


The EVM simulates both the gated and the pulsed modes of the MC6805R2 MCU. In the gated mode, the internal phase 2 (02) clock is gated by the signal applied to the MCU timer input. In the pulsed mode, the timer counts the input pulses applied to the MCU timer input. When the pulsed mode is required, the fabricated jumper must be repositioned between pins 1 and 2.

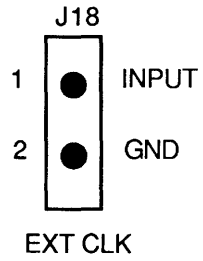
The MC68705R3/U3 MCU timer is software programmable. Both the timer input and prescaler selection capabilities (via jumper headers J15 and J20, respectively) are bypassed when reconfiguration for either the MC68705R3/U3 MCUs are made via the MCU select header J16. (Refer to paragraph 2.3.1.)

2.3.6 Clock Input Select Header (J17)

Jumper header J17 is used to select either internal or external clock source to be used by the EVM. The internal clock source is a 4 MHz crystal. The EVM is factory-configured and shipped with the clock input selected to internal 4 MHz clock operation. This is accomplished by the installation of a fabricated jumper on pins 2 and 3 as shown below. To select an external clock source, remove jumper from pins 2 and 3, and reinstall jumper on pins 1 and 2.

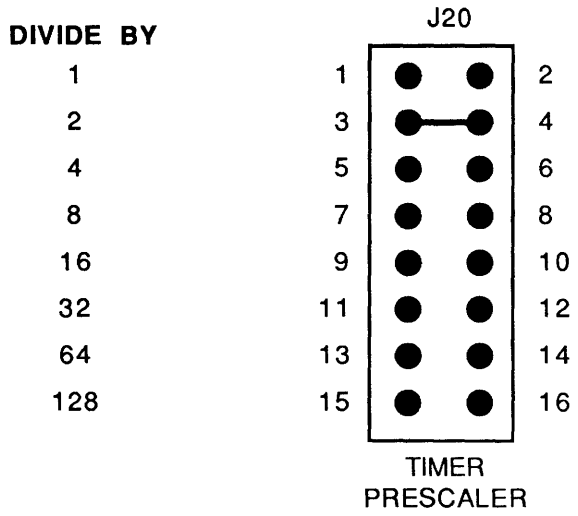


When the external clock source is selected, connector J18 (2-pin header) is used to facilitate interconnection of the external clock source.



2.3.7 Timer Prescaler Select Header (J20)

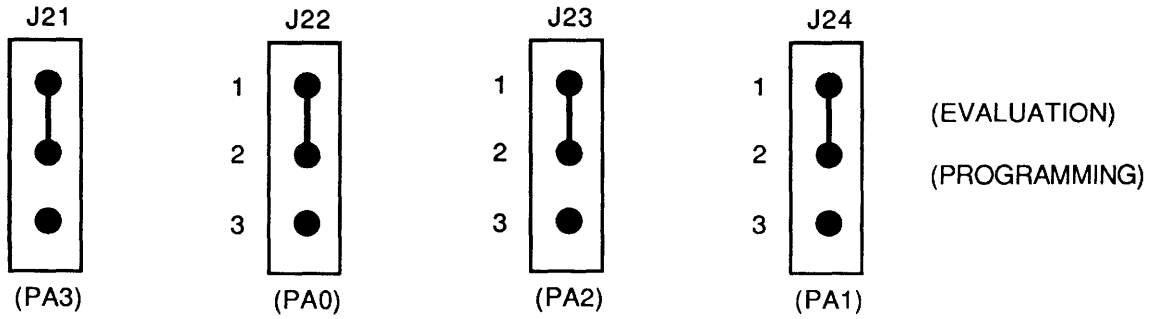
Jumper header J20 is used to select the timer prescaler value for the MC6805R2 MCU, if installed in socket U21. The EVM is factory-configured and shipped with the timer prescaler selected for a divide-by-two operation. This is accomplished by the installation of a fabricated jumper on pins 3 and 4 as shown below.



The MC68705R3/U3 MCU timer is software programmable. Both the timer input and prescaler selection capabilities (via jumper headers J15 and J20, respectively) are bypassed when reconfiguration for either the MC68705R3/U3 MCUs are made via the MCU select header J16. (Refer to paragraph 2.3.1.)

2.3.8 EPROM MCU Programming Select Headers (J21 thru J24)

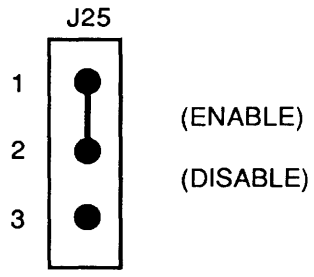
The EVM contains an MC68705P3/R3/U3 EPROM MCU programmer designed to program data into the MCU internal EPROM, and to copy the contents of the programmed EPROM into the EVM pseudo ROM. Jumper headers J21-J24 are used to connect four control lines to the EPROM MCU programming sockets XU57 and XU58 during the programming mode of operation. These four control lines are the MCU I/O port A lines PAO-PA3. The EVM is factory-configured and shipped with fabricated jumpers installed as shown below.



Jumpers positioned between pins 1 and 2 disconnect the loading effect of the programmer from the MCU I/O port PA0-PA3 lines when the EPROM MCU programmer is not in use. When use of the EPROM MCU programmer is desired, fabricated jumpers on headers J21-J24 are to be repositioned between pins 2 and 3.

2.3.9 Write Protect Disable Header (J25)

Jumper header J25 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.



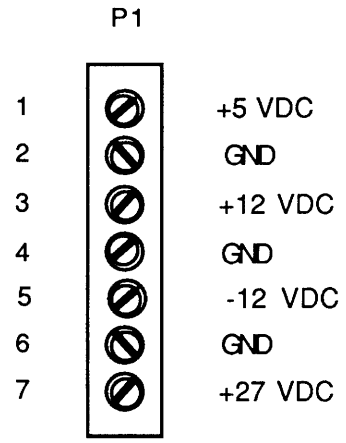
2.4 INSTALLATION INSTRUCTIONS

The EVM is designed for table top operation. A user supplied power supply and RS232C 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.

2.4.1 Power Supply - EVM Interconnection

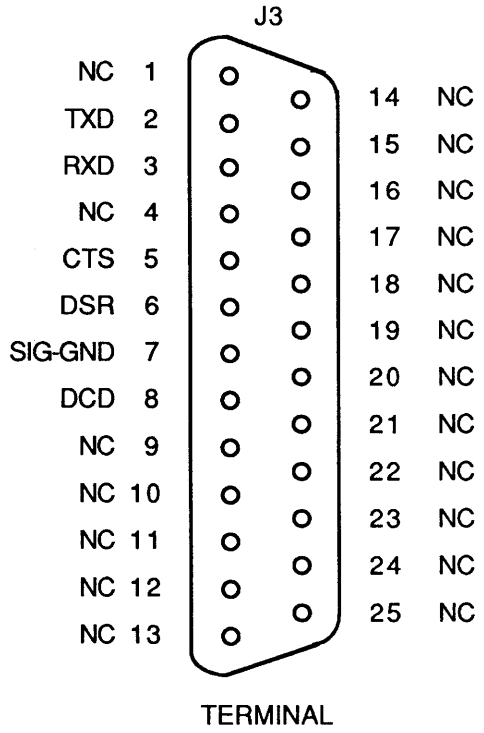
The EVM requires +5 Vdc @ 1.0 A, +12 Vdc @ 0.1 A, and -12 Vdc @ 0.1 A for operation. The +27 Vdc programming voltage (VPP) may be supplied by three 9 volt batteries or +27 Vdc @ 50 mA power supply.

The user supplied power supply is connected to connector P1, which is a terminal block designed to accept 14-22 AWG wire. Interconnection of the power supply wiring to the EVM is shown below.



