



29B UNIVERSAL PROGRAMMER

981-0121-001 REV A

OCTOBER 84

Copyright © Data I/O Corporation, 1984. All rights reserved.

Data I/O has made every attempt to ensure that the information in this document is accurate and complete. However, Data I/O assumes no liability for errors, or for any damages that result from use of this document or the equipment which it accompanies.

Data I/O reserves the right to make changes to this document without notice at any time.

*LogicPak™ and Logic Fingerprint™ are trademarks of Data I/O Corporation.

Applies to: Engineering part No. 990-0013-016
Text Part Number 090-0201-002

SAFETY SUMMARY


General safety information for operating personnel is contained in this summary. In addition, specific **WARNINGS** and **CAUTIONS** appear throughout this manual where they apply and are not included in this summary.

Definitions

WARNING statements identify conditions or practices that could result in personal injury or loss of life.

CAUTION statements identify conditions or practices that could result in damage to equipment or other property.

Symbols

 : This symbol appears on the equipment and it indicates that the user should consult the manual for further detail.

$V \sim$: This symbol stands for Vac. For example, $120V \sim = 120 \text{ Vac}$

Power Source

Check the voltage selector indicator (located on the rear panel) to verify that the product is configured for the appropriate line voltage.

Grounding the Product

The product is grounded through the grounding conductor of the power cord. To avoid electric shock, plug the power cord into a properly wired and grounded receptacle only. Grounding this equipment is essential for its safe operation.

Power Cord

Use only the power cord specified for your equipment.

Servicing

To reduce the risk of electric shock, do not perform any servicing other than that described in this manual.

TABLE OF CONTENTS

SECTION 1. INTRODUCTION

1.1 INTRODUCTION	1-1
1.2 SYSTEM OVERVIEW	1-2
1.3 INTRODUCTION TO OPERATION	1-4
1.3.1 TYPICAL 29B OPERATION	1-5
1.3.2 SAMPLE 29B PROGRAMMER OPERATION WITH UNIPAK™	1-5
1.3.3 REMOTE OPERATION	1-7
1.4 SPECIFICATIONS	1-8
1.4.1 FUNCTIONAL SPECIFICATIONS	1-8
1.4.2 POWER REQUIREMENTS	1-8
1.4.3 PHYSICAL AND ENVIRONMENTAL REQUIREMENTS	1-8
1.5 FIELD APPLICATIONS SUPPORT	1-8
1.6 WARRANTY	1-8
1.7 SERVICE	1-9
1.8 ORDERING	1-9

SECTION 2. INSTALLATION

2.1 INTRODUCTION	2-1
2.2 PAK INSTALLATION	2-1
2.3 PAK REMOVAL	2-2
2.4 POWER CONNECTION	2-2
2.4.1 VERIFYING/CHANGING THE OPERATING VOLTAGE	2-2
2.4.2 VERIFYING/REPLACING THE LINE FUSE	2-4
2.4.3 GROUNDING THE UNIT	2-4
2.4.4 APPLYING POWER	2-5
2.5 SERIAL I/O INTERFACE	2-6
2.5.1 CABLING	2-6
2.5.2 SETTING PARAMETERS	2-8
2.5.3 HOOKING UP A SERIAL PAPER TAPE READER	2-9

SECTION 3. OPERATION

3.1 INTRODUCTION	3-1
3.2 POWER-UP	3-2
3.2.1 GENERAL OPERATING NOTES	3-2
3.2.2 POWER-UP PROCEDURE	3-2
3.3 POWER-DOWN	3-3
3.4 MANUAL KEYBOARD OPERATIONS	3-3
3.4.1 FAMILY AND PINOUT CODES	3-3
3.4.2 DEVICE INSERTION	3-4
3.4.3 DEVICE REMOVAL	3-4
3.5 SOURCE/DESTINATION METHOD OF SYNTAX (COPY AND VERIFY)	3-5
3.5.1 SOURCE/DESTINATION DETERMINATION	3-5
3.5.2 GENERALIZED KEY SEQUENCE SYNTAX	3-6
3.6 FRONT PANEL KEYBOARD DATA TRANSFER OPERATIONS	3-7
3.6.1 LOAD RAM WITH MASTER DEVICE DATA	3-7
3.6.2 LOAD RAM FROM SERIAL PORT	3-8
3.6.3 PROGRAM DEVICE WITH RAM DATA	3-9
3.6.4 OUTPUT TO SERIAL PORT	3-11
3.6.5 BLOCK MOVE	3-12
3.6.6 VERIFY RAM DATA AGAINST MASTER DEVICE DATA	3-13
3.6.7 VERIFY RAM FROM SERIAL PORT	3-15

3.7	KEYBOARD EDIT OPERATIONS	3-16
3.7.1	HEXADECIMAL BASE	3-16
3.7.2	OCTAL BASE	3-17
3.7.3	BINARY BASE	3-18
3.8	SELECT KEY FUNCTIONS	3-19
3.8.1	ACCESSING SELECT FUNCTIONS	3-19
3.8.2	EXTENDED SELECT FUNCTIONS	3-19
3.8.3	DESCRIPTIONS AND KEY SEQUENCES	3-19
3.8.4	RAM DATA MANIPULATION COMMANDS	3-20
3.8.5	UTILITY AND INQUIRY COMMANDS	3-23
3.8.6	SERIAL I/O OPERATIONS	3-32
3.9	REMOTE OPERATION	3-36
3.9.1	COMPUTER REMOTE CONTROL (CRC)	3-36
3.9.2	SYSTEM REMOTE CONTROL (SRC)	3-45

SECTION 4. MAINTENANCE

4.1	OVERVIEW	4-1
4.2	MAINTENANCE	4-1
4.2.1	COVER REMOVAL	4-1
4.2.2	REMOVING THE CONTROLLER BOARD PROTECTIVE SHIELD	4-1
4.2.3	INSTALLING THE PROTECTIVE SHIELD	4-1
4.2.4	CLEANING	4-1
4.2.5	INSPECTION	4-1
4.3	CALIBRATION	4-2
4.3.1	POWER SUPPLY CALIBRATION	4-2
4.3.2	PERFORMANCE CHECK/DC CALIBRATION SET-UP	4-4
4.3.3	CALIBRATION OF PROGRAMMING PAKS AND MODULES	4-6
4.4	TROUBLESHOOTING	4-7
4.4.1	PRELIMINARY TROUBLESHOOTING	4-7
4.4.2	TROUBLESHOOTING SPECIFIC AREAS	4-8

SECTION 5. CIRCUIT DESCRIPTION

5.1	INTRODUCTION	5-1
5.2	GENERAL CIRCUIT DESCRIPTION	5-1
5.3	DETAILED CIRCUIT DESCRIPTION	5-1
5.3.1	POWER SUPPLIES	5-3
5.3.2	KEYBOARD/DISPLAY	5-5
5.3.3	CONTROLLER	5-5

APPENDIX A. DATA TRANSLATION FORMATS

A.1	INTRODUCTION	A-1
A.2	DATA VERIFICATION	A-1
A.3	CODES	A-3
A.4	LEADER AND NULL OUTPUT	A-3
A.5	TRANSLATION FORMATS	A-3
A.5.1	BINARY TRANSFER, CODE 10	A-3
A.5.2	DEC BINARY FORMAT, CODE 11	A-3
A.5.3	ASCII BINARY FORMAT, CODES 01, 02 AND 03 (OR 095, 06, AND 07)	A-4
A.5.4	THE 5-LEVEL BNPF FORMAT, CODES 08 OR 09	A-4
A.5.5	SPECTRUM FORMAT, CODES 12 OR 13	A-4
A.5.6	ASCII OCTAL AND HEX FORMATS, CODES 30-37 AND 50-58	A-5
A.6	MICROPROCESSOR FORMATS	A-7

APPENDIX B. TERMINAL REMOTE CONTROL (TRC)

B.1 INTRODUCTION.....	B-1
B.2 SYSTEM VARIATIONS	B-1
B.3 COMMAND GROUPS	B-2
B.4 SYNTAX	B-7
B.5 PROGRAMMER RESPONSES TO TERMINAL DATA ENTRY	B-7
B.6 SERIAL I/O INTERFACE	B-7
B.7 ACCESS TO REMOTE CONTROL	B-7
B.7.1 ENTRY	B-7
B.7.2 EXIT	B-7
B.8 COMMAND SUMMARY.....	B-8
B.8.1 CONTROL-KEY COMMANDS.....	B-8
B.8.2 UTILITY COMMANDS	B-8
B.8.3 DEVICE COMMANDS	B-10
B.8.4 I/O COMMANDS	B-11
B.8.5 EDITING COMMANDS.....	B-15
B.8.6 MEMORY MODE COMMANDS	B-16

APPENDIX C. OPTIONAL HANDLER INTERFACE

C.1 INTRODUCTION.....	C-1
C.2 COMPATIBILITY	C-1
C.3 HANDLER PORT INFORMATION.....	C-1
C.4 SYSTEM COMPONENTS	C-4
C.5 PERFORMANCE BOARD SELECTION.....	C-4
C.6 MCT HANDLER SET-UP	C-5
C.7 SYSTEM SET-UP.....	C-5
C.7.1 CONNECTING THE HANDLER UniPak™ TO THE PERFORMANCE BOARD.....	C-5
C.7.2 INSTALLING THE PERFORMANCE BOARD.....	C-6
C.7.3 CONNECTING THE 29B TO THE HANDLER.....	C-6
C.7.4 CONNECTING A COMPUTER OR TERMINAL TO THE 29B	C-7
C.8 POWER-UP	C-7
C.9 BASIC DATA TRANSFER OPERATIONS.....	C-9
C.9.1 ENTERING CRC MODE (OPTIONAL)	C-9
C.9.2 ENTERING FAMILY AND PINOUT CODES	C-9
C.9.3 DEVICE INSERTION	C-9
C.9.4 BINNING CONTROL	C-9
C.9.5 PREVENTING NONBLANK DEVICES FROM BEING PROGRAMMED	C-10
C.9.6 LOAD RAM WITH MASTER DEVICE DATA.....	C-10
C.9.7 PROGRAM A DEVICE WITH 29B RAM DATA.....	C-10
C.9.8 VERIFY 29B'S RAM DATA WITH MASTER DEVICE DATA	C-11
C.19 OTHER COMMANDS	C-11

APPENDIX D. REFERENCE MATERIAL

APPENDIX E. ERROR CODES

APPENDIX F. SCHEMATICS

LIST OF FIGURES

1-1	29B System Components	1-3
1-2	29B Front Panel Functions	1-4
1-3	Sample 29B Programmer Operation	1-6
1-4	Sample Sum-Check Calculation	1-7
2-1	Installing/Removing a Programming Pak	2-2
2-2	Back Panel	2-2
2-3	Voltage Wheel Selector	2-3
2-4	Accessing the Line Fuse	2-4
2-5	Programmer Power Switch Location	2-5
2-6	Sample Interconnection Methods	2-6
2-7	Baud Rate Switch and RS232 Serial Port	2-8
2-8	Status Switch Setting	2-8
2-9	Interconnection Method of Serial Paper Tape Reader	2-9
3-1	Power Switch and Power Cord Locations	3-2
3-2	Device Installation (Multiuse Sockets)	3-4
3-3	Block Move	3-12
3-4	Computer Remote Control Components	3-37
3-5	Inputting Remote Control Parameters	3-46
4-1	Calibration	4-2
4-2	Power Supply Test and Adjustment Points	4-3
4-3	Front Panel and Protective Shield Removal	4-4
4-4	29B with Programming Module Prepared for Calibration	4-5
4-5	Troubleshooting	4-7
4-6	Checking Baud Rate on Pin 2	4-9
5-1	Functional Diagram of 29B Universal Programmer	5-1
5-2	Interconnect Diagram for 29B Universal Programmer	5-2
5-3	Circuit Board Locations	5-3
5-4	Power Supply Circuitry	5-4
5-5	Block Diagram FIP Display, Driver, and Keyboard	5-6
5-6	Controller Board Block Diagram	5-7
5-7	Write Timing	5-8
5-8	Read Timing	5-8
5-9	Expansion RAM Board Block Diagram	5-9
5-10	Optional Handler Interface Block Diagram	5-10
A-1	Formatting the Instrument Control Code and Data Translation Format Code	A-2
A-2	Input or Output Binary Tape	A-3
A-3	ASCII Binary Formats	A-4
A-4	Spectrum Format	A-4
A-5	ASCII-Octal and Hex Formats	A-5
A-6	Optional Address Field in ASCII-Octal and Hex Formats	A-6
A-7	Syntax of the Sum-Check Field in I/O Operation	A-6
A-8	Specifications for RCA Cosmac Data Files	A-7
A-9	Specifications for Fairchild Fairbug Data Files	A-8
A-10	Specifications for MOS Technology Data Files	A-9
A-11	Specifications for Motorola Data Files	A-10
A-12	Specifications for Intel Intellec 8/MDS Data Files	A-11
A-13	Specifications for Signetics Absolute Object Data Files	A-12
A-14	Specifications for Tektronix Hexadecimal Data Files	A-13
A-15	Specifications for Motorola Exormax Data Files	A-14
A-16	Specifications for Intel MCS-86 Hexadecimal Object (Code 88) Data Files	A-15
A-17	Calculating an Address using the Intel MCS-86 (Code 88) Extended Address Board	A-17
A-18	Specifications for Hewlett Packard Absolute Format Data Files	A-18
A-19	Specifications for Texas Instruments SDSMAC Data Files	A-19
B-1	System Configuration Between Programmer and Terminal Remote Control	B-1
C-1	29B Handler Port Pin Assignment	C-1
C-2	Handler Control Signal Timing Diagram	C-3
C-3	The 29B/Handler System	C-4
C-4	Connecting the Handler UniPak™ and Performance Board	C-5

C-5	Installing the Performance Board-MCT Handler	C-6
C-6	Installing the Performance Board-Delta Handler	C-7
C-7	Handler/Programmer Connection-MCT Handler	C-8
C-8	Handler/Programmer Connection-Delta Handler	C-8

LIST OF TABLES

1-1	Using the 29B Manual	1-1
1-2	Available Translation Formats for the 29B Universal Programmer	1-8
2-1	Line Fuse Ratings	2-4
2-2	Serial Interface Connector Pin Assignments	2-7
3-1	Copy and Verify Operations	3-5
3-2	Address Parameters	3-6
3-3	Select Commands	3-19
3-4	Response Characters	3-37
3-5	Error Status Word	3-38
3-6	Command Summary	3-40
3-7	Control Commands	3-41
3-8	Utility Commands	3-41
3-9	Device Commands	3-42
3-10	I/O Commands	3-43
3-11	Editing Commands	3-44
3-12	Handler Commands	3-44
3-13	Command Entry in SRC	3-45
3-14	Select Commands	3-57
4-1	Calibration Voltage Ranges	4-2
4-2	Displays and Instructions for Calibrating Standard Programming Modules	4-6
4-3	Power Supply Voltages	4-8
5-1	Address Map	5-5
A-1	Instrument Control Codes	A-2
A-2	Data Translation Formats	A-2
B-1	Control-Key Commands	B-2
B-3	Device Commands	B-3
B-4	I/O Commands	B-4
B-5	Editing Commands	B-5
B-6	Memory Mode Commands	B-6
C-1	Handler Interface Connector Pin Assignment	C-2
C-2	Handler-Specific Commands (CRC Mode)	C-9

SECTION 1 INTRODUCTION

1.1 INTRODUCTION

Data I/O's 29B Universal Programmer utilizes modular programming paks to reliably program most commercially available programmable memory and logic devices. The family code and pinout code table at the back of the Pak manual lists the devices (by manufacturer).

Standard features of the 29B include a 64K x 8 data RAM, an RS232C serial port and a 16-character alphanumeric display. The 29B also offers 27 data translation formats which enable you to communicate between the 29B and development systems, such as Texas Instruments, Motorola, Intel, Tektronix, Signetics, Fairchild, RCA and Hewlett-Packard. In addition, the standard system includes two remote control protocols. The Data I/O Computer Remote Control (CRC) communication protocol provides CRC operation that is compatible with other Data I/O products. System Remote Control (SRC) utilizes a simple command protocol that is similar to the syntax featured in the 29B front panel control to initiate rapid start-up in remote operation. Both of these remote-control features are fully documented in section 3 (Operation).

The 29B also offers the following optional features. If you wish to purchase any of these features, contact your nearest Data I/O sales representative; a list of sales representatives is provided at the back of this manual.

- Terminal Remote Control (TRC)—an optional package, compatible with Data I/O's System 1903.
- Handler Interface—an optional feature that allows you to connect the 29B to a Delta or MCT device handler. The combination of a 29B, Handler UniPak™, and device handler enables you to program large quantities of both MOS (metal oxide semiconductor) and bipolar PROMs (programmable read only memories) with minimum operator handling.

NOTE

Although other programming Paks can be used, neither the Pak/handler interface nor the 29B Handler port cable is provided. The Handler port is required.

When using the Exatron 2500 handler, the optional handler interface is not necessary and other programming paks (other than the Handler UniPak™) can be used. Exatron provides the interface necessary for operation with the 29B.

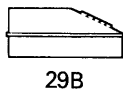
- Serial Paper Tape Reader (950-1950)—can be used to transmit data from a paper tape to your 29B.

This manual describes the components and operation of the 29B. Subjects addressed in this manual and the corresponding sections are listed in table 1-1. Use this list as a quick-reference point of the major sections in this manual.

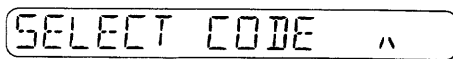
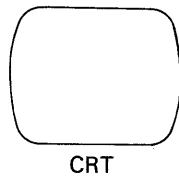
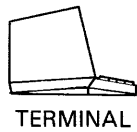
Table 1-1. Using the 29B Manual

SUBJECT	SECTION
Safety Summary	Front of manual
Pak Installation	2.2
Power Connection	2.4
Serial I/O Interface	2.5
Power-Up	3.2
Power-Down	3.3
Manual Keyboard Operations	3.4
Selecting the Family and Pinout Codes	See Pak Manual.
Device Insertion	3.4.2
Source/Destination Method of Syntax	3.5
Front Panel Keyboard Data	
Transfer Operations	3.6
Edit Data	3.7
Perform a Select Function	3.8
Remote Operations	3.9
Maintenance of 29B	4.2
Maintenance of Pak	See Pak manual.
Calibration of 29B Power Supplies	4.3.1
Calibration of 29B and Pak	See Pak Manual.
Troubleshooting 29B	4.4
Troubleshooting Pak	See Pak Manual.
Circuit Description	Section 5
Data Translation Formats	Appendix A
Terminal Remote Control	Appendix B
Optional Handler Interface	Appendix C
Reference Material	Appendix D
Error Codes	Appendix E
Schematics	Appendix F
Data I/O Service Centers	Back of Manual
Warranty Information	Back of Manual

Throughout this manual, the entries that you are to make from either the programmer or a terminal are indicated by an equipment symbol and key symbol. For example,



indicates the "COPY" key on the programmer keyboard should be pressed. In addition, the symbols shown below are used to indicate terminal operation, prompts that will appear on the CRT, or prompts that will appear on the programmer display.

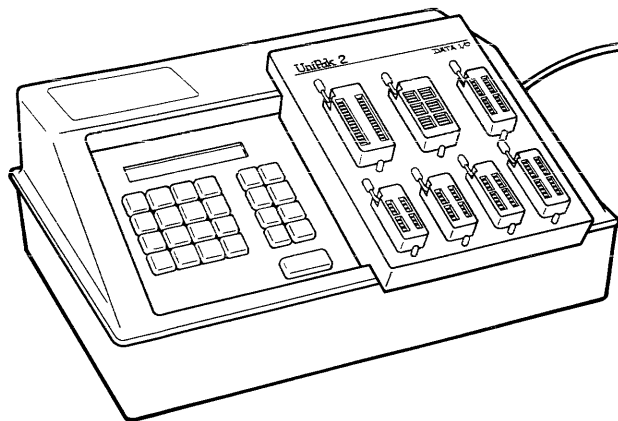


29B DISPLAY

1.2 SYSTEM OVERVIEW

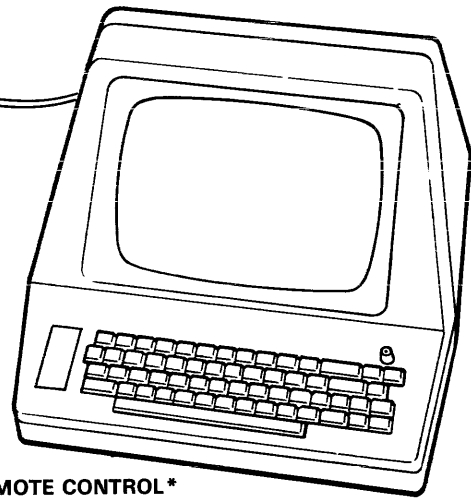
The 29B provides a universal means of programming, testing, and verifying a variety of memory and logic devices. You can tailor the 29B to your programming needs by selecting the appropriate programming Pak and plugging it into the 29B. The various programming components from which you can choose to complete your programming system are listed below and shown in figure 1-1.

- UniPak 2™—programs more than 600 devices, including MOS and CMOS EPROMs and EEPROMs, fuse link, AIM and DEAP bipolar PROMs. Programming algorithms are software selectable and no additional personality modules are required. Simple pinout adapters are available for 40-pin microcomputers and parts with non-standard pinouts.
- LogicPak™—combined with appropriate plug-in adapters, allows you to design, program and functionally test more than 80 different logic devices. The high-level software translates your logic design from truth tables or Boolean equations into the correct fuse map.
- GangPak™—gives you the capability of programming, in a single operation, identical sets of MOS EPROMs or EEPROMs. Set programming allows you to partition a program into one or more sets of PROMs. The GangPak™ can also be used for conventional gang programming of up to eight devices at a time.
- MOSPak™—programs more than 145 MOS EPROMs and EEPROMs, requiring no additional hardware.



29B PROGRAMMER

Base unit with keyboard; accepts Paks; contains control electronics. 29B also contains standard System Remote Control (SRC) and Computer Remote Control (CRC) software. An optional handler interface can be used for fast, high-volume programming of MOS and bi-polar PROMs.

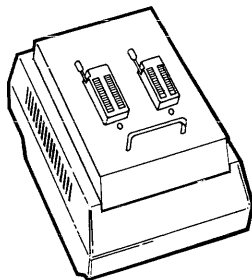


REMOTE CONTROL *

The standard Computer Remote Control (CRC) allows you to save programs on disk or tape which may be downloaded to the 29B. System Remote Control (SRC) allows you to send commands to the 29B from a terminal, and an optional Terminal Remote Control (TRC) is a terminal remote mode which is compatible with that used in System 19 programmers.

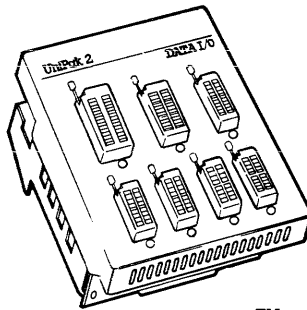
*Terminal not included.

PROGRAMMING PAKS



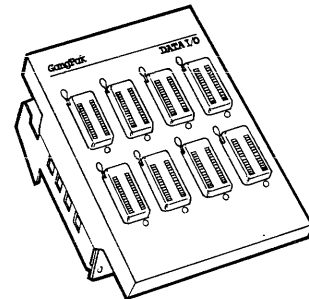
LogicPak™

Combined with appropriate plug-in adapters, the LogicPak™ allows you to design, program and functionally test more than 80 logic devices.



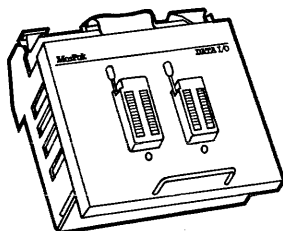
UniPak 2™

Programs more than 600 popular bipolar PROMs, MOS EPROMs, and MOS EEPROMs.



GangPak™

Programs multiple MOS PROM sets and gang programs MOS PROMs for high-volume programming operations.



MOSPak™

Programs more than 145 MOS EPROMs and EEPROMs with no additional hardware changes.

PROGRAMMING MODULES

Support device family applications not covered by programming Paks.

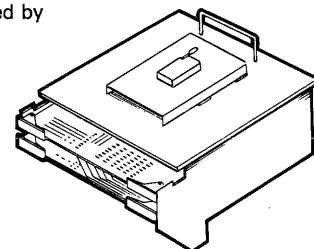


Figure 1-1. 29B System Components

1.3 INTRODUCTION TO OPERATION

With the 29B, you can perform four basic operations (listed below) by using the four dedicated mode (operation) keys on the front panel keyboard. These mode keys and their

uses are discussed in the following subsections and shown in figure 1-2. Section 3 (Operation) provides specific step-by-step instructions on how to execute the operations listed in the following subsections.

