

M1468705EVM/D1

**M1468705EVM
Evaluation Module
User's Manual**

MICROSYSTEMS

QUALITY • PEOPLE • PERFORMANCE

M1468705EVM
EVALUATION MODULE
USER'S MANUAL

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, Motorola reserves the right to make changes to any products herein to improve reliability, function, or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights or the rights of others.

The computer program stored in the Read Only Memory of this device contains material copyrighted by Motorola Inc., first published 1983, and may be used only under a license such as the License For Computer Programs (Article 14) contained in Motorola's Terms and Conditions of Sale, Rev. 1/79.

EXORbus and EXORciser are trademarks of Motorola Inc.

First Edition

Copyright 1983 by Motorola Inc.

PREFACE

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

An asterisk (*) following the signal name for signals which are level significant denotes that the signal is true or valid when the signal is low.

An asterisk (*) following the signal name for signals which are edge significant denotes that the actions initiated by that signal occur on a high to low transition.

TABLE OF CONTENTS

	<u>Page</u>
CHAPTER 1	GENERAL INFORMATION
1.1	INTRODUCTION 1-1
1.2	FEATURES 1-1
1.3	SPECIFICATIONS 1-1
1.4	GENERAL DESCRIPTION 1-4
CHAPTER 2	HARDWARE PREPARATION AND INSTALLATION INSTRUCTIONS
2.1	INTRODUCTION 2-1
2.2	UNPACKING INSTRUCTIONS 2-1
2.3	HARDWARE PREPARATION 2-1
2.3.1	I/O Connector (J1) 2-3
2.3.2	Audio Cassette Playback Phase Select Header (J2) 2-4
2.3.3	HOST Port (J4) CTS and DCD Signal Enable Headers (J5, J18) 2-4
2.3.4	User Map 2K or 8K Select Header (J8) 2-5
2.3.5	Write Protect Disable Header (J10) 2-5
2.3.6	TERMINAL Port and HOST Port Baud Rate Select Headers (J11, J12) 2-6
2.3.7	On-Board or External Clock Select (J13) 2-6
2.3.8	External Clock Connection (J14) 2-7
2.3.9	EPROM Programming Select Headers (J15, J16, J17) 2-7
2.4	INSTALLATION INSTRUCTIONS 2-7
2.4.1	Power Supply Installation 2-9
2.4.2	Terminal Installation 2-9
2.4.3	Host Computer (Modem) Installation 2-11
2.4.4	Printer Installation 2-12
2.4.5	Audio Cassette Player Installation 2-12
CHAPTER 3	OPERATING INSTRUCTIONS
3.1	INTRODUCTION 3-1
3.2	SWITCHES AND INDICATORS 3-1
3.2.1	Switches 3-1
3.2.2	Indicators 3-2
3.3	MEMORY MAP CONCEPT 3-2
3.3.1	Monitor Map 3-2
3.3.2	User Map 3-2
3.4	MAP SWITCHES 3-4
3.4.1	Program Execution 3-4
3.5	LIMITATIONS 3-4
3.6	OPERATING PROCEDURE 3-5
3.6.1	System Initialization 3-5
3.6.2	System Operation 3-6
3.6.3	Command Line Format 3-6
3.6.4	Primitive Commands 3-7
3.6.5	Assembler/Disassembler (Interactive) 3-8

TABLE OF CONTENTS (cont'd)

		<u>Page</u>
CHAPTER 3	OPERATING INSTRUCTIONS (cont'd)	
3.6.6	Block Fill Memory with Data	3-9
3.6.7	Breakpoint Set/Breakpoint Remove	3-10
3.6.8	Go (Execute Program)	3-11
3.6.9	Download Memory (S-Records) from Host File	3-12
3.6.10	Memory Display	3-13
3.6.11	Memory Modify (Interactive)	3-14
3.6.12	Proceed (from Breakpoint)	3-15
3.6.13	Printer Attach/Printer Detach	3-16
3.6.14	Program EPROM	3-17
3.6.15	Register Display	3-18
3.6.16	Read EPROM Contents	3-19
3.6.17	Register Modify (Interactive)	3-20
3.6.18	Trace	3-21
3.6.19	Transparent Mode	3-22
3.6.20	Dump Memory to Cassette Tape	3-23
3.6.21	Load Memory from Cassette Tape	3-24
3.7	PROGRAMMING THE EPROM	3-25
3.8	READING THE EPROM CONTENTS	3-26
CHAPTER 4	FUNCTIONAL DESCRIPTION	
4.1	INTRODUCTION	4-1
4.2	BLOCK DIAGRAM DESCRIPTION	4-1
4.2.1	Decode and Map Switch Logic	4-1
4.2.2	Monitor I/O	4-2
4.2.3	User Map Area	4-3
4.2.4	User I/O Ports	4-3
4.2.5	EPROM Programmer and Reader	4-3
CHAPTER 5	SUPPORT INFORMATION	
5.1	INTRODUCTION	5-1
5.2	INTERCONNECT SIGNALS	5-1
5.2.1	Terminal Port J3 Interconnect Signals	5-1
5.2.2	Host Port J4 Interconnect Signals	5-2
5.2.3	Printer Port J9 Interconnect Signals	5-2
5.3	PARTS LIST	5-4
5.4	SCHEMATIC DIAGRAM	5-7
APPENDIX A	S-RECORD OUTPUT FORMAT	A-1

LIST OF ILLUSTRATIONS

		<u>Page</u>
FIGURE 1-1.	Typical EVM Module	1-2
2-1.	Option Locations	2-2
2-2.	I/O Connector (J1)	2-3
2-3.	Audio Cassette Playback Phase Select (J2)	2-4
2-4.	HOST port (J4) CTS and DCD Signal Enable Headers (J5,J18)	2-4
2-5.	User Map 2K or 8K Select Header (J8)	2-5
2-6.	Write Protect Disable Header (J10)	2-5
2-7.	TERMINAL Port and HOST Port Baud Rate Select Headers (J11, J12)	2-6
2-8.	On-Board or External Clock Select (J13)	2-6
2-9.	External Clock Connection (J14)	2-7
2-10.	EPROM Programming Select Headers (J15, J16, J17)	2-7
2-11.	System Configuration	2-9
2-12.	Power Supply Installation	2-9
2-13.	Terminal Cable Details	2-10
2-14.	Host Cable Signal Line Connections	2-11
2-15.	Printer Cable Interconnection Diagram	2-13
3-1.	EVM Memory Map	3-3
3-2.	Dump Program Flow	3-26
3-3.	Dump Program Listing	3-28
4-1.	EVM Module Block Diagram	4-4
5-1.	EVM Module Parts Location	5-8
5-2.	EVM Module Schematic Diagram (7 Sheets)	5-9

LIST OF TABLES

TABLE 1-1.	Evaluation Module Specifications	1-3
3-1.	Monitor Primitive Commands	3-7
5-1.	Terminal Port J3 Interconnect Signals	5-1
5-2.	Host Port J4 Interconnect Signals	5-2
5-3.	Printer Port J9 Interconnect Signals	5-3
5-4.	EVM Module Parts List	5-4

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

This manual provides general information, hardware preparation, programming considerations, and functional description for the M1468705EVM Module, hereinafter called the EVM. A typical EVM is shown in Figure 1-1.

1.2 FEATURES

The features of the EVM include:

- . Economical means of evaluation for the MC1468705 MCU's
- . Requires only power supply and terminal for operation
- . Monitor/debugger
- . One line assembler/disassembler
- . Cassette download/upload
- . Host download
- . Printer port
- . Dual memory maps: 4K ROM - monitor map
4K RAM - user map
- . Programmer for on-chip EPROM
- . User MCU I/O available
- . Power-on reset
- . User reset/abort switches
- . External clock input
- . Wire-wrap area

1.3 SPECIFICATIONS

The EVM specifications are identified in Table 1-1.

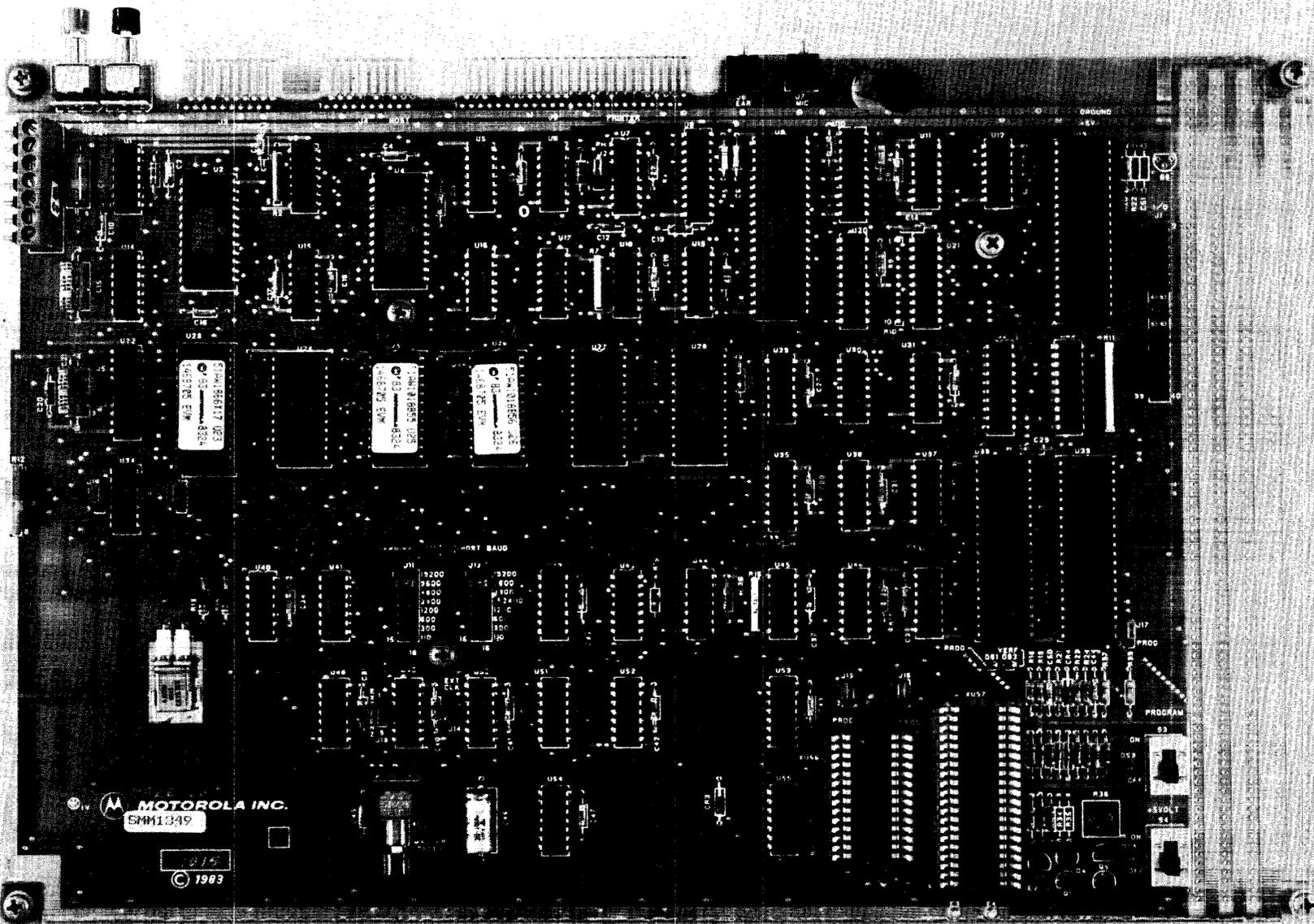


FIGURE 1-1. Typical EVM Module

TABLE 1-1. Evaluation Module Specifications

CHARACTERISTICS	SPECIFICATIONS
Microcomputer	Simulates MCl468705 MCU operation
Clock signal	Crystal controlled 4.0 MHz or external clock
Bus frequency	1.0 MHz or external frequency divided-by-four
Memory size	
RAM	112 bytes
Pseudo ROM	3968 bytes
Interrupts	
Four vectored	External, timer, timer in wait state, and software
User input/output signals	
Ports A, B, D	TTL voltage compatible input/output
Port C	CMOS-compatible input/output
Monitor input/output signals	
Terminal port	RS-232 (RxD and TxD only)
Host port	RS-232 (RxD and TxD only)
Printer port	Parallel Centronics interface
Cassette port	Audio cassette interface
Operating temperature	0° to 55° C
Power requirements	
Module	+5 Vdc at 1.0 A (max) +12 Vdc at 0.1 A (max)
Programmer	-18 Vdc at 50 mA (max)
Dimensions	
Width x height	14.50 x 9.75 in. (36.83 x 24.76 cm)
Board thickness	1.20 in. (3.05 cm)

1.4 GENERAL DESCRIPTION

The EVM provides a tool for generating, executing, and debugging MCU code. By providing all of the essential timing and I/O circuitry, the EVM simplifies user evaluation of the MCU.

Operation of an MCL468705 MCU is simulated by the resident MCL46805G2 MCU. Data transfer within the EVM is controlled by the monitor ROM firmware. In turn, this ROM is controlled from an RS-232C compatible user-supplied terminal. User object code may be generated using the resident one-line assembler/disassembler, or may be downloaded to the user program RAM through the host port or cassette port. User object code may then be executed using various debug commands in the monitor. Code may also be executed from reset using the USER RESET switch. MCU EPROM is simulated by write-protecting user program RAM during program execution.

Simulated MCU parallel I/O ports are available for external connection. These lines are also used to control the MCL468705 on-chip EPROM programmer. The EPROM is programmed and verified by inserting the MCL468705 device into the correct socket and entering a monitor command. The object code in the user program RAM is programmed into the MCU EPROM.

Programming is different for each MCU. The monitor firmware PROM must be changed by the user, depending on which MCU is to be programmed. A jumper select option allows either the on-board clock or a user-supplied external clock to be used. A printer port is provided which is Centronics-compatible. User circuitry may be added on the wire-wrap area of the module.

Optional kits of firmware and software are available for operation of the EVM with an EXORset host. The Motorola part numbers for the kits are M68EVMSETG2 and M68EVMSETF2.

CHAPTER 2

HARDWARE PREPARATION AND INSTALLATION INSTRUCTIONS

2.1 INTRODUCTION

This chapter provides unpacking, hardware preparation, 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/inspection of equipment.

CAUTION

AVOID TOUCHING AREAS OF CMOS CIRCUITRY;
STATIC DISCHARGE CAN DAMAGE INTEGRATED CIRCUITS.

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

2.3 HARDWARE PREPARATION

Figure 2-1 illustrates the location of the switches, LED indicators, sockets, power connections, and headers on the EVM. The module is shipped with factory-installed jumper configurations. The EVM is operational (except programming) with the factory-installed jumpers. It is necessary to make changes in the jumper arrangements for the following conditions:

- a. I/O connector (J1)
- b. Audio cassette playback phase select (J2)
- c. HOST port $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ signal enable (J5, J18)
- d. User map 2K or 8K select (J8)
- e. TERMINAL port and HOST port baud rate select (J11, J12)
- f. Write protect disable for user memory (J10)
- g. On-board or external clock select (J13)
- h. External clock connection (J14)
- i. EPROM programming select (J15, J16, J17)

2.3.1 I/O Connector (J1)

The I/O connector may be used as a convenient means of connecting user circuitry to MCl468705 interrupt and timer inputs and port C I/O. The I/O connector also connects to the simulated user I/O port A, port B, and port D. Signal names shown in Figure 2-2 are the MCl468705 equivalent.