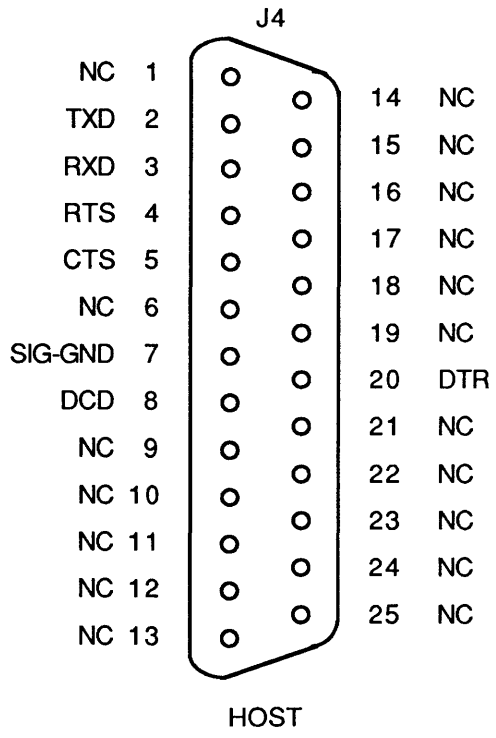
2.4.2 Terminal - EVM Interconnection

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 J3 (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 J3, refer to Chapter 5.



2.4.3 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 J4 (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 J4, refer to Chapter 5.



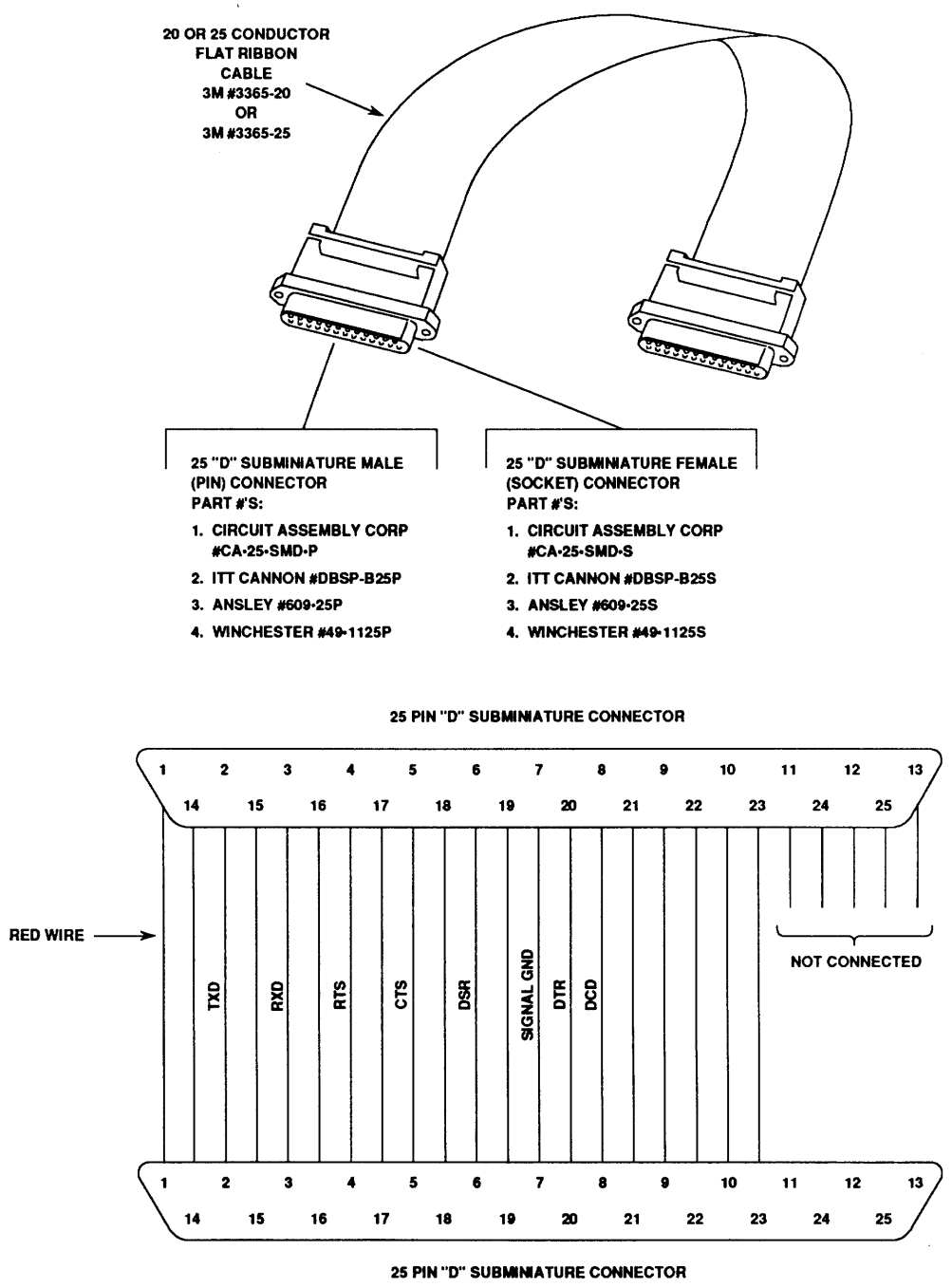
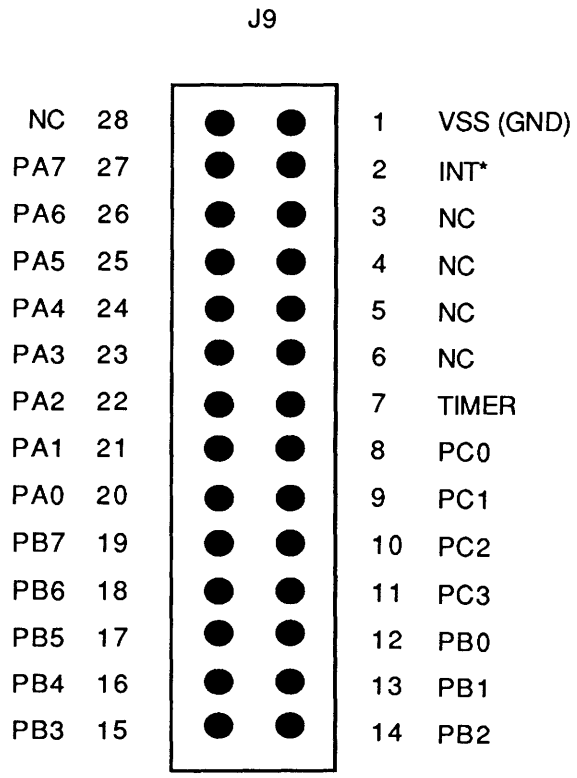


FIGURE 2-2. Terminal or Host Computer Cable Assembly

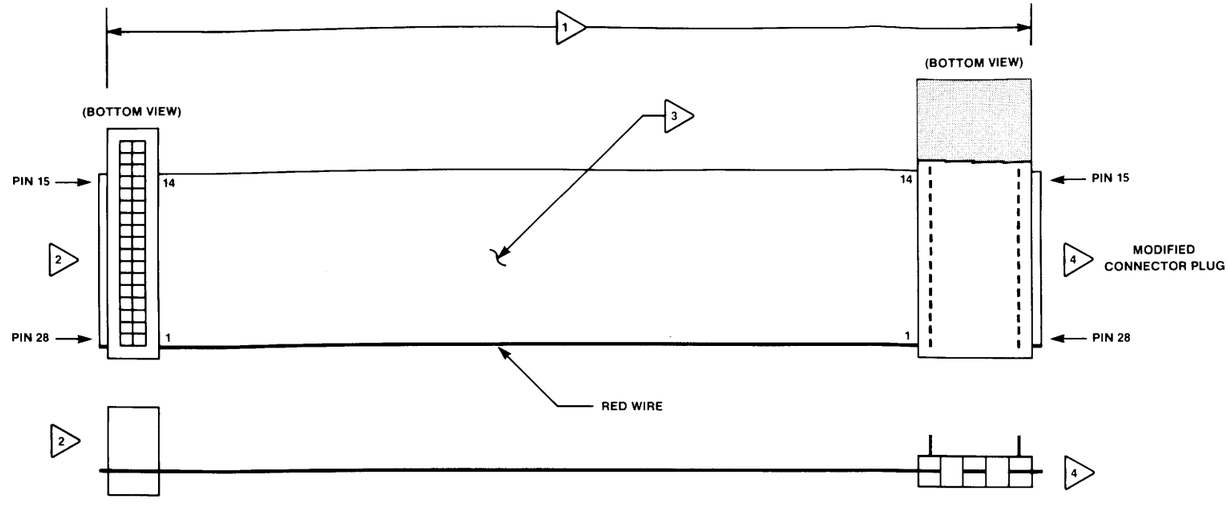
2.4.4 Target System - EVM 28-Pin Interconnection