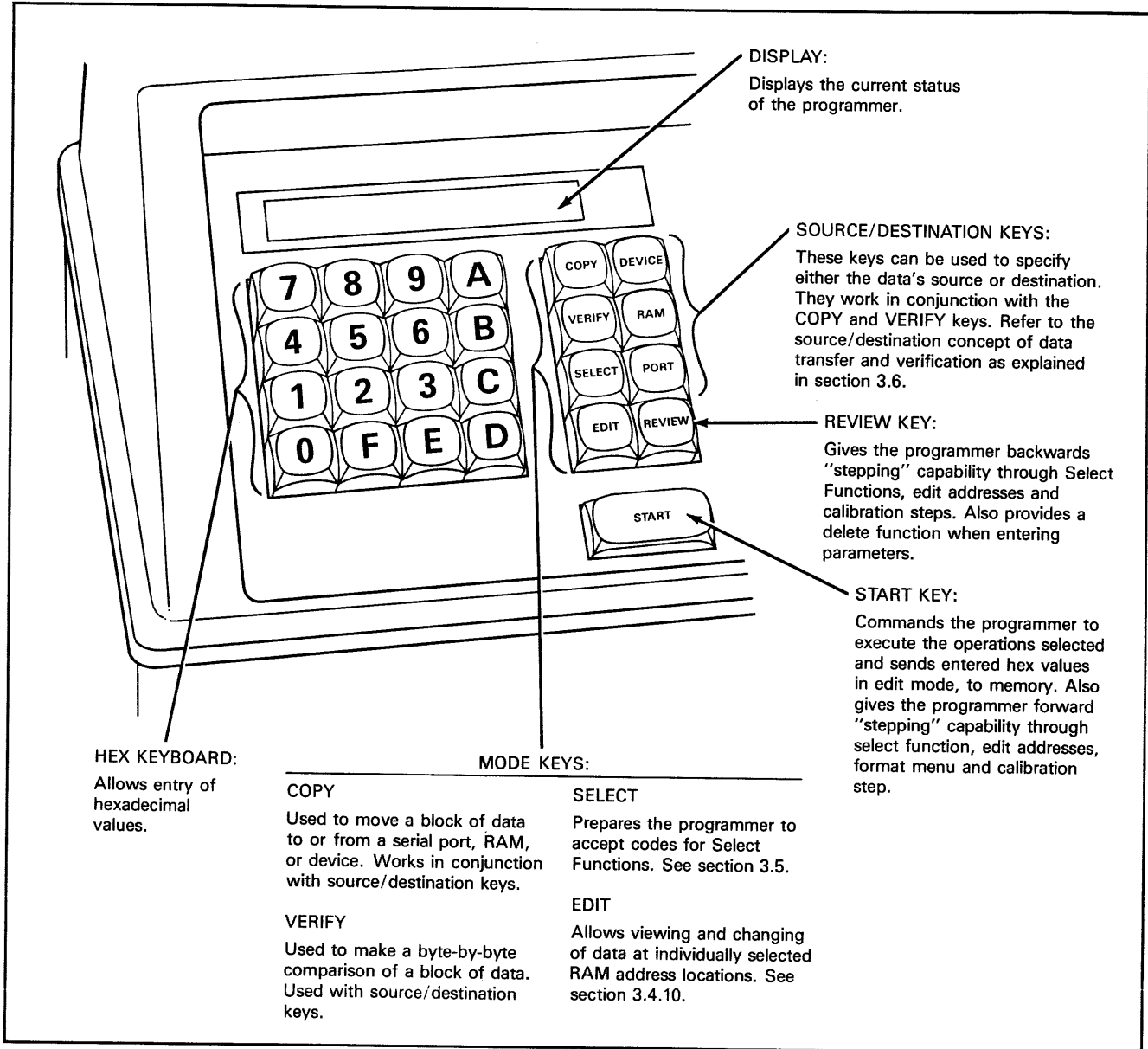


Figure 1-2. 29B Front Panel Functions

1.3.1 TYPICAL 29B OPERATION

COPY (Data Transfers)

To transfer data from one device to another or from one RAM location to another, use the COPY key. When you press the COPY key, the programmer moves data from the source (a device, the programmer RAM, or the serial port) to the destination (a device, the programmer RAM, or the serial port); for example, from the programmer data RAM to a blank device in the socket. At the completion of this operation, the device is programmed; that is, it contains a copy of the data in the programmer RAM. The "source/destination" concept is explained in section 3.5.

There are five basic types of copy operations. These operations are summarized below and described in detail in section 3.

- Load RAM with Master Device Data—transfers programming data from a master device and to the programmer RAM. When the data transfer is complete, the 29B calculates and displays the sum-check (see Glossary for definition) of the loaded data.
- Load RAM from Serial Port—translates data received at the serial port and transfers it to the programmer RAM. When completed, the programmer calculates and displays the sum-check of the data. If a sum-check has been sent with the data from the serial port, the programmer will compare the two and signal an error if they do not match.
- Program—puts the data in the programmer RAM into a device. Programming is automatic, starting with an illegal-bit test and a blank check to ensure that the device can be programmed. Data is then programmed into the device in the socket one byte at a time. This continues until all data bytes have been programmed into the device. After programming is completed, the data in the device is automatically compared with the RAM data to ensure correct programming.
- Output RAM to Serial Port—translates data from the programmer RAM and transfers it to the serial port.
- Block move—rearranges blocks of data within RAM.

VERIFY (Data Verification)

The VERIFY key is used to make a byte-by-byte comparison of data in two locations, one referred to as the "source" and one as the "destination"; see section 3-5 for details. There are two types of Verify operations.

- Verify Device—compares data twice from the device (the destination) byte-by-byte with the data in RAM (the source). On the first pass (first comparison), the 29B checks the parameters by lowering V_{CC} within the manufacturer's specified lower level. On the second pass, V_{CC} is raised to the upper level specification range.
- Verify RAM from Serial Port—compares incoming data from the serial port byte-by-byte with the data in RAM.

EDIT (Editing Data)

The EDIT key allows you to view and change data at specified RAM addresses. Section 3 provides specific details.

SELECT (Select Functions)

The SELECT key allows you to change certain operational parameter default values, perform RAM data manipulations, and access certain less frequently used operations. These operations are referred to in this manual as Select Functions. Detailed Select Function information, including operation, is located in section 3.8.

1.3.2 SAMPLE 29B PROGRAMMER OPERATION WITH UNIPAK™

The 29B is a versatile tool for programming data into blank devices. Figure 1-3 illustrates the process of programming data from RAM into a blank device by using the programmer's front panel keyboard. In this example, the source of data is a master device; however, data can also be input from the programmer keyboard or through the serial port.

When using a master device as your source of data, you need only a few steps to program this data into a blank device. With the appropriate pak installed (section 2.2) and the machine powered on (section 2.4), the steps you need to perform this operation are provided below and diagrammed in figure 1-3.

WARNING

Refer to the safety instructions of section 2.4 before powering up the 29B.

1. Press: COPY + DEVICE + RAM + START.
2. Enter the family and pinout codes for your device. These codes are included in your Pak manual.
3. Insert the master device into the appropriate Pak socket.
4. Press START.
5. Remove master device from the socket.
6. Press: COPY + RAM + DEVICE + START. If the device to be programmed has different family and pinout codes from the master, enter the correct codes. Otherwise, go to the next step.
7. Insert the blank device into the appropriate Pak socket.
8. Press START.

Step 8 will initiate an illegal-bit and blank check test (see Glossary) of the device followed by the programming and verify sequence. The programmer will signal you when the device has been programmed.

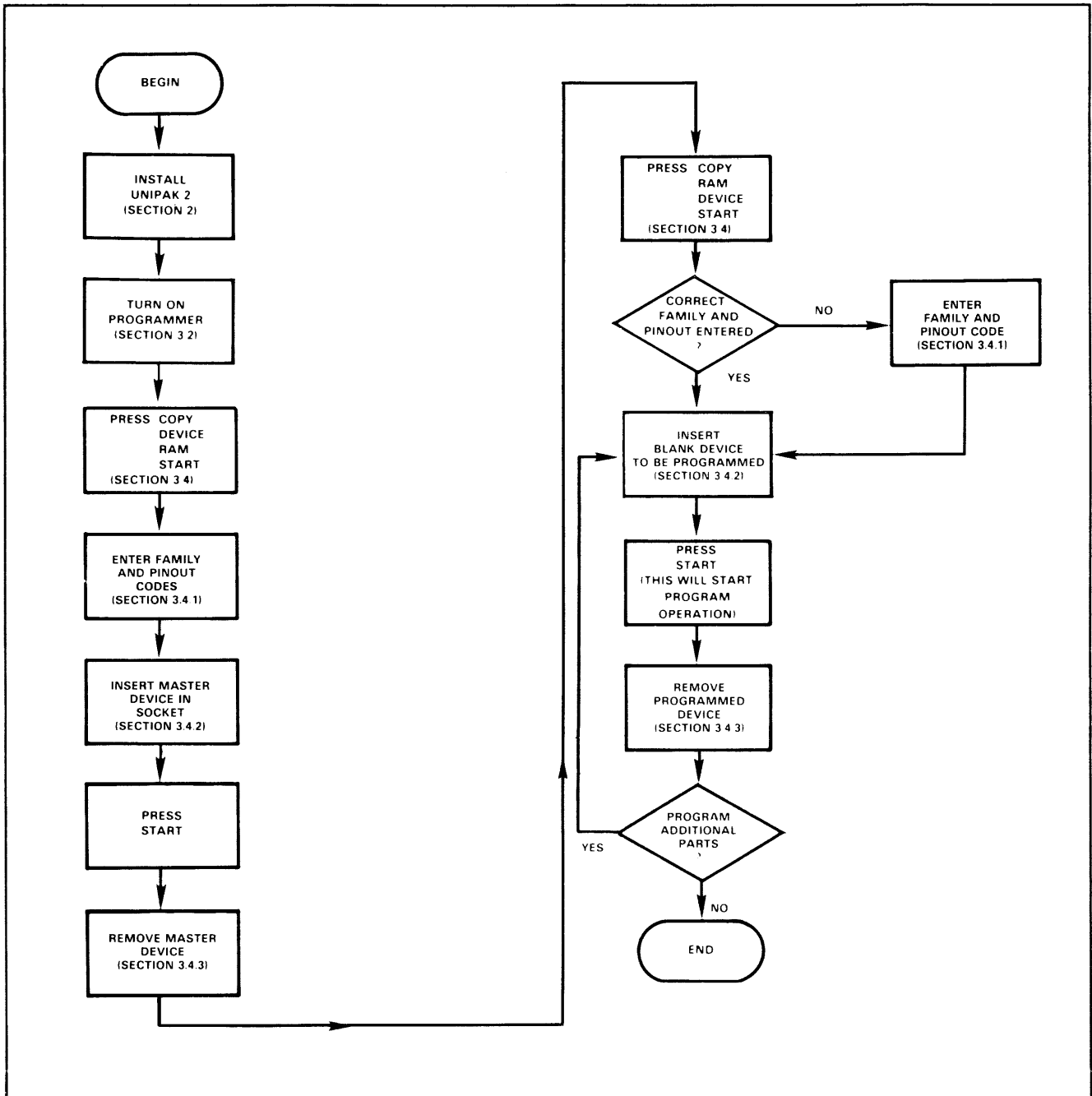


Figure 1-3. Sample 29B Programmer Operation

After data have been loaded into RAM from the serial port or the master device, the programmer calculates the sum-check of the data (figure 1-4) and displays it (when in terminal mode, it will be displayed on the terminal). The sum-check, which is used to verify the integrity of data transfers, is a summation of 8-bit bytes of fuse data expressed as a 4-digit hexadecimal number (see figure 1-4).

84	1000100
C1	11000001
62	01100010
24	00100100
01CB	0000 0001 1100 1011
HEX DATA	BINARY DATA
Sum-check in hexadecimal notation	Sixteen-bit binary sum-check

Figure 1-4. Sample Sum-Check Calculation

If data were loaded through the serial port and a sum-check was sent with it, the programmer will compare the sum-check with its own calculation. If they agree, the correct sum-check is displayed. If they do not agree, the programmer will signal an error. Data from the serial port will also be checked for correct parity if the program parity switch is on (see subsection 2.5.2).

The System 19, 100A, 29A, and 29B programmers have a data-lock feature that allows you to limit the accessibility of your RAM data. This is useful in production environments to protect the integrity of your RAM data. Refer to section 3 for information on data lock.

For a program operation with the 29B programmer, programming is automatic and starts with a series of tests: backward device test, illegal-bit test, and blank check. During the backward device test, the programmer automatically checks the device's orientation in the programmer module's socket(s) and displays an error if it is inserted backwards. The 29B displays:

DEV BACKWARDS 32

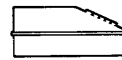
The illegal-bit test checks for previously programmed bits in a nonblank device that cannot be programmed according to the data in RAM. If illegal bits exist, the 29B displays:

ILLEGAL BIT 21

During the blank check, the programmer searches the device for programmed bits. If any are found (and the bits are legal), the programmer will signal the operator and the 29B displays:

NON BLANK 20

Nonblank parts can be over-programmed by again pressing



If the device passes these tests, data are transferred from the programmer RAM to the Pak one byte (eight fuse states) at a time. The Pak then applies the programming pulses to the first fuse and tests its condition. If the address fails to program, the 29B displays:

PROGRAM FAIL 22

Otherwise, programming proceeds to the next address until all have been programmed; with some P/T adapters, programming algorithms vary and may display only a verify error. Refer to the Pak manuals for manufacturer-specific programming algorithms and waveform pictures.

1.3.3 REMOTE OPERATION

The 29B can also be operated in remote control. Computer Remote Control (CRC) and System Remote Control (SRC) are explained in section 3 (Operation). An optional Terminal Remote Control (TRC), compatible with System 1903, is documented in appendix B.

1.4 SPECIFICATIONS

The 29B's specifications are listed below.

1.4.1 FUNCTIONAL SPECIFICATIONS

Functional specifications for the 29B are as follows:

- General Architecture: Microprocessor controlled (6808)
- Data RAM: 64K x 8
- Programming Support: GangPak™, LogicPak™, UniPak 2™, MOSPak™ and programming modules
- Keyboard: 16-key hexadecimal and 9-key functional
- Display: 16-character alphanumeric
- Input/Output: Serial RS232C and 20 mA current loop
- Baud Rates: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19,200
- Remote Control: Computer Remote Control (CRC), System Remote Control (SRC), Optional Terminal Remote Control (TRC)
- Translation Formats: See table 1-2.
- Handler Capability: Optional handler port is available for binning and control signals.

Table 1-2. Available Translation Formats for the 29B Universal Programmer

ASCII-B10F	DEC Binary
ASCII-BHLF	Fairchild Fairbug
ASCII-BNPF	Hewlett-Packard 64000 Absolute
ASCII-Hex (Apostrophe)	Intel Intellec 8/MDS
ASCII-Hex (Comma)	Intel MCS-86 Hexadecimal
ASCII-Hex (Percent)	Object
ASCII-Hex SMS	MOS Technology
ASCII-Hex (Space)	Motorola Exorciser
ASCII-Octal (Apostrophe)	Motorola Exomax
ASCII-Octal (Percent)	RCA Cosmac
ASCII-Octal SMS	Signetics Absolute Object
ASCII-Octal (Space)	Spectrum
Binary	Tektronix Hexadecimal
BNPF (5-Level)	Texas Instruments SDSMAC

1.4.2 POWER REQUIREMENTS

Power requirements for the 29B are as follows:

- Operating Voltages: 100, 120, 220 or 240 Vac + 5% or - 10%.
- Frequency Range: 50 ± 2 Hz or 60 ± 2 Hz for 100V and 120V units, 50 ± Hz for 220V and 240V units.
- Power Consumption: 115W/175 VA
- Fuse Protection: primary and secondary fuse protection

1.4.3 PHYSICAL AND ENVIRONMENTAL

Physical and Environmental Requirements for the 29B are as follows:

- Dimensions: 31.1 x 15.2 x 27.3 cm (15 x 6 x 10.8 in.)
- Weight: 6.4 kg (14.1 lb)
- Operating Temperature: +5° to 45°C (41° to 113°F)
- Storage Temperature: -40° to 70°C (-40° to 158°F)
- Humidity: to 95% (noncondensing)
- Operational Altitude: to 10,000 ft.

1.5 FIELD APPLICATIONS SUPPORT

Data I/O has field applications engineers (FAEs) who can provide you additional information about interfacing Data I/O products with other equipment and answer your questions about problems you may have with your equipment. The location of the FAE nearest you is given in the back of this manual. Call your FAE if you have any questions or problems.

1.6 WARRANTY

Data I/O equipment is warranted against defects in materials and workmanship. The warranty period is one year and begins when you receive the programmer. The warranty card at the back of this manual explains the length and conditions of the warranty. For warranty service, contact your nearest Data I/O Service Center.

1.7 SERVICE

Data I/O maintains Service Centers throughout the world, each staffed with factory-trained technicians to provide prompt, quality service. A list of all Service Centers is located at the back of this manual. In addition to making repairs, they clean and calibrate all serviced equipment.

If you must ship the 29B to the Data I/O factory or a Service Center for servicing, do the following:

1. Contact the Service Center to let them know you will be sending in equipment.
2. Attach a tag to the equipment, describing the work required and identifying the owner and the purchase order number for the servicing.
3. Wrap the instrument with heavy paper or plastic.
4. Place it in the original shipping container or some other suitable heavy carton with foam packing material and seal the container with strong tape.
5. Mark the container "DELICATE INSTRUMENT" and "FRAGILE."

In correspondence, identify the unit by serial number, model number and name.

For more information about servicing Data I/O products, contact your nearest Data I/O Service Center or sales representative.

1.8 ORDERING

To place an order for equipment, contact your Data I/O representative. Orders for shipment must include the following:

- Description of the equipment (see latest Data I/O price list or contact your sales representative for equipment and part numbers).
- Purchase order number
- Desired method of shipment
- Quantity of each item ordered
- Shipping and billing address of firm, including zip code
- Name of person ordering equipment

SECTION 2

INSTALLATION

2.1 INTRODUCTION

This section describes how to set up your programmer for operation. For best results in getting your programmer ready for use, we suggest that you follow the procedures described in the following subsections.

- Pak Installation (section 2.2)
- Power Connection (section 2.4)
- Power and Fuse Requirements (sections 2.4.1 through 2.4.4)
- Serial I/O Interface (section 2.5)

Your 29B was thoroughly calibrated, tested, and inspected before shipment. The unit was carefully packaged to prevent damage and should arrive free of any defects and in perfect operating condition. Carefully inspect it for any damage that may have occurred during transit. If you note any damage, file a claim with the carrier and notify Data I/O.

Check to make sure that you have received the following required equipment:

Item	Part Number	Quantity
• 29B Universal Programmer	990-0013	1
• Power Cord		1
-Europe	416-0010	
-USA/Canada/Japan	416-1577	
• Serial Port Mating Connector	401-3064	1
• Serial Port Connector Hood	401-3069	1
• Operator's Guide	12-990-0013	1

Check the operation of the unit after you have completed the installation instructions in this section.

2.2 PAK INSTALLATION

Any of the Data I/O programming Paks may be installed and removed with the programmer's power on; this feature allows you to retain data in the 29B RAM when changing programming Paks. If the programmer power is turned on before a Pak is installed, you will hear a "beep" until the Pak is properly installed.

CAUTION

Voltage transients can cause device damage. Be sure that all sockets are empty when switching power on or off or installing or removing the Pak.

To install a Pak into the 29B, refer to figure 2-1 and follow this installation procedure.

1. Slide the Pak into the opening in the programmer.
2. Tilt the Pak up, and gently push it back to hook its flange over the top back edge of the programmer opening (figure 2-1a).
3. Lower the Pak into position as shown in figure 2-1b.

CAUTION

Be careful when inserting the Pak. If the connector at the bottom of the Pak (see figure 2-1b) has bent contact pin(s), forcing the Pak could break the pin(s) or damage the connector.

4. Press down gently on the front edge of the Pak to ensure a good connection (figure 2-1c).

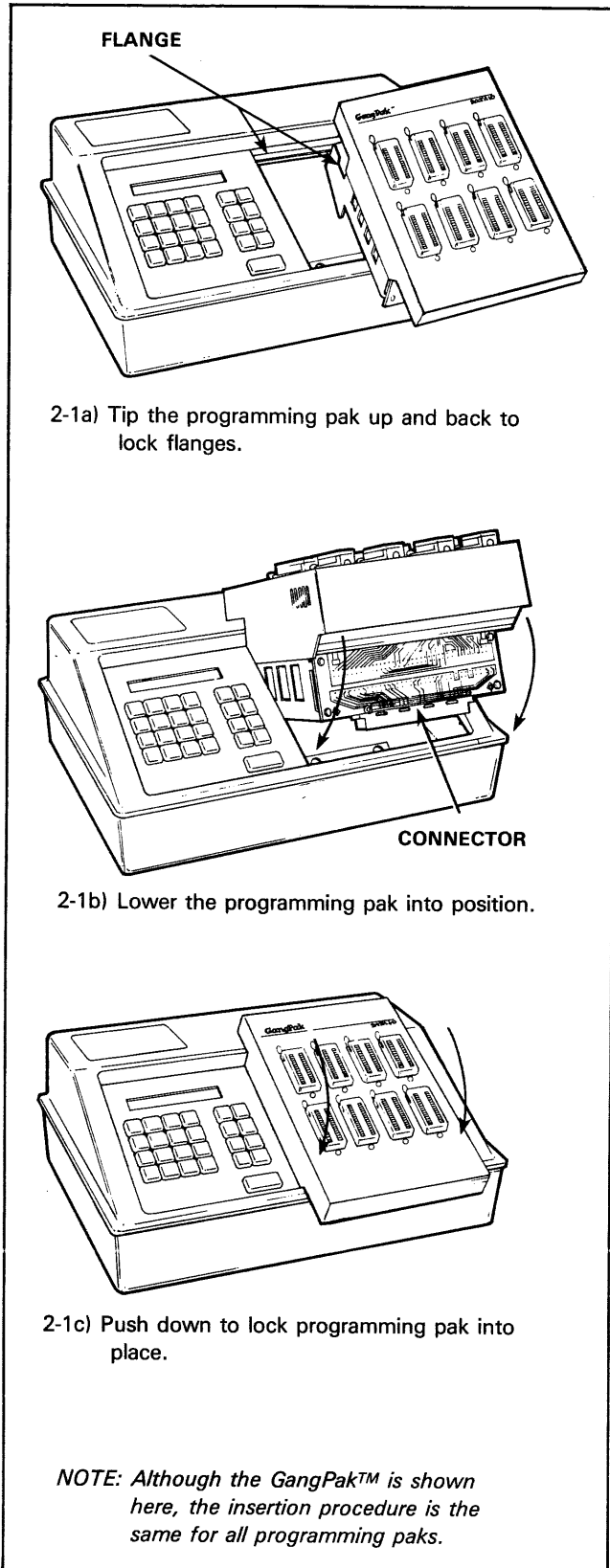


Figure 2-1. Installing/Removing a Programming Pak

2.3 PAK REMOVAL

To remove the Pak from the 29B:

1. Check to make sure the programmer is not in the middle of an operation. If it is, wait until the operation is complete. The operation is complete when the action symbol on the display disappears.
2. Check to make sure a device is not in a socket. If one is in a socket, remove it as described in subsection 3.4.3.
3. Tilt the front portion of the Pak up and gently remove it from the programmer.

2.4 POWER CONNECTION

Before applying power to your programmer, make sure that the operating voltage is correct (section 2.4.1), that the line fuse is intact (section 2.4.2), and that the unit is properly grounded (section 2.4.4). Finally, connect your power cord and apply power to the programmer (section 2.4.5).

2.4.1 VERIFYING/CHANGING THE OPERATING VOLTAGE

The factory has selected the proper voltage according to your specification. A voltage reading is visible through a window in the door that covers the voltage wheel selector, located on the back panel, as shown in figure 2-2. This voltage should be the same as the line voltage on which the machine will operate. If the voltage that appears in the window is incorrect, change the operating voltage according to the following procedure.

CAUTION

This instrument may be damaged if operated with the wrong line voltage.

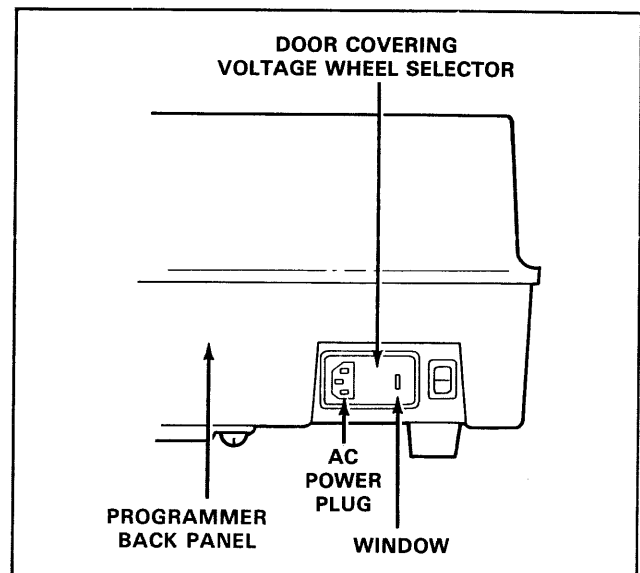


Figure 2-2. Back Panel

The procedures to verify and or change the operating voltage are described here and illustrated in figure 2-3.