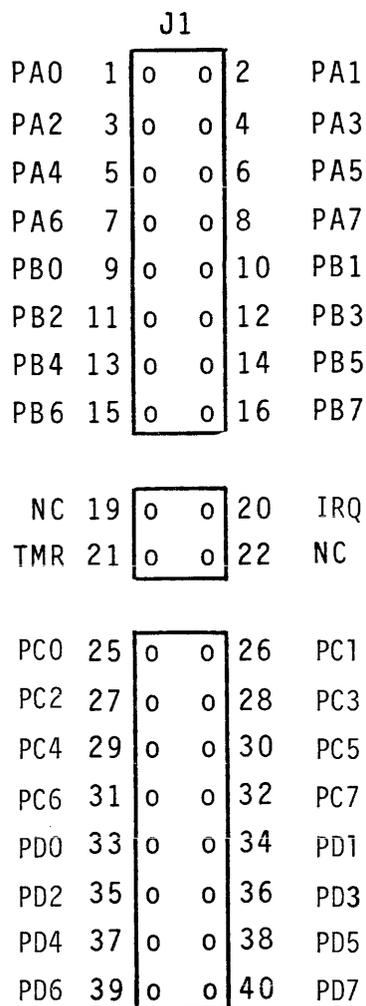


FIGURE 2-2. I/O Connector (J1)

2.3.2 Audio Cassette Playback Phase Select Header (J2)

Some audio tape cassette recorders invert the recorded signal. Header J2 allows the signal to be inverted to a compatible phase for the EVM. The EVM is factory-configured with the jumper positioned between pins 1 and 2 for a non-inverted playback signal (see Figure 2-3). For a recorder with an inverted playback signal, position the jumper between pins 2 and 3.

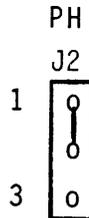


FIGURE 2-3. Audio Cassette Playback Phase Select (J2)

2.3.3 HOST Port (J4) $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ Signal Enable Headers (J5, J18)

The EVM is shipped with the HOST port (J4) $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ input signals held high, and with the jumpers on header J5 and J18 positioned between pins 1 and 2 (see Figure 2-4). Should the host computer or modem have full handshake capability, the jumpers may be positioned between pins 2 and 3 on headers J5 and J18.

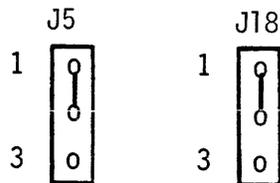


FIGURE 2-4. HOST port (J4) $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ Signal Enable Headers (J5, J18)

2.3.4 User Map 2K or 8K Select Header (J8)

Header J8 allows the user to select between a 2K user map for evaluating an MC1468705F2 and an 8K user map for evaluating an MC1468705G2. The EVM is factory-configured in the 8K mode with the jumper between pins 1 and 2 (see Figure 2-5).

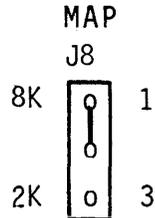


FIGURE 2-5. User Map 2K or 8K Select Header (J8)

2.3.5 Write Protect Disable Header (J10)

The EVM is shipped with the jumper between pins 1 and 2 (see Figure 2-6), which write protects the user program space 080-FFF during user program execution (pseudo ROM). To disable the write protect function, move the jumper to between pins 2 and 3.

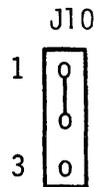


FIGURE 2-6. Write Protect Disable Header (J10)

2.3.6 TERMINAL Port and HOST Port Baud Rate Select Headers (J11, J12)

Headers J11 and J12 are provided to allow the baud rates to be set for compatibility with the user terminal and host computer. The EVM is factory-configured with the jumpers positioned for 9600 baud rate operation (see Figure 2-7). For other baud rate operation, refer to Figure 2-7.

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 = host 1200). If both baud rates are the same, a character may be lost occasionally by the terminal. Download is not affected by equal baud rates on the host and terminal.

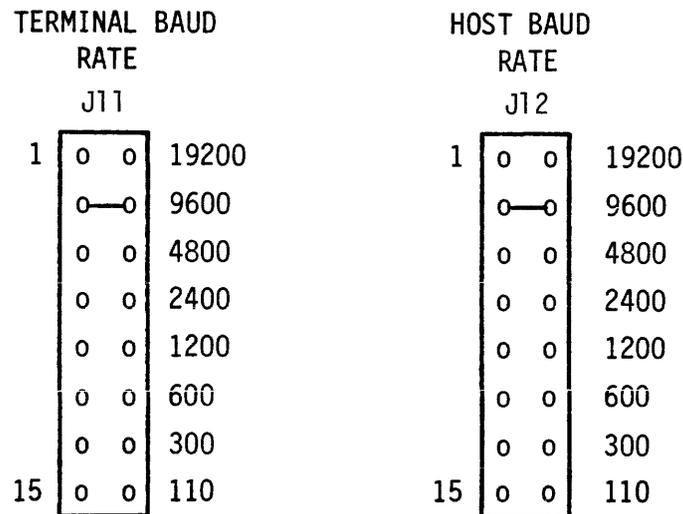


FIGURE 2-7. TERMINAL Port and HOST Port Baud Rate Select Headers (J11, J12)

2.3.7 On-Board or External Clock Select (J13)

The on-board clock source is a 4 MHz crystal. The EVM is factory-configured for 4 MHz operation with the jumper positioned between pins 2 and 3 (4 MHz) on header J13 (see Figure 2-8). Should the user wish to use a user-supplied external clock, the jumper should be positioned between pins 1 and 2 (EXT). The EXT position is used in conjunction with header J14.

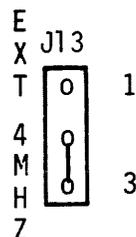


FIGURE 2-8. On-Board or External Clock Select (J13)

2.3.8 External Clock Connection (J14)

A two-pin header is provided for connection of an external TTL compatible clock (see Figure 2-9). Header pin 2 is ground (GND).

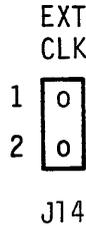


FIGURE 2-9. External Clock Connection (J14)

2.3.9 EPROM Programming Select Headers (J15, J16, J17)

The EVM is designed to program the MC1468705G2 and MC1468705F2. Headers are provided to connect the three control lines needed to handshake with the MC1468705 during programming. These three lines are also user port D lines (PD0, PD1, PD3 on J1). Jumpers positioned between pins 1 and 2 on headers J15-J17 disconnect the loading effect of the programmer from user I/O port D. The EVM is shipped with the four jumpers in the disconnected position (see Figure 2-10). Refer to Chapter 3 for programming information.

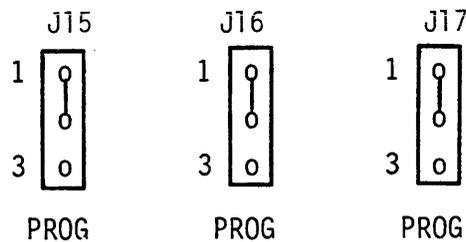


FIGURE 2-10. EPROM Programming Select Headers (J15, J16, J17)

2.4 INSTALLATION INSTRUCTIONS

The EVM is a stand-alone system which is designed for table-top operation. A user-supplied power supply is required for operation and programming. A user-supplied terminal is required for entering and changing data. A host computer (e.g., EXORciser), a printer, and/or an audio cassette may be connected, but are not required for normal operation (see Figure 2-11).

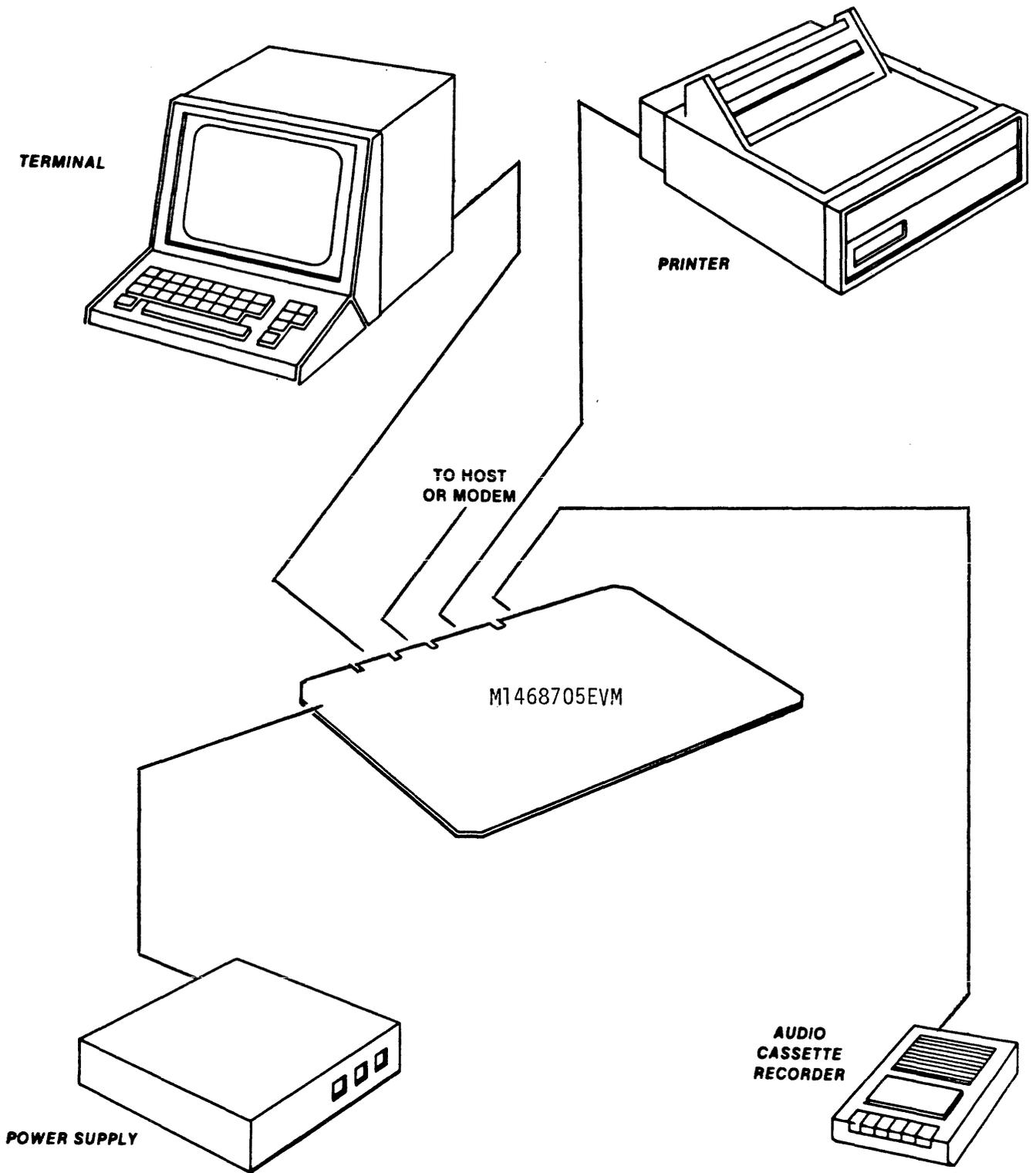


FIGURE 2-11. System Configurations

2.4.1 Power Supply Installation

The EVM requires +5 Vdc @ 1 A, +12 Vdc @ 0.1 A, -12 Vdc @ 0.1 A for operation, and -18 Vdc @ 50 mA for programming. A recommended power supply for +5 V, +12 V is a Condor model TAA-16W. The -18 programming voltage may be supplied by two 9 V batteries or a good -18 V, 50 mA power supply.

The terminal block P1 on the EVM is designed for 14-22 AWG wire. See Figure 2-12 for connections.

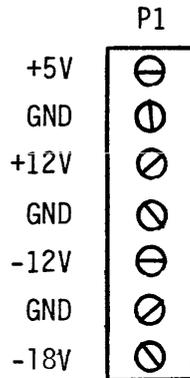


FIGURE 2-12. Power Supply Installation

2.4.2 Terminal Installation

An RS-232C compatible terminal is required to enter data, debug a user program, print data, and program an MCU EPROM. Most terminals can be connected to the EVM with a cable requiring a 20-contact edge connector on the EVM end and a 25-contact D-type subminiature connector on the terminal end. The subminiature connector can be either pin (male) or socket (female), as required by the user terminal. Both of these cable types are available from Motorola:

<u>PART NUMBER</u>	<u>DESCRIPTION</u>
M68RS232M	RS-232C Cable, Edge/Male
M68RS232F	RS-232C Cable, Edge/Female

The user may manufacture a cable suitable for this purpose. Figure 2-13 shows a detail of the cable, lists several suitable vendor part numbers (any equivalent part may be used), and shows the conductor line designations present on the EVM. The cable requires a 20- or 25-conductor flat ribbon; connectors should be installed according to manufacturer's instructions. The edge connector may be keyed to prevent incorrect cable connection.

The cable is connected between the EVM edge connector J3 (TERMINAL) and the user terminal. Be sure to orient pin 1 of the cable with pin 1 of J3.

2.4.3 Host Computer (Modem) Installation

The EVM can be operated with a host computer directly or by modem. The host computer (e.g., EXORciser) or modem is connected to the EVM with a cable which is similar to the terminal cable. Refer to paragraph 2.4.2 for details of purchased or user-manufactured cable. The edge connector may be keyed to prevent incorrect cable connection. Figure 2-14 shows the host computer cable signal line connections which are present on the EVM.

The cable is connected between the EVM edge connector J4 (HOST) and the user host computer or modem. Be sure to orient pin 1 of the cable with pin 1 of J4.

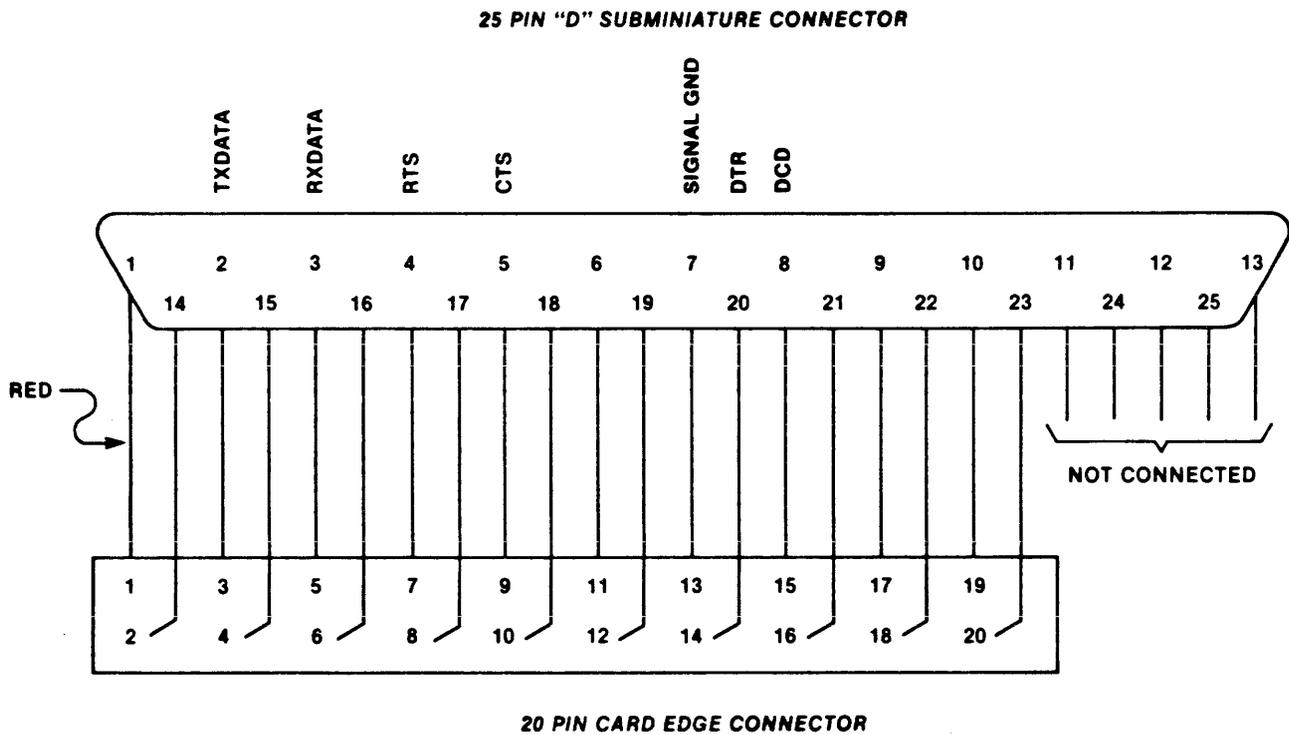


FIGURE 2-14. Host Cable Signal Line Connections

2.4.4 Printer Installation

The EVM is designed to drive any Centronics interface printer. When purchased from Motorola, the printer is supplied with a cable. The cable may be purchased from Motorola by ordering part number MEX68PIC.