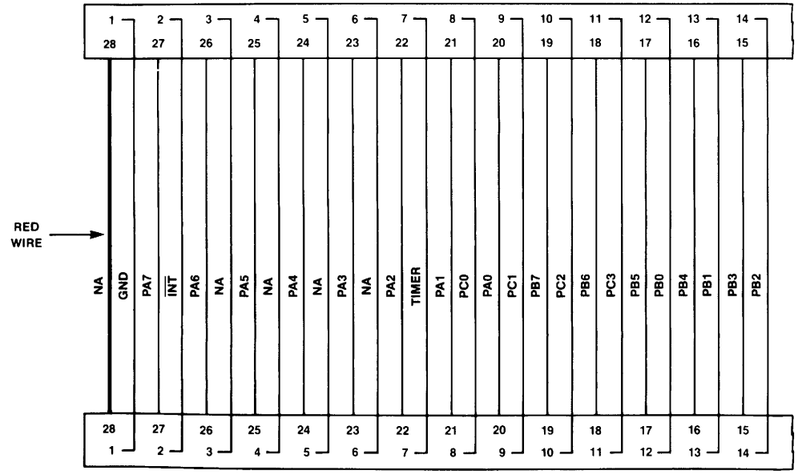
Target system to EVM interconnection for the MC68(7)05PX operation is accomplished via EVM connector J9, and a user supplied cable assembly as shown in Figure 2-3. MCU I/O port connector J9 is a 28-pin header (shown below) that facilitate the interconnection of the cable assembly for evaluation purposes. For connector pin assignments and signal descriptions of the EVM connector J9, refer to Chapter 5.



28-Pin MCU I/O Port Connector

2





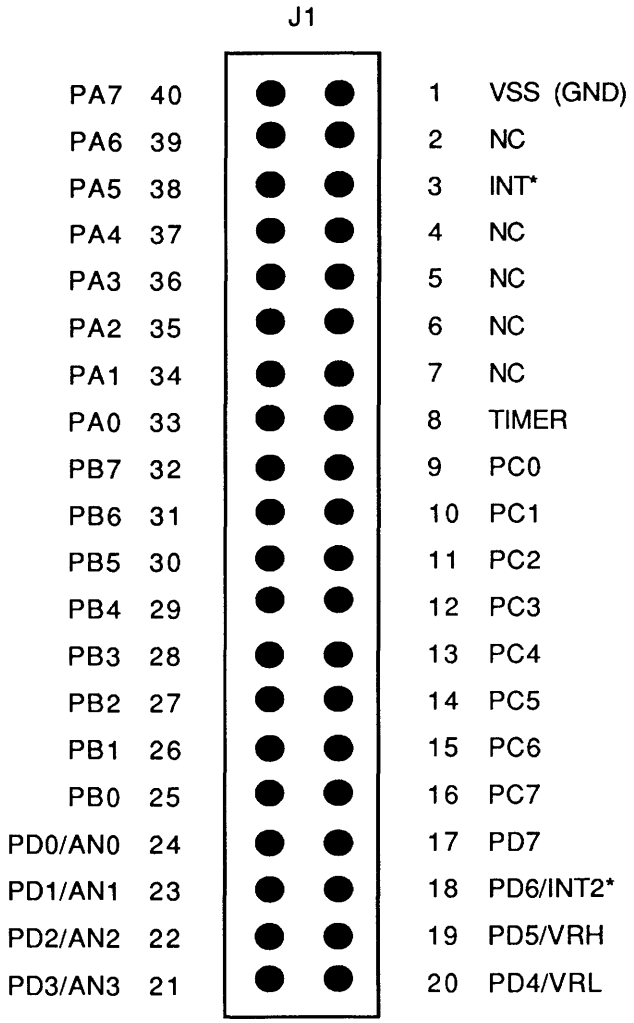
NOTES:

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 # 3508-0000)
5. NA DENOTES NOT APPLICABLE. SIGNAL NOT USED ON EVM.

FIGURE 2-3. 28-Pin MCU I/O Port Extension Cable Assembly Diagram

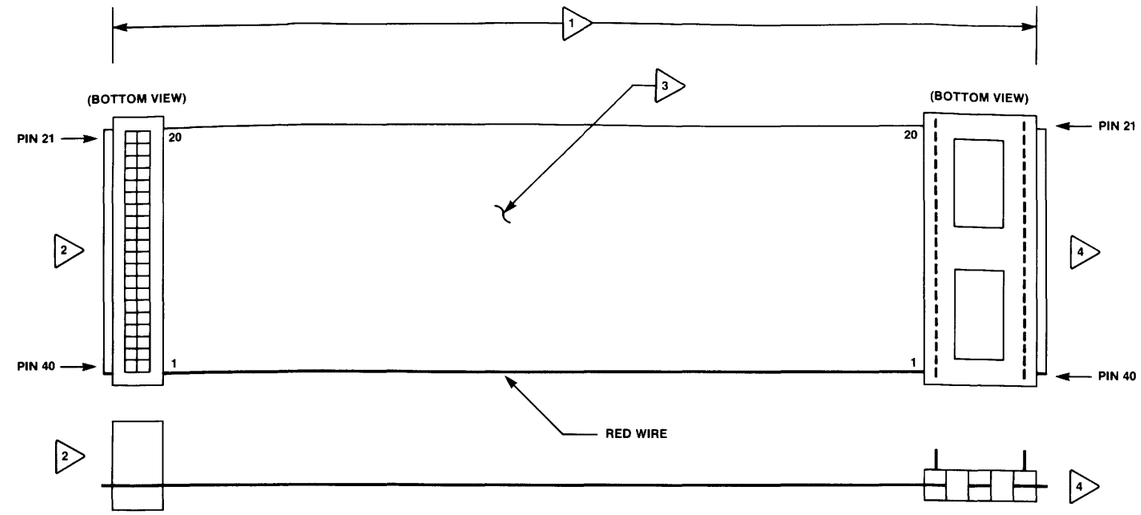
2.4.5 Target System - EVM 40-Pin Interconnection

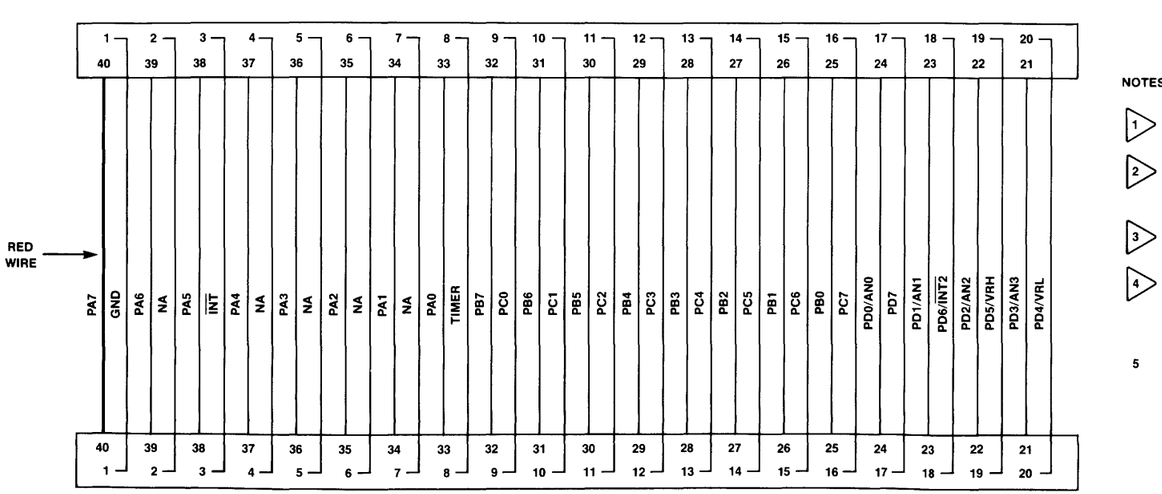
Target system to EVM interconnection for the MC68(7)05RX/UX operation is accomplished via EVM connector J1, and a user supplied cable assembly as shown in Figure 2-4. MCU I/O port connector J1 is a 40-pin header (shown below) that facilitate the interconnection of the cable assembly for evaluation purposes. For connector pin assignments and signal descriptions of the EVM connector J1, refer to Chapter 5.