1. Gently pry open the door that covers the voltage wheel selector with a flat-blade screwdriver (figure 2-3a).
2. Pull the voltage wheel selector out of its slot (figure 2-3b).
3. Rotate the selector until the correct operating voltage points toward you (figure 2-3c).

4. Insert the selector back into its slot.

NOTE

If you wish to access the line fuse at this point, proceed to step 2 in section 2.4.2.

5. Snap the door closed.
6. The correct voltage reading will now appear in the window (figure 2-3d).

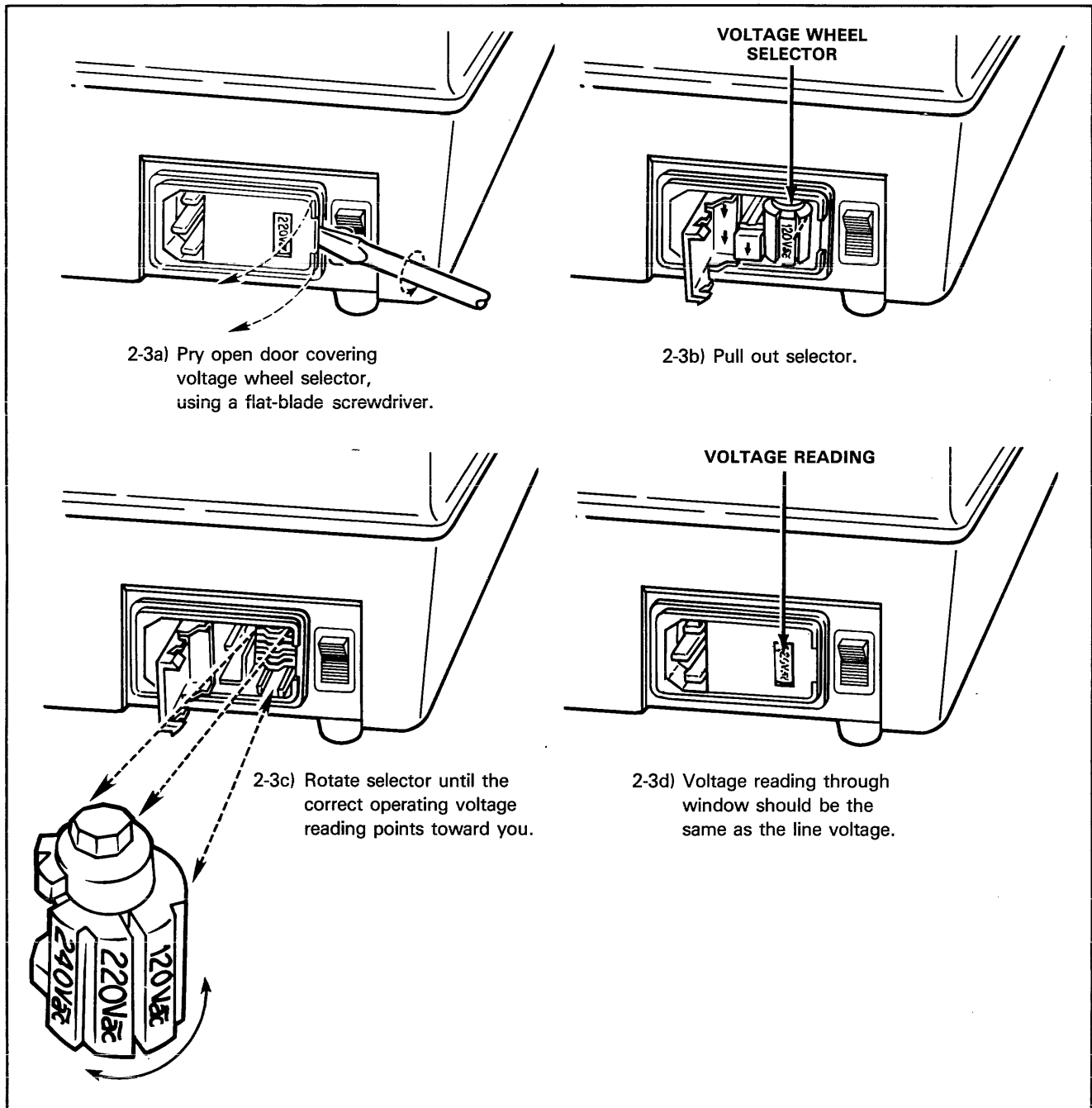


Figure 2-3. Voltage Wheel Selector

2.4.2 VERIFYING/REPLACING THE LINE FUSE

The line fuse is located behind the same door that covers the voltage wheel selector. After you determine that your 29B is set to the proper operating voltage, perform the following procedure to verify that the line fuse is correct and intact. In the event that the fuse is blown, replace it with one of the correct size. Procedure steps are illustrated in figure 2.4.

1. Gently pry open the door that covers the fuse holder using a flat-blade screwdriver (figure 2-4a).

NOTE

There are two fuse receptacles; only the one on the bottom is connected to the programmer's circuitry. The top receptacle is a spare fuse tray. See the next subsection (2.4.3) for more information on this spare tray.

2. Pull the bottom fuse holder out of its slot (figure 2-4b).
3. Check to determine whether the fuse is intact. If it is intact, proceed to step 4. If it is blown, install a new fuse. See table 2-1 for line fuse ratings.

CAUTION

For continued protection against the possibility of fire, replace only with a fuse of specified voltage, current and type ratings.

Table 2-1. Line Fuse Ratings

Operating Voltage	Line Fuse Rating			Data I/O Part Number
	Current	Voltage	Type	
100	2.5 amp	250V	slow-blow ^(a)	416-1240-001
120	2.5 amp	250V	slow-blow ^(a)	416-1240-001
220	1 amp	250V	slow-blow ^(b)	416-1571-001
240	1 amp	250V	slow-blow ^(b)	416-1571-001

^(a)Littlefuse type "313," Bussman type "MDA"
^(b)Littlefuse type "218," Bussman type "GDC," Schurter type "001-2504."

4. Insert the fuse holder into its slot so that the arrow on the fuse holder points in the same direction as the arrows on the door (figure 2-4c).
5. Snap the door closed.

2.4.3 SPARE LINE FUSE TRAY

All 29B programmers are equipped with two line fuse trays (see figure 2-4 (b)). The white fuse tray accepts 1/4 x 1 1/4 inch fuses; the black tray accepts 5 x 20 millimeter fuses, commonly available in Europe. Only the right-hand fuse receptacle is connected to the programmer's circuitry.

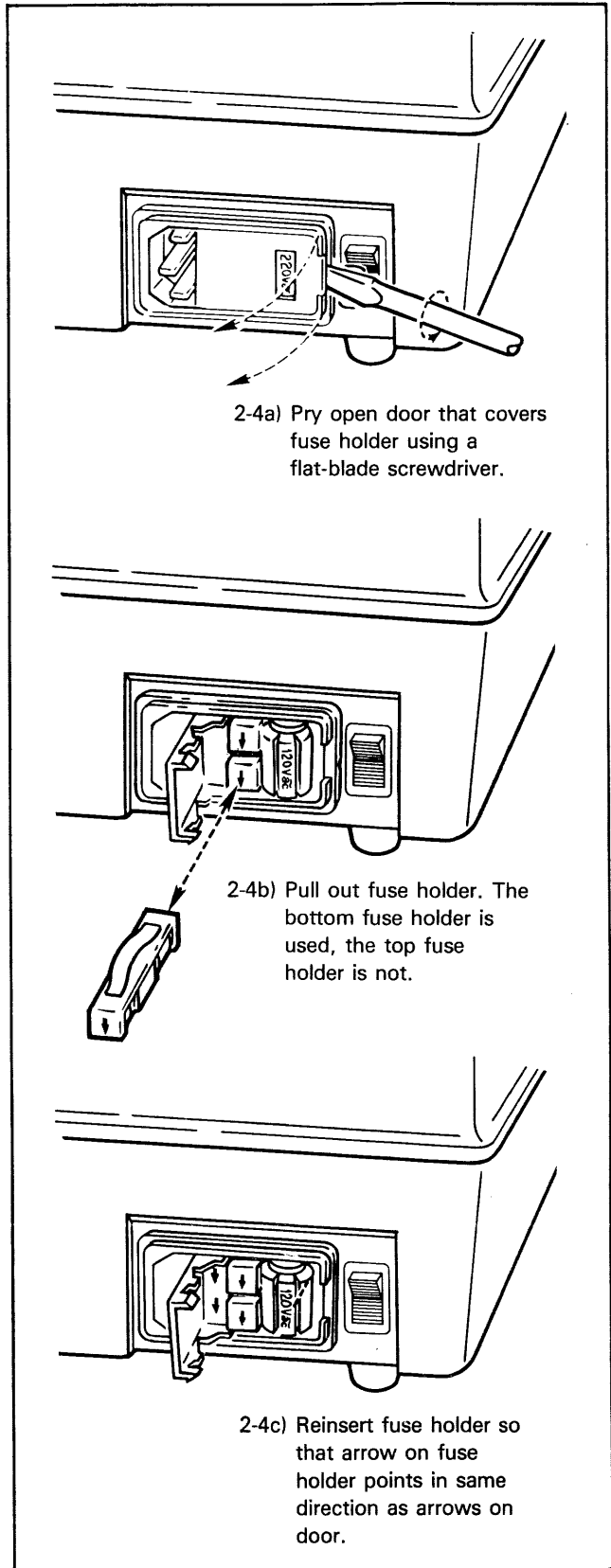


Figure 2-4. Accessing the Line Fuse

2.4.4 GROUNDING THE UNIT

The 29B is shipped with a three-wire power cable. This cable connects the chassis of the unit to the earth ground when the cable is connected to a three-wire (grounded) receptacle.

WARNING

Continuity of the grounding circuit is vital for the safe operation of the unit. Never operate this equipment with the grounding conductor disconnected.

2.4.5 APPLYING POWER

Now that you have verified the programmer's voltage setting, fuse and grounding, you are ready to apply ac power. Follow this procedure:

1. Check to make sure a device is not plugged into any Pak socket. If a device is in a socket, flip up the socket's locking arm and remove the device from the socket. See subsection 3.4.3.
2. Connect to ac power by plugging the power cord into the back panel and into an ac receptacle.
3. Press the power switch to the ON position; figure 2-5 shows the power switch location and indicates the ON position.

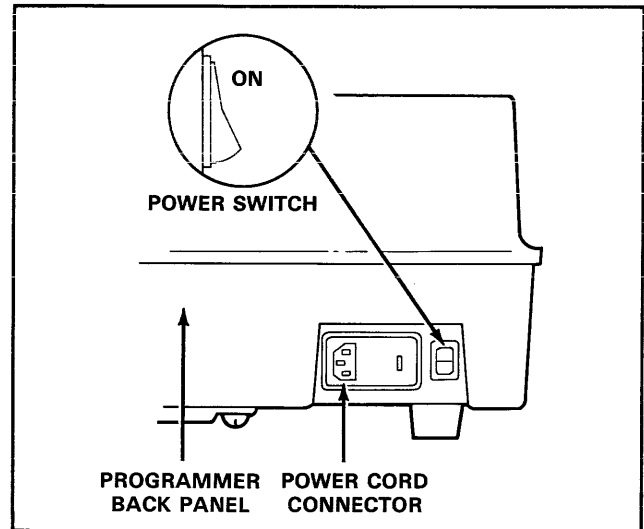


Figure 2-5. Programmer Power Switch Location

2.5 SERIAL I/O INTERFACE

An RS232C serial port interface is used to connect the 29B to computer systems and other peripherals. This requires installing the proper serial interface cabling and setting the appropriate operational parameters.

NOTE

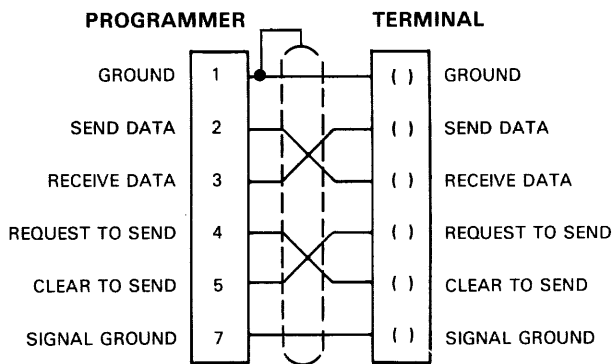
Any computer or any other peripheral device that interfaces to the 29B's serial port must allow for duplex operation consistent with the 29B programmer's remote control software. System remote control (SRC) and computer remote control (CRC) are half duplex, whereas terminal remote control (TRC) is full duplex.

2.5.1 CABLING

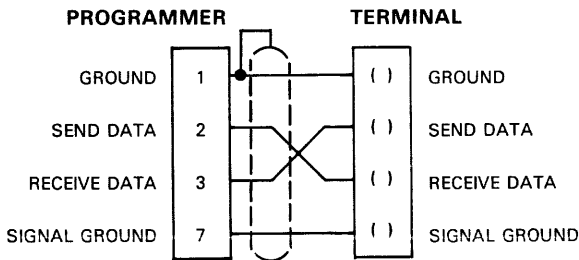
To connect the 29B to other instruments, you must use the serial interface connector pin assignments called out in table 2-2; figure 2-6 shows sample interconnections in the serial interface for half/full duplex with handshake (figure 2-6a) and without handshake (figure 2-6b) and current loop connection for full duplex (figure 2-6c) and half-duplex (figure 2-6d). See figure 2-7 for the location of the RS232C serial port connector.

NOTE

To reduce electromagnetic interference (EMI), we recommend using a shielded cable.



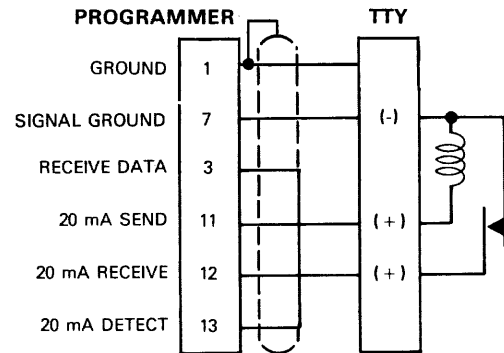
2-6a) RS232C Connection, Half/Full Duplex, with Handshake



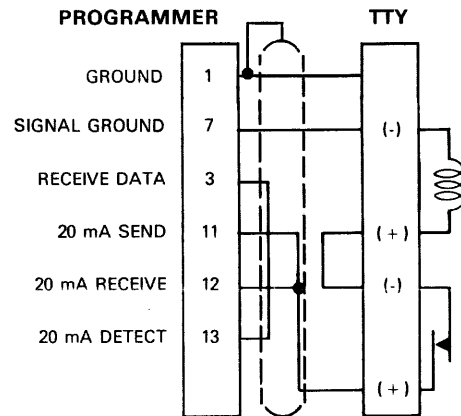
2-6b) RS232C Connection, Half/Full Duplex, without Handshake

NOTES:

1. All signals are named with respect to the originating unit.
2. All undesignated pins are to be left open.
3. For applications that do not require handshaking, the programmer's clear to send line is pulled up internally.



2-6c) 20 mA Current Loop Connection, Full Duplex



2-6d) 20 mA Current Loop Connection, Half Duplex

Figure 2-6. Sample Interconnection Methods

Table 2-2. Serial Interface Connector Pin Assignments

PIN NO.	SIGNAL MNEMONIC	DESCRIPTION
1	Ground	Provides a safety ground connection.
2	Send Data	Transmits data within RS232C voltage levels (+ 12V and -5V).
3	Receive Data	Accepts data within RS232C voltage levels.
4	Request to Send	This line is normally held high by the programmer. It is dropped to inhibit data transmission from a remote source.
5	Clear to Send ^(a)	A high level on this line allows the programmer to transfer data. A low level inhibits data transfer.
6	Data Set Ready	Connected by internal jumper to Data Ready (pin 20). Simulates indication that the programmer is operating.
7	Signal Ground	This line provides a common signal connection to the RS232C remote source.
8	Carrier Detect ^(a)	This line is positive when modem detects a carrier signal. The line is sampled by the programmer if used.
9	+ 24 Vdc	Available for external use if required (500 mA maximum).
10		Not used.
11	20 mA Send	Transmits data using active 20-mA current loop.
12	20 mA Receive	Accepts data using active 20-mA current loop.
13	Detect	20-mA receive data (pin 12) is internally converted to RS232C levels. Output on pin 13 should be jumpered externally to receive data (pin 3).
14-19		Not used.
20	Data Ready	Connected by internal jumper to data set ready (pin 6). A high level on this line from the RS232C data terminal indicates that the data terminal is ready.
21		Not used.
22	+ 5 Vdc	Available for external use if required (200 mA maximum).
23	- 5 Vdc	Available for external use if required (200 mA maximum).
24-25		Not used.

^(a) Pins 5 and 8 have internal pull-ups and need no connection if unused.

2.5.2 SETTING PARAMETERS

Before the 29B can operate with another system, three parameters must be set: parity, stop bits and the baud rate. These parameters must be the same for both systems.

Baud Rate

To set the baud rate, refer to figure 2-7 and follow this procedure:

1. Locate the baud rate rotary switch on the programmer's rear panel.
2. Locate the desired baud rate on the chart in figure 2-7 and the switch position required for that baud rate.
3. With a flat-blade screwdriver, turn the switch to the numbered switch position that corresponds to your desired baud rate.

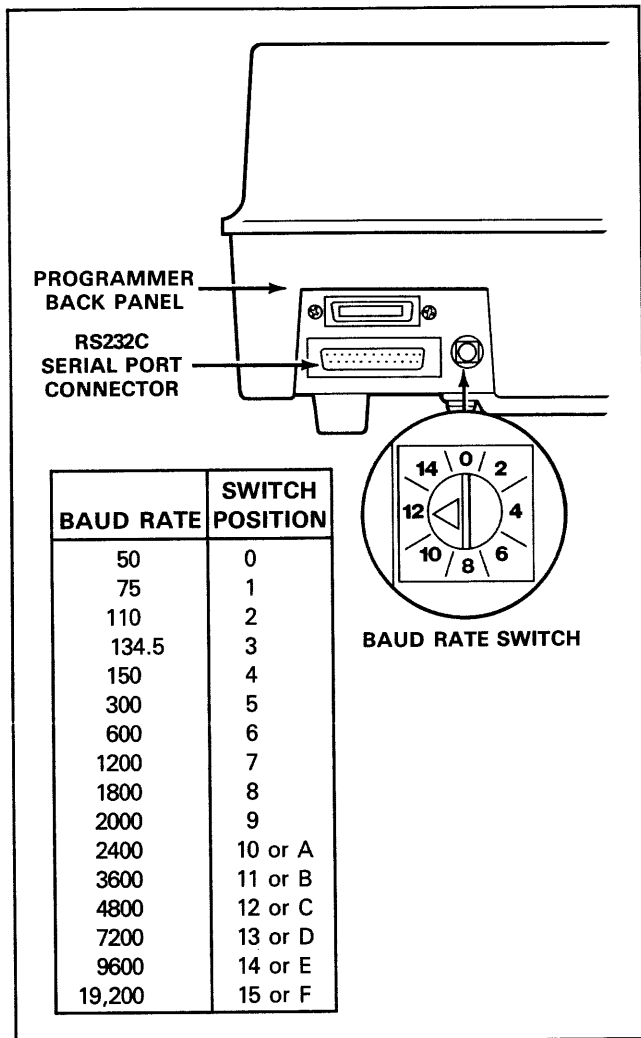


Figure 2-7. Baud Rate Switch and RS232 Serial Port

Parity and Stop Bits

To set parity and stop bits, refer to figure 2-8 and follow this procedure:

CAUTION

Make sure that all device(s) have been removed from the Pak socket(s) before removing the Pak or turning off the programmer. Voltage transients caused by removing the Pak could damage the device(s).

1. Turn off programmer.
2. Remove the Pak from the programmer (see subsection 2.3).
3. Access the controller board's status switch (U53) through the cut-out in the lower right corner of the protective shield. See figure 2-8.
4. Flip the switches to the desired setting; figure 2-8 shows the switch positions set by the factory.
5. Reinstall the Programming pak and turn on the programmer.

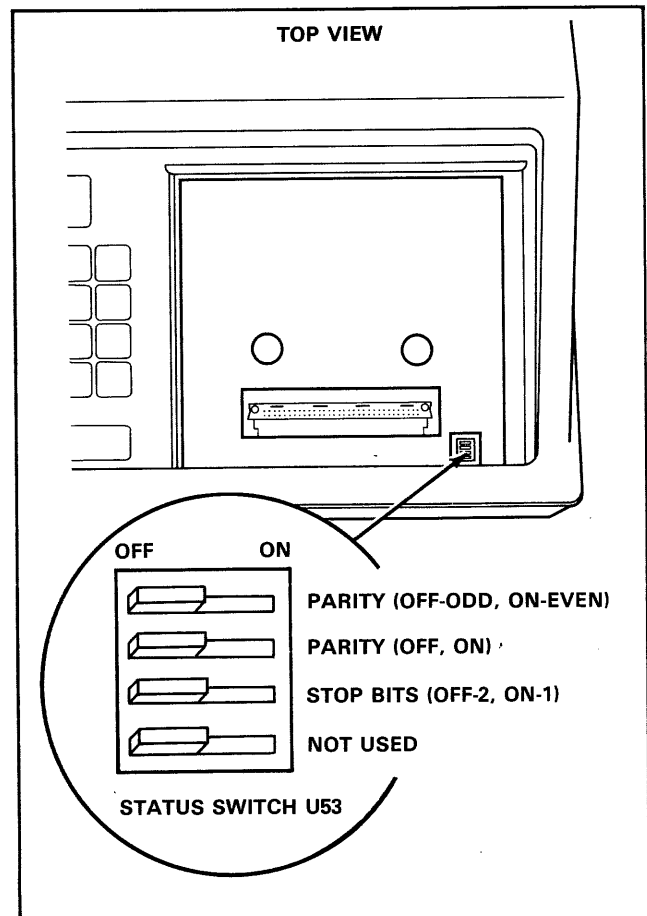


Figure 2-8. Status Switch Setting

2.5.3 HOOKING UP A SERIAL PAPER TAPE READER

A Data I/O Serial Paper Tape Reader (950-1950) can be connected to your programmer. A direct connection, to the programmer's serial port, is made using the existing serial paper tape reader cable. It will connect according to the specifications in figure 2-9. Set the baud rate at 2400. Two operations are possible using the serial paper tape reader: load RAM from the serial port and verify RAM from the serial port.

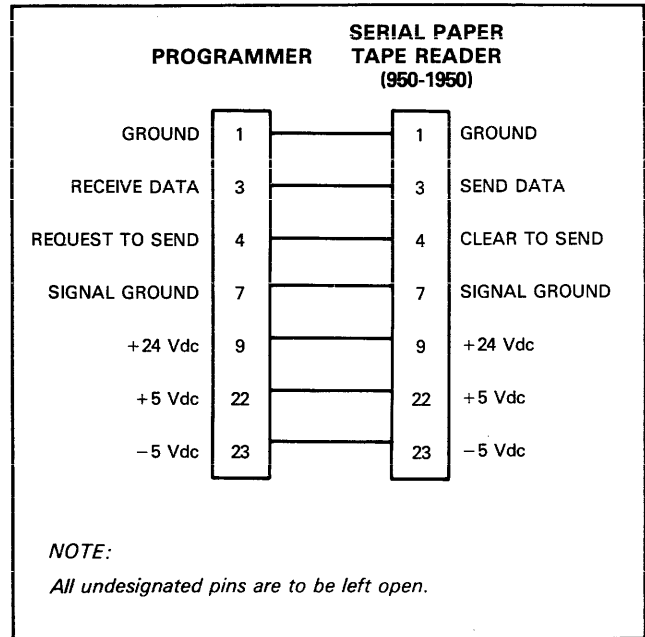


Figure 2-9. Interconnection method of Serial Paper Tape Reader

SECTION 3 OPERATION

3.1 INTRODUCTION

The 29B can be operated from the front panel keyboard or from remote control via a terminal or computer.

Subsections 3.2 through 3.8 provide basic operation procedures from the front panel keyboard. Procedures for using the 29B from Remote Control are documented as follows:

Computer Remote Control (CRC)	Section 3.9.1
System Remote Control (SRC)	Section 3.9.2
Terminal Remote Control (TRC)	Appendix B

The 29B obtains data for the basic operations from any of three sources: a master device, the serial port, or data entered manually from the programmer front panel keyboard. The basic operations of the 29B are: copy, verify, edit, and select functions. These are briefly described below and the step-by-step procedures are described in the following subsections.

- Copy - moves data from source (device, RAM, or port) to a destination (device, RAM, or port). The specific copy operations are:
 - Load RAM with master device data. Subsection 3.6.1.
 - Load RAM from serial port. Subsection 3.6.2.
 - Program device with RAM data. Subsection 3.6.3.
 - Output RAM data to port. Subsection 3.6.4.
 - RAM to RAM block move. Subsection 3.6.5.
- Verify - compares data between a source and destination. The verify operations include:
 - Verify device data (comparing device data with RAM data) subsection 3.6.6.
 - Verify input data (comparing serial input data with RAM data) subsection 3.6.7.
- Edit - changes data at selected addresses within the programmer RAM. The edit operation is documented in subsection 3.7.
- Select Function - allows either the manipulating of RAM data or the changing of various operating parameters. These functions are listed and described in subsection 3.8.