The user may manufacture this cable, although it is a more complex assembly than ribbon cable and connectors. Figure 2-15 shows the interconnection diagram of the cable and lists suitable connector part numbers. The cable may be keyed on the edge connector to prevent incorrect cable connection.

The cable is connected between the EVM edge connector J9 (PRINTER) and the user printer. Be sure to orient pin 1 of the cable with pin 1 of J9.

2.4.5 Audio Cassette Player Installation

An audio cassette player may be used to save program object code from the EVM. Two signal lines plus ground must be connected to the cassette player from EVM J6 (EAR) and J7 (MIC). An inexpensive cassette player, GE model 3-5153, may be used.

The user must buy or make a cable suitable for the cassette player used. An audio patch cord (Radio Shack part number 42-2420) and an adapter (Radio Shack part number 274-327) may be used with most recorders which use miniature phone plugs.

The cassette player is connected to the EVM by connecting the adapter to EVM J6 or J7 and the patch cord between the adapter and the cassette player. The data out signal at EVM J7 (MIC) should be connected to the MICROPHONE input of the cassette recorder for saving programs. The data in signal at EVM J6 (EAR) should be connected to the EARPHONE output of the cassette player for loading programs into the EVM memory. Ground connections are part of the patch cords.

EXAMPLE:
3M #3415-0001

EXAMPLE:
AMPHENOL #57-10360-13
CINCH #57-10360

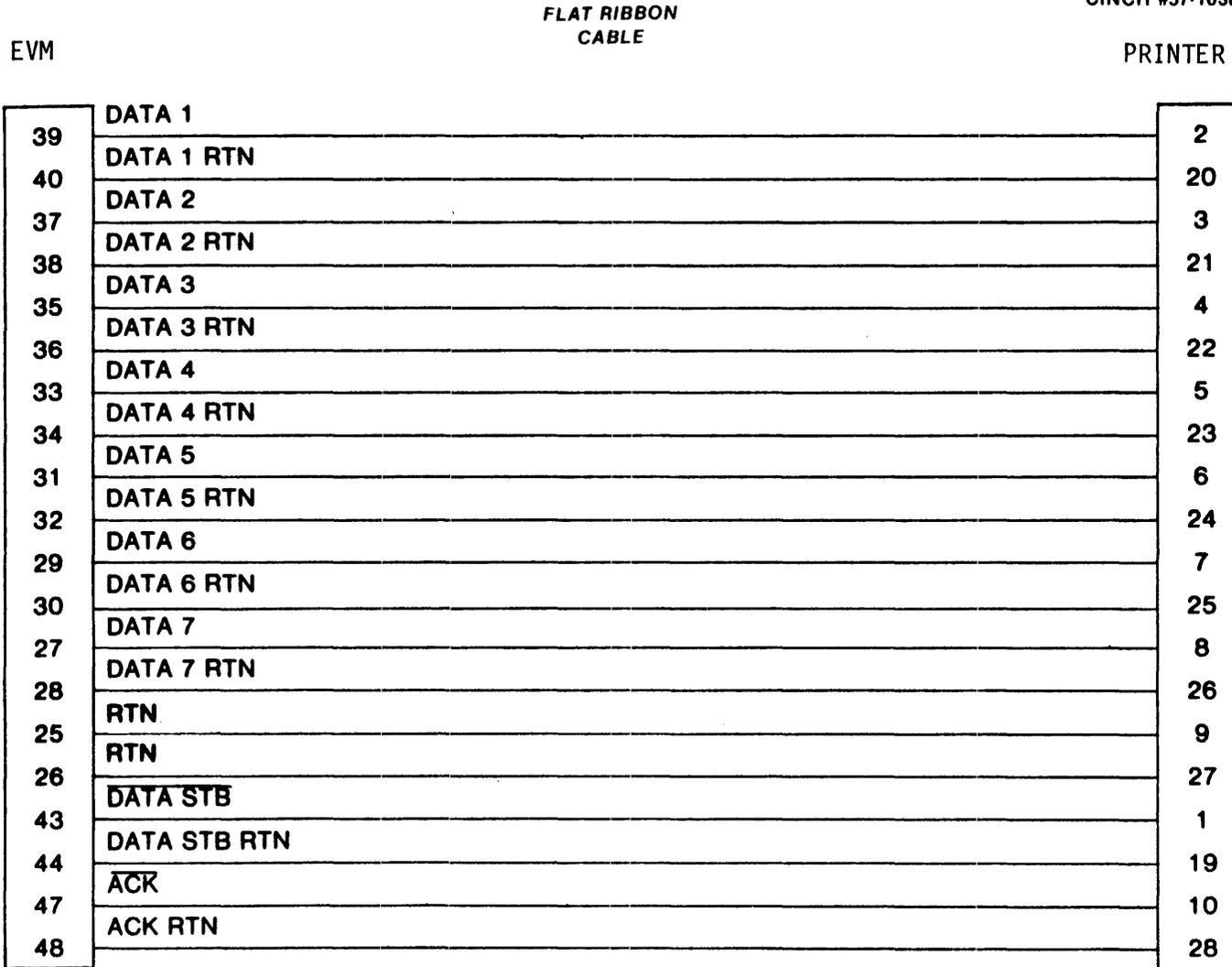


FIGURE 2-15. Printer Cable Interconnection Diagram

2-13/2-14

CHAPTER 3

OPERATING INSTRUCTIONS

3.1 INTRODUCTION

This chapter describes the operating procedures, MCU EPROM programming procedures, and the debug/monitor commands for the EVM.

3.2 SWITCHES AND INDICATORS

The EVM is provided with five switches and four LED indicators. The indicators show the status of certain functions during EPROM programming, and may prompt the user to perform some function. See Figure 2-1 for location of switches and indicators.

3.2.1 Switches

Two switches (S1, S2) control the user reset and abort functions of the module. Two switches (S3, S4) control the EPROM programming operation, together with the LED status indicators. A master reset switch (S5) is also provided.

USER RESET - The USER RESET pushbutton switch (S1), located on the upper left corner of the EVM, is used to reset the MPU and all user I/O, and switch to the user map. The USER RESET switch may be used as a permanent map switch for execution of user code from the user reset vector (see user map description).

ABORT - The ABORT pushbutton switch (S2), located on the upper left corner of the EVM, is used to return to the monitor map from the user map (see user map description). The ABORT switch has no effect when operating in the monitor map.

PROGRAM - The PROGRAM slide switch (S3), located on the lower right corner of the EVM, is used to activate +12 Vdc and +27 Vdc power used to program the EPROM. LED indicator (CR7) is lit when this switch is in the ON position.

+5VOLT - The +5VOLT slide switch (S4), located on the lower right corner of the EVM, is used to activate +5 Vdc power used during the programming and read operations. LED indicator (CR8) is lit when this switch is in the ON position.

MASTER RESET - The MASTER RESET switch (S5), located on the lower left of the EVM, is used to perform a power on reset of the entire EVM.

3.2.2 Indicators

Four LED indicators show the status of the programmer circuitry.

PROG - Red LED indicator (DS1), when lit, informs the user that the device in socket XU56 or XU57 is being programmed.

VERF - Red LED indicator (DS2), when lit, informs the user that the device in socket XU56 or XU57 has been programmed and verified.

PROGRAM - Red LED indicator (DS3), when lit, informs the user that PROGRAM switch (S3) is in the ON position, applying -12 Vdc and -18 Vdc to the device to be programmed.

+5VOLT - Red LED indicator (DS4), when lit, informs the user that +5VOLT switch (S4) is in the ON position, applying +5 Vdc to the device to be programmed or read.

3.3 MEMORY MAP CONCEPT

The EVM is designed to operate in either one of two maps (monitor map or user map). The monitor map contains 4K bytes of RAM/ROM which includes the monitor ROM, monitor RAM/stack, monitor I/O, a monitor control register, and a user stack pointer (see Figure 3-1). The user map, which simulates the MC1468705 memory map, contains the user I/O ports, user RAM/stack, and user pseudo ROM (see Figure 3-1).

The memory maps are decoded with a field programmable gate array.

NOTE

The entire 4K-byte memory map is available to the user, although all MC1468705 MCU devices do not have EPROM throughout the entire 4K-byte map. Refer to the specific device data sheet for valid program locations.

3.3.1 Monitor Map

The terminal, host, cassette, and printer ports are available only in the monitor map. User programs in the user map can not access the peripheral ports.

3.3.2 User Map

All user I/O ports (A, B, C, D) are available in the user map for evaluation. All ports are reconstructed off chip except port C.

User Pseudo ROM - The user program space (080-FFF) is RAM, which 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.

Header J8 on the EVM provides a simulated 8K user map for evaluating the MC1468705G2, and a simulated 2K user map for evaluating the MC1468705F2. Although the EVM has only a 4K map, the user vectors fetched from 1FF6-1FFF or 7F6-7FF for the MC1468705G2 and MC1468705F2, respectively, appear to the EVM MCU at 1FF6-1FFF. This is accomplished by shadowing the lower 4K map into the upper 4K map (8K option), or by multiple shadowing the lower 2K map into the upper 6K map (2K options).

3.4 MAP SWITCHES

Two types of map switches are provided -- a temporary switch which allows modification of user memory under monitor control, and a permanent switch which allows execution of user programs.

3.4.1 Program Execution

A permanent map switch for execution of user code can be initiated in two ways:

Command from the monitor

The monitor commands G (execute from address), P (proceed from breakpoint), and T (trace instructions) allow execution of user programs.

Pressing the USER RESET switch (S1)

When the USER RESET switch is pressed, the MCU and all user I/O are reset, maps are immediately switched to the user map, and the reset vector is fetched from the user map by the MCU. Breakpoints are ignored during this mode of execution. Execution can be halted only with the ABORT switch.

Return to monitor control occurs in two ways:

A breakpoint or the end of a trace function in the monitor

Execution of user code continues until a breakpoint (SWI) is encountered or the ABORT switch is pressed. The map is switched back to the monitor map after the registers are saved on the user stack. An SWI which occurs when no breakpoint is set -- and when the ABORT switch is not being pressed -- does not cause a map switch. This allows user SWI's to be executed in real time.

Pressing the ABORT switch (S2)

The ABORT switch, when pressed, forces an SWI on the data bus until the next LIR. The registers are saved on the user stack and maps switched to the monitor. The ABORT switch has no effect on the monitor map.

3.5 LIMITATIONS

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

Although an external clock input may be used to evaluate the MCU at reduced speeds, LOAD, TPLD, and TPDU functions may be used only while operating at full 1-MHz rate. To operate at reduced speed with an external clock, the following procedure should be performed.

- a. Load the program (LOAD, TPLD, or TPDU commands) at full 1-MHz clock rate.
- b. Press and hold the MASTER RESET switch while positioning the jumper at J13 to pins 1 and 2. When the jumper is in place, release the MASTER RESET switch.

- c. The user program can now be executed and debugged at reduced clock speed.
- d. To return to full clock rate, repeat steps b and c moving jumper on header J13 to pins 2 and 3.

The baud rates of the terminal and host 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.

The audio cassette format is unique to the EVM and is not compatible with any other tape format.

The printer port does not support any of the error detect lines. Only the data, strobe, and acknowledge lines are implemented.

3.6 OPERATING PROCEDURE

The monitor is the resident firmware 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. The commands fall into five general categories:

- a. Commands which allow the user to display or modify memory.
- b. Commands which allow the user to display or modify the various internal registers of the MCU.
- c. Commands which allow the user to execute a program under various levels of control.
- d. Commands which control access to the various input/output peripherals connected to the EVM.
- e. Commands which program or read the MCU EPROM.

The monitor requires five bytes of user stack for user register storage on monitor (re)entry. No limitation is placed on the user as to other uses of the stack.

All user registers are in an unknown state on monitor power-up.

3.6.1 System Initialization

Turning power on initializes the EVM system. The MCU is reset and the monitor invoked. The terminal displays:

```
EVMbug705 REV X.XX   S=XX   C=XX   A=XX   X=XX   P=XXXX
?
```

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

Other means can be used to reinitialize the EVM firmware (see paragraph 3.2.1).

3.6.2 System Operation

After initialization or return of control to the monitor, the terminal displays the prompt ? and waits for a response. If an invalid response is entered, the terminal displays a "What?", and rings the bell.

A dollar sign (\$) may precede any number input except bit number specifications in the BCLR/BSET and BRCLR/BRSET assembler mnemonics. All numbers are interpreted as hexadecimal -- the (\$) is not required.

The user can call any of the commands supported by the monitor. A standard input routine controls the system while the user types a line of input. Command processing begins only after the line has been terminated by a carriage return (see note 6, paragraph 3.6.3).

3.6.3 Command Line Format

The command line format is:

? [NO]<command> [<parameters>]

where:

?	Is the prompt from the EVM monitor.
NO	Is the negative (opposite) of some primitive commands.
command	Is the primitive command.
parameters	Are separated by a single space.

NOTES

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

< >	Enclose a syntactical variable.
[]	Enclose optional fields.
[]...	Enclose optional fields which may be repeated.

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

2. Fields are separated by a single space.
3. A maximum of 30 characters may be entered on a command line. After the 30th character is entered, the monitor automatically terminates the line and begins processing the 30-character line.

4. RETURN (CR) (carriage return) enters the command line and causes processing to begin.
5. CTRL-X cancels the entire input line.
6. Parameters are interpreted to be the last two or three characters in the parameter field. Parameter errors may be corrected by re-entering the parameter without a space.

3.6.4 Primitive Commands

Monitor primitive commands are listed in Table 3-1.

TABLE 3-1. Monitor Primitive Commands

COMMAND	DESCRIPTION
ASM <address>	Assembler/disassembler (interactive)
BF <address1> <address2> <data>	Block fill memory with data
[NO]BR [<address>]...	Breakpoint set/reset/remove
G [<address>]	Go (execute program)
LOAD [<offset>]<= text>	Download memory (S-records) from host file
MD <address1> [<address2>]	Memory display
MM <address>	Memory modify (interactive)
P [<count>]	Proceed (from breakpoint)
[NO]PA	Printer attach/reset printer attach
PROG	Program EPROM
RD	Register display
READ	Read EPROM contents
RM	Register modify (interactive)
T [<count>]	Trace (instructions)
TM [<exit character>]	Transparent mode
TPDU <address1> <address2>	Dump memory to cassette tape
TPLD [<offset>]	Load memory from cassette tape

Each of the individual commands is described in the following paragraphs.

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 translated 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 instructions are translated.

The ASM command allows the user to create, modify, and debug MC6805 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.

EXAMPLE:

- a. ?LDA ,X
- b. ?STA 10,X
- c. ?BRCLR 3,\$52,200
- 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. It is left to the user to watch for this.

Each of the instruction pairs ASLA/LSLA and ASLX/LSLX have the same opcode, so disassembly always reveals LSLA or LSLX, respectively.

A dollar sign (\$) may precede any number input except bit specifications in the BCLR/BSET and BRCLR/BRSET assembler mnemonics.