40-Pin MCU I/O Port Connector

2





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 # 3508-0000)

5 NA DENOTES NOT APPLICABLE. SIGNAL NOT USED ON EVM.

FIGURE 2-4. 40-Pin MCU I/O Port Extension Cable Assembly Diagram

CHAPTER 3

OPERATING INSTRUCTIONS

3.1 INTRODUCTION

This chapter provides the necessary information to initialize and operate the EVM in a target system environment. EPROM MCU erasing and programming, assembling/disassembling, and downloading procedures are also provided. Information contained in this chapter will be presented in the following order:

a. Control switch and indicator descriptions

b. Limitations

c. Operating procedures

d. EPROM MCU erasing procedure

e. EPROM MCU programming procedures

f. Assembly/disassembly procedures

g. Downloading procedures

3.2 SWITCHES AND INDICATORS

The EVM contains five switches and four LED indicators that allow the user to control and monitor specific functions. (Refer to Figure 2-1 for switch and indicator locations.) Switches S1, S2, and S5 control the reset and abort functions. Switches S3 and S4 control the EPROM MCU programming power. Table 3-1 identifies these switches by name, description, and function. Table 3-2 identifies the four indicators by name, description, and function.

TABLE 3-1. EVM Control Switches

NAME	DESCRIPTION AND FUNCTION
USER RESET switch (S1)	Momentary action pushbutton switch - resets EVM MCU and user I/O, and enables map switching to the user map. This switch can be used as a map switch for execution of user code from the user reset vector.
ABORT switch (S2)	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.
PROGRAM switch (S3)	On-off slide switch - applies +12 Vdc and +27 Vdc power to the EPROM MCU programmer circuitry. LED indicator CR7 is illuminated when this switch is placed in the ON position.
+5VOLT switch (S4)	On-off slide switch - applies +5 Vdc power to the EPROM MCU programmer circuitry. LED indicator CR8 is illuminated when this switch is placed in the ON position.
MASTER RESET switch (S5)	Momentary action pushbutton switch - when pressed, places EVM operation in the monitor map and enables the EVM prompt to be displayed on terminal CRT.

TABLE 3-2. EVM Indicators

NAME	DESCRIPTION AND FUNCTION
PROGRAM indicator (CR1)	Red LED indicator - illuminated when EPROM MCU device installed in socket XU57 or XU58 is programmed.
VERIFY indicator (CR2)	Red LED indicator - illuminated when EPROM MCU device installed in socket XU57 or XU58 is programmed and verified.
Program indicator (CR7)	Red LED indicator - illuminated when PROGRAM switch (S3) is placed in the ON position, applying +12 Vdc and +27 Vdc to the programmer circuitry.
+5 Volt indicator (CR8)	Red LED indicator - illuminated when +5VOLT switch (S4) is placed in the ON position, applying +5 Vdc to the EPROM MCU device installed in socket XU57 or XU58.

3.3 LIMITATIONS

The MC6805R2 MCU timer is not software controllable.(see paragraph 2.3.1).

The resident MC6805R2 MCU timer is stopped when the monitor is invoked. When the MC68705R3 is installed as the resident MCU, the timer is software controllable and the timer stop feature is lost.

The user may not trace a return from interrupt (RTI) or a clear interrupt mask bit (CLI) instruction with an interrupt enabled and pending due to MCU interrupt handling. Attempting this will cause multiple interrupts in the user map. 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 occurance of multiple interrupts.

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/EPROM available. The user must be aware of the memory map of the MCU being simulated and ensure that only valid ROM/EPROM 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.

3.4 OPERATING PROCEDURE

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 the following functions:

a. Display or modify memory

b. Display or modify MCU internal registers

c. Execute program under various levels of control

d. Control access to various I/O peripherals connected to the EVM

e. Control programming and reading of EPROM MCU

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

```
EVMbug-HMOS05 X.X
S=7F    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) 111HINZC 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) 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 3-3. 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 3-3. Monitor (EVMbug) Commands

COMMAND	DESCRIPTION
ASM <address>	Assembler/disassembler (interactive)
BF <address1> <address2> <data>	Block fill memory with data
BR [<address>]...	Breakpoint set
COPY	Copy EPROM contents to user pseudo ROM
G [<address>]	Go (execute program)
LOAD <port> [=<text>]	Load (S-records*) from I/O port
MD <address1> [<address2>]	Memory display
MM <address>	Memory modify (interactive)
NOBR [<address>]...	Remove breakpoint
P [<count>]	Proceed (thru breakpoint)
PROG	Program EPROM MCU
RD	Register display
RM	Register modify (interactive)
T [<count>]	Trace
TM [<exit character>]	Transparent mode

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

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
100 A6 55       LDA  #$55

102 C1 00 9D    CMP  $009D   >STA $60
102 B7 60       STA  $60

104 9D          NOP          >LDA 0,X
104 E6 00       LDA  $00,X

106 FB          ADD  ,X      >BRA $100
106 20 F8       BRA  $0100

108 FF          STX  ,X       >.
>
```

Immediate mode addressing, requires # before operand.

Direct mode addressing, may have $ but not necessary.

Index mode, if not offset (,X) will be accepted.

Branch offsets calculated automatically, address required as conditional branch operand.

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 MC6805 MCU device (i.e., port addresses, timer registers, etc.). The monitor examines each modified user memory location to insure that valid memory exists.

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 Copy

COPY

The COPY command allows the user to copy the contents of the programmed MC68705R3/U3/P3 MCU internal EEPROM (installed in programming sockets XU57 or XU58) into the EVM user pseudo ROM. This copy operation can only be performed if the dump program resides in the MC68705R3/U3/P3 MCU.

Refer to paragraph 3.9 for information pertaining to the implementation of the dump program which enables the monitor to copy EPROM contents into user pseudo ROM.

The COPY command is now entered via the terminal keyboard to enable the MCU EPROM contents to be copied into the EVM user map. The EVMbug prompt is displayed on the terminal CRT upon completion of the copying operation.

EXAMPLES	DESCRIPTION
>COPY	Copy MCU EPROM contents.
>	Prompt indicates copy sequence completed.

3.6.5 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 S2 is activated (pressed), or the MASTER RESET switch S5 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 80 Abort S=7F P=0104 A=55 X=FF C=E8 111.I...	Transfer of EVMbug monitor control. ABORT switch S2 is used to restore EVMbug monitor control if no breakpoints were preset prior to the G command entry.

3.6.6 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.

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.

3.6.7 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.

NOTE

The EVM does not support any type of data flow control on the terminal I/O port. If the external terminal connected to the EVM cannot support 9600 baud, the terminal CRT display may become garbled after one full display of data.

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   ................
>
```

3.6.8 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.
>	

3.6.9 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 Brkpts=0321 0080 0420 >	Removes breakpoint located at 200.
>NOBR 321 80 Brkpts=0420 >	Removes breakpoints located at 321 and 80.
>NOBR Brkpts= >	Removes all breakpoints.

3.6.10 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.

3.6.11 Program

PROG

The PROG command is used to program the MC68705 MCU internal EPROM installed in programming socket XU57 or XU58.

CAUTION

Do not attempt to program a mask programmed MCU device (e.g., MC6805Px, MC6805Rx, or MC6805Ux). Damage will occur to the device.

Prior to entering this command, the user must follow the EPROM programming procedure as described in paragraph 3.8. This procedure removes the reset condition applied to the MCU, and enables the MCU EPROM to be programmed.

NOTE

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

EXAMPLES	DESCRIPTION
>PROG >	Program all EPROM MCU locations. Prompt indicates program/verify operation has been completed.

3.6.12 Register Display

RD

The RD command displays the MCU register contents.

COMMAND FORMAT	DESCRIPTION
>RD	Contents of the following registers are displayed: S = Stack pointer P = Program counter A = Accumulator A X = Index register C = Condition codes

3

EXAMPLE

```
>RD
Regs
S=7F    P=0103    A=21    X=00    C=E8        111.I...
>
```

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=E0	111.....
Specific CCR bit set (I)	C=E8	111.I...
Specific CCR bits set (H, I, and Z)	C=F9	111HI..C

3.6.13 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=7F	
P=0103>0080	Change P register and go to A register.
A=FF>AA	Change A register.
X=FF>	Examine X register.
C=E8>.	Examine C register and terminate command.
>	

3.6.14 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=7F    P=0101    A=00    X=FF    C=EA          111.I.Z.
>
```

MULTIPLE TRACE EXAMPLE

```
>T 2
0102  9D          NOP
S=7F    P=0102    A=01    X=FF    C=E8          111.I...
0103  20 FC       BRA   $0101
S=7F    P=0103    A=01    X=FF    C=E8          111.I...
>
```

3.6.15 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.