These operations and their uses are further described in Introduction to Operation (see section 1). All front panel copy and verify operations are presented in subsection 3.6

in a step-by-step key sequence format. Copy and verify operations follow the source/destination method of data transfer and verification (see subsection 3.5). Entries on the 29B front panel keyboard are indicated by the 29B symbol and the key to be pressed. For example; instructions to press the RAM key on the 29B front panel keyboard would be illustrated as follows:



Section 3 consists of nine subsections. The title of each subsection and a brief explanation of each is listed below:

- 3.1 INTRODUCTION: Lists the major items covered in this section.
- 3.2 POWER-UP: Gives the procedures to apply power to the programmer.
- 3.3 POWER DOWN: Gives the procedures to remove power from the programmer.
- 3.4 MANUAL KEYBOARD OPERATIONS: Gives an explanation of the family/pinout code and explains the method of entering the code. Gives the procedures to insert and remove a device from the programmer pak or module sockets.
- 3.5 SOURCE/DESTINATION METHOD OF SYNTAX (COPY AND VERIFY): Shows the syntax used when executing a copy or verify instruction.
- 3.6 FRONT PANEL KEYBOARD DATA TRANSFER OPERATIONS: Gives the procedures to manually execute the copy and verify instructions from the programmer front panel keyboard.
- 3.7 KEYBOARD EDIT OPERATIONS: Gives the procedures to execute edit instructions from the programmer front panel keyboard.
- 3.8 SELECT KEY FUNCTIONS: Gives the procedures to execute select instructions from the programmer front panel keyboard.
- 3.9 REMOTE OPERATIONS: Explains the two standard remote control systems used with the 29B Programmer and gives the command structure and operating procedures for each.

3.2 POWER-UP

3.2.1 GENERAL OPERATION NOTES

Keep the following items in mind while operating the 29B.

- **Aborting an Operation.**

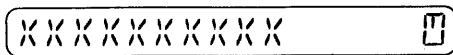
Most operations (except for a few select functions) can be aborted by pressing one of the four mode keys (**COPY**, **VERIFY**, **SELECT**, or **EDIT**). If an operation is in progress when one of these keys is pressed, the 29B momentarily displays the message:

FUNCTION ABORT

then assumes the mode selected and displays the message appropriate for the mode selected.

- **Action Symbol.**

Copy, verify, and some select operations in progress are indicated by a rotating action symbol in the display as shown below.



- **Changing an Address or Block Size.**

Some operations prompt for address or block size information by displaying a “^” character followed by “ADDR” (in some cases “ADDRESS”), or “SIZE” on the front panel keyboard display. For example; a copy operation might prompt for a block size or beginning address. A specific block size, RAM address, device address, or serial data address offset can be entered from the programmer front panel keyboard with the numeric keys, or the default values can be entered by pressing the start key. The default states for the various address and block size parameters associated with device, RAM, and port-related operations are identified in subsection 3.5 (Source/Destination Method of Syntax).

- **I/O Operation.**

When executing Input/Output related operations, be sure the correct operational parameters (i.e., baud rate, parity bits, and stop bits) are set (see subsection 2.5).

3.2.2 POWER-UP PROCEDURE

Use the following procedure to power-up the programmer:

1. Install a programming Pak or module into the programmer if this has not already been done (see subsection 2.2).

NOTE

If a programming Pak or module is not installed in the programmer when power is turned on, the 29B will “beep” until a programming Pak is properly installed.

CAUTION

Be careful when installing the Pak to assure that the connector pins on the bottom of the Pak are not damaged.

Check to make sure the sockets on the Pak are empty. If a device is in the socket, lift up the socket lever (see subsection 3.4.3) then gently lift the device out of the socket. See subsection 2.2 for Pak installation procedures.

2. Plug the AC power cord into the rear of the programmer and into a power receptacle.
3. Press the power switch at the back of the programmer to the “ON” position (see figure 3-1).

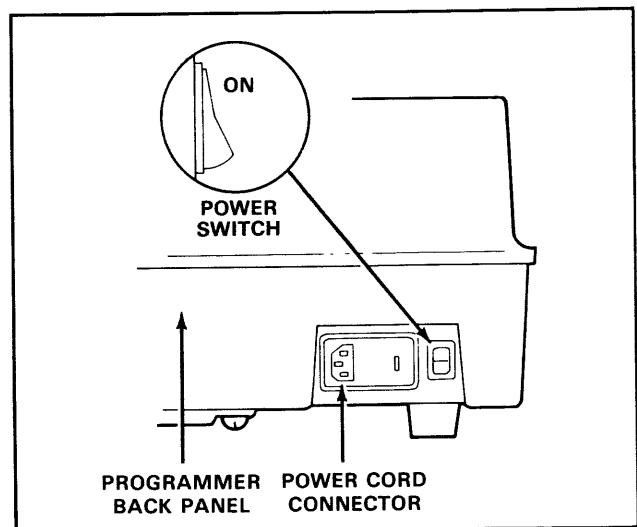
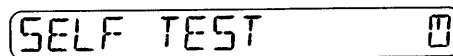


Figure 3-1. Power Switch and Power Cord Locations

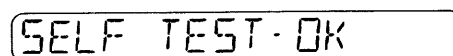
When power is fully applied, the programmer automatically performs the self-test routine which initializes the 29B hardware and checks the scratch RAM, firmware and data RAM. While the self test routine is being performed the programmer will display:



NOTE

The action symbol (□) has a rotating hand which rotates several times to indicate the programmer is performing the self test.

When the self test is complete the programmer will display:



3.3 POWER DOWN

CAUTION

Do not turn the programmer power off while a device is in a socket; voltage transients may damage the device.

Use the following procedure to remove power from the programmer:

1. Insure the programmer is not executing an operation. If it is, wait until the operation is completed.
2. If a device is installed in a socket, remove it using the procedure described in subsection 3.4.3 before continuing.
3. Press the power switch to the OFF position (see figure 3-1).

3.4 MANUAL KEYBOARD OPERATIONS

The 29B performs the following basic front panel keyboard functions:

- Copy Commands
 - Load RAM with master device data (subsection 3.6.1).
 - Load RAM from serial port (subsection 3.6.2).
 - Program device with RAM data (subsection 3.6.3).
 - Output RAM data to port (subsection 3.6.4).
 - RAM to RAM block move (subsection 3.6.5).
- Verify Commands
 - Verify device data (subsection 3.6.6).
 - Verify RAM from serial port (subsection 3.6.7).
- Edit Commands (see subsection 3.7)
- Select Commands (see subsection 3.8)

In addition to the basic front panel keyboard functions listed several operating functions may be selected using the keyboard edit (subsection 3.7) and select (subsection 3.8) functions.

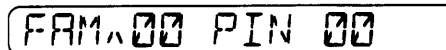
3.4.1 FAMILY AND PINOUT CODES

Some Programming Paks use family and pinout codes to identify the programming algorithms used to program various programmable devices. The codes are listed in the programming Pak manual for those Paks that require a family/pinout code entry. Once the codes have been entered in the 29B they remain in effect until they are changed for a new operation or power is removed from the programmer.

CAUTION

Be sure you enter only those family/pinout codes published in the Pak manual(s). Invalid codes can cause unpredictable results at the device socket, which may damage the device. Also, a valid family code and a valid pinout code can be combined to produce an invalid (illegal) combination. The correct family/pinout combination for a device can be determined by referring to the applicable pak manual(s). Any family and pinout combinations not contained in these manuals are considered illegal. Data I/O assumes no responsibility or liability for results produced by entry of illegal family/pinout code combinations.

Paks that have the capability for use with more than one type of device will prompt the operator to enter or change the family/pinout code combination whenever device related operations are performed. If a family/pinout combination has never been entered, the 29B displays:



FAM:00 PIN 00

NOTE

If this display appears, a family/pinout code must be entered before any device related operation can continue.

The family and pinout codes are published in the manual for the Pak being used with the programmer. Use the Pak manual and the following procedure to determine the family and pinout codes for the device being programmed.

1. Locate the manufacturer and part number stamped on the device.
2. Go to the family and pinout code table in the Pak manual and find the manufacturer's name.
3. Go to the column entitled "Device Part Number" and find the number corresponding to the number on the device.
4. Go to the column labeled "Family Code" and "Pinout Code" to find the code numbers corresponding to the device number for the manufacturer of the device.
5. Enter the Family and Pinout code you selected from the table when prompted by the programmer or terminal.

If a family/pinout combination has been entered previously, that code combination appears as a hex code in the 29B display as follows:

FAM,HH PIN HH

If the family/pinout code is correct for the device to be programmed, press the **START** key to continue. If it is not, press the applicable four keys on the hexadecimal keypad to enter the correct code. If an error is made while entering the family and pinout codes the copy or verify operation must be cancelled and restarted from the beginning.

NOTE

If the Pak or programming module is dedicated to one device, the family/pinout code is not required and the display prompt does not appear.

3.4.2 DEVICE INSERTION

After the applicable family and pinout codes have been entered, a device(s) can be inserted in the Pak device socket(s) or programming module socket adapter. The procedures presented here describe the insertion of a device in the 28-pin socket on a GangPak™. The basic insertion steps; however, apply to all devices and their socket receptacles.

A good electrical connection between the device and the socket is essential. Use the following procedure to ensure the device is properly installed in the socket:

1. Check to make sure the programmer is not executing an operation. If it is, wait until the operation is completed.
2. Be sure the correct family and pinout code for the device is entered or available for entry when prompted.

NOTE

For some Paks such as the UniPak 2™ once the family/pinout has been entered an LED will light up to indicate the socket used for the device.

3. Lift the lever on the side of the socket (see figure 3-2). The lever stays in the upright position.
4. Gently set the device in the socket. Be sure that pin 1 of the device is aligned with pin 1 of the socket as shown in Figure 3-2. If the socket has more contacts than the device has pins, insert the device so that the extra contacts are above the device; i.e., the bottom pins of the device must mate with the bottom pins of the socket.
5. Push the lever down to lock the device in the socket.

3.4.3 DEVICE REMOVAL

Use the following procedure to remove the device from the socket:

1. Check to make sure the programmer is not executing an operation. If it is, wait until the operation is completed.
2. Lift the lever on the side of the socket (see figure 3-2). The lever stays in the upright position.
3. Lift the device out of the socket.

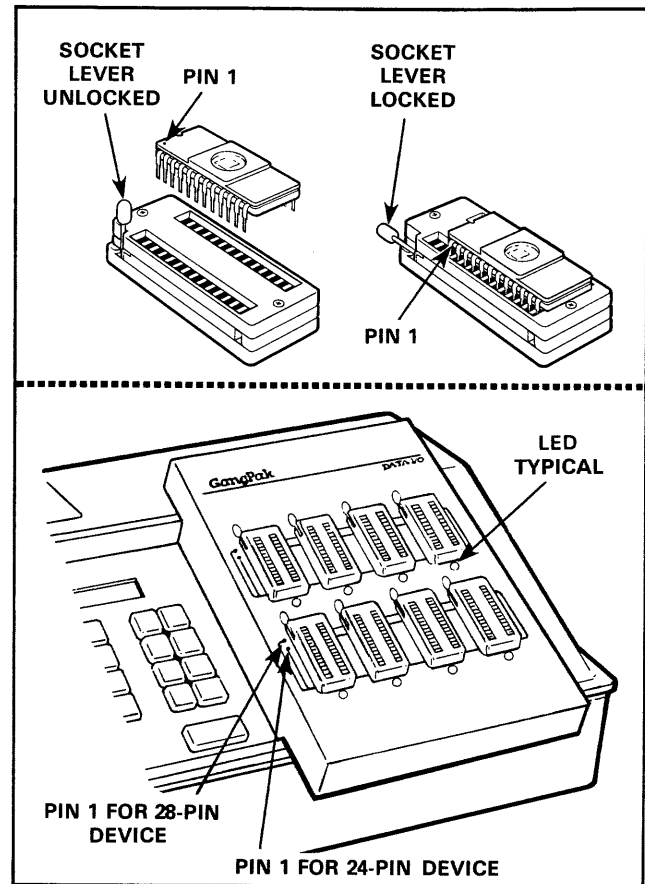


Figure 3-2. Device Installation (Multiuse Sockets)

3.5 SOURCE/DESTINATION METHOD OF SYNTAX (COPY AND VERIFY)

The 29B uses a source/destination method of syntax when operations are executed with the COPY and VERIFY commands. This syntax is used for all copy and verify operations commanded directly from the front panel keyboard or indirectly via the System Remote Control. This syntax is not used on either the Computer Remote Control or the optional Terminal Remote Control.

3.5.1 SOURCE/DESTINATION DETERMINATION


A COPY command operation directs the 29B to transfer data from one medium to another. A VERIFY command operation directs the 29B to verify data at one medium against data at another to assure they are identical.

The copy and verify commands are used in conjunction with the source/destination keys (DEVICE, RAM, and PORT). The operator initiates the operation by choosing either the COPY or VERIFY key, then specifying the source of the data (DEVICE, RAM, or PORT) and then the destination for the data (again, either DEVICE, RAM, or PORT). By using table 3-1, you can determine the operation you wish to execute based on the source/destination concept. This table also provides you with the location of the operational procedure section, within this manual. To determine the source and destination from the table go to the portion dealing with the operation (copy or verify) to be performed and find the box with the desired mode. Then read in the far left column the source required to perform the operation and read up to the top line for the destination required.

Table 3-1. COPY and VERIFY Keyboard Operations

COPY OPERATION			
DESTINATION SOURCE	DEVICE (blank device)	RAM (programmer data RAM)	PORT (peripheral)
DEVICE (master device)	/	LOAD FROM DEVICE See subsection 3.6.1	/
RAM (programmer data RAM)	PROGRAM DEVICE See subsection 3.6.3	BLOCK MOVE See subsection 3.6.5	OUTPUT TO PORT See subsection 3.6.4
PORT (peripheral)	/	INPUT FROM PORT See subsection 3.6.2	/

VERIFY OPERATION			
DESTINATION SOURCE	DEVICE (previously programmed device)	RAM (programmer data RAM)	PORT (peripheral)
DEVICE (master device)	/	VERIFY DEVICE See subsection 3.6.6	/
RAM (programmer data RAM)	VERIFY DEVICE See subsection 3.6.6	/	INPUT VERIFY See subsection 3.6.7
PORT (peripheral)	/	INPUT VERIFY See subsection 3.6.7	/

 Illegal Source/Destination Combination

When the COPY command is used, data is moved from the source to the destination; for example, from the programmer data RAM to a blank device in the socket. At the completion of this operation the device will contain a copy of the data in the programmer RAM. The device is now "programmed."

When the VERIFY command is used, the programmer makes a byte-by-byte comparison of the data in RAM with the data in a programmed device or the data input from the serial port. In a verify operation, data in two mediums is compared, rather than transferred.

3.5.2 GENERALIZED KEY SEQUENCE SYNTAX

The generalized key sequence syntax for the COPY and VERIFY commands is:

[function] [source] XXXX/YYYY [destination] ZZZZ
[START]

In the sequence format above function is either copy or verify, and both source and destination can be either RAM, device, or port. However, it should be noted that either the source or the destination must be RAM in any sequence. It is not possible to go directly between the device and the port in either direction. The data must always pass through RAM. The character strings XXXX, YYYY, and ZZZZ represent parameters associated with the source/destination keys. XXXX is the beginning source address. YYYY is the source block size and ZZZZ is the beginning destination address. Table 3-2 provides the usage, definition, and default values for these parameters. If the default values are correct, it is not necessary to enter them, just press [START] in the key sequence following the function (copy or verify). If a mistake is made while entering one of the parameters, press the [REVIEW] key to erase the entry for that parameter so the correct value can be re-entered.

Table 3-2 Address Parameters

Key	Source	Parameter ^(a)		Name	Default Value	Definition
		Dest.				
Device	XXXX	ZZZZ		Begin Device Address	0 ^(b)	First device address from which data is output or to which data is input.
RAM	XXXX	ZZZZ		Begin RAM Address	0 ^(b)	First data RAM address from which data is output or to which data is input.
Port	XXXX ^(c)	ZZZZ ^(c)		Address Offset	Output: ^(d) 0 Input: ^(d) First Address	The address offset is subtracted from all port addresses for input and added to all RAM addresses for output.
Device, RAM, or Port	YYYY	N/A		Block Size	^{(b)(e)}	The number of bytes to be transferred.

Notes:

- (a) When defaults are in effect, ADDR is displayed for address parameters and SIZE is displayed for Block Size.
- (b) To return to the default value, press REVIEW or enter 0.
- (c) Six or eight digits for 16-bit translation formats.
- (d) To return to the default condition enter FFFF (up to 8 characters).
- (e) The default value is the device size in device-related operations. In port-related operations it is from the first RAM address specified to the end of RAM.

3.6 FRONT PANEL KEYBOARD DATA TRANSFER OPERATIONS

The following procedures describe how to perform data transfer operations from the 29B front panel keyboard.

3.6.1 LOAD RAM WITH MASTER DEVICE DATA

Use the following procedure to load the 29B RAM with data from a master device using the programmer front panel:



to select the mode.

29B Displays

COPY DATA FROM



to select the source of the data.

29B Displays

DEV ^ ADDR / SIZE TO

The prompt (^) preceding "ADDR" in the display means that you can change the begin device address to any address within the range of device word limit by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered begin address.

- If the hex value that appears in place of "ADDR" is correct, go to step 3.
- If it is incorrect, enter the correct hex value.
- If you do not wish to change the begin device address or block size (step 3), go to step 4.



29B Displays

DEV ADDR ^ SIZE TO

The prompt (^) preceding "SIZE" means that you can change the block size by entering the hex value with

the keyboard. If an entry is not made, the value defaults to the device size. If a hex value appears in place of "SIZE," it is from a previously entered block size.

NOTE

If a begin device address is specified then the size must be specified also. The size must be equal to or less than the device size minus the begin device address.

- If the hex value that appears in place of "SIZE" is correct, go to step 4.
- If it is incorrect, enter the correct hex value, then go to step 4.



to select the destination for the data.

29B Displays

CO DEV ^ RAM ^ ADDR

The prompt (^) preceding "ADDR" means that you can change the begin RAM address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered RAM begin address.

- If the hex value that appears in place of "ADDR" is correct, go to step 5.
- If it is incorrect, enter the correct hex value, then go to step 5.

NOTE

If the Pak or programming module installed does not require a family/pinout code combination the 29B automatically skips the next step.



29B Displays

FAM ^ 00 PIN 00

Enter the four digit hex family/pinout code combination for the device to be copied. The codes are listed in manual for the Pak or programming module installed. If a code combination other than

zeroes appears on the display, it is from a previously entered code combination.

- a. If the hex value that appears in place of the family/pinout combination is correct, go to step 6.
- b. If it is incorrect, enter the correct family/pinout code hex value, then go to step 6.

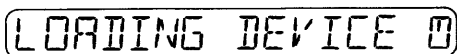
NOTE

If the Pak installed has more than one socket, the LED next to the correct socket will illuminate.

6. Insert and lock the master device into the appropriate socket. (See subsection 3.4.2).



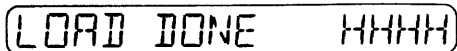
29B Displays



NOTE

The amount of time the programmer will require to perform this operation will vary depending on the device size. The action symbol character rotates while the operation is taking place.

8. When the operation is complete, the following display signals the programmer's readiness. If an error code is displayed, see appendix E.



NOTE

HHHH is the hex sumcheck of all the device data. If block limits are set the sumcheck (HHHH) will be calculated only for the block size.

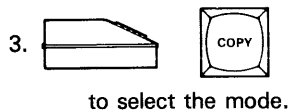
9. Remove the master device from the socket (see subsection 3.4.3).
10. To repeat the load operation from another device with the same family and pinout codes return to step 7.

3.6.2 LOAD RAM FROM SERIAL PORT

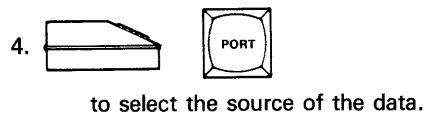
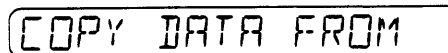
Use the following procedure to load the 29B RAM from front panel keyboard with incoming serial port data:

1. Set-up the serial port. Refer to section 2.

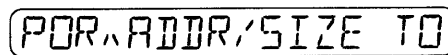
2. Select the appropriate data translation format from appendix A and execute the format select function B3 (see subsection 3.8).



29B Displays



29B Displays

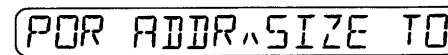


The prompt (^) preceding "ADDR" in the display means that you can change the begin address offset to any address within the range of the data format by entering the hex value with the front panel keyboard. If an entry is not made, the value defaults to the first incoming address. If a hex value appears in place of "ADDR," it is from a previously entered begin address. Enter FFFF (up to 8 characters) to reset the default.

- a. If the hex value that appears in place of "ADDR" is correct, go to step 5.
- b. If it is incorrect, enter the correct hex value.
- c. If you do not wish to change the begin device address or block size (step 5), go to step 6.



29B Displays



The prompt (^) preceding "SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "SIZE," it is from a previously entered block size.

- a. If the hex value that appears in place of "SIZE" is correct, go to step 6.

- b. If it is incorrect, enter the correct hex value, then go to step 6.



to select the destination for the data.

29B Displays

CO POR: RAM ^ ADDR

The prompt (^) preceding "ADDR" means that you can change the begin RAM address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered RAM begin address.


- If the hex value that appears in place of "ADDR" is correct, go to step 7.
- If it is incorrect, enter the correct hex value, then go to step 7.



29B Displays

INPUT PORT 0

NOTE

The amount of time the programmer will require to perform this operation will vary depending on the data format, block size, and baud rate. The action symbol  character rotates while the operation is taking place.

8. When the operation is complete the following display signals the programmer's readiness. If an error code is displayed see appendix E.

INPUT DONE HHHH

NOTE

HHHH is the hex sumcheck of all the received data.

9. To repeat the load operation, return to step 7.

3.6.3 PROGRAM DEVICE WITH RAM DATA

When programming a device, the system automatically performs illegal bit tests and blank checks at nominal V_{CC} to verify the ability of the PROM to accept programming before it begins the operation. Use the following procedure to program a blank device with data from RAM using the programmer front panel keyboard:



to select the mode.

29B Displays

COPY DATA FROM



to select the source of the data.

29B Displays

RAM ^ ADDR / SIZE TO

The prompt (^) preceding "ADDR" in the display means that you can change the begin RAM address to any address within the range of RAM word limit by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered begin address.

- If the hex value that appears in place of "ADDR" is correct, go to step 3.
- If it is incorrect, enter the correct hex value.
- If you do not wish to change the begin device address or block size (step 3), go to step 4.



29B Displays

RAM ADDR ^ SIZE TO

The prompt (^) preceding "SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to the device size. If a hex value appears in place of "SIZE," it is from a previously entered block size.

- a. If the hex value that appears in place of "SIZE" is correct, go to step 4.
- b. If it is incorrect, enter the correct hex value, then go to step 4.



to select the destination for the data.

29B Displays

CO RAM; DEV ^ ADDR

The prompt (^) preceding "ADDR" means that you can change the begin device address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered device begin address.

- a. If the hex value that appears in place of "ADDR" is correct, go to step 5.
- b. If it is incorrect, enter the correct hex value, then go to step 5.

NOTE

If the Pak or programming module installed does not require a family/pinout code combination the 29B automatically skips the next step.



29B Displays

FAM ^ 00 PIN 00

Enter the four digit hex family/pinout code combination for the device to be programmed. The codes are listed in the manual for the Pak or programming module installed. If a code combination other than zeroes appears on the display, it is from a previously entered family and pinout code.

- a. If the hex value that appears in place of the family/pinout combination is correct, go to step 6.
- b. If it is incorrect enter the correct family/pinout code hex value, then go to step 6.

NOTE

If the Pak installed has more than one socket, the LED next to the correct socket will illuminate.

6. Insert and lock the blank device into the appropriate socket. (See subsection 3.4.2).



29B Displays

TEST DEVICE 0

If the device passes the blank check and illegal bit test the 29B automatically begins to program the device.

29B Displays

PROGRAM DEVICE 0

When the device has been successfully programmed the 29B automatically verifies the programmed device.

29B Displays

VERIFY DEVICE 0

NOTE

The amount of time the programmer will require to perform these operations will vary depending on the device size, technology, and programming algorithm. The action symbol character □ rotates while the operation is taking place.

8. When the operation is complete the following display signals the programmer's readiness. If an error code is displayed, see appendix E.

29B Displays

PRG DONE 01 HHHH

NOTE

The number (01) following PRG DONE (program done) is the sequence number. The sequence number increments by 1 for each device programmed. HHHH is the hex sumcheck of all the device data. If block is defined, the sumcheck (HHHH) will be calculated only for the block size.