The assembler is terminated by entering a period (.) followed by a carriage return. 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.

EXAMPLE

<u>?ASM 200</u>			
0200	FF	STX ,X	<u>?BRSET 7,10,210</u>
0200	0E 10 0D	BRSET 7,\$10,\$0210	
0203	FF	STX ,X	<u>?LDA #\$55</u>
0203	A6 55	LDA #\$55	
0205	FF	STX ,X	<u>?STA ,X</u>
0205	F7	STA ,X	
0206	FF	STX ,X	<u>?LDX 654</u>
0206	CE 06 54	LDX \$0654	
0209	FF	STX ,X	<u>?BRA 209</u>
0209	20 FE	BRA \$0209	
020B	FF	STX ,X	<u>?.</u>

BF <address1> <address2> <data>

where:

address1 Is the lower limit for fill operation.
 address2 Is the upper limit for fill operation.
 data Is the 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 0-F or 60-7F. The user I/O configurations and interrupt masks are located at 0-F. 60-7F contain the user stack area. The user registers are stored in the five bytes immediately below the user stack.

No operation is performed if address1 is greater than address2.

EXAMPLE

COMMENTS

?BF 101 200 FF

Fill each byte of memory from 101 through 200 with the pattern FF.

?BF 10 10 0

Sets location 10 to 0.

3.6.7 Breakpoint Set
Breakpoint Remove

BR
NOBR

BR [<address>]...
NOBR [<address>]

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

The NOBR command is used to remove one or all breakpoints from the breakpoint table.

User should not put a breakpoint on an SWI instruction as this is the instruction that the monitor uses to breakpoint/single-step a user program. The user may use the SWI instruction in the user program.

When a new breakpoint is entered, the breakpoint table is first cleared. A maximum of three breakpoints may be set.

After setting or removing a breakpoint, the current breakpoint addresses, if any, are displayed.

COMMAND FORMAT

DESCRIPTION

BR	Display all current breakpoints.
BR <address>	Set a breakpoint.
BR <address1> <address2>...	Set several breakpoints.
NOBR	Remove all current breakpoints.
NOBR <address>	Remove a breakpoint.

EXAMPLE

COMMENTS

? <u>BR</u> 080	Sets a breakpoint at 080.
? <u>BR</u> 084 09A 984	Sets three breakpoints.
? <u>NOBR</u>	Removes all breakpoints.
? <u>BR</u>	Displays all current breakpoints.

G [<address>]

where:

address Is the starting address where execution begins.

The G command allows the user to initiate user program execution in free run in real time. The user may optionally specify a starting address where execution is to begin. Execution starts at current PC unless starting address is specified. Program execution continues until a breakpoint is encountered or the ABORT switch is pressed.

EXAMPLE

COMMENT

?G

Begin execution at current PC location.

?G 100

Begin execution at location 100.

LOAD [<offset>]<=text>

where:

offset Is the optionally specified offset to be applied to all load addresses retrieved from the loaded S-records.

=text The text following the = is a host command sent to the host port, which instructs the host to download S-records.

The load command moves object data in S-record format (see Appendix A) from an external host to memory.

When loading data from a host file, the data received by the monitor is echoed to the terminal so that the user may monitor the load progress. S-records are not echoed to the printer due to timing considerations. If the terminal is running at a baud rate less than the host, the S-record echo may be scrambled but the data is entered correctly to memory.

As the monitor processes only valid S-record data, it is possible for the monitor to hang up during a load operation (there is no internal timeout because the terminal and host may be running at different baud rates). Therefore, prior to the load operation, the user should define a reset vector to execute a branch-to-self instruction. Then, if the monitor should become hung up during a load operation, the user may do a reset/abort on the processor and return the monitor to the command state.

The load operation may optionally specify an offset to be applied to all load addresses retrieved from the loaded S-records. No check is made that the address does not roll over from high memory to low memory after adding the offset.

EXAMPLE

COMMAND

?LOAD =COPY FILE.LX:1,#CN

Loads the data from the host (EXORciser) to the EVM.

?LOAD 100=COPY FILE.MX,#

Loads, with offset, the data from the host (EXORmacs) to the EVM.

MD <address1> [<address2>]

where:

address1 Is the beginning address of the memory to be displayed.

address2 Is the ending address of the memory to be displayed.

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

No operation is performed if address1 is greater than address2.

EXAMPLE

?MD 100 110

```
100 0C 00 00 30 6D 28 0C 00 00 39 6E 10 02 80 00 00   ...0m(...9n.....
110   00 0F 11 C0 10 38 1E 3C 00 E4 4E 4E 0C 00 00 41   ...@.8.<.dNN...A
```

?MD 10A

```
10A 6E 10 02 80 00 00 00 0F 11 C0 10 38 1E 49 00 E4   n.....8.<..
```

MM <address>

where:

address Is the memory location at which to start display/modify.

The MM command allows the user to examine/modify user memory locations in an interactive manner.

Caution should be used when modifying the user stack area because the user registers are stored in the five bytes immediately below the user stack.

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

[<data>](CR) - Update location and sequence forward.

[<data>^(CR) - Update location and sequence backward.

[<data>]=(CR) - Update location and reopen same location.

[<data>].(CR) - Update location and terminate.

An entry of only ".(CR)" terminates the register modify interactive mode.

CTRL X may be used to cancel any input line; the monitor remains in the MM command.

EXAMPLE

COMMENT

?MM 100
0100=FF?00=
0100=00?
0100=FF?00=
0100=00?.

Display memory location 100, modify FF to 00, and display same location, then terminate.

3.6.12 Proceed (from Breakpoint)

P

P [<count>]

where:

count Is the number (in hexadecimal) 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.

EXAMPLE

?P 5

3.6.13 Printer Attach
Printer Detach

PA
NOPA

PA
NOPA

The PA command allows the user to software attach the line printer so that information sent to the terminal is also printed. The printer is physically attached to EVM J9 (PRINTER).

The NOPA command allows the user to detach the printer from the terminal.

The PA command should only be entered when a printer is physically connected to the EVM. Otherwise, the monitor hangs up while waiting for a response from the printer. To recover from this hangup, the printer must be attached, the power removed from the EVM, or the MASTER RESET switch (S5) pressed.

PROG

The PROG command is used to program the MC1468705 MCU internal EPROM. Prior to entering this command, jumpers must be positioned, switches must be set, and the unprogrammed MCU installed in the proper socket. Paragraph 3.7 describes the procedure for programming the EPROM in detail. Because of the limited monitor memory size, only the MC1468705G2 can be programmed with the G2 version of the monitor program, and only the MC1468705F2 can be programmed with the F2 version of the monitor program.

RD

The RD command displays the MCU register contents.

COMMAND FORMATDESCRIPTION

?RD

The contents of the following registers are displayed:

S = stack pointer
C = condition code
A = accumulator
X = index
P = program counter

EXAMPLE

?RD

S=7F C=FA A=FF X=10 P=0047

3.6.16 Read EPROM Contents

READ

READ

The READ command allows the user to display the contents of the programmed internal EPROM of the MC1468705 MCU.

Paragraph 3.8 describes the procedure for reading the programmed EPROM.

RM

The RM command is used to modify the MPU registers. The RM command takes no parameters and starts by displaying the S (stack pointer) register and then displaying the C (condition code) register and allowing changes to be made to the condition code register. The order of the register display follows:

S (stack pointer) - No modification allowed; moves to next line.

C (condition code) - Only bits 0-4 are significant.

A (user A register)

X (user X register)

P (program counter) - Two bytes displayed/allowed.

The user may not change the stack pointer using the RM command.

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

[<data>](CR) - Update register and sequence forward.

[<data>^(CR) - Update register and sequence backward.

[<data>]=(CR) - Update register and reopen same location.

[<data>).(CR) - Update register and terminate.

An entry of only ").(CR)" terminates the register modify interactive mode.

The user may use CTRL-X to cancel any input line; the monitor remains in this command.

<u>EXAMPLE</u>	<u>COMMENT</u>
?RM	Command entered.
S=7D	Shows S status and skips to C.
C=00?CC	Change C register.
A=AA?FE=	Change A register. Re-examine register.
A=FE?	Look at A register change.
X=10?77	Change X register.
P=0056?100^	Change P register. Sequence backward.
X=77?._	Look at X register. Terminate command.
?=RD	Enter RD command.
Regs S=7D C=CC A=FE X77 P=0100	

T [<count>]

where:

count Is the number (in hexadecimal) 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 user program execution stops upon the PC encountering a breakpoint address. Depending upon the user program execution results, the monitor issues one of the following target status messages before displaying a prompt for the next user input.

Trace	S=nn	C=nn	A=nn	X=nn	P=nnnn
Brkpt	S=nn	C=nn	A=nn	X=nn	P=nnnn

The user should not try to trace an instruction that branches to itself (e.g., BRCLR Y,XX,*). Because the monitor puts 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 executes. The user may enter a G command while the PC points to this type of instruction as long as the instruction is not at a breakpoint address.

The monitor issues an error message if the user attempts to trace at an address that contains an invalid opcode.

EXAMPLE

COMMENT

?T

Trace S=7D C=FC A=00 X=10 P=0051

Single trace.

?T 5

Trace S=7D C=FA A=00 X=10 P=0054

Trace S=7D C=FC A=00 X=10 P=0056

Trace S=7D C=FC A=00 X=10 P=0051

Trace S=7D C=FA A=00 X=10 P=0054

Trace S=7D C=FC A=00 X=10 P=0056

Multiple trace with count of 5 entered.

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. All input/output between them is 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 (e.g., ?TM !). The exit character can be any keyboard character (printable or non-printable). The default exit character is CTRL-A. When the user task is completed, the transparent mode is terminated by entering the same exit character.

<u>EXAMPLES</u>	<u>COMMENTS</u>
<u>?TM</u> <u>MDOS</u> <u>=DIR</u> . . .	Command followed by a carriage return. CTRL-A is default exit character. MDOS program entered.
<u>=CTRL-A</u> ?	Task complete. Enter exit character CTRL-A. Transparent mode terminated.
<u>?TM !</u> <u>MDOS</u> <u>=DIR</u> . . .	Command followed by exit character (!). MDOS program entered. Directory called up.
<u>=!</u> ?	Task complete. Enter exit character (!). Transparent mode terminated.

TPDU <address1> <address2>

where:

address1 Is the beginning address of the memory to be dumped.
 address2 Is the ending address of the memory to be dumped.

The TPDU command outputs memory contents (S-records) (see Appendix A) from beginning address through ending address to an audio cassette through EVM J7. The recording technique is unique to the EVM.

When dumping data to cassette tape, the data generated by the monitor is echoed to the terminal so that the user may monitor the dump process. S-records are not echoed to the printer due to timing considerations. If the terminal is running at a baud rate less than the cassette (cassette runs at approximately 300 baud), the S-record echo is scrambled but the data is entered on the cassette tape correctly.

When saving data to cassette tape, the monitor writes a leader to the cassette tape. The leader takes approximately 15 seconds to write. Therefore, no data echo is seen on the screen during this delay. The leader allows the user time to start the cassette in record mode and not miss any data when loading memory back from cassette. A 4K memory dump to the cassette takes approximately two minutes.

No operation is performed if the starting address is greater than the ending address.

EXAMPLE

```
?TPDU 200 20F
S0030000FC
S11302006654BBCE6704610013521E3C004447F899
S9030000FC
```

COMMENT

Dump to be performed starting at memory location 200.

TPLD [<offset>]

where:

offset Is the optionally-specified offset to be applied to all load addresses retrieved from the loaded S-records.

The TPLD command loads data in S-record format from cassette tape to memory.

The load operation may optionally specify an offset to be applied to all load addresses retrieved from loaded S-records. No check is made that the address does not roll over from high memory to low memory after adding the offset.

When loading data from cassette tape, the data received by the monitor is echoed to the terminal so that the user may monitor the load process. S-records are not echoed to the printer due to timing considerations. If the terminal is running at a baud rate less than the cassette (cassette runs at approximately 300 baud), the S-record echo is scrambled but the data is entered to memory correctly. A 2K memory load from cassette tape takes approximately one minute.

Because the monitor processes only valid S-record data, it is possible for the monitor to hang up during a load operation (there is no internal timeout because the terminal and cassette may be running at different baud rates). Therefore, prior to the load operation, the user should define a reset vector to execute a branch-to-self instruction. Then, if the monitor should become hung up during a load operation, the user may do a reset/abort on the processor and return the monitor to the command state.

EXAMPLE

```
?TPLD
S0030000FC
S11302006654BBCE6704610013521E3C004447F899
S9030000FC
```

COMMENT

Loads data from cassette to EVM with no offset.

```
?TPLD 100
S0030000FC
S11302006654BBCE6704610013521E3C004447F899
S9030000FC
```

Loads data with offset of 100.

3.7 PROGRAMMING THE EPROM

The monitor firmware allows the user to program MC1468705G2 and MC1468705F2 internal EPROM (see paragraph 3.6.14). When the user wants to program the MCU internal EPROM, jumpers must be repositioned, switches set, and indicators observed. Refer to device data sheets for details of the internal EPROM.

Paragraph 3.8 describes the READ function and the read program.

It is assumed that the user program has been debugged and resides in the EVM memory.

The following steps are used to program the internal EPROM:

- a. Check switches S3 and S4 at the lower right of the EVM to ensure that the switches are in the OFF position.
- b. Connect -18 Vdc source (see paragraph 2.4.1).
- c. On headers J15, J16, and J17, remove the jumpers from between pins 1 and 2 and install between pins 2 and 3 (see paragraph 2.3.9).
- d. Insert unprogrammed MC1468705 into socket XU56 for the 28-pin device, or socket XU57 for the 40-pin device.
- e. Position switch S4 (+5VOLT) to the ON position and observe LED indicator DS4 is lit.
- f. Position switch S3 (PROGRAM) to the ON position and observe LED indicator DS3 is lit.
- g. Enter the command PROG (see paragraph 3.6.14).
- h. LED indicator PROG (DS1) lights while programming is in progress. Approximately one second later, LED indicator VERR (DS2) lights, indicating the transfer has been verified. A monitor prompt (?) indicates program/verify function is complete. If the VERR indicator does not light, the programming is not correct. It is recommended that a second attempt at programming the same part be made. This is often successful when verification fails.
- i. Position switches S3 and S4 to the OFF position.
- j. Remove the programmed device. Repeat the procedure from step d to program additional devices.
- k. When programming task is complete, position the jumpers on headers J15, J16, and J17 to pins 1 and 2.

3.8 READING THE EPROM CONTENTS

The monitor firmware allows the user to read the contents of the programmed EPROM. A special dump program must be included with the user program 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 80 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 (listed in Figure 3-3) is written for an MC1468705G2 (reset vector at 1FFE-1FFF).

User I/O bit PC3 must not be held low by the user application circuitry during reset. After reset, all user I/O is available for applications (see Figure 3-2).

The dump program for the MC1468705F2 is available with the F2 monitor program.