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

3.7 EPROM MCU ERASING PROCEDURE

MC68705Px/Rx/Ux 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.

3.8 EPROM MCU PROGRAMMING PROCEDURE

The EVM contains programming circuitry that enables the user to program MC68705Px/Rx/Ux MCU internal EPROM. When performing the programming procedure the user must be aware of the following cautions and general warnings/limitations (notes) to prevent damage to the EPROM MCU device being programmed:

CAUTIONS

Do not program non-EPROM MCU device locations. Damage will occur to the non-EPROM MCU device.

NOTES

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

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

The monitor firmware allows the user to program MC68705R3, MC68705U3, and MC68705P3 internal EPROM. To program the specific EPROM MCU, jumpers must be repositioned, switches set, and indicators observed as described in the following procedure. It is assumed that the user program has been debugged and resides in EVM memory.

The following steps are used to program the EPROM MCU:

a. Place programmer switches S3 and S4 in the OFF position.

b. Connect +27 Vdc source to P1 connector. (Refer to paragraph 2.4.1.)

c. Reposition fabricated jumpers on headers J21-J24 between pins 2 and 3. Jumper repositioning will reconfigure the EVM for the programming mode of operation. (Refer to paragraph 2.3.8.)

d. Press MASTER RESET switch S5.

e. Insert MC68705 EPROM MCU into programming socket XU57 or XU58.

   XU57 for 28-pin device (MC68705P3/P5)

   XU58 for 40-pin device (MC68705R3, MC68705U3/U5)

f. Place switch S4 (+5 Vdc programming power) to the ON position and observe that LED indicator CR8 is illuminated.

g. Place switch S3 (Programming VPP) to the ON position and observe that LED indicator CR7 is illuminated.

h. Enter PROG command. (Refer to paragraph 3.6.11.)

i. PROGRAM LED indicator (CR1) will illuminate upon completion of the programming sequence.

   Depending upon the length of the user program, CR1 will illuminate from a number of seconds to 3 minutes depending upon the length of the user program. Approximately 2 seconds later, the VERIFY LED indicator (CR2) will illuminate upon completion of the verify sequence. The terminal CRT will display the monitor prompt ">" upon completion of the program/verify sequence. If the programming sequence cannot be verified, the VERFIFY indicator will not illuminate.

j. Press MASTER RESET switch S5.

k. Place switch S3 to the OFF position.

l. Place switch S4 to the OFF position.

m. Remove programmed EPROM MCU device from programming socket. Repeat the procedure from step (d.) to program additional devices.

n. When EPROM MCU programming is complete, fabricated jumpers on headers J21-J24 are reposition between pins 1 and 2. Jumper repositioning will reconfigure the EVM for the evaluation mode of operation. (Refer to paragraph 2.3.8)

3.9 EPROM MCU COPYING PROCEDURE

The monitor firmware in conjunction with a special dump program allows the user to copy the contents of the programmed MC68705R3/U3/P3 MCU internal EPROM into user pseudo ROM.

This special dump program must be included in the user program (stored in memory) to be transferred to the EPROM. Approximately 30 bytes of memory are required for the dump program. This dump program may be relocated anywhere in the valid EPROM space by changing the reset vector from $C0 to the new start location.

Caution must be used in calculating branch offsets since this dump program executes from EPROM and cannot be debugged by the monitor. The dump program (shown on the following page) is written for an MC68705R3/U3 (reset vectors located at $FFE-$FFF). If an MC68705P3 is to be copied, reset vectors are moved to $7FE-7FF.

User I/O bit PB3 must not be held low by the user application circuitry during reset. After reset, all user I/O is available for applications (see Figure 3-1).

The mask option register (MOR) must be programmed for the crystal oscillator option (b7=0). It is not possible to read an MCU programmed for the RC oscillator option using the EVM.

```
0001
0002 0000              PORTA    EQU    $00             DATA PORT
0003 0004              PADR     EQU    $04             PORT A DIRECTION
0004 0001              PORTB    EQU    $01             CONTROL PORT
0005 0005              PBDR     EQU    $05             PORT B DIRECTION
0006 0060              FETCH    EQU    $60             SELF-MODIFYING CODE
0007 0061              PAGE     EQU    FETCH+1         ADDRESS POINTER
0008 0062              BTYE     EQU    PAGE+1
0009 00C0              RESET    EQU    $C0             RESET VECTOR
0010                   *
0011                   ****************************************
0012                   *
0013                   *  SELF-MODIFYING CODE AT RAM $60
0014                   *
0015                   *  FETCH   LDA   $0080
0016                   *          RTS
0017                   *
0018                   ****************************************
0019                   *
0020 00C0                       ORG    RESET
0021 00C0 06 01 2B     START    BRSET  3,PORTB,USER    IF PB3=1,GO TO USER CODE
0022 00C3 A6 C6        READ     LDA    #$C6            TRANSFER SELF-MODIFYING CODE
0023 00C5 B7 60                 STA    FETCH           TO RAM
0024 00C7 3F 61                 CLR    PAGE
0025 00C9 A6 80                 LDA    #$80
0026 00CB B7 62                 STA    BYTE
0027 00CD 4C                    INCA
0028 00CE B7 63                 STA    BYTE+1
0029                   *
0030 00D0 A6 FF                 LDA    #$FF            PORT A IS OUTPUT
0031 00D2 B7 04                 STA    PADR
0032 00D4 A6 10                 LDA    #$10            PORT B BIT 4 IS OUTPUT
0033 00D6 B7 05                 STA    PBDR
0034 00D8 19 01        LOOP     BCLR   4,PORTB         SEND: DATA INVALID
0035 00DA 07 01 FD              BRCLR  3,PORTB,*       WAIT FOR DATA REQUEST
0036 00DD AD 81                 BSR    FETCH           GET NEXT BYTE
0037 00DF B7 00                 STA    PORTA           AND SEND IT
0038 00E1 18 01                 BSET   4,PORTB         SEND: DATA VALID
0039 00E3 3C 62                 INC    BYTE            INCREMENT POINTER
0040 00E5 26 02                 BNE    L1
0041 00E7 3C 61                 INC    PAGE
0042 00E9 06 01 FD     L1       BRSET  3,PORTB,*       WAIT FOR : GOT DATA
0043 00EC 20 EA                 BRA    LOOP
0044                   *
0045                   * USER CODE STARTS HERE
0046                   *
0047 00EE 20 FE        USER     BRA    *
0048                   *
0049                   * THE RESTART VECTOR FOR THE MC68705R3/U3 IS $FFE AND FOR THE
0050                   * MC68705P3 IS $7FE.
0051                   *
0052 0FFE                       ORG    $FFE            VECTOR FOR 68705R3/U3
0053 0FFE 00 C0        VECTOR   FDB    RESET
0054
0055
```