9. Remove the device from the socket.
10. To program additional identical devices using the data stored in RAM return to step 6.

3.6.4 OUTPUT TO SERIAL PORT

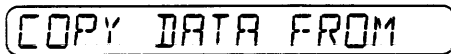
Use the following front panel keyboard procedure to output data to the serial port:

1. Set-up the serial port. Refer to section 2.
2. Select the appropriate data translation format from appendix A and execute the format select function B3 (see section 3.8).



to select the mode.

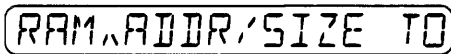
29B Displays





to select the source of the data.

29B Displays

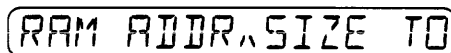


The prompt (^) preceding "ADDR" in the display means that you can change the begin RAM address to any address within the range of RAM word limit by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered begin address.

- a. If the hex value that appears in place of "ADDR" is correct, go to step 5.
- b. If it is incorrect, enter the correct hex value.
- c. If you do not wish to change the begin device address or block size (step 5), go to step 6.



29B Displays



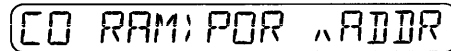
The prompt (^) preceding "SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to RAM size minus the begin RAM address. If a hex value appears in place of "SIZE," it is from a previously entered block size.

- a. If the hex value that appears in place of "SIZE" is correct, go to step 6.
- b. If it is incorrect, enter the correct hex value, then go to step 6.



to select the destination for the data.

29B Displays

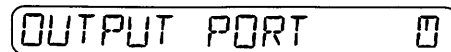


The prompt (^) preceding "ADDR" means that you can change the address offset (see the glossary in appendix D) by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered address offset.


- a. If the hex value that appears in place of "ADDR" is correct, go to step 7.
- b. If it is incorrect, enter the correct hex value, then go to step 7.



29B Displays



NOTE

The amount of time the programmer will require to perform this operation will vary depending on the data format, block size, and baud rate. The action symbol  character rotates while the operation is taking place.

8. When the operation is complete the following display signals the programmer's readiness. If an error code is displayed, see appendix E.

OUTPUT DONE HHHH

NOTE

HHHH is the hex sumcheck of all the transmitted data.

9. To repeat the output operation, return to step 7.

3.6.5 BLOCK MOVE

A block move copies data in one block of RAM locations to another block of RAM locations, beginning at a defined address (see figure 3-3). Use the following front panel keyboard procedure to copy data from one location in RAM to another location in RAM:



selects the mode.

29B Displays

COPY DATA FROM



to select the source of the data.

29B Displays

RAM ^ ADDR / SIZE TO

The prompt (^) preceding "ADDR" in the display means that you can change the begin RAM address to any address within the range of RAM word limit by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered begin address.

- a. If the hex value that appears in place of "ADDR" is correct, go to step 3.
- b. If it is incorrect, enter the correct hex value.
- c. If you do not wish to change the begin device address or block size (step 3), go to step 4.



29B Displays

RAM ADDR ^ SIZE TO

The prompt (^) preceding "SIZE" means that you can change the block size by entering the hex value with the keyboard. An entry must be made or data can not be moved. If a hex value appears in place of "SIZE," it is from a previously entered block size.

- a. If the hex value that appears in place of "SIZE" is correct, go to step 4.
- b. If it is incorrect, enter the correct hex value, then go to step 4.

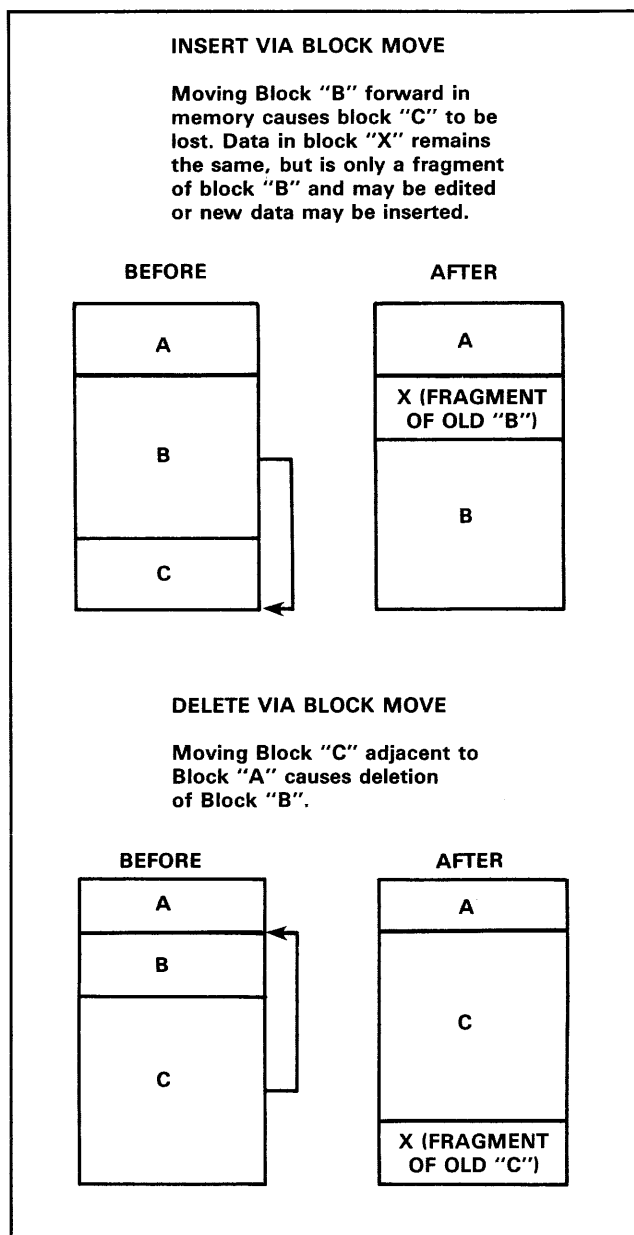
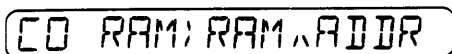


Figure 3-4. Block Move



to select the destination for the data.

29B Displays

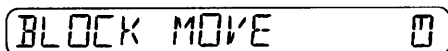


The prompt (^) preceding "ADDR" means that you can change the destination address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered destination address.


- If the hex value that appears in place of "ADDR" is correct, go to step 5.
- If it is incorrect, enter the correct hex value, then go to step 5.



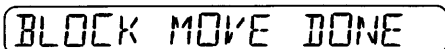
29B Displays



NOTE

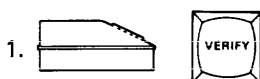
The amount of time the programmer will require to perform this operation will vary depending on the block size. The action symbol character  rotates while the operation is taking place.

- When the operation is complete the following display signals the programmer's readiness. If an error code is displayed, see appendix E.



3.6.6 VERIFY RAM DATA AGAINST MASTER DEVICE DATA

Use the following front panel keyboard procedure to verify that the data in the 29B RAM is the same as the data in the master device.



to select the mode.

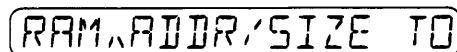
29B Displays





to select the source of the data.

29B Displays

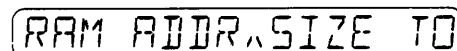


The prompt (^) preceding "ADDR" in the display means that you can change the begin RAM address to any address within the range of the RAM word limit by entering the address hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered begin address.

- If the hex value that appears in place of "SIZE" is correct, go to step 3.
- If it is incorrect, enter the correct hex value, then go to step 3.



29B Displays



The prompt (^) preceding "SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to the device size. If a hex value appears in place of "SIZE," it is from a previously entered block size.

NOTE

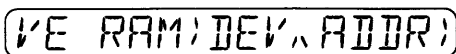
If a begin device address is specified, then the size must be specified also. The size must be equal to or less than the device size minus the begin device address.

- If the hex value that appears in place of "SIZE" is correct, go to step 4.
- If it is incorrect, enter the correct hex value, then go to step 4.



to select the destination for the data.

29B Displays



The prompt (^) preceding "ADDR" means that you can change the begin device address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered device begin address.

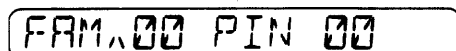
- If the hex value that appears in place of "ADDR" is correct, go to step 5.
- If it is incorrect, enter the correct hex value, then go to step 5.

NOTE

If the Pak or programming module installed does not require a family/pinout code combination, the 29B automatically skips the next step.



29B Displays



Enter the four digit hex family/pinout code combination for the device to be verified. The codes are listed in the manual for the Pak module installed. If a code combination other than zeroes appears on the display, it is from a previously entered code combination.

- If the hex value that appears in place of the family/pinout combination is correct, go to step 6.
- If it is incorrect, enter the correct family/pinout code hex value, then go to step 6.

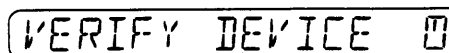
NOTE

If the Pak installed has more than one socket, the LED next to the correct socket will illuminate.

- Insert and lock the device into the appropriate socket. (See subsection 3.4.2).



29B Displays:

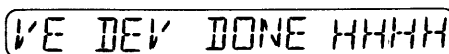


The 29B will perform a two-pass verify.

NOTE

The amount of time the programmer will require to perform this operation will vary depending on the device size. The action symbol line rotates 360 degrees while the operation is taking place.

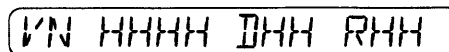
- When the operation is complete the following display signals the programmer's readiness.



NOTE

HHHH is the hex sumcheck of all the device data. If block size is defined the sumcheck (HHHH) will be calculated only for the block size.

- If the device does not verify the following error message format will appear:



NOTE

The N in VN is the number of the verify pass, either 1 or 2. HHHH is the address where the verify error was found. DHH is the device hex data and RHH is the RAM hex data.

If a verify error is displayed, press the START key to resume verifying at the next address. Verification will resume until all addresses have been verified or another error is found. Press the START key as many times as is required to locate all of the addresses where there is a verify error.

NOTE

The LogicPak™ verify error message is different. Refer to the LogicPak™ manual for the error message format.

- Remove the device from the socket.
- To repeat the verify operation, return to step 6.

3.6.7 VERIFY RAM FROM SERIAL PORT

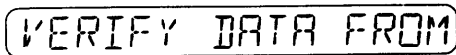
Use the following front panel keyboard procedure to verify that the RAM data is the same as the incoming serial port data:

1. Set-up the serial port. Refer to section 2.
2. Select the appropriate data translation format from appendix A and execute the format select function B3 (see subsection 3.8).



to select the mode.

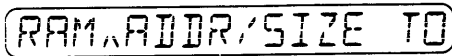
29B Displays





to select the source of the data.

29B Displays

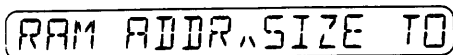


The prompt (Λ) preceding "ADDR" in the display means that you can change the begin RAM address to any address within the range of RAM word limit by entering the address hex value with the keyboard. If an entry is not made, the value defaults to the first incoming address. If a hex value appears in place of "ADDR," it is from a previously entered begin address.

- a. If the hex value that appears in place of "ADDR" is correct, go to step 5.
- b. If it is incorrect, enter the correct hex value.
- c. If you do not wish to change the begin device address or block size (step 5), go to step 6.



29B Displays



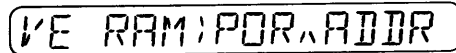
The prompt (Λ) preceding "SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to RAM size minus the begin RAM address. If a hex value appears in place of "SIZE," it is from a previously entered block size.

- a. If the hex value that appears in place of "SIZE" is correct, go to step 6.
- b. If it is incorrect, enter the correct hex value, then go to step 6.



to select the destination for the data.

29B Displays



The prompt (Λ) preceding "ADDR" means that you can change the address offset by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered address offset.

- a. If the hex value that appears in place of "ADDR" is correct, go to step 7.
- b. If it is incorrect, enter the correct hex value, then go to step 7.



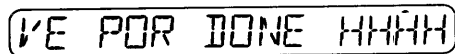
29B Displays



NOTE

The amount of time the programmer will require to perform this operation will vary depending on the data format, block size, and baud rate. The action symbol character rotates while the operation is taking place.

8. When the operation is complete, the following display signals the programmer's readiness. If an error code is displayed, see appendix E.



NOTE

HHHH is the hex sumcheck of all the device data. If block limits are set the sumcheck (HHHH) will be calculated only for the block size.

9. To repeat the verify operation, return to step 7.

3.7 KEYBOARD EDIT OPERATIONS

The EDIT key allows the operator to change the stored data at a specified RAM address. The data may be presented in either binary, octal, or hexadecimal, as determined by a select code (F5, F6, or F7 respectively). The programmer automatically comes up in hexadecimal, which is the default condition. However, if either octal or binary is desired, it may be selected using the applicable select code.

If edit addresses are outside the range of the device addresses two asterisks (**) are displayed for the device data.

Since the display varies when an edit operation is performed in the three numbering systems an example of each is given below.

NOTE

The symbols H, Q, or B are used in subsections 3.7.1 through 3.7.3 to denote hexadecimal, octal, or binary numeric characters. Therefore, an "H" symbol on a key means the operator can press any key on the numeric key pad. A "B" symbol implies only 0 or 1 can be pressed, since binary values are composed of only 0's and 1's.

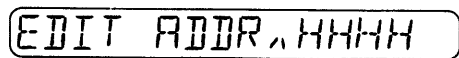
3.7.1 HEXADECIMAL BASE

Use the following procedure to edit data in a RAM address when the hexadecimal base has been selected, either by select code F7 or default:



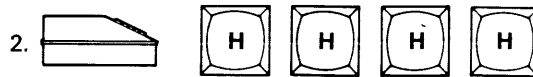
1. to select the operating mode.

29B Displays



The prompt (^) after "ADDR" in the display means that you can change the RAM address to be edited by entering any hex address within the range of the RAM. The "HHHH" in the display represents the last hex address edited (power-up default is 0000).

- a. If the hex value address displayed is correct go to step 3.
- b. If it is not correct go to step 2.



2. hex address to be edited.

29B Displays

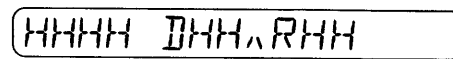


entered hex address.



3.

29B Displays



The "HHHH" in the display represents the hex address in RAM to be edited. The device data is represented by the "DHH" symbols; The "D" representing the device and the "HH" the hex data in the device at the address defined. The RAM data to be edited is represented by the symbols " ^ RHH." The prompt symbol " ^ " to show that this data can be edited, the "R" to show it is in RAM, and "HH" to show the hex value of the data presently entered in that RAM address.

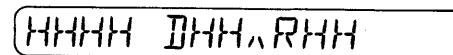
NOTE

When the 29B initially powers up, device data is undefined since the device family/pinout code is also undefined. Therefore; under these conditions the edit command will show "D**". After a device type has been defined (usually as the result of a copy or verify device operation) device data can be compared with RAM data; however, device data is only shown for the block of data beginning at the begin RAM address and with a length equal to the device size minus the begin device address.



4. hex data to be entered in RAM address.

29B Displays



new data is displayed following the ^ R.

5. The procedure may be continued in any of the following ways:

- Press START to increment (+1) the address to edit the next higher RAM location and then resume at either step 4 or 5, as applicable.
- Press REVIEW to decrement (-1) the address to edit the next lower RAM location and then resume at either step 4 or 5, as applicable.
- Press EDIT to select another RAM address for editing and then resume at either step 2 or 3, as applicable.

6. Press any other function (blue) key (COPY, VERIFY, SELECT) to exit from the EDIT mode of operations.

3.7.2 OCTAL BASE

Use the following procedure to edit data in a RAM address when the octal base has been selected (select code F6):



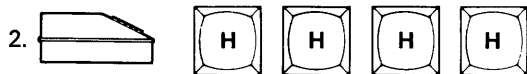
to select the operating mode.

29B Displays

EDIT ADDR ^ HHHH

The prompt (^) following "ADDR" in the display means that you can change the RAM address to be edited by entering any hex address within the range of the RAM. The "HHHH" in the display represents the last hex address edited (power-up default is 0000).

- If the hex value address displayed is correct go to step 3.
- If it is not correct go to step 2.



hex address to be edited.

29B Displays

EDIT ADDR ^ HHHH

entered hex address.



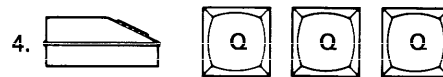
29B Displays

HHHH DQQQ ^ RQQQ

The "HHHH" in the display represents the hex address in RAM to be edited. The defined device data is represented by the "DQQQ" symbols; The "D" representing the device and the "QQQ" the octal data in the device at the address defined. The RAM data to be edited is represented by the symbols "^ RQQQ." The prompt symbol "^" to show which data can be edited, the "R" to show it is in RAM, and "QQQ" to show the octal value of the data presently residing in that RAM address.

NOTE

*When the 29B initially powers up, device data is undefined since the device family/pinout code is also undefined. Therefore, under these conditions the edit command will show "D***." After a device type has been defined (usually as the result of a copy or verify device operation) device data can be compared with RAM data; however, device data is only shown for the block of data beginning at the begin RAM address and with a length equal to the device size minus the begin device address.*



octal data to be entered in RAM address.

29B Displays:

HHHH DQQQ ^ RQQQ

new data is displayed following the ^ R.

5. The procedure may be continued in any of the following ways:

- Press START to increment (+1) the address to edit the next higher RAM location and then resume at either step 4 or 5, as applicable.
- Press REVIEW to decrement (-1) the address to edit the next lower RAM location and then resume at either step 4 or 5, as applicable.
- Press EDIT to select another RAM address for editing and then resume at either step 2 or 3, as applicable.

6. Press any other function (blue) key (COPY, VERIFY, SELECT) to exit from the EDIT mode of operations.

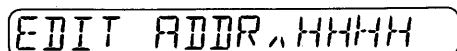
3.7.3 BINARY BASE

Use the following procedure to edit data in a RAM address when the binary base has been selected (select code F5):



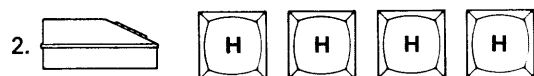
to select the operating mode.

29B Displays



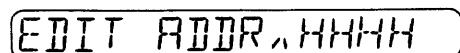
The prompt (^) following "ADDR" in the display means that you can change the RAM address to be edited by entering any hex address within the range of the RAM. The "HHHH" in the display represents the last hex address edited (power-up default is 0000).

- a. If the hex value address displayed is correct, go to step 3.
- b. If it is not correct, go to step 2.



hex address to be edited.

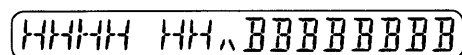
29B Displays:



entered hex address.



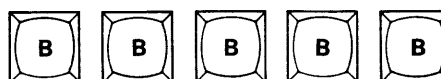
29B Displays



The "HHHH" in the display represents the hex address in RAM to be edited. The defined device data is represented by the "HH" symbols; representing the hex data in the device at the address defined. The RAM data to be edited is represented by the symbols " ^BBBBBBBB." The prompt symbol " ^" to show which data can be edited and "BBBBBBBB" to show the binary value of the data presently entered in that RAM address.

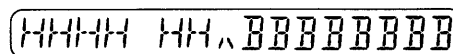
NOTE

When the 29B initially powers up, device data is undefined since the device family/pinout code is also undefined. Therefore; under these conditions the edit command will show "D**." After a device type has been defined (usually as the result of a copy or verify device operation) device data can be compared with RAM data; however, device data is only shown for the block of data beginning at the begin RAM address and with a length equal to the device size, minus the begin device address.



binary data to be entered in RAM address.

29B Displays:



new data is displayed following the " ^".

5. The procedure may be continued in any of the following ways:

- Press START to increment (+ 1) the address to edit the next higher RAM location and then resume at either step 4 or 5, as applicable.
- Press REVIEW to decrement (- 1) the address to edit the next lower RAM location and then resume at either step 4 or 5, as applicable.
- Press EDIT to select another RAM address for editing and then resume at either step 2 or 3, as applicable.

6. Press any other function (blue) key (COPY, VERIFY, SELECT) to exit from the EDIT mode of operations.

3.8 SELECT KEY FUNCTIONS

The SELECT key allows the operator to perform additional operations, called select functions, that are used for:

- RAM data manipulations
- Utility and inquiry commands
- Serial I/O commands

3.8.1 ACCESSING SELECT FUNCTIONS

The select functions may be accessed by either direct entry, stepping, or scrolling.

For direct entry, press SELECT. The 29B will display "SELECT CODE." Enter the hex code for the desired function or data translation format and then press START. (Some functions, e.g. those to enter terminal remote control, require the START key to be pressed twice.) The display will prompt any additional entries required, or indicate an invalid entry.

To access the select functions by stepping, depress the SELECT key repeatedly until the function desired is displayed. Then press the START key to initiate the operation. Press the REVIEW key to step backwards through the select functions. The functions are displayed in hexadecimal order.

To access the select functions by scrolling, depress SELECT and then START. Each function is momentarily displayed in turn. When the desired function is displayed press any key to stop the scrolling. To back up, press REVIEW. Once the desired select function is displayed, initiate the operation by pressing START.

The programmer signals that the operation is complete by displaying two asterisks (**) in the last two display positions.

3.8.2 EXTENDED SELECT FUNCTIONS

Some Paks feature specific operations, resident in their Pak software, called extended select functions. These functions generally are of a utility nature and are documented in their respective Pak manuals. Use the applicable Pak manual for specific step-by-step instructions.

3.8.3 DESCRIPTIONS AND KEY SEQUENCES

Following are the select functions, their descriptions, and key sequences. The key sequences are for direct entry. When stepping or scrolling simply skip the steps preceding the display of the desired select function and enter the operation at that point (normally step 3 of the procedure). The command descriptions and key sequences are arranged in hexadecimal order within the three major headings, i.e., RAM data manipulation (3.8.4), utility and inquiry (3.8.5), and serial I/O (3.8.6). Table 3-3 is a summary of the select commands, in hexadecimal order.

Table 3-3. Select Commands

HEX CODE	TITLE	COMMAND GROUP
01-99	Format Codes	Format Codes
A1	Swap Nibbles	RAM Data Manipulation
A2	Fill RAM	RAM Data Manipulation
A3	Invert RAM	RAM Data Manipulation
A4	Clear All RAM	RAM Data Manipulation
A5	Split RAM	RAM Data Manipulation
A6	Shuffle RAM	RAM Data Manipulation
B0	Device Size	Utility and Inquiry
B1	SumCheck RAM	Utility and Inquiry
B2	System Config	Utility and Inquiry
B3	Format Number	Utility and Inquiry
B4	Nonblank Fail	Utility and Inquiry
B9	Display Test	Utility and Inquiry
C1	Calibration	Utility and Inquiry
D7	Leader Output	Serial I/O
D8	Size Record	Serial I/O
D9	Null Count	Serial I/O
F0	Program Count	Utility and Inquiry
F1	Remote Mode CRC	Utility and Inquiry
F2	29B TRC	Utility and Inquiry
F3	Lock Data On	Utility and Inquiry
F4	Nibble Mode	Utility and Inquiry
F5	Binary Base	Utility and Inquiry
F6	Octal Base	Utility and Inquiry
F7	Hex Base	Utility and Inquiry
F8	Byte/Nib Mode	Utility and Inquiry
F9	Timeout Off	Serial I/O
FA	Char Output	Serial I/O
FB	Enable Port	Serial I/O
FC	Remote On Off	Serial I/O

3.8.4 RAM DATA MANIPULATION COMMANDS:

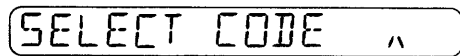
A1 SWAP NIBBLES Command

Use the following procedure to exchange high- and low-order nibbles of every byte:



to select the select function.

29B Displays

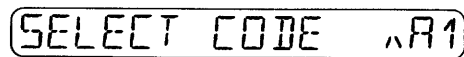


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

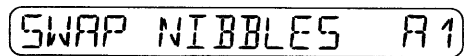


to select the swap nibbles function.

29B Displays

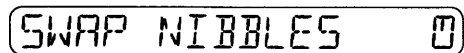


29B Displays

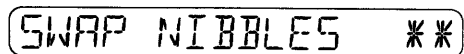


to exchange high and low order nibbles of every byte.

29B Displays



Action symbol rotates.



Swap nibble function is complete.

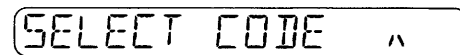
A2 FILL RAM Command

Use the following procedure to fill RAM from the last EDIT address to the end of RAM with variable hex data. The default data value is 00. If select function F4 has been specified, or if a 4-bit device is selected, it will fill only the lower order nibbles of RAM; otherwise it will default to the word width of the selected device.



to select the select function.

29B Displays

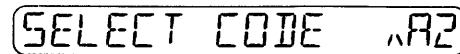


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

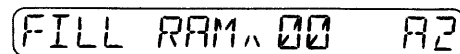


to select the fill RAM function.

29B Displays

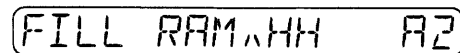


29B Displays



Hex data to be placed in RAM.

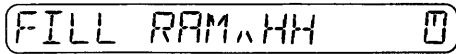
29B Displays



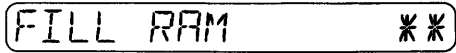


to write the entered hex data in RAM.