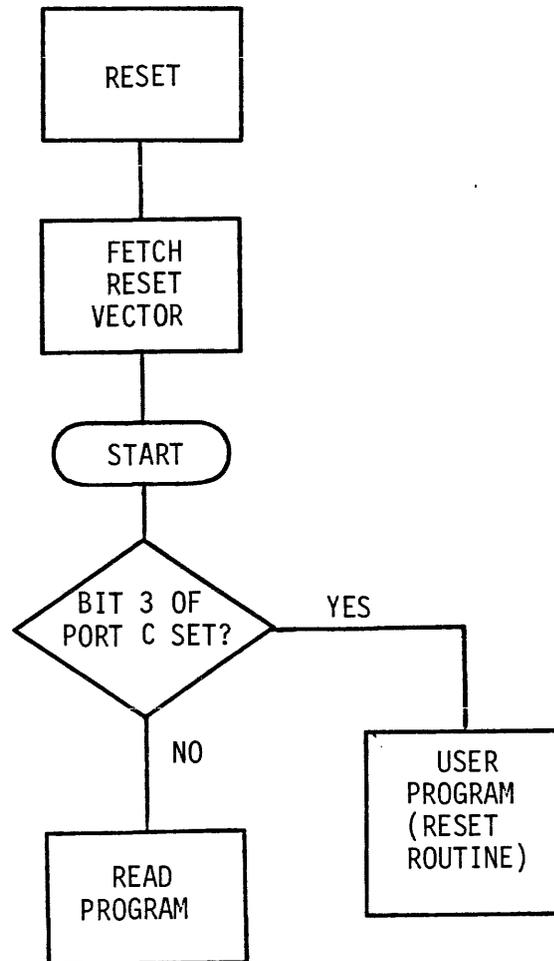


FIGURE 3-2. Dump Program Flow

The procedure to perform the READ function is as follows.

NOTE

It is important that user circuitry connected to I/O connector J1 does not pull down any of the user port lines during programming or reading. Full CMOS levels are necessary on these lines.

- a. Check switches S3 and S4 at the lower right of the EVM to assure that the switches are in the OFF position.
- b. On headers J15, J16, and J17, remove the jumpers from between pins 1 and 2 and install between pins 2 and 3 (see paragraph 2.3.9).
- c. Insert programmed MC1468705 into socket XU56 for the 28-pin device, or socket XU57 for the 40-pin device.
- d. Position switch S4 (+5VOLT) to the ON position and observe LED indicator DS4 is lit.
- e. Enter the command READ (see paragraph 3.6.16).
- f. The monitor displays the prompt (?) when the READ function is complete.
- g. Position switch S4 to the OFF position.
- h. Remove the programmed device. Repeat the procedure from step c to read additional devices.
- i. When read task is completed, position the jumpers on headers J15, J16, and J17 to pins 1 and 2.

```

1 P *****
2 P *
3 P * This program dumps the 1468705G2 ROM data from $80 - $BF4 and from *
4 P * $1FF5 - $1FFF to the EVM, if the DATA REQUEST bit (PC3) is held low *
5 P * at reset. If PC3 is held high at reset, the USER program is executed. *
6 P * After reset, PC3 can be used without this restriction. *
7 P *
8 P * DUMP 1468705G2 I/O DEFINITIONS *
9 P *
10 P * DATA PBO - PB7 ---> EVM *
11 P * RESET RESET <--- EVM *
12 P * DATA REQUEST PC3 <--- EVM *
13 P * DATA VALID PDO ---> EVM *
14 P *
15 P *****
16 P
17 A 0001 DATA EQU $01 Data Bus (PORT B)
18 A 0005 DATADR EQU $05 Data Bus direction register
19 A 0002 REQUEST EQU $02 Data Request (PORT C)
20 A 0003 VALID EQU $03 Valid Data (PORT D)
21 A 0007 VALIDDR EQU $07 Valid Data direction register
22 A 0080 RESET EQU $80 Reset Vector
23 A 00F5 ROMEND EQU $F5 LSB of the last ROM address to dumped
24 A 001F VECTRS EQU $1F MSB of Vector addresses
25 A 0060 FETCH EQU $60 Start of self-modifying code
26 A 0061 PAGE EQU FETCH+1 Address pointer MSB
27 A 0062 BYTE EQU FETCH+2 Address pointer LSB
28 P
29 P *****
30 P *
31 P * SELF-MODIFYING CODE AT RAM $60 *
32 P *
33 P * FETCH LDA $0080 *
34 P * RTS *
35 P *
36 P *****
37 P
38 A 0080 ORG RESET
39 A 0080 060238 START BRSET 3,REQUEST,USER Execute user program ?
40 A
41 A 0083 A6C6 READ LDA #$C6 No, transfer self-modifying code to RAM
42 A 0085 B760 STA FETCH
43 A 0087 3F61 CLR PAGE
44 A 0089 A680 LDA #$80
45 A 008B B762 STA BYTE
46 A 008D 4C INCA
47 A 008E B763 STA BYTE+1
48 A
49 A 0090 A6FF LDA #$FF Configure Data Bus and Valid Data ports
50 A 0092 B705 STA DATADR as outputs
51 A 0094 B707 STA VALIDDR
52 A 0096 AEF5 LDX #ROMEND
53 A
54 A 0098 1103 LOOP BCLR 0,VALID Send: Data invalid
55 A 009A 0702FD WAIT1 BRCLR 3,REQUEST,WAIT1 Wait for Data Request
56 A 009D ADC1 BSR FETCH Get next byte
57 A 009F B701 STA DATA Send it
58 A 00A1 1003 BSET 0,VALID Send: Data valid

```

FIGURE 3-3. Dump Program Listing (Sheet 1 of 2)

```

59 A
60 A 00A3 3C62      INC      BYTE      Increment address pointer
61 A 00A5 270D      BEQ      MSB        Roll over to MSB ?
62 A 00A7 B362      CPX      BYTE      No, end of program ROM LSB ?
63 A 00A9 260B      BNE      WAIT2     No, continue
64 A 00AB 076108    BRCLR   3,PAGE,WAIT2 End of program ROM ?
65 A 00AE AE1F      LDX     #VECTRS    Yes, change address pointer MSB
66 A 00B0 BF61      STX     PAGE
67 A 00B2 2002      BRA     WAIT2      and continue
68 A
69 A 00B4 3C61      MSB     INC      PAGE Increment address pointer MSB
70 A 00B6 0602FD    WAIT2  BRSET   3,REQUEST,WAIT2 Wait for: Got data
71 A 00B9 20DD      BRA     LOOP       Continue
72 A
73 A 00BB 20FE      USER   BRA     USER   User program starts here
74 A
75 A      1FFE      ORG     $1FFE
76 A 1FFE 0080      VECTOR FDB     RESET   Reset vector location
77 A
78 A      0080      END     START

```

**** TOTAL ERRORS 0-- 0
**** TOTAL WARNINGS 0-- 0

SYMBOL TABLE LISTING

SYMBOL NAME	SECT	VALUE	SYMBOL NAME	SECT	VALUE
BYTE	A	0062	ROMEND	A	00F5
DATA	A	0001	START	A	0080
DATADR	A	0005	USER	A	00BB
FETCH	A	0060	VALID	A	0003
LOOP	A	0098	VALIDDR	A	0007
MSB	A	00B4	VECTOR	A	1FFE
PAGE	A	0061	VECTRS	A	001F
READ	A	0083	WAIT1	A	009A
REQUEST	A	0002	WAIT2	A	00B6
RESET	A	0080			

FIGURE 3-3. Dump Program Listing (Sheet 2 of 2)

CHAPTER 4

FUNCTIONAL DESCRIPTION

4.1 INTRODUCTION

This chapter provides a block diagram description of the EVM module. A block diagram is shown in Figure 4-1.

4.2 BLOCK DIAGRAM DESCRIPTION

The EVM is designed to simulate an MC1468705 MCU with the resident MC146805G2. Data transfer within the EVM is controlled by the monitor ROM firmware controlled from an external RS-232C compatible user terminal. The contents of the monitor ROM are not available to the user.

After system initialization, 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 on the desire of the user. For the alternate case, the command is executed under control of the monitor, and the system returns to a waiting condition after command completion. During command execution, additional user input may be required, depending on the command function.

The monitor requires five bytes of user stack for user register storage upon monitor (re)entry. No limitation is placed on the user as to other uses of the stack.

All user registers are in an unknown state on monitor power-up.

4.2.1 Decode and Map Switch Logic

The EVM operates in either one of two maps (monitor map or user map) (see Figure 3-1). The monitor map contains 4K bytes of RAM/ROM, which includes a user stack pointer, a monitor control register, monitor I/O, monitor RAM/stack, and the monitor ROM. The user map contains the user I/O ports, user RAM/stack, and user pseudo ROM. A temporary map switch which allows modification of user memory, and a permanent switch which allows execution of user programs are provided on the EVM.

Temporary Map Switch

The temporary map switch executes any three-byte instruction in the user map and returns immediately to the monitor map. The opcode and operand are fetched from the monitor map, maps are switched, and the instruction is executed in user space. Maps are switched automatically back to the monitor map at the beginning of the next load instruction register (LIR) from the processor.

Permanent Map Switch

The permanent map switch is initiated by command from the monitor, or by the USER RESET switch. Return to the monitor occurs at a breakpoint, at the end of a trace function in the monitor, or the ABORT switch.

The return from interrupt (RTI) opcode is fetched from the monitor map, maps are switched, and the registers are fetched from the user stack. Execution proceeds from the program counter (PC) fetched from the user stack.

When the USER RESET switch is pressed, the MCU and all user I/O are reset, maps are immediately switched to the user map, and the reset vector is fetched from the user map by the MCU. Breakpoints are ignored during this mode of execution.

Execution of user code continues until an SWI is decoded on the data bus during an LIR, and either the breakpoint flip-flop is set or the ABORT switch is pressed. The map is switched back to the monitor map after the registers are saved on the user stack. An SWI which occurs when no breakpoint is set -- and when the ABORT switch is not being pressed -- does not cause a map switch. This allows user SWI's to be executed in real time.

The ABORT switch, when pressed, forces an SWI on the data bus until the next LIR. The registers are saved on the user stack and maps switched to the monitor. The ABORT switch has no effect on the monitor map.

When the USER RESET switch is pressed, the MCU and user I/O are reset into the user map. When the ABORT switch is pressed while in the user map, the monitor map is reenetered.

4.2.2 Monitor I/O

The monitor I/O consists of two serial RS-232C ports, a cassette port, and a parallel printer port.

Terminal Port

The terminal port communicates with an RS-232C compatible terminal in character mode. Baud rates are selectable through the use of a jumper on header J12. A software transparent mode allows direct communications between the terminal and the host.

Host Port

The host port provides downloading of S-records from an RS-232C compatible host computer (EXORciser or any modem). Baud rates are selectable through the use of a jumper on J12.

Cassette Port

The cassette port provides a means of dumping and loading S-records to and from an audio cassette recorder. The recording technique is unique to the EVM.

Printer Port

The printer port interfaces with any standard Centronics type parallel printer. Seven data lines, strobe, and acknowledge are the only signal lines supported. When the printer port is activated, all information sent to the terminal is echoed to the printer.

4.2.3 User Map Area

The user map area consists of the user RAM/stack and the user pseudo ROM.

User RAM/Stack

The user stack is resident in memory from 010-07F. The stack pointer value is displayed when trace and breakpoint functions are executed. For detailed information on the stacking order, see the specific device data sheet.

User Pseudo ROM

The user program space 080-FFF is RAM, which 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.

A jumper on header J18 provides selection of a 2K-byte user map for evaluating the MC1468705. Vectors fetched from 7F8-7FF in the user program appear to the resident MCU at 1FF8-1FFF. This is accomplished by shadowing the lower 2K map into the upper 6K bytes during this option.

4.2.4 User I/O Ports

All user I/O ports (A, B, C, D) are available in the user map for evaluation. All ports are reconstructed off chip except port C.

The on chip timer, A/D, and interrupts are available to the user.

4.2.5 EPROM Programmer and Reader

The MC1468705 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 takes data from the EVM I/O ports and transfers the data into the internal EPROM of the MCU device.

The contents of the programmed EPROM may be read by the EVM. A small program can be added to the user code, which allows the contents of the EPROM to be downloaded to EVM memory.

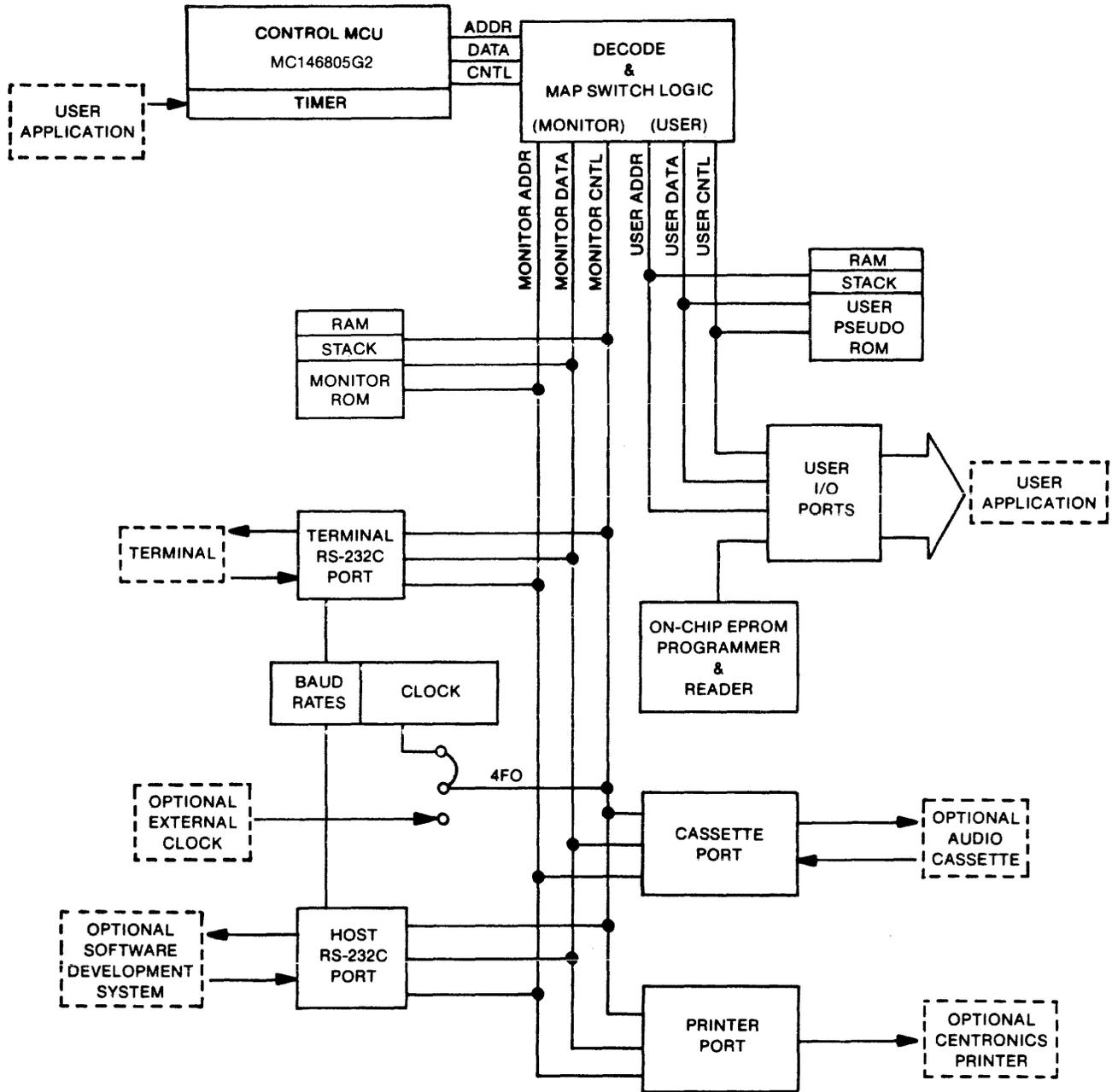


FIGURE 4-1. EVM Module Block Diagram

CHAPTER 5

SUPPORT INFORMATION

5.1 INTRODUCTION

This chapter provides the interconnect signals, parts list with parts location illustration, and schematic diagram of the EVM Module.

5.2 INTERCONNECT SIGNALS

The EVM interconnects through cables with a terminal, a host computer, a printer, and a cassette recorder. See paragraph 2.4.5 for cassette connectors J6 and J7.

5.2.1 Terminal Port J3 Interconnect Signals

The signals at edge connector J3 on the EVM module are RS-232C compatible. Table 5-1 lists each pin connection, signal mnemonic, and signal characteristic for the connector.