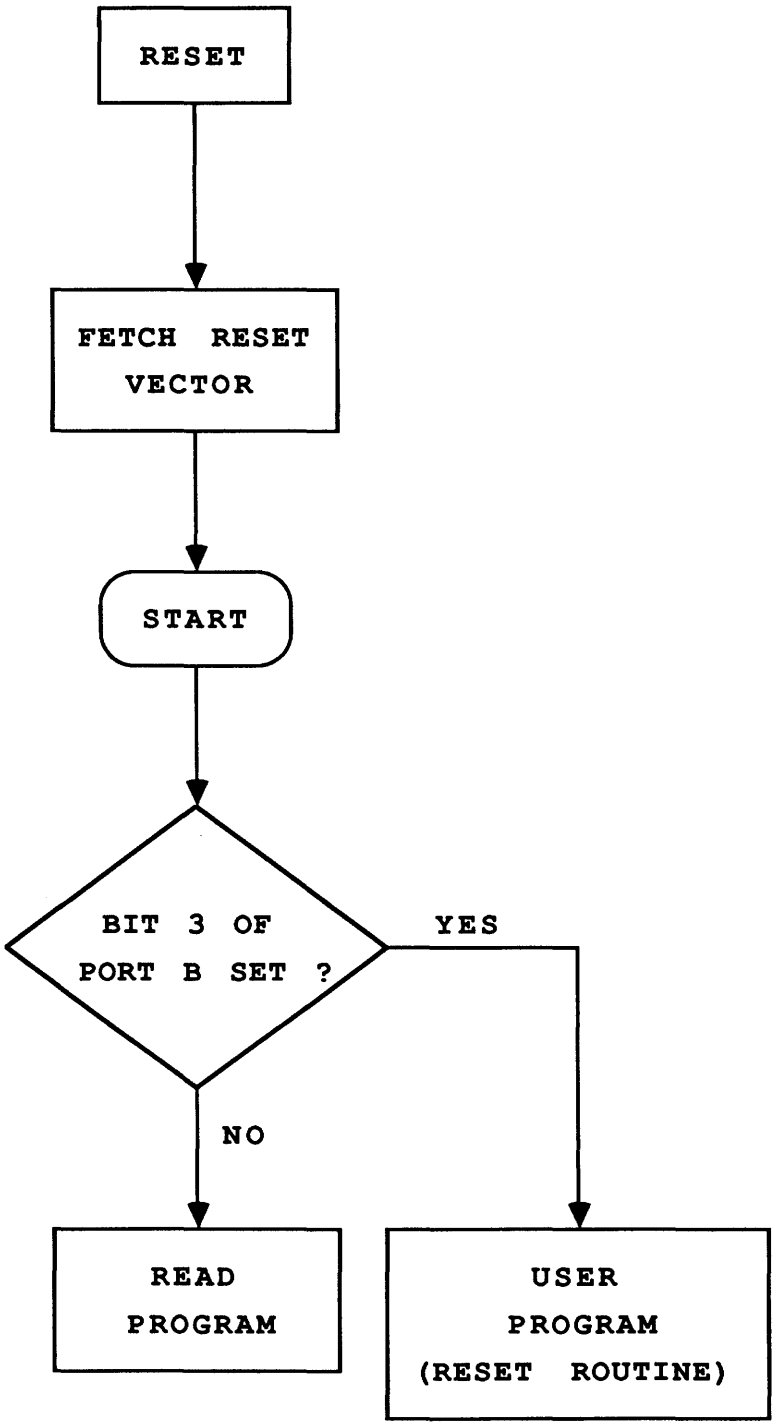


FIGURE 3-1. Dump Program Flow Chart

The following steps are used to copy the contents of the programmed MC68705R3/U3/P3 internal EPROM into user pseudo ROM:

a. Place programmer switches S3 and S4 in the OFF position.

b. Reposition fabricated jumpers on headers J21-J24 between pins 2 and 3. Jumper repositioning will reconfigure the EVM for the programming mode of operation. (Refer to paragraph 2.3.8.)

c. Press MASTER RESET switch S5.

d. Insert MC68705 EPROM MCU into programming socket XU57 or XU58.

   XU57 for 28-pin device (MC68705P3/P5)

   XU58 for 40-pin device (MC68705R3, MC68705U3/U5)

e. Place switch S4 (+5 Vdc programming power) to the ON position and observe that LED indicator CR8 is illuminated.

f. Enter COPY command. (Refer to paragraph 3.6.4.)

   The terminal CRT will display the monitor prompt ">" upon completion of the copying sequence. The EPROM contents now reside in user pseudo ROM and can be observed by displaying memory.

g. Press MASTER RESET switch S5.

h. Place switch S4 to the OFF position.

i. Remove programmed EPROM MCU device from programming socket. Repeat the procedure from step (d.) to copy additional devices.

j. When EPROM MCU copying is complete, fabricated jumpers on headers J21-J24 are reposition between pins 1 and 2. Jumper repositioning will reconfigure the EVM for the evaluation mode of operation. (Refer to paragraph 2.3.8.)

3.10 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. 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.

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=7F P=0FF8 A=60 X=FF C=E8 111.I...	
>	
>RM	*Modify program counter register.*
S=7F	
P=0FF8>100=	
P=0100>(RETURN)	
A=60>.	
>BR 103 106	*Set breakpoints.*
Brkpts=0103 0106	
>G	*Begin execution of program.*
Brkpt	
S=7F P=0103 A=01 X=00 C=E8 111.I...	
>	
>P 45	*Proceed 45 times within loop.*
Brkpt	
S=7F P=0103 A=46 X=00 C=E8 111.I...	
>	
>NOBR 103	*Remove breakpoint.*
Brkpts=0106	
>G	*Continue program execution.*
Brkpts	
S=7F P=0106 A=00 X=01 C=E8 111.I...	
>	
>T 2	*Monitor program execution.*
0102 4C INCA	
S=7F P=0102 A=00 X=01 C=E8 111.I...	
0103 26 FD BNE $0102	
S=7F P=0103 A=01 X=01 C=E8 111.I...	
>	

3.11 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 (or target system memory) 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.

3.11.1 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.*

3

```
>LOAD H=COPY EXORT.LX,#CN
COPY EXORT.LX,#CN
S0030000FC
S110001F424547204C4F414420484552457E
S1110100243130302057494C4C204C4F4144A0
S1110200243230302057494C4C204C4F41449E
S1110300243330302057494C4C204C4F41449C
S1110400243430302057494C4C204C4F41449A
S1110500243530302057494C4C204C4F414498
S1110600243530302057494C4C204C4F414496
S1110700243530302057494C4C204C4F414494
S9030000FC
>
```

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

3.11.2 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 J3 for downloading operations.

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

EXAMPLE	DESCRIPTION
C>KERMIT IBM-PC Kermit-MS VX.XX Type ? for help	*IBM-PC prompt. Enter Kermit program.*
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 (CTRL)]C Kermit-MS>PUSH	*EVM download command (via terminal port) entered.*
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.*

3

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. The EVM memory map description and switching concept are supported by the memory map diagram (Figure 4-2).

4.2 EVM DESCRIPTION

The EVM is designed to evaluate an MC68705 MCU based target system via the resident MC6805R3 MCU. The EVM contains two memory maps (monitor or user map) that are switchable 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. EPROM MCU programmer

4.2.1 MCU and Control Circuits

The EVM contains an MC6805R3 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 S1. 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 S1 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 S2 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 until the next 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).