29B Displays



Action symbol rotates.

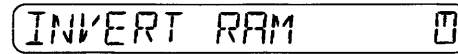


Fill RAM function is complete.

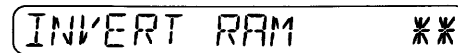


to perform a ones complement on data stored in RAM.

29B Displays



Action symbol rotates.



Invert RAM function is complete.

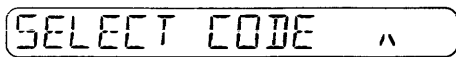
A3 INVERT RAM Command

Use the following procedure to perform the ones complement of 4 or 8 bits of each word, as determined by the word size in effect:



to select the select function.

29B Displays

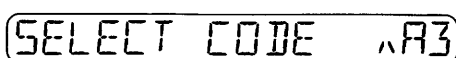


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

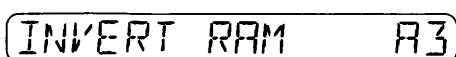


to select the invert RAM function.

29B Displays



29B Displays



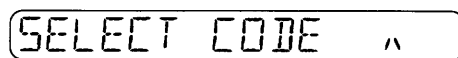
A4 CLEAR ALL RAM Command

Use the following procedure to clear all RAM to zeros:



to select the select function.

29B Displays

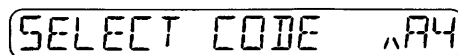


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

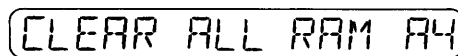


to select the clear all RAM function.

29B Displays



29B Displays



4.   to clear all RAM to zeros.

29B Displays

CLEAR ALL RAM 0



action symbol rotates.

CLEAR ALL RAM **

clear all RAM function is complete.

A5 SPLIT RAM Command




Use the following procedure to split odd- and even-addressed bytes in RAM about a center point, dividing them into two adjacent blocks occupying the same original amount of RAM. The center point must be a power of two between 0 and the RAM midpoint. The default center point is the RAM total midpoint.

1.   to select the select function.

29B Displays

SELECT CODE ^

The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

2.    to select the split RAM function.






29B Displays

SELECT CODE ^AS

3.   to split the RAM around the designated midpoint.


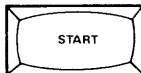
29B Displays

SPLIT RAM ^HHHH

4.      sets the hex midpoint. (Step can be skipped if the total RAM midpoint is correct.)

29B Displays

SPLIT RAM HHHH

5.   to split the RAM around the designated midpoint.

29B Displays

SPLIT RAM 0

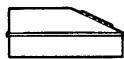

action symbol rotates.

SPLIT RAM **

split RAM function is complete.

A6 SHUFFLE RAM Command

Use the following procedure to shuffle the block of RAM addresses immediately above the center point with the block below, placing the lower-block bytes at even-numbered addresses starting with 0 and the upper-block addresses at odd-numbered addresses starting with 1. The center point must be a power of two between 0 and the RAM midpoint. The default center point is the total RAM midpoint.

1.   to select the select function.

29B Displays

SELECT CODE ^

The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

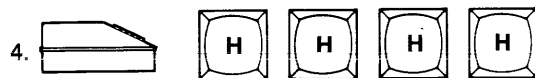
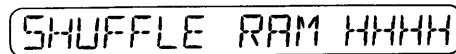


to select the shuffle RAM function.

29B Displays

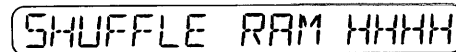


29B Displays



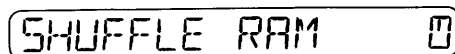
to set the hex midpoint. (Step can be skipped if the total RAM midpoint is correct.)

29B Displays

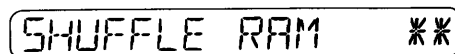


to shuffle the RAM around the designated midpoint.

29B Displays



Action symbol rotates.



Shuffle RAM function is complete.

3.8.5 UTILITY AND INQUIRY COMMANDS

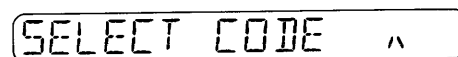
B0 DEVICE SIZE Command

Use the following procedure to display the device size and word width currently entered. The default values vary with the Pak or programming module in use.



to select the select function.

29B Displays

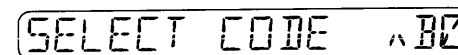


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.



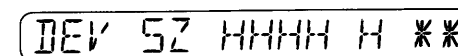
to select the device size function.

29B Displays



to display in hex the current or default device size and word size.

29B Displays



HHHH represents the hex value of the device size and H represents the hex value of the word size.

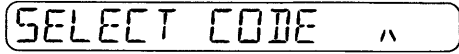
B1 SUMCHECK RAM Command

Use the following procedure to display the total RAM sumcheck:

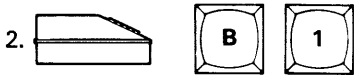


to select the select function.

29B Displays

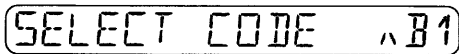


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.



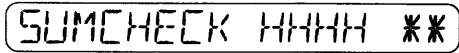
to select the RAM sumcheck function.

29B Displays



to display the RAM sumcheck in hex.

29B Displays



HHHH represents the hex value of the total RAM sumcheck.

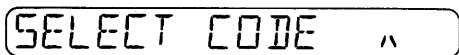
B2 SYSTEM CONFIG Command

Use the following procedure to display the four character system configuration code:



to select the select function.

29B Displays

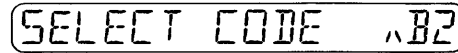


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

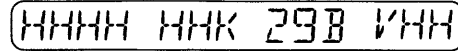


to select the system configuration function.

29B Displays



29B Displays



to display the four character system configuration code (HHHH), RAM size (HHK), Model (29B), and version number (VHH).

B3 FORMAT NUMBER Command

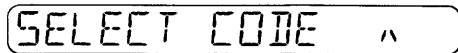
The I/O format can be selected in either of three ways. If the desired format number is known the select mode can be accessed and the code entered directly. The indirect methods allows the operator to display the current or default entry and then change the entry by either stepping through the formats, or entering the format code. Refer to Appendix A to determine which instrument control code and format code is desired.

Use the following procedure to enter a new data translation format number directly:

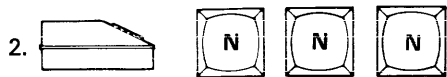


to select the select function.

29B Displays

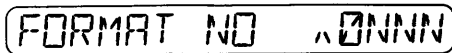


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

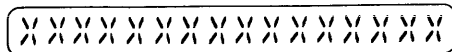


to select the format number; where the first number is the optional instrument control code (default is zero if not entered) and the second and third numbers are the format code.

29B Displays



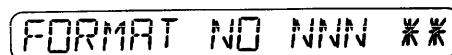
29B Displays



to display the mnemonic for the selected format.



29B Displays



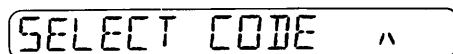
the format number selection is complete.

Use the following procedure to display and change the current data translation format number:



to select the select function.

29B Displays

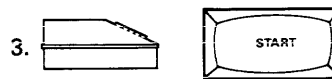
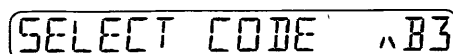


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

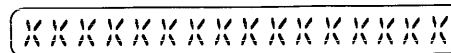


to select the format number function.

29B Displays

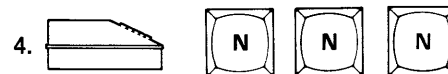


29B Displays



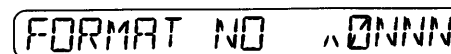
to display the mnemonic for the current (or default) format number.

If the displayed format mnemonic is the desired format go to step 6. If it is not, go to step 4.

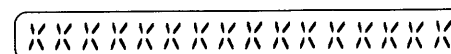


to select the format number; where the first number is the optional instrument control code (default is zero if not entered) and the second and third numbers are the format code.

29B Displays



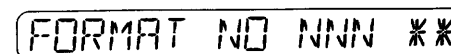
29B Displays



to display the mnemonic for the selected format.



29B Displays



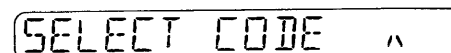
the format number selection is complete.

Use the following procedure to step through the display to change the data translation format number:



to select the select function.

29B Displays

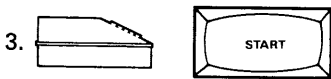
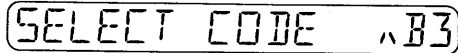


The prompt (Λ) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

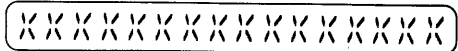


to select the format number function.

29B Displays



29B Displays



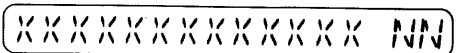
to display the mnemonic for the current (or default) format number.

If the displayed format mnemonic is the desired format go to step 5. If it is not use go to step 4.

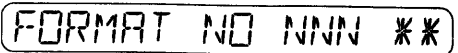


Press and release until the desired format appears on the keyboard display. Press REVIEW to step the display backwards.

29B Displays



29B Displays



the format number selection is complete.

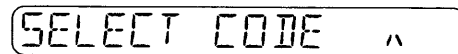
B4 NON BLANK FAIL Command

Use the following procedure to automatically fail non-blank devices when the 29B is equipped with the optional handler. If the optional handler is not connected the command has no effect on the system. When power is initially applied to the 29B with the optional handler installed the default condition is to attempt to program any non-blank devices detected.



to select the select function.

29B Displays

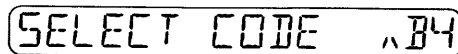


The prompt (Λ) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

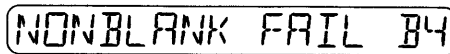


to select the non blank fail function.

29B Displays

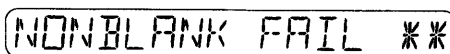


29B Displays



to place the non-blank fail function in effect if the optional handler is connected.

29B Displays



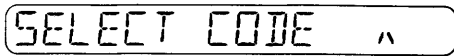
B9 DISPLAY TEST Command

Use the following procedure to perform the display test (all 14 segments of the 16 indicators illuminate for approximately four seconds):



to select the select function.

29B Displays



The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

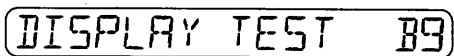


to select the display test function.

29B Displays



29B Displays



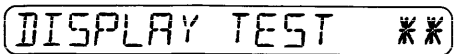
to illuminate all display segments for 4 seconds.

29B Displays



for 4 seconds.

29B Displays



after 4 seconds.

C1 CALIBRATION Command

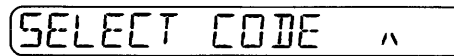
The calibration command response differs between programming Paks and programming modules. In addition some Paks will prompt for additional features.

Use the following procedure to put the programmer in the calibration mode (inhibited in remote operations):



to select the select function.

29B Displays

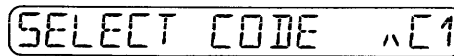


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

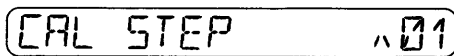


to select the calibration function.

29B Displays




29B Displays



See section 4 and the programming pak or programming module manual for the calibration procedures.

F0 PROGRAM COUNT Command

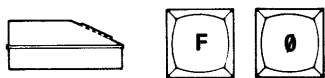
Use the following procedure to display the number of devices programmed since the last power-up or reset:

1.  to select the select function.

29B Displays


SELECT CODE ^

The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

2.  to select the program count function.

29B Displays


SELECT CODE ^F0

3.  to display the decimal contents of the program counter.

29B Displays

PROGRAM COUNT NN

To simply observe the contents of the program counter stop here. To reset the program counter to 00 continue to step 4.


4.  to reset the program counter.

29B Displays

PROGRAM COUNT **

F1 REMOTE MODE Command


Use the following procedure to select the Computer Remote Control (CRC) mode.

1.  to select the select function.

29B Displays


SELECT CODE ^

The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

2.  to select the computer remote control function.


29B Displays

SELECT CODE ^F1

3.  to place the programmer in Computer Remote Control.

29B Displays

REMOTE MODE F1

4.  to place the programmer in Computer Remote Control.

29B Displays

REMOTE MODE 0

Refer to subsection 3.9.1 for further information on Computer Remote Control.

F2 29B TRC (Terminal Remote Control) Command

Use the following procedure to select the optional Terminal Remote Control (TRC):



to select the select function.

29B Displays

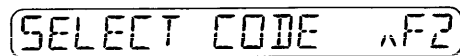


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

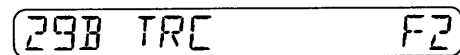


to select the terminal remote control function.

29B Displays

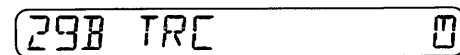


29B Displays



to place the programmer in the optional Terminal Remote Control mode.

29B Displays



Refer to appendix B for further information on Terminal Remote Control.

F3 LOCK DATA ON Command

When the data Lock On function is selected the data in RAM is protected because the keys used to alter data are disabled. This allows an operator to program or verify without accidentally altering data. Only the functions listed below are permitted when the Lock Data On function is selected:

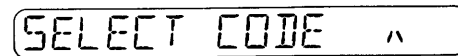
- Copy operations that move data from RAM to the port or device
- Verify operations
- Abort the operation in progress
- Release data lock

Use the following procedure to select the Data Lock On function:



to select the select function.

29B Displays

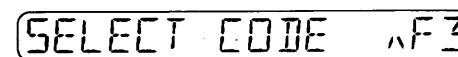


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

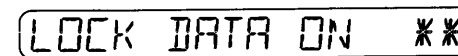


to select the data lock on function.

29B Displays





29B Displays



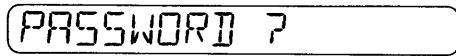
Lock Data On is enabled.



The Lock Data On feature is now enabled. RAM altering operations cannot now be performed until the Data Lock On function is released.

Use the following procedure to release the Data Lock On function:

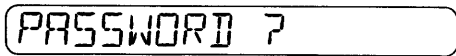
1.   to select the select function.

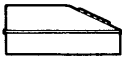

29B Displays



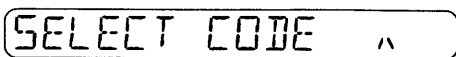
2.   to prepare for release data lock.

29B Displays



3.   to release data lock.



29B Displays



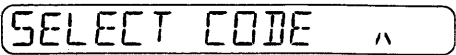
Enter a select code for a select command if desired; otherwise, press either the COPY, VERIFY or EDIT key to begin the next operation. Data Lock On has now been released, and RAM data can be altered.

F4 NIBBLE MODE Command




Use the following procedure to select a 4-bit word size for I/O transfers regardless of the word size defined by default, or the programming module family/pinout. Use the F8 (BYTE/NIB MODE) select command to return to the word size defined by the programming module when the Nibble Mode is no longer desired.

1.   to select the select function.

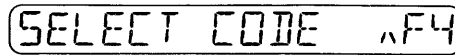
29B Displays



The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

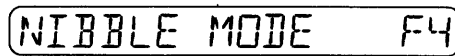
2.    to select the nibble mode function.

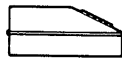

29B Displays



3.   to enable the nibble mode.

29B Displays



4.   to enable the nibble mode.



29B Displays



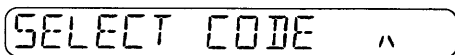
the nibble mode is enabled.

F5 BINARY BASE Command

Use the following procedure to set the number base for Edit operations to binary:

1.   to select the select function.

29B Displays

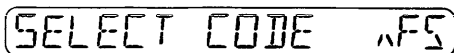


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

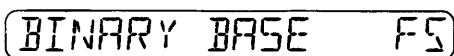


to select the binary base function.

29B Displays

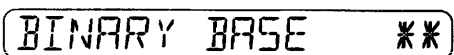


29B Displays



to set the number base to binary for Edit operations.

29B Displays



the binary base selection is complete.

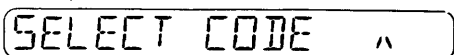
F6 OCTAL BASE Command

Use the following procedure to set the number base for Edit operations to octal:



to select the select function.

29B Displays

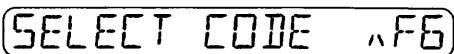


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

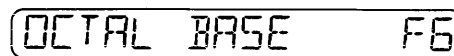


to select the octal base function.

29B Displays

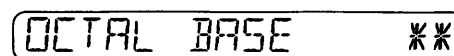


29B Displays



to set the number base to octal for Edit operations.

29B Displays



the octal base selection is complete.

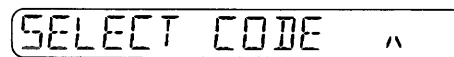
F7 HEX BASE Command

Use the following procedure to set the number base for Edit operations to hexadecimal:



to select the select function.

29B Displays

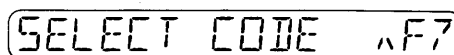


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

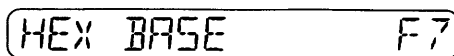


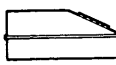
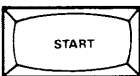
to select the hex base function.

29B Displays

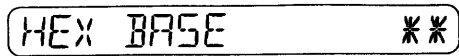


29B Displays



4.   to set the number base to hex for Edit operations.

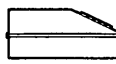

29B Displays



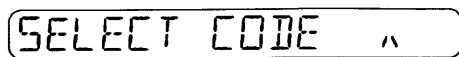
the hex base selection is complete.

F8 BYTE/NIB MODE command




Use the following procedure to nullify select function F4, allowing the word size defined by the programming module to take effect:

1.   to select the select function.

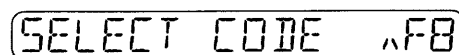
29B Displays



The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

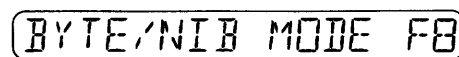
2.    to select the byte/nibble mode function.

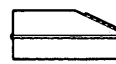

29B Displays



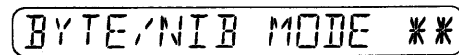
3.   to output 50 nulls from the serial port.

29B Displays



4.   to disable the nibble mode (F4) returning the system to the device word size.



29B Displays



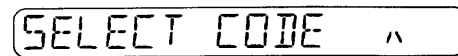
3.8.6 SERIAL I/O OPERATIONS

D7 LEADER OUTPUT Command

Use the following procedure to send 50 nulls from the serial port:

1.   to select the select function.

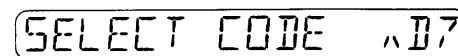
29B Displays



The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

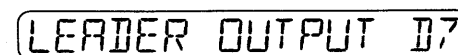
2.    to select the leader output function.


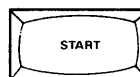
29B Displays



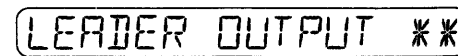
3.   to output 50 nulls from the serial port.

29B Displays



4.   to output 50 nulls from the serial port.

29B Displays



D8 SIZE RECORD Command

Use the following procedure to change the number of bytes per data record on the serial output (the value entered must be in hexadecimal):



to select the select function.

29B Displays



The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

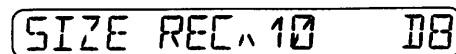


to select the size record function.

29B Displays



29B Displays

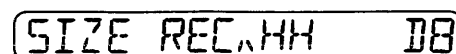


the default entry in the hex field is 10 (16 decimal).



to set the record size. (Step can be skipped if the default or current entry is correct.)

29B Displays



29B Displays



the size record selection is complete.

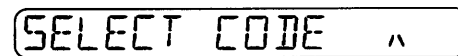
D9 NULL COUNT Command

Use the following procedure to set the null count following each data record or line (carriage return and line feed) on the output. The hex entry selects from 0 (00) to 254 (FE) nulls. An entry of 255 (FF) sends no nulls and suppresses the line feed. The default entry on power-up is 1 (01) null.



to select the select function.

29B Displays

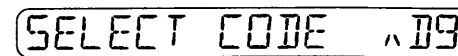


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

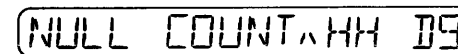


to select the output null count function.

29B Displays



29B Displays





to set the hex null count. (Step can be skipped if the default entry is correct.)

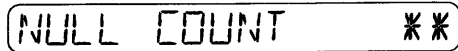
29B Displays





the null count is fixed until a new entry is made.

29B Displays



F9 TIMEOUT OFF Command

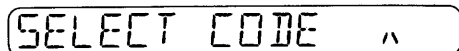
The Timeout Off command is used to disable the 25-second timeout during I/O operations. The 25-second timeout (active unless disabled) is the maximum time the 29B will wait for data when receiving or wait for the receiver of data before sending. Once the timeout has been disabled the only way it can be enabled is to turn the 29B power off, and then reapply power.

Use the following procedure to disable the 25-second timeout feature:



to select the select function.

29B Displays

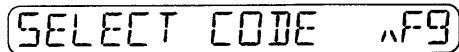


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.



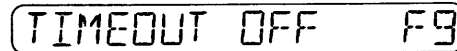
to select the timeout function

29B Displays



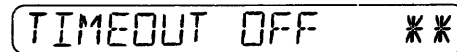


29B Displays





29B Displays



the timeout function is disabled.

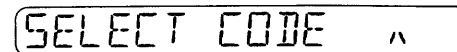
FA CHAR OUTPUT Command

Use the following procedure to enter the hex code for an ASCII character (see Appendix D) which will be transmitted to the port each time START is pressed (this function is inhibited in system remote control):



to select the select function.

29B Displays

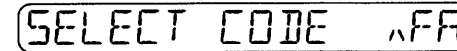


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.



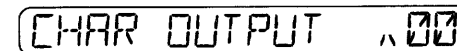
to select the character output function.

29B Displays





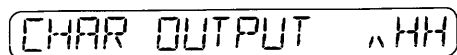
29B Displays



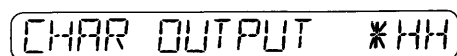


to enter the hex code for the ASCII character to be transmitted. (Step can be skipped if the default entry is correct.)

29B Displays

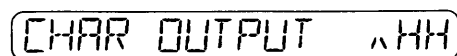


29B Displays



the prompt flashes momentarily.

29B Displays



6. To retransmit the same character return to step 5. To transmit a different character return to step 4.

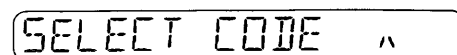
FB ENABLE PORT Command

Use the following procedure to enable system remote control and input interrupt. This also forces the RTS line high at all times for remote control from peripherals requiring a hardware handshake. The default at power-up is RTS low and system remote control and serial interrupts disabled.



to select the select function.

29B Displays

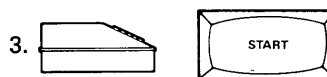
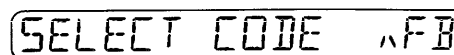


The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.

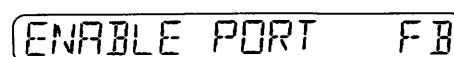


to select the enable port function.

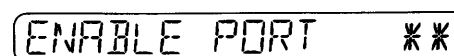
29B Displays



29B Displays



29B Displays



the system remote control is enabled.

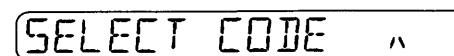
FC REMOTE ON OFF Command

Use the following procedure to enter hex codes for ASCII characters that can be used to turn remote control on or off (may be used with CRC, SRC or TRC):



to select the select function.

29B Displays

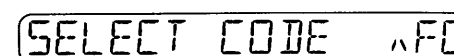


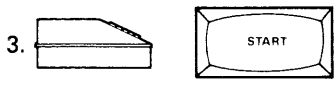
The prompt (^) following "SELECT CODE" on the display means that the two character hex select function code must be entered before the procedure can continue.



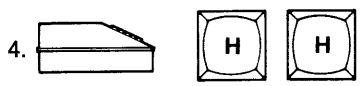
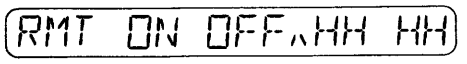
to select the remote on or off function.

29B Displays



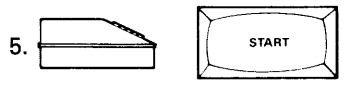
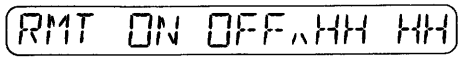


29B Displays



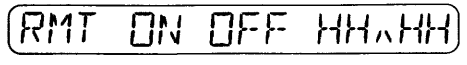
to select the hex ON code. (Step can be skipped if the default entry is correct.)

29B Displays



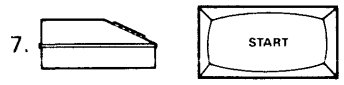
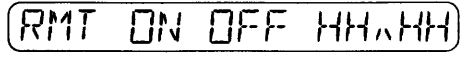
to enter the hex ON code.

29B Displays



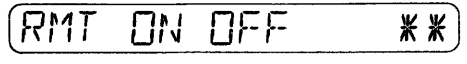
to select the hex OFF code. (Step can be skipped if the default entry is correct.)

29B Displays



the remote on off entry is complete.

29B Displays



3.9 REMOTE OPERATION

The programmer can be operated externally using three different remote control modes. Two of the modes are standard to the programmer and are described in this section. They are the Computer Remote Control (CRC) and System Remote Control (SRC). The third mode is the optional Terminal Remote Control (TRC) mode which is covered in Appendix B of this manual.