TABLE 5-1. Terminal Port J3 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1,2		Not used.
3	TxD	TRANSMIT DATA - This line accepts input data from an RS-232C terminal device.
4		Not used.
5	RxD	RECEIVE DATA - This line transfers data to an RS-232C terminal device.
6,7,8		Not used.
9	CTS	CLEAR TO SEND - This line is held high to be compatible with terminals requiring this signal.
10		Not used.
11	DSR	DATA SET READY - This line is held high to be compatible with terminals requiring this signal.
12		Not used.
13	GND	SIGNAL GROUND - This line provides a common signal ground connection to the RS-232C device.
14		Not used.
15	DCD	DATA CARRIER DETECT - This line is held high to be compatible with terminals requiring this signal.
16-20		Not used.

5.2.2 Host Port J4 Interconnect Signals

The signals at edge connector J4 on the EVM module are RS-232C compatible. Table 5-2 lists each pin connection, signal mnemonic, and signal characteristic for the connector.

TABLE 5-2. Host Port J4 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1,2		Not used.
3	TxD	TRANSMIT DATA - This line transfers data to an RS-232C modem device.
4		Not used.
5	RxD	RECEIVE DATA - This line accepts input data from an RS-232C modem device.
6		Not used.
7	RTS	REQUEST TO SEND - This output signal line requests permission from the RS-232C device to send data.
8		Not used.
9	CTS	CLEAR TO SEND - This input line grants permission to send data (see paragraph 2.3.4).
10,11,12		Not used.
13	GND	SIGNAL GROUND - This line provides a common signal ground connection to the RS-232C device.
14	DTR	DATA TERMINAL READY - This output line is held high to be compatible with devices requiring this signal.
15	DCD	DATA CARRIER DETECT - This input line indicates that the carrier signal has been detected (see paragraph 2.3.4).
16-20		Not used.

5.2.3 Printer Port J9 Interconnect Signals

The signals at connector J9 are Centronics type printer compatible. Table 5-3 lists each pin connection, signal mnemonic, and signal characteristic for the connectors.

TABLE 5-3. Printer Port J9 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1-24		Not used.
25,26	GND	GROUND
27	PR7	PRINTER DATA line (bit 7) - One of seven output data lines to the printer.
28	GND	ROUND - PR7 signal return line.
29	PR6	PRINTER DATA line (bit 6) - Same as PR7, pin 27.
30	GND	GROUND - PR6 signal return line.
31	PR5	PRINTER DATA line (bit 5) - Same as PR7, pin 27.
32	GND	GROUND - PR5 signal return line.
33	PR4	PRINTER DATA line (bit 4) - Same as PR7, pin 27.
34	GND	GROUND - PR4 signal return line.
35	PR3	PRINTER DATA line (bit 3) - Same as PR7, pin 27.
36	GND	GROUND - PR3 signal return line.
37	PR2	PRINTER DATA line (bit 2) - Same as PR7, pin 27.
38	GND	GROUND - PR2 signal return line.
39	PR1	PRINTER DATA line (bit 1) - Same as PR7, pin 27.
40	GND	GROUND - PR1 signal return line.
41,42		Not used.
43	STB*	DATA STROBE - A low level output signal used to strobe the data on PR1-PR7 into the printer.
44	GND	GROUND - Signal return line for STB*.
45,46		Not used.
47	ACK*	ACKNOWLEDGE - An activae low level input signal indicating acceptance of the data into the printer.
48	GND	GROUND - Signal return line for ACK*.
49-50		Not used.

5.3 PARTS LIST

Table 5-4 lists the components of the EVM module. The parts locations are illustrated in Figure 5-1. This parts list reflects the latest issue of hardware at the time of printing.

TABLE 5-4. EVM Module Parts List

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
	84-W8143B01	Printed wiring board
CR3,CR4	48NW9607A01	Rectifier, 1N4001
C1,C15,C21	23NW9618A33	Capacitor, electrolytic, 22 uF @ 25 Vdc
C2-C14,C16-C20, C22-C45,C47-C50	21SW992C025	Capacitor, fixed, ceramic, 01 uF @ 50 Vdc
C46	23NW9618A22	Capacitor, electrolytic, 50 uF @ 16 Vdc
C51	21SW992C010	Capacitor, fixed, ceramic, 470 pF @ 50 Vdc
DS1-DS4	48NW9612A24	Diode, light emitting, red
J1	28NW9802C29	Header, double row post, 4 pin
J2,J5,J8,J10, J13,J15-J18	28NW9802D04	Header, single row post, 3 pin
J6,J7	09NW9803A27	Jack, phone, subminiature, 2 contact
J11,J12	28NW9802B34	Header, double row post, 16 pin
J14	28NW9802D01	Header, double row post, 2 pin
P1	31NW9509A27	Terminal block, 7 position, screw contact
Q1,Q3-Q6	48NW9611A01	Transistor, 2N4403
Q2	48NW9610A01	Transistor, 2N4401
R1,R13,R14, R15,R17	06SW-124A37	Resistor, fixed, film, 330 ohm, 5%, 1/4 W
R2	51NW9626A92	Resistor network, 5/39k ohm
R3,R8	06SW-124A89	Resistor, fixed, film, 47k ohm, 5%, 1/4 W
R4	06SW-124A97	Resistor, fixed, film, 100k ohm, 5%, 1/4 W
R5	06SW-124A81	Resistor, fixed, film, 22k ohm, 5%, 1/4 W
R6,R25,R37,R39	06SW-124A65	Resistor, fixed, film, 4.7k ohm, 5%, 1/4 W

TABLE 5-4. EVM Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
R7,R16	51NW9626A46	Resistor network, 5/4.7k ohm
R9,R18,R19, R33,R38	06SW-124A41	Resistor, fixed, film, 470 ohm, 5%, 1/4 W
R11	51NW9626A37	Resistor network, 9/10k ohm
R12	51NW9626A93	Resistor network, 7/470 ohm
R20	06SW-124A32	Resistor, fixed, film, 200 ohm, 5%, 1/4 W
R21	06SW-124A69	Resistor, fixed, film, 6.8k ohm, 5%, 1/4 W
R23	06SW-124A77	Resistor, fixed, film, 15k ohm, 5%, 1/4 W
R24,R32	06SW-124A49	Resistor, fixed, film, 1.0k ohm, 5%, 1/4 W
R26	06SW-124A60	Resistor, fixed, film, 3.0k ohm, 5%, 1/4 W
R27	06SW-124A71	Resistor, fixed, film, 8.2k ohm, 5%, 1/4 W
R28	06SW-124A44	Resistor, fixed, film, 620 ohm, 5%, 1/4 W
R29	06SW-124A42	Resistor, fixed, film, 510 ohm, 5%, 1/4 W
R30	06SW-124A20	Resistor, fixed, film, 62 ohm, 5%, 1/4 W
R31	06SW-124A75	Resistor, fixed, film, 12k ohm, 5%, 1/4 W
S1,S2,S5	40NW9801A54	Switch, pushbutton, SPDT
S3,S4	40NW9801B33	Switch, slide, DPDT
U1,U14, U19,U46	51NW9615C22	I.C. SN74LS08N
U2,U4	51NW9615B94	I.C. MC6850P
U3	51NW9615B30	I.C. MC1489AL
U5,U52	51NW9615E88	I.C. SN74LS10N
U6,U44,U53	51NW9615E91	I.C. SN74LS00N
U7	51NW9615E72	I.C. MC14583CP
U8,U10,U32	51NW9615F02	I.C. 74LS244N
U9	51NW9615B27	I.C. MC6821P

TABLE 5-4. EVM Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
U11,U51	51NW9615E89	I.C. SN74LS260N
U12,U31,U48	51NW9615F76	I.C. SN74LS11N
U13	51NW9615H50	I.C. SC82027P
U15	51NW9615B29	I.C. MC1488L
U16,U36	51NW9615C20	I.C. SN74LS02N
U17,U40,U43	51NW9615C21	I.C. SN74LS04N
U18	51NW9615C56	I.C. 74S08N
U20	51NW9615E99	I.C. SN74LS374N
U21	51NW9615E96	I.C. SN74LS245N
U22	51NW9615G98	I.C. SC89146P
U23	(See note)	I.C. programmed
U24	51NW9615E80	I.C. MCM68B10P
U25,U26	(See note)	I.C. programmed
U27,U28	51NW9615H59	I.C. HM6116P-4
U29	51NW9615E77	I.C. SN74LS27N
U30,U49	51NW9615F38	I.C. SN74LS393N
U33	51NW9615M97	I.C. MC74HC374N
U34,U35,U41	51NW9615C24	I.C. SN74LS32N
U37,U42, U45,U47	51NW9615C25	I.C. SN74LS74N
U38,U39	51NW9615H51	I.C. SYP6522
U50	51NW9615B51	I.C. MC14024BCP
U54	51NW9615M96	I.C. MC74HCU04N
U55	51NW9615E93	I.C. SN74LS14N
VR1	48NW9608A14	Diode, zener, 1N5244B
VR2	48NW9608A06	Diode, zener, 1N963B

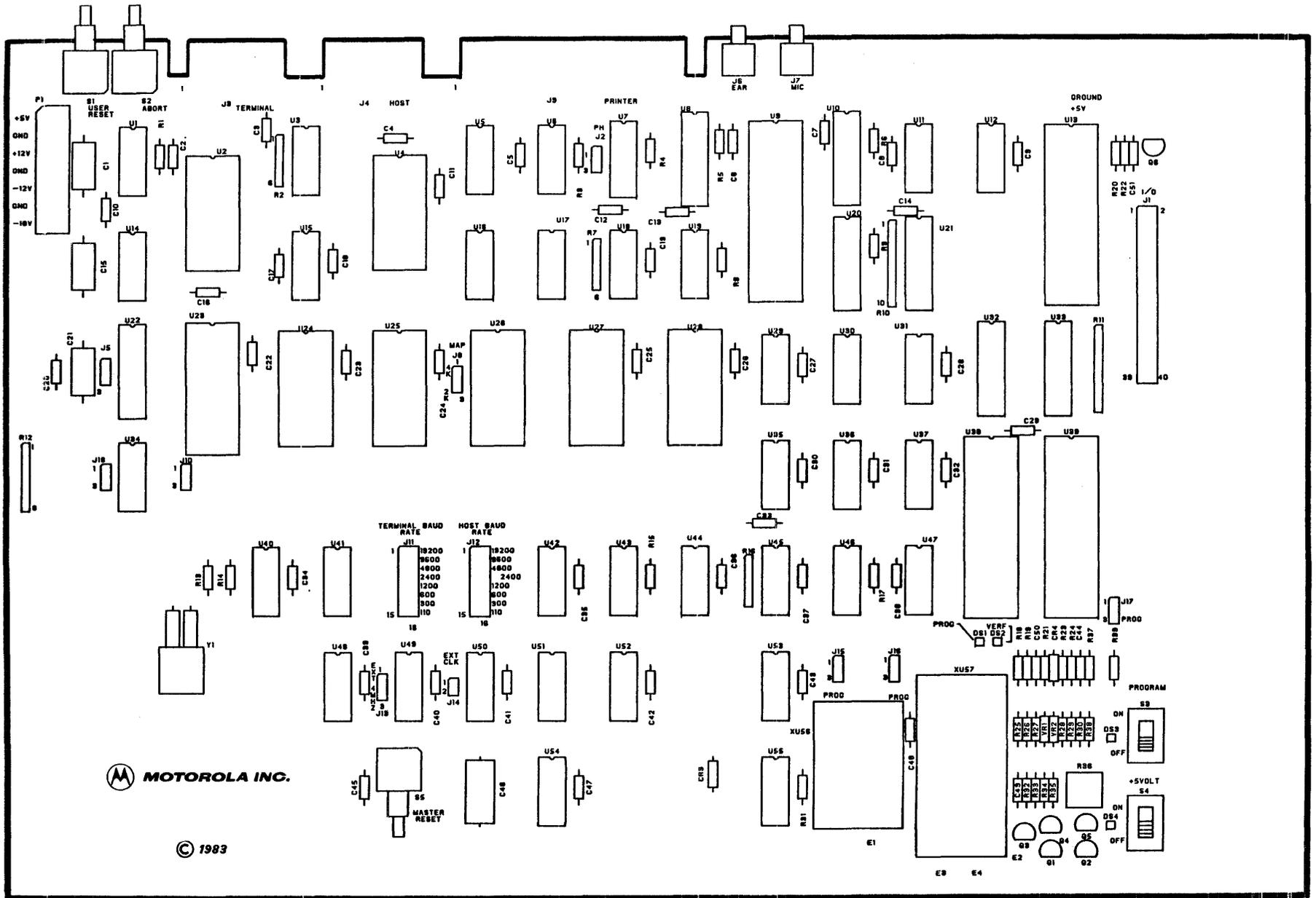
TABLE 5-4. EVM Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
Y1	48NW9606A32	Crystal, 4.0 MHz
	09NW9811A15	Socket, DIL, 24 pin (use at U24-U28)
	09NW9811A39	Socket, DIL, 40 pin (use at XU57)
	09NW9811A21	Socket, DIL, 28 pin (use at U23)
	09NW9811A38	Socket, DIL, 28 pin (used at XU56)
	09NW9811A22	Socket, DIL, 40 pin (use at U13, U38, U39)
	29NW9805B17	Jumper, shorting (use at J1, J2, J5, J8, J10, J11, J12, J13, J15-J18)

NOTE: When ordering, use number labeled on part.

5.4 SCHEMATIC DIAGRAM

Figure 5-2 illustrates the schematic diagram for the EVM module.