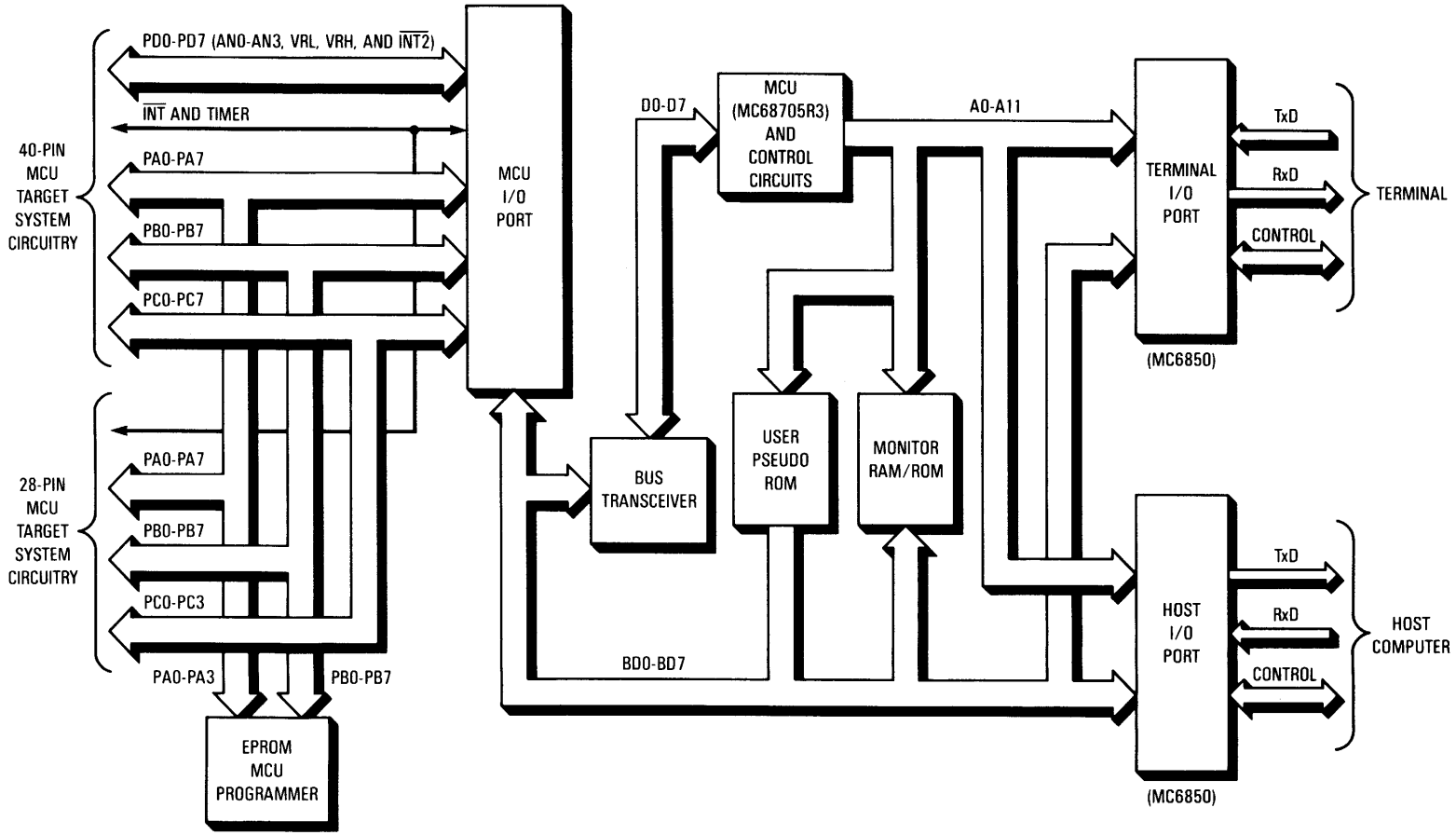


FIGURE 4-1. EVM Block Diagram

4.2.1.2 Abort. The abort circuitry generates internal abort signals upon the activation of the ABORT switch S2. 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. Both memory maps are 4K bytes, and are decoded via the FPLA.

NOTE

The entire 4K-byte memory map is available to the user, although all M68705 MCU family devices do not have EPROM throughout the entire 4K-byte memory map. Refer to specific device data sheet for valid program space locations.

MONITOR MAP

$000	TERMINAL CONTROL
001	TERMINAL DATA
002	HOST CONTROL
003	HOST DATA
004	RESERVED
00B	
00C	USER STACK POINTER
00D	MONITOR CONTROL
00E	RESERVED
00F	RESERVED
010	MONITOR RAM/STACK
07F	
080	2K MONITOR EPROM
7FF	
800	2K MONITOR EPROM
$FFF	

USER MAP

$000	PORT A DATA REGISTER
001	PORT B DATA REGISTER
002	PORT C DATA REGISTER
003	PORT D DATA REGISTER
004	PORT A DDR
005	PORT B DDR
006	PORT C DDR
007	RESERVED
008	TIMER DATA REGISTER
009	TIMER CTRL REGISTER
00A	MISCELLANEOUS REGISTER
00B	PROGRAM CTRL REGISTER
00C	RESERVED
00D	RESERVED
00E	A/D CONTROL REGISTER
00F	A/D RESULT REGISTER
010	USER RAM/STACK
07F	
080	2K USER PSEUDO ROM
7F6	
7F7	(MC68705P3 VECTORS)
7FF	
800	2K USER PSEUDO ROM
FF7	
FF8	(MC68705R3/U3 VECTORS)
$FFF	

Notes:

1. User map locations $080-$FFF are write protected during user program execution.

2. User map locations $003, and $007-$00F are MCU reserved.

FIGURE 4-2. EVM Memory Map

4.2.2.1 Monitor Map Area. As shown in Figures 4-2, the monitor map area contains the monitor I/O (terminal/host) data and control registers, user stack pointer, monitor control register, monitor RAM/stack, and 4K bytes monitor EPROM. The EVMbug monitor EPROM contents are not available to the user.

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

The monitor RAM/stack is used by the monitor for general monitor operations such as temporary data storage, command entries, S-record downloading, etc..

The monitor control register is located at $00D, 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 control register is only controlled by the monitor.

4.2.2.2 User Map Area. As shown in Figure 4-2, the user map area consists of the user RAM/stack and user pseudo ROM. All user I/O ports A, B, C, and D ($000-$003) are available in the user map for evaluation purposes. Port D (analog/digital port) and locations $007-$00F are MCU reserved.

The user RAM/stack is resident in memory locations $010 through $07F. The stack pointer value is displayed on the terminal CRT when the trace and breakpoint functions are executed.

User program space (user pseudo ROM) $080-$FFF is RAM. This RAM is write protected during user program execution. This feature requires all programs to be ROMable and protects against program errors which would otherwise overwrite the program space.

Jumper header J10 provides selection of a 4K-byte user map for evaluating the MC68705R3/U3 EPROM MCU. Header J10 also provides selection of a 2K-byte user map for evaluating the MC68705P3 EPROM MCU. Vectors fetched from 7F8-7FF in the user program appear to the resident MCU at FF8-FFF. This is accomplished by the hardware reconfiguration of the lower 2K map into the upper 2K map.

4.2.3 Terminal and Host Computer I/O Ports

The EVM terminal I/O port communicates with an RS-232C compatible terminal via the TERMINAL connector J3 and a user supplied cable assembly. The MC6850 Asynchronous Communications Interface Adapter (ACIA) - based terminal interface circuitry provides communication and data transfer operations for the EVM and user terminal. Hardware selectable 300-19.2k baud rate generation capabilities, and RS-232C drivers/receivers are also implemented for this port. Refer to Chapter 5 for additional port information.

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.

The EVM host computer I/O port communicates with an RS-232C compatible host computer directly or by modem via the HOST connector J4 and a user supplied cable assembly. The MC6850 ACIA - based host interface circuitry provides communications and data transfer operations for the EVM and user host computer. Hardware selectable 300-19.2k baud rate generation capabilities, and RS-232C drivers/receivers are also implemented for this port. Refer to Chapter 5 for additional port information.

4.2.4 MCU Extension I/O Ports

The EVM provides two user HMOS compatible MCU extension I/O ports for target system evaluation of 28-pin and 40-pin M6805 HMOS family of MCU devices. These device types include MC6805P2/P4/P6, MC6805R2/R3, MC6805U2/U3, MC68705P3/P5, MC68705R3, and MC68705U3/U5.

Target system to EVM interconnection for the MC68(7)05PX operation is accomplished via EVM connector J9, and a user supplied cable assembly. MCU I/O port connector J9 is a 28-pin header that facilitate the interconnection of the cable assembly for evaluation purposes. For connector pin assignments and signal descriptions of the EVM connector J9, refer to Chapter 5.

Target system to EVM interconnection for the MC68(7)05RX/UX operation is accomplished via EVM connector J1, and a user supplied cable assembly. MCU I/O port connector J1 is a 40-pin header that facilitate the interconnection of the cable assembly for evaluation purposes. For connector pin assignments and signal descriptions of the EVM connector J1, refer to Chapter 5.

The M68705EVM contains two 6522 Versatile Interface Adapters (VIAs) located at IC locations U37 and U38. These VIAs replace the resident MCU I/O port pins.

Normally, MCU ports are high impedance when configured as an input port. Since the ports have been re-constructed with VIAs, the input ports are not high impedance because of the VIAs input circuitry containing internal pullups.

4.2.5 EPROM MCU Programmer

The EPROM MCU programmer accommodates and programs two types of MC68705 EPROM MCU device packages. The device packages are 28-pin and 40-pin Dual-In-line Packages (DIPs). Programming socket located at XU57 is used for the 28-pin DIP package, and the socket located at XU58 is used for the 40-pin DIP package.

MC68705 EPROM MCU devices have an internal boot programmer program. When the appropriate voltages are applied externally and reset is lifted, the boot program is entered by a vector. The program receives data from the EVM MCU I/O port and transfers the data into the EPROM MCU device.

4

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 J3 and J4 which are used to interconnect the EVM to an RS-232C compatible terminal and host computer, respectively.

Two MCU I/O port connectors J1 and J9 are used to interconnect the EVM to the target system equipment. Connector J9 is used for 28-pin MCU device evaluation (e.g., MC68(7)05PX). Connector J1 is used for 40-pin MCU device evaluation (e.g., MC68(7)05RX/UX).

Connector P1 interconnects an external power supply to the EVM.