CRC is the preferred mode of operation and should be used for all new driver designs. Both SRC and TRC are included mainly to accommodate user's whose existing operations have software and/or hardware requiring either of those modes of operation.

3.9.1 COMPUTER REMOTE CONTROL

CRC is designed to enable control of the 29B Programmer by a computer. Linked directly to the programmer, the computer generates and sends commands to the programmer, determines variables for setting programming parameters (where needed), and reacts to information returned to it from the programmer.

While these commands may be sent by an operator at a terminal, the commands and syntax described in this manual were designed to be easily incorporated into a computer program.

Installation

The 29B with CRC must be connected to the computer according to RS232C specifications. The function of each serial port pin connection is described in subsection 2.5 of this manual. Refer to table 2-2 to determine the necessary connector pins for serial data transfers. The programmer's baud rate, parity, and stop bit settings are described in subsection 2.5.2.

Overview

Figure 3-4 illustrates the basic components of the 29B under computer remote control. 29B CRC remote control operation is controlled by the user's computer. The user must provide application software which will allow his computer to issue CRC commands and to interpret CRC responses.

Data transferred between the computer and the 29B is generally in ASCII notation, encoded in the selected data translation format (see Appendix A), although straight binary transfer is also possible.

Commands are generated by the computer according to the computer's software or in response to keyboard entries. The computer sends commands to the 29B which executes the command (or tries to) and then sends back a response character.

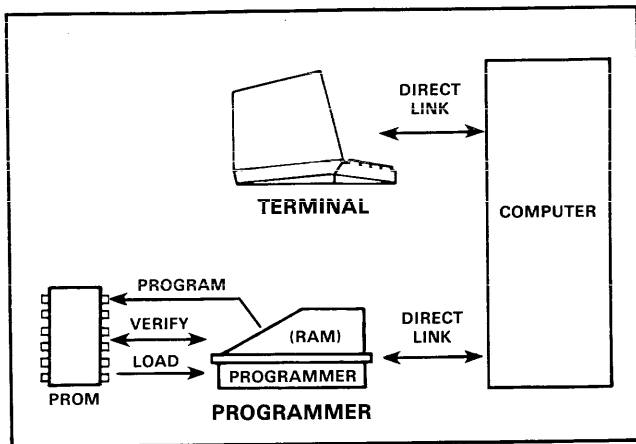


Figure 3-4. Computer Remote Control Components

Response Characters

The programmer sends a response character to the computer after every command; table 3-4 summarizes these characters. A ">" symbol for a response means that the command was successfully executed. Whenever an error occurs, the 29B will send an F to the computer. The computer or the operator can respond by interrogating the programmer with the X or F command. The X command causes the programmer to send the computer a complete list of the last sixteen error codes (described in Appendix E of this manual) that have occurred. The F command codes all errors into a 32-bit error status word.

The error status word format is shown in table 3-5. The table shows that the eight-character word is broken into four two-character groups. The first two-character group defines receive errors, the second group defines programming errors, the third group defines I/O errors and the last group defines RAM errors. Each two character group contains a maximum of eight bits, with each bit representing the presence of the defined error or error type. Multiple errors within a character do occur; therefore, if all four bits within the character are used and present the transmitted character would be F, while if there were no

errors present for that character, zero would be transmitted. The example error status word shown below (also shown on the table) shows how to use the table to decode the error-status word format.

Use the following procedure to decode the error status word 80C80081:

1. The first character in the example is 8. This decodes to the most significant bit high (message bit 31) as the only bit used in the first character. This bit is present whenever there is an error in the message; however it does not define whether the error is a programming, I/O, or RAM error. Since only bit 31 is used in the first character, it will always be either 8 or 0.
2. The second character is 0 showing there are no receive errors. Message bits 24, 25, and 26 only carry information in this group; therefore, it will range in magnitude between 0 (no errors) and 7 (all three possible receive errors).
3. The third character (first in the programming errors group) is C. This decodes as errors present for message bits 22 (start line not set high) and 23 (device-related error). All four message bits (20 through 23) are used so the possible combinations range from 0 (no errors) to F (all four errors).
4. The fourth character is 8. This decodes as an error present for message bit 19 (device not blank). All four message bits (16 through 19) are used so the possible combinations range from 0 to F.
5. The fifth character (first in the I/O errors group) is 0. This shows that there are no errors for this character. Only message bits 12 and 15 are used in this character; therefore, the only characters possible are 0, 1, 8, and 9.
6. The sixth character is also 0, showing again that there are no errors for this character. All four bits are used (message bits 8 through 11) so the characters possible range from 0 to F. Since both the fifth and sixth characters are 0 there are no I/O errors.

Table 3-4. Response Characters

Character	Name	Description
>	Prompt	Sent on entering remote control, after an ESCAPE or BREAK key has halted a command, or after a command has been successfully executed. The programmer then transmits a carriage return.
F	Fail	Informs the computer that the programmer has failed to execute the last command entered. The programmer then transmits a carriage return.
?	Question	Informs the computer that the programmer does not understand a command or the command was invalid. The programmer then transmits a carriage return.

Table 3-5. Error-Status Word

Type of Error	Bit Number	Value	Description
RECEIVE ERRORS	31	8	ANY ERROR. If the word contains any errors, the most significant bit (bit 31) will be high.
	30	4	Not used
	29	2	Not used
	28	1	Not used
	27	8	Not used
	26	4	Serial-overflow error (42)
	25	2	Serial-framing error (41, 43)
	24	1	Buffer overflow, i.e., > 15 characters (48)
PROGRAMMING ERRORS	23	8	Any device-related error
	22	4	Start line not set high (26)
	21	2	L2 + L3> Device
	20	1	Composite DAC error
	19	8	Device not blank (20)
	18	4	Illegal bit (21)
	17	2	Nonverify (23, 24, 29)
	16	1	Incomplete programming, or no card set (22, 25, 30-39)
I/O ERRORS	15	8	I/O error (46, 50, 58, 59, 94, 95 or any I/O error)
	14	4	Not used
	13	2	Not used
	12	1	Compare error (52)
	11	8	Sum-check error (82)
	10	4	Record-count error, MOS Technology (93) Address-check error, Signetics and Tek Hex (92) Record-type error, Intel Intellec 8/MDS (94)
	9	2	Address error, i.e., > word limit (27, 28, 51, 56, 57)
	8	1	Data not hexadecimal (84, 85, 91) Insufficient data received, ASCII-Hex and Octal (54)
RAM ERRORS	7	8	RAM-hardware error (64, 66 or any RAM error)
	6	4	Not used
	5	2	L2 + L3> RAM (in RAM-RAM block move)
	4	1	Invalid center point for split or shuffle
	3	8	Illegal split or shuffle
	2	4	No RAM or insufficient RAM resident (61)
	1	2	RAM write error, or program-memory failure (63)
	0	1	RAM and not on 1K boundary (62)
EXAMPLE: What errors are indicated in this error status word: 80C80081		NOTES	
<p>8 — the word contains error information</p> <p>0 — no receive errors</p> <p>C — (= 8 + 4): 8 = Device-related error 4 = Start line not set high (error 26)</p> <p>8 — device is not blank (error 20)</p> <p>0 — no input errors</p> <p>0 — no input errors</p> <p>8 — RAM error (error 62, and possible 64 and 66)</p> <p>1 — RAM end is not on 1K boundary (error 62)</p>		<p>(a.) The numbers in parentheses are 29B error codes, defined in Appendix E.</p> <p>(b.) An error can cause as many as 3 bits to be high: the bit which represents the error, the most significant bit of the 8-bit word in which the error bit occurs, and bit 31.</p> <p>(c.) After being read, the error-status word resets to zeros.</p>	

7. The seventh character (first in the RAM errors group) is 8. This decodes as an error present for message bit 7 (RAM hardware error). Message bit 6 is not used so possible combinations are 0, 1, 2, 3, 8, 9, 10, and 11.
8. The eighth character is 1 which decodes for an error in message bit 0 (RAM end not on 1k boundry). All four bits are used (message bits 0 through 3) so the characters possible range from 0 to F.

When a command is invalid or not understood by the programmer, it will send a "?." When this occurs, examine the last command entered, check its validity, make the necessary changes or corrections, and reenter the new or corrected command.

Entering and Exiting Remote Control

To enter CRC use select function F1 (see section 3-8). While in CRC the programmer display will show REMOTE MODE. On entering remote control, the programmer will retain all RAM data.

To exit CRC via the 29B keyboard, press any of the four blue mode keys. To exit via the computer, use Z RETURN. The programmer will retain all RAM data and operating parameters except the address offset.

Command Summary

A summary of CRC commands is given in table 3-6. The table response column ends with the symbols ">CRLF." These represent the prompt (>) that appears on the terminal display, and the carriage return (CR) and line feed (LF) that are transmitted after the command has been successfully completed. The line feed is present only if the null command (U) has defined the null count to any hex value between "00" and "FE." The default null count is hex "FF" which results in line feeds being suppressed. A detailed explanation of the CRC commands follow the summary table.

Table 3-6. Command Summary

Type of Command	Command	Name	Response*
CONTROL COMMANDS	RETURN	Execute	
	ESCAPE	Abort	> CRLF
	BREAK	Abort	> CRLF
UTILITY COMMANDS	G	Software Configuration	HHHH> CRLF
	HHHH<	Set Begin RAM	> CRLF
	HHHH;	Set Block Size	> CRLF
	HHHH:	Set Begin Device Address	> CRLF
	HH]	Select Extended Function	> CRLF
	S	Sum-check	HHHH> CRLF
	F	Error-Status Inquiry	HHHHHHHH> CRLF
	X	Error-Code Inquiry	HHHH> CRLF
	H	No Operation	> CRLF
	Z	Escape Remote Control	None
DEVICE COMMANDS	T	Illegal-Bit Test	> CRLF
	B	Blank Check	> CRLF
	[Family and Pinout Inquiry	FFPP> CRLF
	FFPP@	Select Family	> CRLF
	R	Respond	AAAA/B/C> CRLF
	L	Load	> CRLF
	P	Program	> CRLF
	V	Verify	> CRLF
I/O COMMANDS	D	Select Odd Parity	> CRLF
	E	Select Even Parity	> CRLF
	N	Select No Parity	> CRLF
	J	Set 1 Stop Bit	> CRLF
	K	Set 2 Stop Bits	> CRLF
	CFFA	Select Translation Format	> CRLF
	HHM	Select Record Size	> CRLF
	HHU	Set Nulls and Line Feeds	> CRLF
	HHHHW	Set Address Offset	> CRLF
	=	Disable Timeout	> CRLF
	I	Input	> CRLF ^(a)
	O	Output	> CRLF ^(a)
	C	Compare	> CRLF ^(a)
	Y	Parity-Error	0000> CRLF
	EDITING COMMANDS	Q	Swap Nibbles
\		RAM-RAM Block Move	> CRLF
HHHH?		Split RAM Data	> CRLF
HHHH>		Shuffle RAM Data	> CRLF
^		Clear All RAM	> CRLF
HANDLER COMMANDS	%	Handler Start	> CRLF
	H!	Binning Control	> CRLF

* Line feed is optional
(a) Response occurs after data transmission with the proper termination.

Command Groups

CRC commands are divided into the following six categories:

- Control Commands (Table 3-7)
- Utility Commands (Table 3-8)
- Device Commands (Table 3-9)
- I/O Commands (Table 3-10)
- Editing Commands (Table 3-11)
- Handler Commands Table 3-12)

The handler commands are used only when the optional Handler is installed on the 29B. Refer to the table listed for each category of command for a detailed explanation of those commands. The characters (CR) are throughout the tables to represent the carriage return key.

Table 3-7. Control Commands

Title	Command	Description
Return	(CR)	Carriage return character which executes each command. It must be sent to the programmer immediately after the command. All commands are ignored if not followed by a RETURN.
Escape or Break	ESC	These commands cause the programmer to unconditionally halt (abort) any operation in progress, output a > , and await further instructions from the computer.

Table 3-8. Utility Commands

Title	Command	Description
Software Configuration Number	G(CR)	On command, the programmer sends the 4-digit hex number representing the particular configuration or revision level of software resident in the 29B.
Set Begin RAM Address	HHHH<(CR)	This command, preceded by a 4-digit hex address (HHHH), defines the first RAM address to be used for data transfers. It is also the RAM source address when used in a Block Move. The default value is 0.
Set Block Size	HHHH;(CR)	Sets the hex number of bytes (HHHH) to be transferred. The default value is the programming module word limit for device-related operations or the RAM limit less the Begin RAM Address for I/O operations; there is no default value for Block Moves.
Set Begin Device	HHHH:(CR)	This command, preceded by a 4-digit Address hex address (HHHH), defines the first device address to be used for data transfers. It is also used as the RAM destination address when used in a Block Move. The default value is 0.
Select Extended Function	HH](CR)	Accesses Select Functions or Select Codes (HH) carried in the extended software of some programming Paks. Consult the programming Pak manuals for more information.
Sum-Check	S(CR)	This command instructs the programmer to calculate the 4-digit hex sum-check of RAM from 0 or Begin RAM to RAM word limit, device word limit, or the limit defined by the ";" (Block Size) command, whichever is smaller. Sum-check is defined in the Glossary in Appendix A.
Error-Status	F(CR)	On this command, the programmer Inquiry returns a 32-bit word, displayed as 8 hex characters, that codes errors accumulated. The error-status word resets to all zeroes after interrogation. See table 3-5.

Table 3-8. Utility Commands (Continued)

Title	Command	Description
Error Code Inquiry	X(CR)	The programmer responds to this command with hex error codes previously stored. After execution, the error codes are cleared from memory. Appendix E of this manual lists and describes all the error codes.
No Operation	H(CR)	This is a null command and always returns a prompt (>).
Escape Remote	Z(CR)	This command returns control to the Control 29B keyboard. All RAM data and operating parameters except the address offset are retained.

Table 3-9. Device Commands

Title	Command	Description
Illegal-Bit Test	T(CR)	Instructs the programmer to perform an illegal-bit test and stores the error code and returns an F if an illegal-bit occurs (programmed device bit whose corresponding RAM bit is unprogrammed).
Blank Check	B(CR)	Instructs the programmer to do a blank check (search the device for programmed bits) and store the error code and return an F if the device is nonblank.
Family and Pinout Inquiry	[(CR)	The programmer responds to this command with the Family and Pinout Codes of the selected device. Returns an F when using a programming module not requiring Family and Pinout Codes.
Select Family and Pinout	FFPP@(CR)	Selects a 2-digit hex Family Code (FF) and a 2-digit hex Pinout Code (PP). This command is only valid when using programming Paks requiring Family and Pinout Codes.
Respond	R(CR)	The programmer checks the programming module or Pak and outputs: AAA/ (device word limit) B/ (byte size), C 1 (VOL) or 0 (VOH). A VOL device is one in which the programmed state is LOW, whereas a VOH device is one in which the programmed state is HIGH.
Load	L(CR)	This command instructs the programmer to load data into RAM from the device in the programming module, within the parameters defined by the Begin RAM Address, Block Size, and Begin Device Address.
Program	P(CR)	This command instructs the programmer to program the data in RAM into the device in the programming module, within the parameters defined by the Begin RAM Address, Block Size, and Begin Device Address.
Verify	V(CR)	This command instructs the programmer to compare RAM data with the data of the device in the programming module, within the parameters defined by the Begin RAM Address, Block Size, and Begin Device Address.

Table 3-10. I/O Commands

Title	Command	Description
Select Odd Parity	D(CR)	Instructs the programmer to set odd parity for output data and inspect incoming data for odd parity. This command overrides the programmer's parity switch. The default value is the programmer's parity-switch setting.
Select Even Parity	E(CR)	Instructs the programmer to set even parity for output data and inspect incoming data for even parity. The E command overrides the programmer's parity switch. The default value is the programmer's parity-switch setting.
Select No Parity	N(CR)	Instructs the programmer to not check incoming data for parity, and to output data without parity. The N command overrides the programmer's parity switch. The default value is the programmer's parity-switch setting.
Set 1 Stop Bit	J(CR)	On receiving this command, the programmer sets one stop bit for serial data transfers. The default value is the programmer's stop-bit switch setting.
Set 2 Stop Bits	K(CR)	On receiving this command, the programmer sets two stop bits for serial data transfers. The default value is the programmer's stop-bit switch setting.
Set Translation	CFFA(CR)	This command selects the instrument control code (C) and the format code (FF) used to define handshaking and data translation during serial I/O operations. The default values are instrument control code zero (0) and MOS Technology Format -81. All the data translation formats available are detailed in Appendix A of this manual.
Select Record Size	HHM(CR)	The 2 hex characters (HH) define the number of data bytes per record in serial-output operations. The default value is 16 bytes per record for data translation formats with a variable record size (all formats except ASCII-Binary, Spectrum, and Fairchild Fairbug).
Clear All RAM	^ (CR)	Clears all of the programmers data RAM to zeroes.
Set Nulls	HHU(CR)	The 2 hex characters (HH) before U set the number of nulls to be output following the carriage return in serial-output operations, and enable line feeds. The default value is no nulls or line feeds. Entering FF before U will also invoke the default value.
Disable Timeout	= (CR)	This command disables the 25-second I/O timeout. The timeout can be restored only by turning off the programmer and then turning it on again.
Input Data	I(CR)	This command instructs the programmer to accept formatted data from the computer.
Output Data	O(CR)	This command instructs the programmer to translate RAM data into the selected data translation format and output this data to the computer. The programmer will stop outputting on receipt of the X-OFF character, DC-3 (CTRL S), and will resume on receipt of the X-ON character, DC-1 (CTRL Q).
Compare Data	C(CR)	This command instructs the programmer to compare data in RAM with data from the computer.
Parity-Error	Y(CR)	This command instructs the programmer to output the hex number of parity errors (up to FFF) encountered since power-on, since the last Y command, or since the last parity command (D, E, or N).

Table 3-11 Editing Commands

Title	Command	Description
Swap Nibbles	Q(CR)	Instructs the programmer to exchange high- and low-order halves of every word in RAM. This is useful when programming 4-bit devices with only one-half of RAM at a time.
RAM-RAM Block Move	\ (CR)	This command moves a specified number of bytes (as specified by the Block Size) from one RAM location (as specified by the Begin RAM Address) to another (specified with the Begin Device Address Command).
Split RAM Data	HHHH?(CR)	<p>For 16-bit microprocessor data; complement of Shuffle RAM Data (below). After a block of data is input or loaded to RAM (each sequential pair of 8-bit bytes representing a 16-bit word), the command "splits" the block into two adjacent blocks, separated by the specified center point (HHHH). The split stores the even-numbered 8-bit bytes of each byte pair in sequence from address 0 to the center point; odd-numbered bytes are stored in sequence at addresses beginning at the center point. The reorganized data occupies the same original block in RAM.</p> <p>Each block of data can then be programmed into an 8-bit device, and the two devices can be addressed in parallel (while in use) to deliver 16-bit words to the processor.</p> <p>Typically, the center point will equal the number of words in the 8-bit device to be programmed. In any event, it must meet two requirements:</p> <ol style="list-style-type: none"> 1. It must be a power of 2. 2. It must be less than or equal to half the size of the resident RAM. <p>The center-point default value is the RAM midpoint.</p>
Shuffle RAM Data	HHHH> (CR)	<p>For 16-bit microprocessor data. Complement of Split RAM Data, this command merges into one block the two adjacent blocks of data which meet at the specified center point address (HHHH). Two 8-bit devices are first loaded adjacent to each other in RAM, beginning a address 0, to create the two blocks, which are then merged for serial transfer. The center point must be a power of 2 between 0 and RAM midpoint. The center-point default value is the RAM midpoint.</p>
Address Offset	HHHHW(CR)	This command specifies the value to be subtracted from addresses on input and added to them on output. Up to eight characters (in some formats) can be input before this command.

Table 3-12. Handler Commands

Title	Command	Description
Handler Start	%(CR)	This command puts the programmer into a waiting state until the handler indicates a start condition or the operator presses the START key on the programmer. Once the start condition has been received by the programmer it will output a prompt ">" symbol to the serial port and wait for another command.
Binning Control	HI(CR)	Instructs the handler to distribute programmed and verified devices to the numbered (H) bin, which must be from 1 to 4.

3.9.2 SYSTEM REMOTE CONTROL

The programmer's System Remote Control (SRC) capability allows you control of the programmer's operation from a terminal. Once the controlling terminal has been properly interfaced to the programmer (see section 2), select function FB (Port Enable) must be manually entered from the keyboard to enable SRC.

Command Protocol

The syntax for SRC is similar to that of the front panel manual operations, using the source/destination syntax method described in section 3.5. When keying in commands from a terminal, the programmer recognizes the first two characters of each word (except REVIEW), as shown in table 3-13.

Table 3-13. Command Entry in SRC

KEYBOARD COMMAND	REMOTE CONTROL COMMAND
Copy	CO(SP)
Verify	VE(SP)
Select	SE(SP)
Edit	ED(SP)
Device	DE(SP)
RAM	RA(SP)
Port	PO(SP)
Review	/

The space bar [denoted by the letters (SP) as shown in table 3-13] is used after the command as a delimiter, setting the boundaries for that command. The programmer will not define the characters input until the space bar is entered. And, since the programmer only recognizes the first two characters, some variations are possible. For example; the command COPY from DEVICE address (XXXX) for block size of (YYYY) to RAM address (ZZZZ) can be written with the same result in either of the following ways:

```
CO(SP)DE(SP)XXXX(SP)YYYY(SP)TO(SP)RA(SP)ZZZZ(CR)
COPY(SP)DEVICE(SP)XXXX(SP)YYYY(SP)TO(SP)RAM(SP)
ZZZZ(CR)
```

NOTE

The word "TO" must be keyed in before the destination when entering commands with SRC.

In either example, the programmer will load the data in the device into the programmer RAM, in the same way as a Load from Device is done from the front panel keyboard. The carriage return (CR) at the end of the line acts as an execute key. Command characters are input to the programmer until the (CR) is entered, signalling the programmer to execute that line of characters.

Command Entry

There are two methods of command entry while using SRC, direct or interactive. In direct command entry the commands are entered from the terminal keyboard using the space bar between words as shown in the examples above (Command Protocol paragraph). Entry of the X, Y, and Z values is optional, if they are not included the programmer uses either the last valid entry or the default value.

The interactive method streamlines the entries required of the operator. The operator keys in the function, then presses (CR). In return, the terminal displays a prompt similar to those displayed on the programmer display. For example, if the CO (CR) is entered for a Copy command, the terminal would display COPY DATA FROM to prompt the operator of the next entry. Operation occurs just as it does when using the programmer's keyboard except the commands are keyed in at the terminal.

When entering data on the terminal, the slash (/) is used in place of the REVIEW key. When pressed, it causes the previous character(s), prior to a space bar entry, to be ignored. The characters are still displayed on the screen; however, all entries back to the previous space bar are ignored and the operator can enter the replacement characters.

Inputting Parameters

The parameters required are given in figure 3-5. The characters entered must be valid 4-digit hexadecimal values. When the default value is satisfactory, no new value needs to be entered. If it is necessary to skip over the source address (when its default is correct) and change the block size, input a comma (,) or space bar (SP).

SRC Operations

When a command is sent to the keyboard using SRC the programmer reacts the same as if a manual command had originated at the keyboard. When the command is entered, the programmer display reads the same as when the corresponding key is pressed on the front panel keyboard. The terminal display, while not identical with the programmer display, has approximately the same information. The 29B's SRC can display both Select and Format menus on the terminal. The procedures that follow cover the commands available through SRC and show both the terminal and programmer display.

WHEN DEFAULT VALUES ARE ALL CORRECT:

DIRECT ENTRY
CO_vDEV_vTO_vRA(CR)

INTERACTIVE
CO(CR)
(COPY DATA FROM > displayed on terminal)
DE(CR)
(DEV ADDR,SIZE > displayed on terminal)
TO_vRA(CR)
(CO DEV > RAM ADDR > displayed on terminal)

NOTE

The commands shown here are examples only. The complete list of commands is in Table 3-6.

WHEN NO DEFAULT VALUES ARE CORRECT:

DIRECT ENTRY
CO_vDEV_vXXXX_vYYYY_vTO_vRA_vZZZZ(CR)

INTERACTIVE
CO(CR)
(COPY DATA FROM > displayed on terminal)
DE(CR)
(DEV ADDR,SIZE > displayed on terminal)
XXXX_vYYYY_vTO_vRA(CR)
(COPY DEV > RAM ADDR > displayed on terminal)
ZZZZ(CR)

WHEN SOURCE ADDRESS DEFAULT IS CORRECT AND BLOCK SIZE DEFAULT IS INCORRECT:





DIRECT ENTRY
CO_vDEV_v .XXXX_vTO_vRA(CR)

INTERACTIVE
CO(CR)
(COPY DATA FROM > displayed on terminal)
DE(CR)
(DEV ADDR,SIZE > displayed on terminal)
.YYYY_vTO_vRA(CR)
(COPY DEV > RAM ADDR > displayed on terminal)(CR)

Figure 3-5. Inputting Remote Control Parameters

COPY AND VERIFY COMMANDS. Use the following procedures to perform the copy and verify procedures using SRC.