8-5

FIGURE 5-1. EVM Module Parts Location

- NOTES:
- FOR REFERENCE DRAWINGS REFER TO BILL(S) OF MATERIAL 01-W3143B01
 - UNLESS OTHERWISE SPECIFIED:
ALL RESISTORS ARE IN OHMS, ± 5PCT, 1/4 WATT.
ALL CAPACITORS ARE IN UF.
ALL VOLTAGES ARE DC.
 - INTERRUPTED LINES CODED WITH THE SAME LETTER OR LETTER COMBINATIONS ARE ELECTRICALLY CONNECTED.
 - △ DEVICE TYPE NUMBER IS FOR REFERENCE ONLY. THE NUMBER VARIES WITH THE MANUFACTURER
 - SPECIAL SYMBOL USAGE:
* DENOTES - ACTIVE LOW SIGNAL.
 - INTERPRET DIAGRAM IN ACCORDANCE WITH AMERICAN NATIONAL STANDARDS INSTITUTE SPECIFICATIONS, CURRENT REVISION.
 - △ PART TYPES ARE ABBREVIATED IN THE FIELD OF THE DRAWING, FOR FULL PART TYPE, REFER TO TABLE 1.
 - INDICATES EQUIPMENT MARKING.
 - CODE FOR SHEET TO SHEET REFERENCES IS AS FOLLOWS: SHEET 6 ZONE A6

TABLE 1 △

REF DES	TYPE △	GND	+5V	+12V	-12V
U1	74LS08	7	14		
U2	MC6850	1	12		
U3	MC1489A	7	14		
U4	MC6850	1	12		
U5	74LS10	7	14		
U6	74LS00	7	14		
U7	MC14583	8	16		
U8	74LS244	10	20		
U9	MC6821	1	20		
U10	74LS244	10	20		
U11	74LS260	7	14		
U12	74LS11	7	14		
U13	MC146805G2	20	40		
U14	74LS08	7	14		
U15	MC1488	7		14	1
U16	74LS02	7	14		
U17	74LS04	7	14		
U18	74S08	7	14		
U19	74LS08	7	14		
U20	74LS374	10	20		
U21	74LS245	10	20		
U22	74LS373	10	20		
U23	82S103	14	20		
U24	MCM68B10	1	24		
U25	MCM2716	12	24		
U26	MCM2716	12	24		
U27	HM6116	12	24		
U28	HM6116	12	24		
U29	74LS27	7	14		
U30	74LS393	7	14		
U31	74LS11	7	14		
U32	74LS244	10	20		
U33	74HC374	10	20		
U34	74LS32	7	14		
U35	74LS32	7	14		

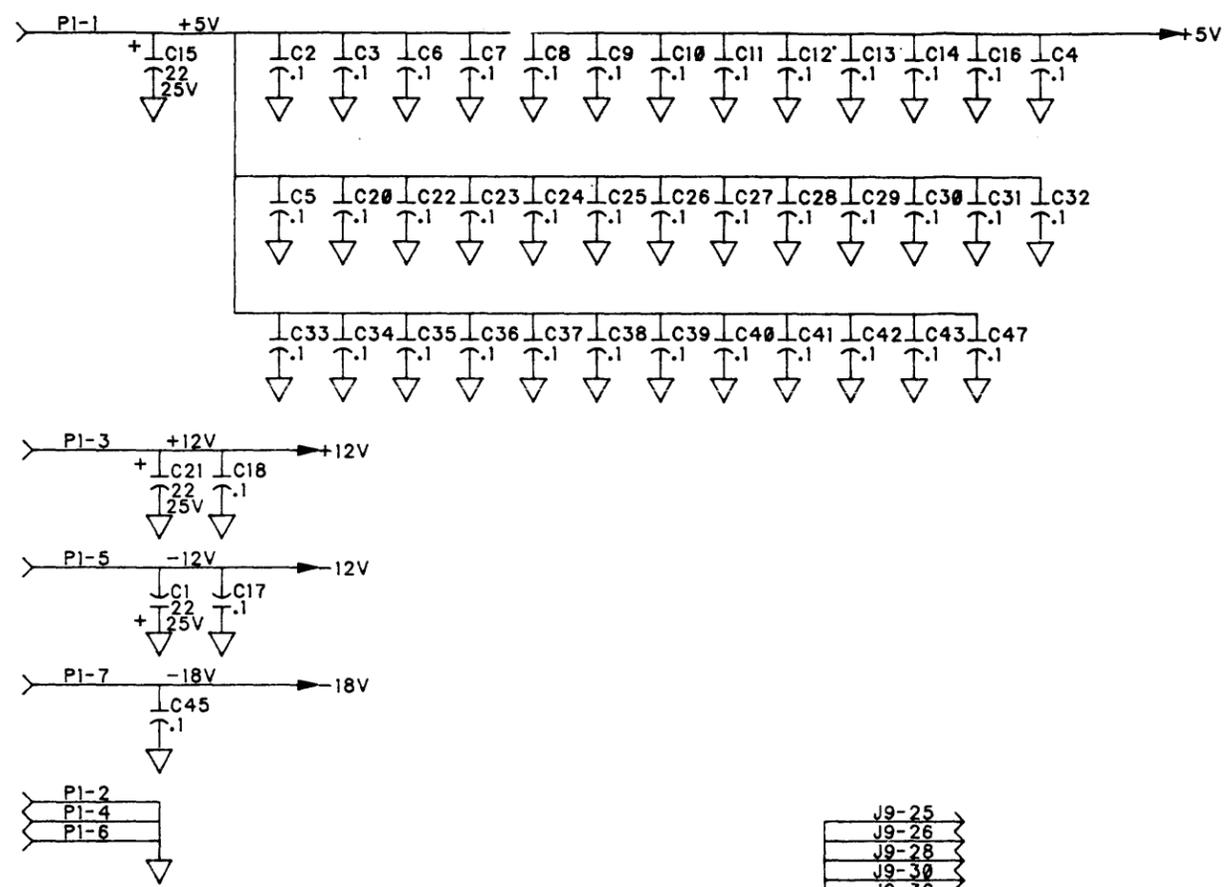
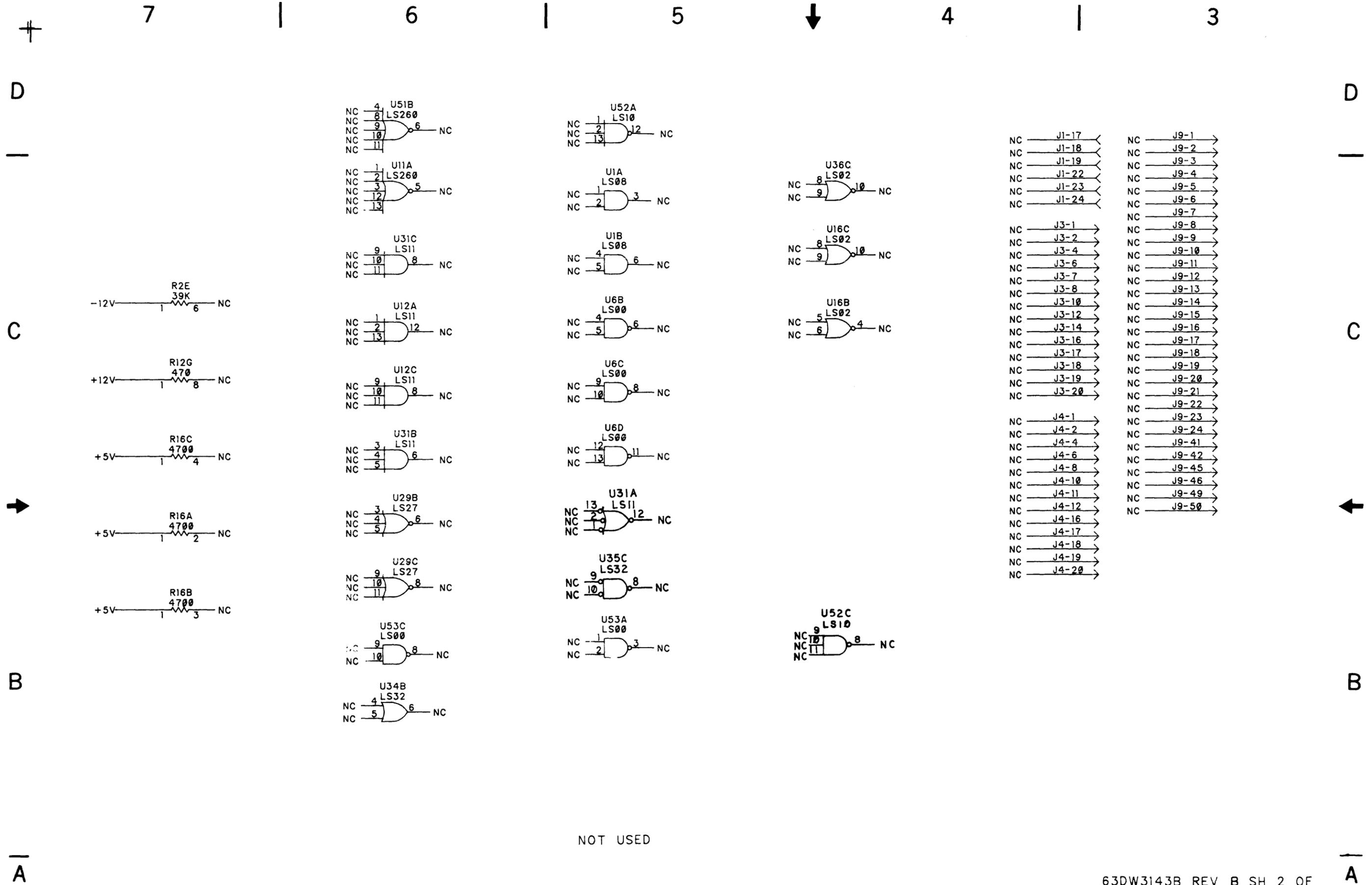


TABLE 1, CONT'D △

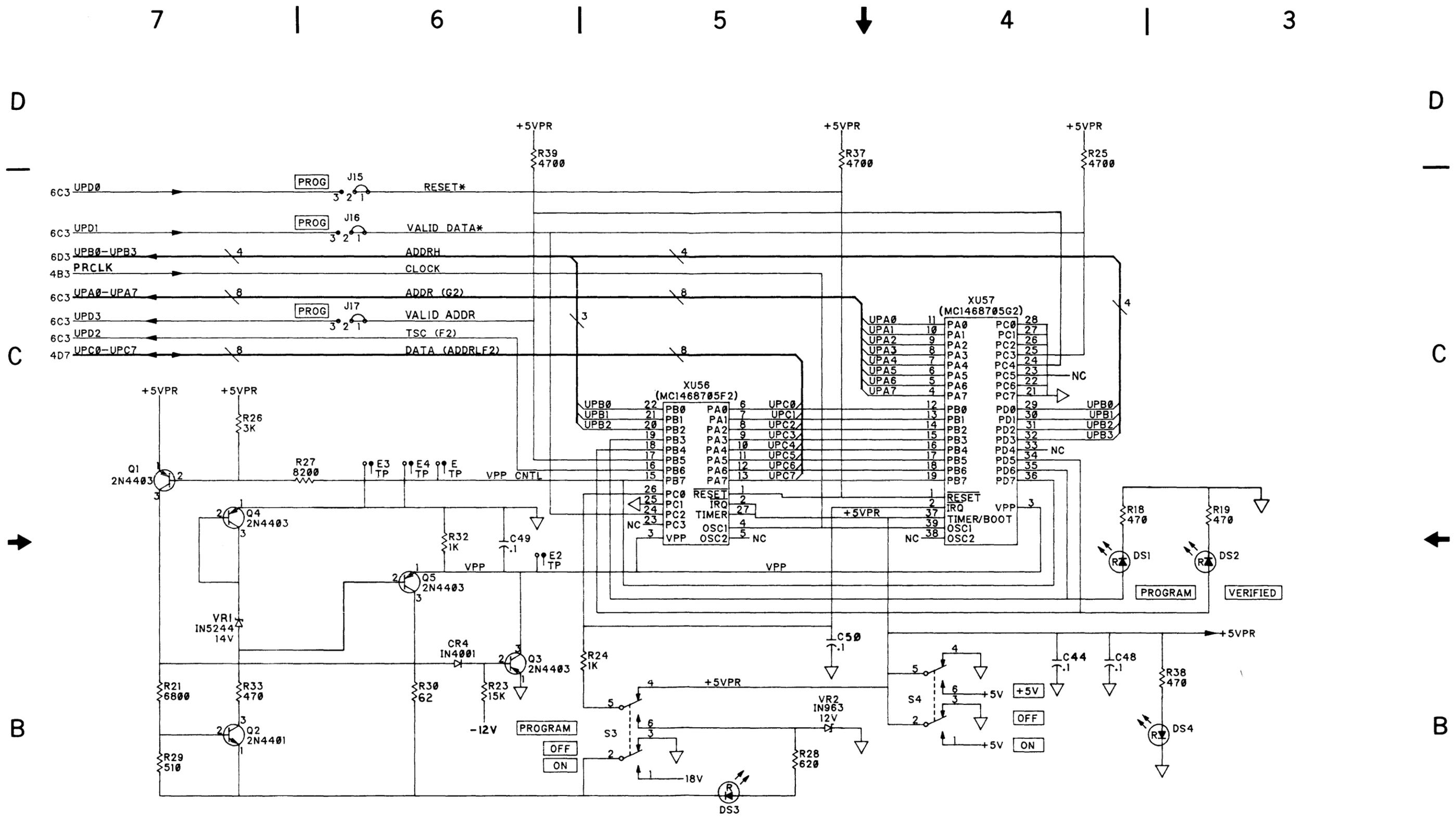
REF DES	TYPE △	GND	+5V	+5VPR
U36	74LS02	7	14	
U37	74LS74A	7	14	
U38	SYP6522	1	20	
U39	SYP6522	1	20	
U40	74LS04	7	14	
U41	74LS32	7	14	
U42	74LS74A	7	14	
U43	74LS04	7	14	
U44	74LS00	7	14	
U45	74LS74A	7	14	
U46	74LS08	7	14	
U47	74LS74A	7	14	
U48	74LS11	7	14	
U49	74LS393	7	14	
U50	MC14024	7	14	
U51	74LS260	7	14	
U52	74LS10	7	14	
U53	74LS00	7	14	
U54	74HC004	7	14	
U55	74LS14	7	14	
XU56	(SOCKET)	14		28
XU57	(SOCKET)	20		40

Y1	XUI THRU XU55
XU57	XUI THRU XU55
VR2	
U55	
S5-	
R39	R10, R15, R22, R34, R35, R36
Q5	
E4	
P1	
DS4	
J18	
CR4	CR1, CR2
C50	
HIGHEST NUMBER USED	NOT USED
REFERENCE DESIGNATIONS	



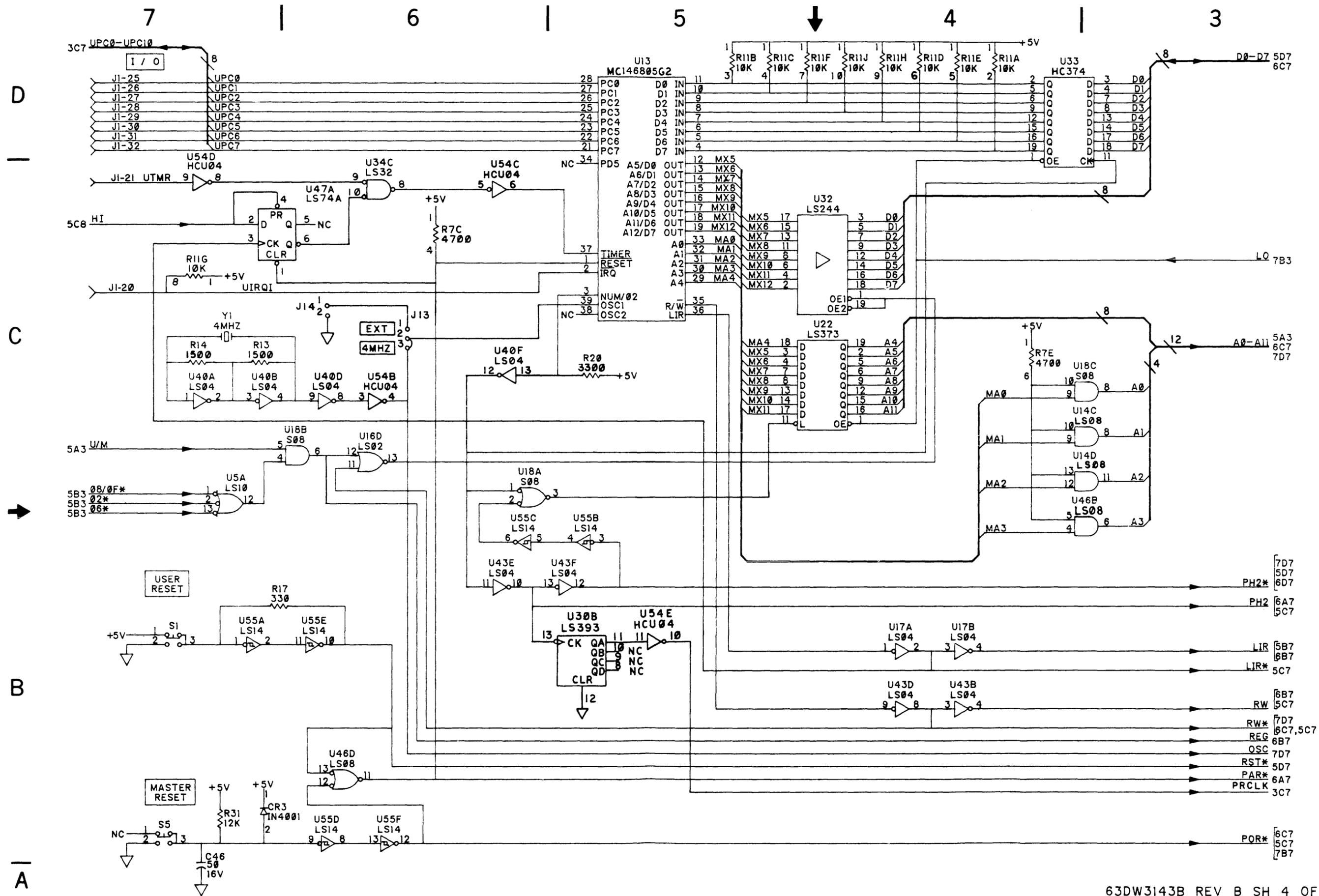
NOT USED

63DW3143B REV B SH 2 OF 7
 FIGURE 5-2. EVM Module Schematic Diagram (Sheet 2 of 7)