Pin assignments for the above connectors (J3, J4, J9, J1, and P1) are identified in Tables 5-1 through 5-5, respectively. Connector signals are identified by pin number, signal mnemonic, and signal name and description.

TABLE 5-1. RS-232C Terminal Port Connector J3 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	------	Not connected.
5	CTS	CLEAR TO SEND - An output signal (held high) 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.

TABLE 5-2. RS-232C Host Port Connector J4 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 2.3.2.)
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	DCD	DATA CARRIER DETECT - An input signal used to indicate an acceptable carrier signal has been detected. (Refer to paragraph 2.3.2.)
9-19	------	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 J9 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	VSS	GROUND
2	INT*	INTERRUPT REQUEST - An input signal that asynchronously applies an MCU interrupt.
3-6	NC	Not connected.
7	TIMER	TIMER - Input mode, serves as timer clock. Output mode, reflects contents of DOUT bit of timer status and control register each time TMZ bit has a low to high transition.
8	PC0	PORT C (bits 0-3) - General purpose I/O lines controlled by software via data direction registers.
9	PC1	
10	PC2	
11	PC3	
12	PB0	PORT B (bits 0-7) - General purpose I/O lines controlled by software via data direction registers.
13	PB1	
14	PB2	
15	PB3	
16	PB4	
17	PB5	
18	PB6	
19	PB7	
20	PA0	PORT A (bits 0-7) - General purpose I/O lines controlled by software via data direction registers.
21	PA1	
22	PA2	
23	PA3	
24	PA4	
25	PA5	
26	PA6	
27	PA7	
28	NC	Not connected.

5

TABLE 5-4. 40-Pin MCU I/O Port Connector J1 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	VSS	GROUND
2	NC	Not connected.
3	INT*	INTERRUPT REQUEST - An input signal that asynchronously applies an MCU interrupt.
4-7	NC	Not connected.
8	TIMER	TIMER - Input mode, serves as timer clock. Output mode, reflects contents of DOUT bit of timer status and control register each time TMZ bit has a low to high transition.
9	PC0	PORT C (bits 0-3) - General purpose I/O lines controlled by software via data direction registers.
10	PC1	
11	PC2	
12	PC3	
13	PC4	
14	PC5	
15	PC6	
16	PC7	
17	PD7	PORT D (bits 0-7) - General purpose input lines; or four analog input lines, two voltage reference input lines, and a second interrupt request line.
18	PD6/INT2*	
19	PD5/VRH	
20	PD4/VRL	
21	PD3/AN3	
22	PD2/AN2	
23	PD1/AN1	
24	PD0/AN0	
25	PB0	PORT B (bits 0-7) - General purpose I/O lines controlled by software via data direction registers.
26	PB1	
27	PB2	
28	PB3	
29	PB4	
30	PB5	
31	PB6	
32	PB7	

5

TABLE 5-4. 40-Pin MCU I/O Port Connector J1 Pin Assignments (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
33	PA0	PORT A (bits 0-7) - General purpose I/O lines controlled by software via data direction registers.
34	PA1	
35	PA2	
36	PA3	
37	PA4	
38	PA5	
39	PA6	
40	PA7	

TABLE 5-5. Input Power Connector P1 Pin Assignments

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	+5 V	+5 Vdc Power - Input voltage (+5 Vdc @ 1.0 A) used by the EVM logic circuits.
2	GND	GROUND
3	+12 V	+12 Vdc Power - Input voltage (+12 Vdc @ 0.1 A) used by the EVM logic circuits.
4	GND	GROUND
5	-12 V	-12 Vdc Power - Input voltage (-12 Vdc @ 0.1 A) used by the EVM logic circuits.
6	GND	GROUND
7	+27 V	+27 Vdc Power - Input voltage (+27 Vdc @ 50 mA) used by the EVM programming circuitry.

5.3 PARTS LIST

Table 5-6 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 5-6. EVM Parts List

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
	Printed Wiring Board (PWB), M68705EVM
C1, C3, C10	Capacitor, electrolytic, 22 uF @ 25 Vdc
C2, C4, C5, C8, C9, C11-C13, C15-C33, C35-C53	Capacitor, fixed, ceramic, 0.1 uF @ 50 Vdc
C7	NA
C34	Capacitor, electrolytic, 50 uF @ 16 Vdc
C54	Capacitor, 24 pF @ 50 Vdc
CR1, CR2, CR7, CR8	Diode, light emitting, red
CR3-CR5	Rectifier, 1N4001
J1	Header, double row post, 40 pin, Aptronics # 929715-01-20
J2, J8, J10, J11, J14-J17, J19, J21-J25	Header single row post, 3 pin, Aptronics # 929705-01-03
J3, J4	Connector, AMP #206584-2 DB25S
J5-J7	NA
J9	Header, double row post, 28 pin, Aptronics # 929715-01-14

TABLE 5-6. EVM Parts List (cont'd)

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
J12, J13	Header, double row post, 14 pin, Aptronics # 929715-01-07
J18	Header, single row post, 2 pin, Aptronics # 929705-01-02
J20	Header, double row post, 16 pin, Aptronics # 929715-01-08
P1	Terminal block 7 position, screw contact Electrovert # 25.104.0753
Q1, Q2	Transistor, 2N4401
R1, R16	Resistor, fixed, film, 330 ohm, 5%, 1/4 W
R2	Resistor network, five 39k ohm, Allen Bradley # 706A393
R3-R6, R8, R9	NA
R7, R15	Resistor network, five 4.7k ohm, Allen Bradley # 706A472
R10	Resistor network, nine 10k ohm, Allen Bradley # 710A103
R11	Resistor network, seven 470 ohm, Allen Bradley # 708A471
R12, R13, R23	Resistor, fixed, film, 1.5k ohm, 5%, 1/4 W
R27, R28	Resistor, fixed, film, 10k ohm, 5%, 1/4 W
R14	Resistor, fixed, film, 1.0k ohm, 5%, 1/4 W
R17	Resistor, fixed, film, 12k ohm, 5%, 1/4 W
R18-R20	Resistor, fixed, film, 470 ohm, 5%, 1/4 W
R21, R22, R24-R26	Resistor, fixed, film, 4.7k ohm, 5%, 1/4 W

TABLE 5-6. EVM Parts List (cont'd)

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
S1, S2, S5	Switch, pushbutton, SPDT, C&K # 8125-SD9R2BE
S3, S4	Switch, slide, DPDT, Switchcraft # C56206L2
U1, U13, U17, U18, U45	I.C. 74LS08N
U2, U4	I.C. MC6850P, ACIA
U3	I.C. MC1489AL
U5, U51	I.C. 74LS10N
U6, U43, U53	I.C. 74LS00N
U7, U8, U9	NA
U10, U32	I.C. 74LS244N
U11, U50	I.C. 74LS260N
U12, U31, U47	I.C. 74LS11N
U14	I.C. MC1488L
U15, U35	I.C. 74LS02N
U16, U39, U42, U54	I.C. 74LS04N
U19	I.C. 74LS374N
U20	I.C. 74LS245N
U21	I.C. MC68705R3S
U22	I.C. 74LS373N
U23	I.C. 82S103, FPGA, address map decoder, programmed (SEE NOTE)
U24	I.C. MC68B10P, monitor stack/RAM

TABLE 5-6. EVM Parts List (cont'd)

REFERENCE DESIGNATION	COMPONENT DESCRIPTION
U25, U26	I.C. 2716, EVMbug monitor, 2Kx8 EPROM, programmed (SEE NOTE)
U27, U28	I.C. 2016P, user pseudo ROM, 2Kx8 RAM
U29	I.C. 74LS27N
U30, U48, U55	I.C. 74LS393N
U33, U34, U40	I.C. 74LS32N
U36, U41, U44, U46, U52	I.C. 74LS74N
U37, U38	I.C. R6522AP
U49	I.C. MC14024BCP
U56	I.C. 74LS14N
VR1	Diode, zener 22V, 1N969B
XU57	Socket, programming, low insertion force, 28-pin DIP, Robinson Nugent # TSN-286-HT
XU58	Socket, programming, low insertion force, 40-pin DIP, Robinson Nugent # TSN-406-HT
Y1	Crystal, 4.0 MHz, Murata/Erie # E400A
	Fabricated jumper, Aptronics # 929955-00 (use with jumper headers J2, J8, J10-J17, J19-J25)

5

5.4 DIAGRAMS

Figure 5-2 is the EVM schematic diagram.