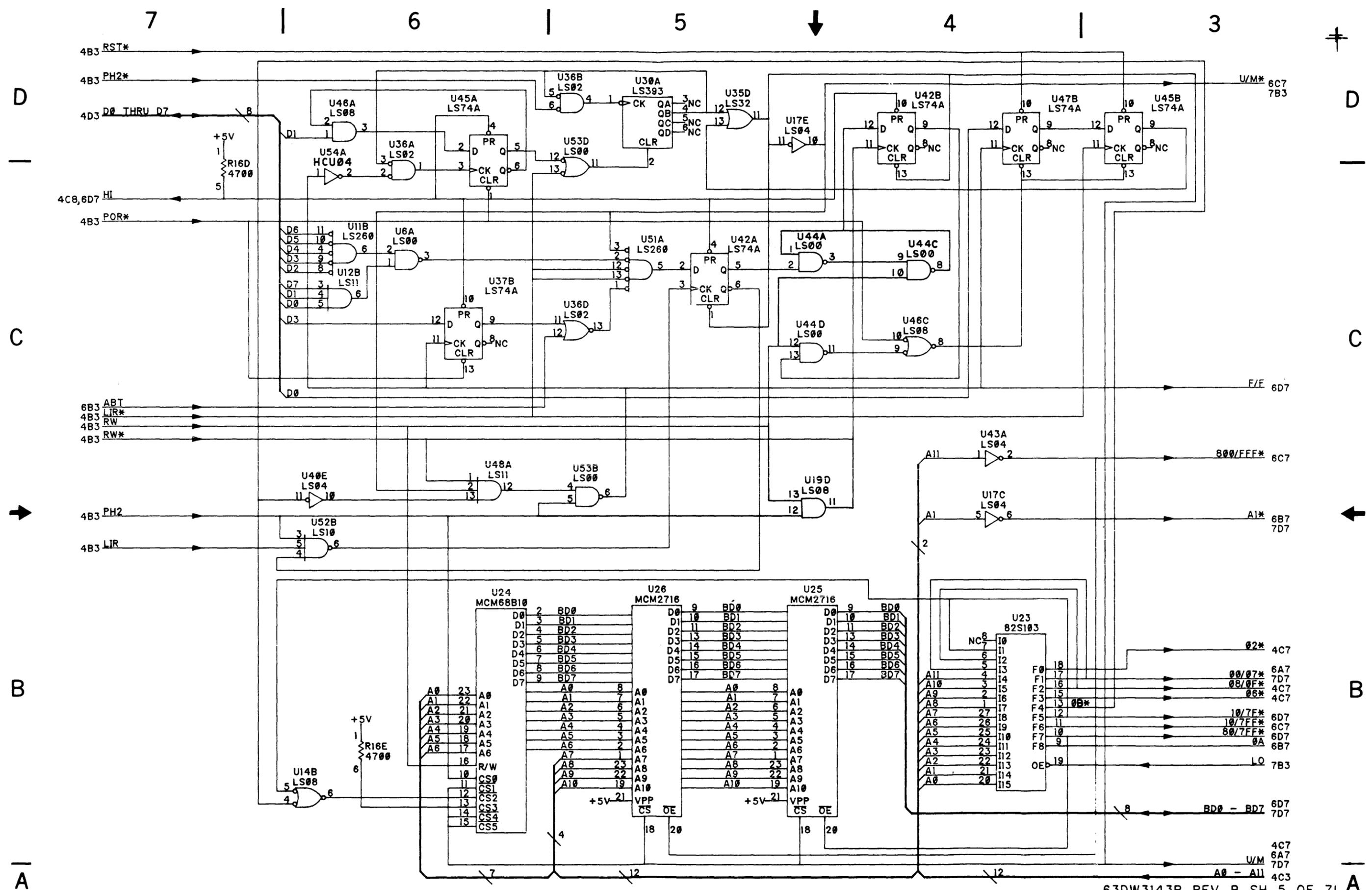


EPROM PROGRAMMER

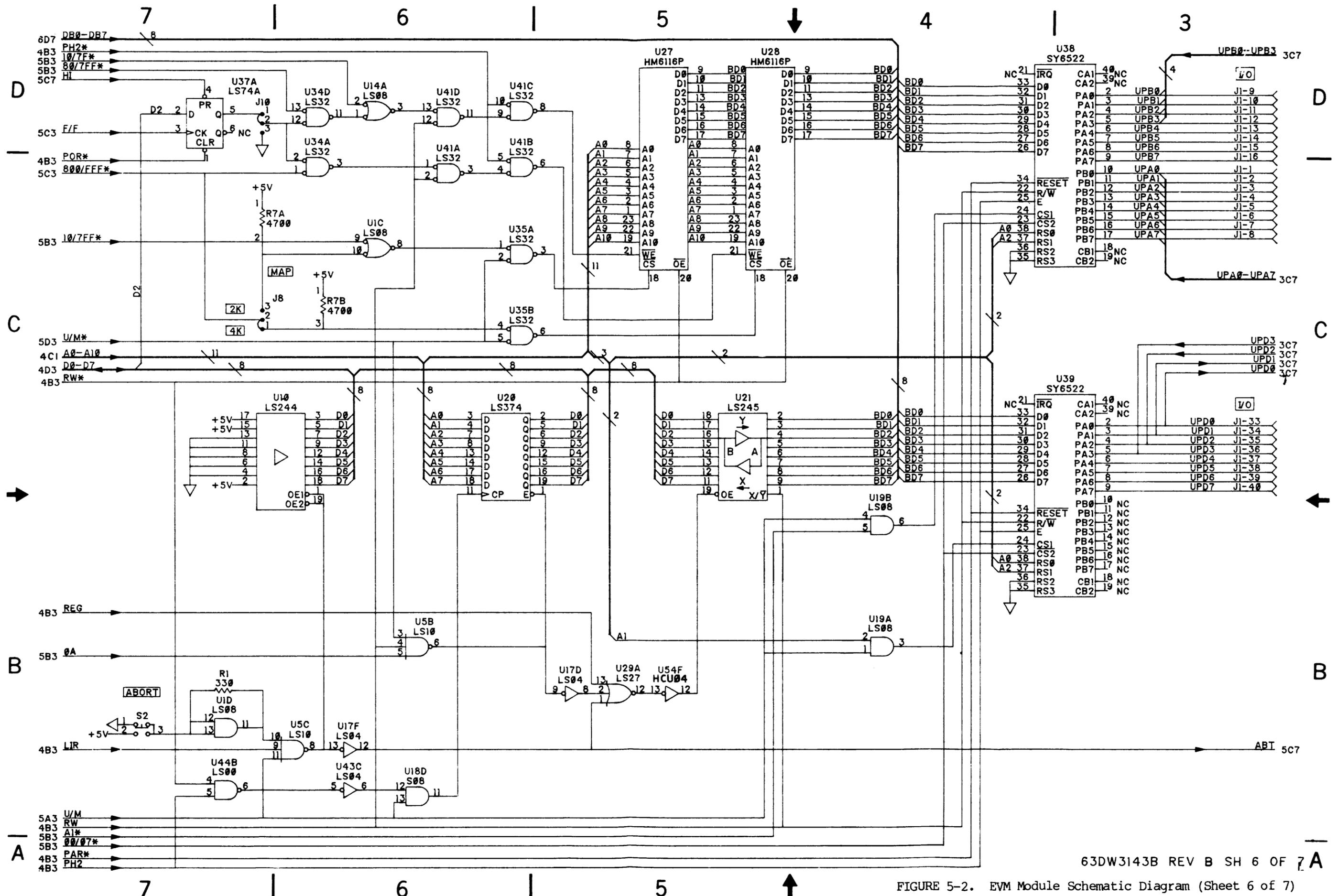
63DW3143B REV B SH 3 OF 7
 FIGURE 5-2. EVM Module Schematic Diagram (Sheet 3 of 7)

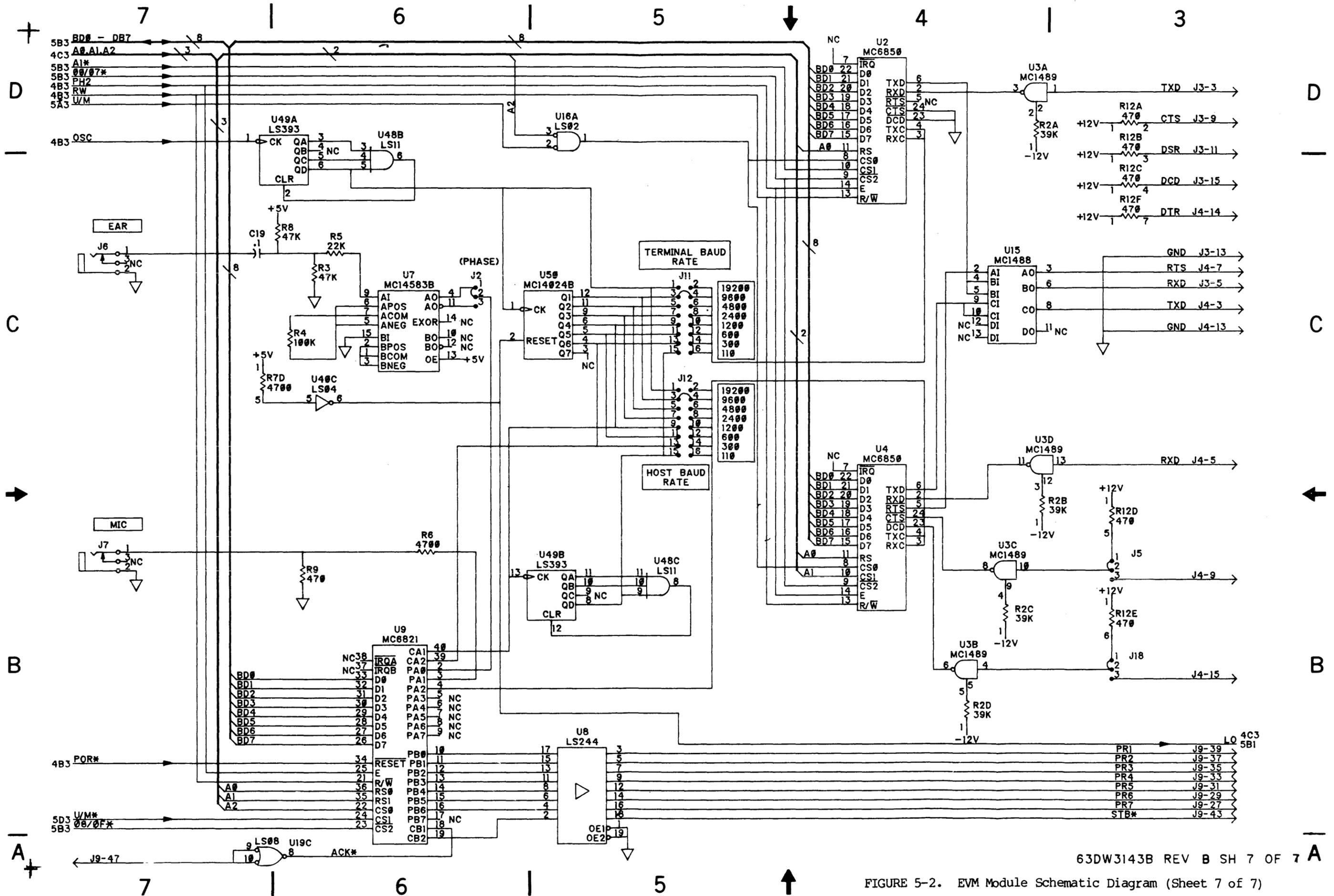


63DW3143B REV B SH 4 OF
 FIGURE 5-2. EVM Module Schematic Diagram (Sheet 4 of 7) 5-15/5-16



63DW3143B REV B SH 5 OF 7
 FIGURE 5-2. EVM Module Schematic Diagram (Sheet 5 of 7)





63DW3143B REV B SH 7 OF 7
 FIGURE 5-2. EVM Module Schematic Diagram (Sheet 7 of 7)

APPENDIX A

S-RECORD OUTPUT FORMAT

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:

type	record length	address	code/data	checksum
------	---------------	---------	-----------	----------

where the fields are composed as follows:

<u>FIELD</u>	<u>PRINTABLE CHARACTERS</u>	<u>CONTENTS</u>
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.

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's manual for that program must be consulted. 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 A record containing code/data and the 3-byte address at which the code/data is to reside.
- S3 A record containing code/data and the 4-byte address at which the code/data is to reside.
- S5 A record containing the number of S1, S2, and S3 records transmitted in a particular block. This count appears in the address field. There is no code/data field.
- S7 A termination record for a block of S3 records. The address field may optionally contain the 4-byte address of the instruction to which control is to be passed. There is no code/data field.
- S8 A termination record for a block of S2 records. The address field may optionally contain the 3-byte address of the instruction to which control is to be passed. There is no code/data field.
- 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.

Only one termination record is used for each block of S-records. S7 and S8 records are usually used only when control is to be passed to a 3- or 4-byte address. Normally, only one header record is used, although it is possible for multiple header records to occur.

CREATION OF S-RECORDS

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 microprocessor-based or a 16-bit microprocessor-based system.

EXAMPLE

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

```
S00600004844521B
S1130000285F245F2212226A000424290008237C2A
S11300100002000800082629001853812341001813
S113002041E900084E42234300182342000824A952
S107003000144ED492
S9030000FC
```

The module consists of one S0 record, four S1 records, and an S9 record.

The S0 record is comprised of the following character pairs:

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

The first S1 record is explained as follows:

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

The next 16 character pairs of the first S1 record 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:

<u>OPCODE</u>	<u>INSTRUCTION</u>
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

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

2A The checksum of the first S1 record.

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

The S9 record is explained as follows:

- S9 S-record type S9, indicating that it is a termination record.
- 03 Hexadecimal 03, indicating that three character pairs (3 bytes) follow.
- 00
- 00 The address field, zeroes.
- FC The checksum of the 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:

<u>type</u>		<u>length</u>				<u>address</u>								<u>code/data</u>								<u>checksum</u>						
S	1	1	3			0	0	0	0			2	8	5	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	6	...	3	2	4	1
0101	0011	0011	0001	0011	0001	0011	0011	0011	0000	0011	0000	0011	0000	0011	0010	0011	1000	0011	0101	0100	0110	...	0011	0010	0100	0001		

SUGGESTION/PROBLEM REPORT



Motorola welcomes your comments on its products and publications. Please use this form.

To: Motorola Inc.
Microsystems
2900 S. Diablo Way
Tempe, Arizona 85282
Attention: Publications Manager
Maildrop DW164

Product: _____ Manual: _____

COMMENTS: _____

Please Print

Name _____ Title _____
Company _____ Division _____
Street _____ Mail Drop _____ Phone _____
City _____ State _____ Zip _____

For Additional Motorola Publications
Literature Distribution Center
616 West 24th Street
Tempe, AZ 85282
(602) 994-6561

Microsystems Field Service Support
(800) 528-1908
(602) 829-3100





MOTOROLA Semiconductor Products Inc.

P.O. BOX 20912 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.