Load RAM with Master Device Data. Use the following procedure to load the 29B RAM with data from a master device using a remote terminal and SRC:





1.    

to select the mode.

29B Displays

COPY DATA FROM

The terminal displays: COPY DATA FROM >

2.    

to select the source of the data.

29B Displays

DEV\ADDR\SIZE TO

The terminal displays the command and data in the following sequence:

COPY DATA FROM >
DEV ADDR,SIZE >

3. "ADDR" means that you can change the begin device address to any address within the range of device word limit by entering the hex value with the terminal. If an entry is not made the value defaults to zero (0). If a hex value appears in place of "ADDR" it is from a previously entered begin address.

"SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made the value defaults to device size. If a hex value appears in place of "SIZE" it is from a previously entered block size.

NOTE

If a begin address is specified then the size must be specified also. The size must be equal to or less than the device size minus the begin address.

If the default or previously entered values are correct, disregard the XXXX and/or YYYY characters in this step representing the device address and block size respectively and enter only the terminal message "TO(SP)RA(CR)." If the values are incorrect, enter the data in the sequence below:

XXXX,YYYY(SP)TO(SP)RA(CR)
or
TO(SP)RA(CR)

29B Displays

CO DEV\RAM\ADDR

The terminal displays:

COPY DATA FROM >
DEV ADDR,SIZE >
CO DEV > RAM ADDR >

4. "ADDR" means that you can change the begin RAM address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered RAM begin address.

If the default or previously entered values are correct, disregard the ZZZZ characters in this step representing the RAM address and enter only the terminal carriage return (CR). If the values are incorrect, enter the data in the sequence below:

ZZZZ(CR) or (CR)

29B Displays

```
FAM,00 PIN 00
```

The terminal displays data in the following sequence:

```
COPY DATA FROM >
DEV ADDR,SIZE >
CO DEV> RAM ADDR >
FAM HH PIN HH >
```

NOTE

If the Pak or programming module installed does not require a family/pinout code combination the 29B skips to the display in step 5.

5. Enter the 4-digit hex family/pinout code combination for the device to be copied. The codes are listed in the Pak or programming module manual. If a code combination appears on the display, it is from a previously entered code.

If a previously entered family/pinout code is correct, disregard the HHHH characters in this step representing the family/pinout code and enter only the terminal carriage return (CR). If the code is incorrect, enter the correct code in the sequence below:

HHHH(CR) or (CR)

29B Displays

```
LOAD DEVICE
```

The terminal displays:

```
COPY DATA FROM >
DEV ADDR,SIZE >
CO DEV> RAM ADDR >
FAM HH PIN HH >
LOAD DEVICE >
```

NOTE

If the pak installed has more than one socket, the LED next to the correct socket will illuminate.

6. Insert and lock the master device into the appropriate socket. (See section 3.4.2).



29B Displays

```
LOADING DEVICE 0
```

The terminal displays data in the following sequence:

```
COPY DATA FROM >
DEV ADDR,SIZE >
CO DEV> RAM ADDR >
FAM HH PIN HH >
LOAD DEVICE >
(Cursor Returns)
```

NOTE

The amount of time the programmer will require to perform this operation will vary depending on the device size. The action symbol line rotates 360 degrees while the operation is taking place. When the operation is complete the following display signals the programmer's readiness. If an error code is displayed, see appendix E.

29B Displays

```
LOAD DONE HHHH
```

The terminal displays data in the following sequence:

```
COPY DATA FROM >
DEV ADDR,SIZE >
CO DEV> RAM ADDR >
FAM HH PIN HH >
LOAD DEVICE >
(Cursor Returns)
LOAD DONE HHHH >
```

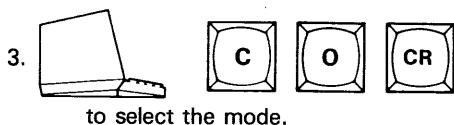
NOTE

HHHH is the hex sumcheck of the data transferred, i.e., the hex sumcheck of RAM data starting at the begin RAM address and extending for a length of address locations equal to the device size minus the begin device address.

8. Remove the master device from the socket.
9. To repeat the load operation, return to step 6.

Load RAM From Serial Port. Use the following procedure to load the 29B RAM from front panel control, with incoming serial port data using a remote terminal and SRC:

1. Set-up the serial port. Refer to Section 2.
2. Select the appropriate data translation format from appendix A and execute the B3 (format number) select function (see section 3.8).

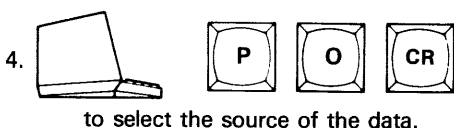


29B Displays

`COPY DATA FROM`

The terminal displays:

COPY DATA FROM >



29B Displays

`POR, ADDR, SIZE TO`

The terminal displays data in the following sequences:

COPY DATA FROM >
POR ADDR, SIZE >

5. "ADDR" in the display means that you can change the begin port address to any address within the range of format word limit (up to eight characters in length, determined by the format selected) by entering the hex value with the terminal. If an entry is not made, the value defaults to the first incoming address. If a hex value appears in place of "ADDR" it is from a previously entered begin address.

"SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to end of RAM address minus the begin RAM address. If a hex value appears in place of "SIZE," it is from a previously entered block size.

If the default or previously entered values are correct disregard the XXXX and/or YYYY characters in this step representing the port address and block size respectively and enter only the terminal message "TO(SP)RA(CR)." If the values are incorrect, enter the data in the sequence below:

XXXX,YYYY(SP)TO(SP)RA(CR)
or
TO(SP)RA(CR)

29B Displays

`CO POR, RAM, ADDR`

The terminal displays data in the following sequence:

COPY DATA FROM >
POR ADDR, SIZE >
CO POR > RAM ADDR >

6. "ADDR" means that you can change the begin RAM address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered RAM begin address.

If the default or previously entered values are correct, disregard the ZZZZ characters in this step representing the RAM address and enter only the terminal carriage return (CR). If the values are incorrect, enter data in the sequence below:

ZZZZ(CR) or (CR)

29B Displays

`INPUT PORT`

The terminal displays data in the following sequence:

COPY DATA FROM >
POR ADDR, SIZE >
CO POR > RAM ADDR >
(Cursor Returns)

NOTE

The amount of time the programmer will require to perform this operation will vary depending on the block size. The action symbol line rotates 360 degrees while the operation is taking place. When the operation is complete the following display signals the programmer's readiness. If an error code is displayed see appendix E.

29B Displays

INPUT DONE HHHH

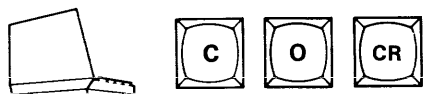
The terminal displays data in the following sequence:

```
COPY DATA FROM >
POR ADDR,SIZE >
CO POR> RAM ADDR >
(Cursor Returns)
INPUT DONE HHHH >
```

NOTE

HHHH is the hex sumcheck of all data transferred to RAM.

Program Device with RAM Data. When programming a device the 29B system performs an illegal bit test and a blank check at nominal Vcc to verify the ability of the PROM to accept programming before it begins the operation. Use the following procedure to program a blank device with data from RAM using a remote terminal and SRC:

- 

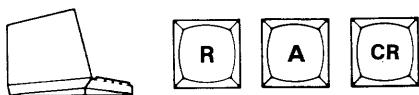
to select the mode.

29B Displays

COPY DATA FROM

The terminal displays:

```
COPY DATA FROM >
```

- 

to select the source of the data.

29B Displays

RAM ADDR,SIZE TO

The terminal displays data in the following sequence:

```
COPY DATA FROM >
RAM ADDR,SIZE >
```

- “ADDR” in the display means that you can change the begin RAM address to any address within the range of RAM word limit by entering the hex value with the terminal. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of “ADDR” it is from a previously entered begin address.

“SIZE” means that you can change the block size by entering the hex value with the keyboard. If an entry is not made the value defaults to the device size. If a hex value appears in place of “SIZE,” it is from a previously entered block size.

NOTE

If the begin device address is specified then the size must be specified also. The size must be equal to or less than the device size minus the begin device address.

If the default or previously entered values are correct, disregard the XXXX and/or YYYY characters in this step representing the RAM address and block size respectively and enter only the terminal message “TO DE(CR).” If the values are incorrect, enter the data in the following sequence:

```
XXXX,YYYY(SP)TO(SP)DE(CR)
or
TO(SP)DE(CR)
```

29B Displays

CO RAM,DEV ADDR

The terminal displays data in the following sequence:

```
COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> DEV ADDR >
```

- “ADDR” means that you can change the begin device address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of “ADDR,” it is from a previously entered device begin address.

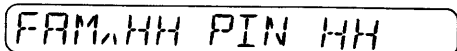
If the default or previously entered values are correct, disregard the ZZZZ characters in this step representing the device address and enter only the terminal carriage return (CR). If the values are incorrect, enter the data in the sequence below:

```
ZZZZ(CR or (CR)
```

NOTE

If the Pak or programming module installed does not require a family/pinout code combination the 29B automatically skips to the display in step 5.

29B Displays



The terminal displays data in the following sequence:

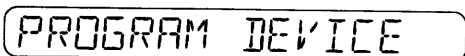
COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> DEV ADDR >
FAM HH PIN HH >

- 5. Enter the four digit hex family/pinout code combination for the device to be copied. The codes are listed in the pak or programming module manual. If a code combination appears on the display, it is from a previously entered code.

If a previously entered family/pinout code is correct, disregard the HHHH characters in this step representing the family/pinout code and enter only the terminal carriage return (CR). If the values are incorrect, enter data in the sequence shown below:

HHHH(CR) or (CR)

29B Displays



The terminal displays data in the following sequence:

COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> DEV ADDR >
FAM HH PIN HH >
PROGRAM DEVICE >

NOTE

If the pak installed has more than one socket the LED next to the correct socket will illuminate.

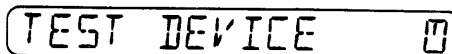
- 6. Insert and lock the device to be programmed into the appropriate socket. (See section 3.4.2).



NOTE

The amount of time the programmer will require to perform the test, program, and verify operations will vary depending on the device size. The action symbol line rotates 360 degrees while the operations are taking place. As each operation is completed in turn, the following displays signal that the programmer has completed each step. If an error code is displayed, see appendix E.

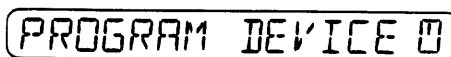
29B Displays



The terminal displays data in the following sequence:

COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> DEV ADDR >
FAM HH PIN HH >
PROGRAM DEVICE >
(Cursor Returns)

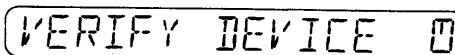
29B Displays



The terminal displays data in the following sequence:

COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> DEV ADDR >
FAM HH PIN HH >
PROGRAM DEVICE >
(Cursor Returns)
(No Change)

29B Displays



The terminal displays data in the following sequence:

COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> DEV ADDR >
FAM HH PIN HH >
PROGRAM DEVICE >
(Cursor Returns)
(No Change)
(No Change)

29B Displays

PRG DONE NN HHHH

The terminal displays data in the following sequence:

COPY DATA FROM >
 RAM ADDR,SIZE >
 CO RAM> DEV ADDR >
 FAM HH PIN HH >
 PROGRAM DEVICE >
 (Cursor Returns)
 (No Change)
 (No Change)
 PRG DONE NN HHHH>

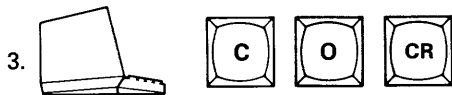
NOTE

NN is the decimal sequence number (number of devices that have been programmed since the last time the counter was reset or power was applied to the programmer). HHHH is the hex checksum of the device data programmed by this operation.

8. Remove the device from the socket.
9. To repeat the load operation, return to step 6.

Output to Serial Port. Use the following procedure to output data to the serial port using the remote terminal and SRC:

1. Set-up the serial port. Refer to Section 2.
2. Select the appropriate data translation format from appendix A and execute the B3 (format number) select function (see section 3.8).



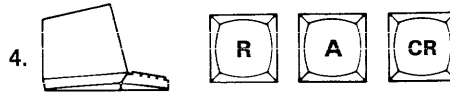
to select the mode.

29B Displays

COPY DATA FROM

The terminal displays:

COPY DATA FROM >



to select the source of the data.

29B Displays

RAM ADDR,SIZE TO

The terminal displays data in the following sequence:

COPY DATA FROM >
 RAM ADDR,SIZE >

5. "ADDR" in the display means that you can change the begin RAM address to any address within the range of RAM word limit by entering the hex value with the terminal. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR" it is from a previously entered begin address.

"SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to the RAM size minus the begin RAM address. If a hex value appears in place of "SIZE," it is from a previously entered block size.

If the default or previously entered values are correct disregard the XXXX and/or YYYY characters in this step representing the device address and block size respectively and enter only the terminal message "TO PO(CR)." If the values are incorrect, enter data in the sequence shown below:

XXXX,YYYY(SP)TO(SP)PO(CR)
 or
 TO(SP)PO(CR)

29B Displays

CO RAM)POR ADDR

The terminal displays data in the following sequence:


COPY DATA FROM >
 RAM ADDR,SIZE >
 CO RAM> POR ADDR >

6. "ADDR" means that you can change the begin port address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered port begin address.

If the default or previously entered values are correct, disregard the ZZZZ characters in this step representing the port address and enter only the terminal carriage return (CR). If the values are incorrect, enter data in the sequence shown below:

ZZZZ(CR) or (CR)

29B Displays

OUTPUT PORT 

The terminal displays data in the following sequence:

COPY DATA FROM >
 RAM ADDR,SIZE >
 CO RAM> POR ADDR >
 (formatted data file scrolls on terminal display)

NOTE

The amount of time the programmer will require to perform this operation will vary depending on the data in RAM. The action symbol line rotates 360 degrees while the operation is taking place.

- When the operation is complete, the following display signals the programmer's readiness. If an error code is displayed, see appendix E.

29B Displays

OUTPUT DONE HHHH





The terminal displays data in the following sequence:

COPY DATA FROM >
 RAM ADDR,SIZE >
 CO RAM> POR ADDR >
 (formatted data file scrolls on terminal display)
 OUTPUT DONE HHHH>

NOTE

HHHH is the hex sumcheck of all data transferred.

Block Move. Use the following procedure to move data from one location in RAM to another using the remote terminal and SRC:

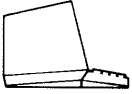



-    
to select the mode.

29B Displays

COPY DATA FROM

The terminal displays:

COPY DATA FROM >

-    
to select the source of the data.

29B Displays

RAM ADDR/SIZE TO

The terminal displays data in the following sequence:

COPY DATA FROM >
 RAM ADDR,SIZE >

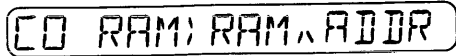
- "ADDR" in the display means that you can change the RAM begin address to any address within the range of RAM word limit by entering the hex value with the terminal. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered begin address.

"SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to 0. Block size must be specified. If a hex value appears in place of "SIZE," it is from a previously entered block size.

If the default or previously entered values are correct, disregard the XXXX and/or YYYY characters in this step representing the device address and block size respectively and enter only the terminal message "TO RA(CR)." If the values are incorrect, enter data in the sequence shown below:

XXXX,YYYY(SP)TO(SP)RA(CR)
 or
 TO(SP)RA(CR)

29B Displays



The terminal displays data in the following sequence:

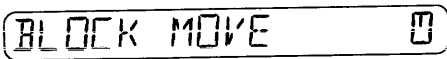
COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> RAM ADDR >

- 4. "ADDR" means that you can change the new (move to) begin RAM address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered RAM begin address.

If the default or previously entered values are correct disregard the ZZZZ characters in this step representing the RAM address and enter only the terminal carriage return (CR). If the values are incorrect, enter data in the sequence shown below:

ZZZZ(CR) or (CR)

29B Displays



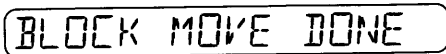
The terminal displays data in the following sequence:

COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> RAM ADDR >
(No Change)

NOTE

The amount of time the programmer will require to perform this operation will vary depending on the device size. The action symbol line rotates 360 degrees while the operation is taking place. When the operation is complete the following display signals the programmer's readiness. If an error code is displayed see appendix E.

29B Displays



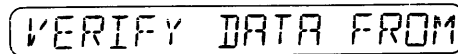
The terminal displays data in the following sequence:

COPY DATA FROM >
RAM ADDR,SIZE >
CO RAM> RAM ADDR >
(No Change)

Verify RAM Data Against Master Device Data. Use the following procedure to verify the data entered in the 29B RAM duplicates the master device data, using the programmer front panel:

- 1. to select the mode.

29B Displays

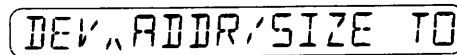


The terminal displays:

VERIFY DATA FROM>

- 2. to select the source of the data.

29B Displays



The terminal displays data in the following sequence:

VERIFY DATA FROM>
DEV ADDR,SIZE >

- 3. "ADDR" in the display means that you can change the begin device address to any address within the range of the device word limit by entering the hex value with the terminal. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered begin address.

"SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to device size. If a hex value appears in place of "SIZE" it is from a previously entered block size.

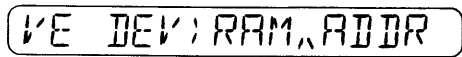
NOTE

If a begin device address is specified then the size must be specified also. The size must be equal to or less than the device size minus the begin device address.

If the default or previously entered values are correct disregard the XXXX and/or YYYY characters in this step representing the device address and block size respectively and enter only the terminal message "TO RA(CR)." If the values are incorrect, enter the data in the sequence shown below:

XXXX,YYYY(SP)TO(SP)RA(CR)
or
TO(SP)RA(CR)

29B Displays



The terminal displays data in the following sequence:

VERIFY DATA FROM>
DEV ADDR,SIZE >
VE DEV> RAM ADDR >

- 4. "ADDR" means that you can change the begin RAM address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered RAM begin address.

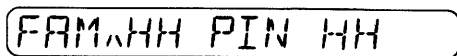
If the default or previously entered values are correct, disregard the ZZZZ characters in this step representing the RAM address and enter only the terminal carriage return (CR). If the values are incorrect, enter data in the sequence shown below:

ZZZZ(CR) or (CR)

NOTE

If the Pak or programming module installed does not require a family/pinout code combination, the 29B automatically skips to the step 5 display.

29B Displays



The terminal displays data in the following sequence:

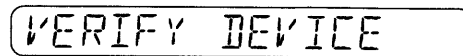
VERIFY DATA FROM>
DEV ADDR,SIZE >
VE DEV> RAM ADDR >
FAM HH PIN HH >

- 5. Enter the 4-digit hex family/pinout code combination for the device to be copied. The codes are listed in the Pak or programming module manual. If a code combination appears on the display, it is from a previously entered code.

If a previously entered family/pinout code is correct, disregard the HHHH characters in this step representing the family/pinout code and enter only the terminal carriage return (CR). If the code is incorrect, enter the correct code in the sequence shown below:

HHHH(CR) or (CR)

29B Displays



The terminal displays data in the following sequence:

VERIFY DATA FROM>
DEV ADDR,SIZE >
VE DEV> RAM ADDR >
FAM HH PIN HH >
VERIFY DEVICE >

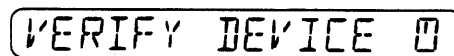
NOTE

If the Pak installed has more than one socket the LED next to the correct socket will illuminate.

- 6. Insert and lock the master device into the appropriate socket. (See section 3.4.2).



29B Displays



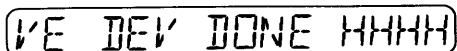
The terminal displays data in the following sequence:

VERIFY DATA FROM >
DEV ADDR,SIZE >
VE DEV> RAM ADDR >
FAM HH PIN HH >
VERIFY DEVICE >
(Cursor Returns)

NOTE

The amount of time the programmer will require to perform this operation will vary depending on the device size. The action symbol line rotates 360 degrees while the operation is taking place. When the operation is complete, the following display signals the programmer's readiness. If an error code is displayed see appendix E.

29B Displays



The terminal displays data in the following sequence:

VERIFY DATA FROM>
DEV ADDR,SIZE >
VE DEV> RAM ADDR >
FAM HH PIN HH >
VERIFY DEVICE >
(Cursor Returns)
VE DEV DONE HHHH

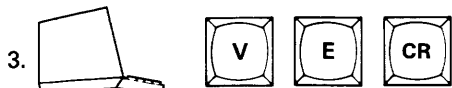
NOTE

HHHH is the hex checksum of the data verified.

- 8. Remove the master device from the socket.
- 9. To repeat the load operation, return to step 7.

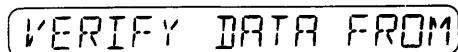
Verify RAM From Serial Port. Use the following procedure to verify the incoming serial port data using the remote terminal and SRC:

- 1. Set-up the serial port. Refer to Section 2.
- 2. Select the appropriate data translation format from appendix A and execute the B3 (format number) select function (see section 3.8).



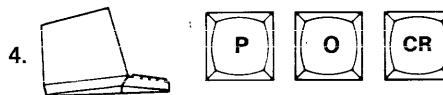
to select the mode.

29B Displays



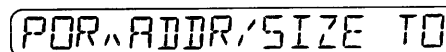
The terminal displays:

VERIFY DATA FROM>



to select the source of the data.

29B Displays



The terminal displays data in the following sequence:

VERIFY DATA FROM>
POR ADDR,SIZE >

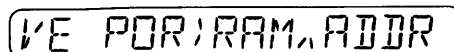
- 5. "ADDR" in the display means that you can change the begin port address to any address within the range of the port word limit by entering the hex value with the terminal. If an entry is not made, the value defaults to the first incoming address. If a hex value appears in place of "ADDR," it is from a previously entered begin address.

"SIZE" means that you can change the block size by entering the hex value with the keyboard. If an entry is not made, the value defaults to RAM size minus the begin RAM address. If a hex value appears in place of "SIZE," it is from a previously entered block size.

If the default or previously entered values are correct, disregard the XXXX and/or YYYY characters in this step representing the port address and block size respectively and enter only the terminal message "TO RA(CR)." If the values are incorrect, enter data in the sequence shown below:

XXXX,YYYY(SP)TO(SP)RA(CR)
or
TO(SP)RA(CR)

29B Displays



The terminal displays data in the following sequence


VERIFY DATA FROM>
POR ADDR,SIZE >
VE POR> RAM ADDR >

- 6. "ADDR" means that you can change the begin RAM address by entering the hex value with the keyboard. If an entry is not made, the value defaults to zero (0). If a hex value appears in place of "ADDR," it is from a previously entered RAM begin address.

If the default or previously entered values are correct, disregard the ZZZZ characters in this step representing the RAM address and enter only the terminal carriage return (CR). If the values are incorrect, enter data in the sequence shown below:

ZZZZ(CR) or (CR)

29B Displays

VERIFY PORT 

The terminal displays data in the following sequence:

VERIFY DATA FROM>
 POR ADDR,SIZE >
 VE POR> RAM ADDR >
 (No Change)

NOTE

The amount of time the programmer will require to perform this operation will vary depending on the device size. The action symbol line rotates 360 degrees while the operation is taking place.

- When the operation is complete, the following display signals the programmer's readiness. If an error code is displayed, see appendix E.

29B Displays

VE POR DONE HHHH

The terminal displays data in the following sequence:





VERIFY DATA FROM>
 POR ADDR,SIZE >
 VE POR> RAM ADDR >
 (No Change)
 VE POR DONE HHHH>

NOTE

HHHH is the hex sumcheck of all the serial port data verified.

EDIT COMMANDS. The contents of the programmer RAM can be edited in remote with either the hexadecimal, octal or binary modes selected, just as from the front panel keyboard. The key sequence is given below for only the hexadecimal mode of operation. If one of the others is used, the operation is the same, the difference is in the manner in which the RAM data is displayed. The address is always displayed in hex, regardless of the mode selected. Refer to the manual procedure in section 3-7 if any question exists on the operation in the octal or binary modes.

Use the following procedure to perform the Edit command while in the hex mode, operating with a remote terminal and using SRC:

-    

to select the mode.

29B Displays

EDIT ADDR,0000

The terminal displays: EDIT ADDR 0000 >

- Enter a hex address to be edited in the sequence shown below:

HHHH(CR)

29B Displays

HHHH DHH,RHH

The terminal displays data in the following sequence:

EDIT ADDR 0000 >
 HHHH DHH,RHH >

- 3. "HHHH" in the display is the hex address in RAM where the data can be edited. "DHH" is the hex data programmed into the device at the selected address (** for data means that address is undefined in the device). 'RHH' is the present hex data in the selected RAM address. To enter the new hex data for the selected address, enter data in the sequence:

HH(CR)

29B Displays

HHHH DHH RHH

New address is previous address plus one. Data is from new address.

The terminal displays data in the following sequence:

```
EDIT ADDR 0000 >
HHHH DHH,RHH >
HHHH DHH,RHH >
```

As long as new data is input followed by a return (CR) the address counter will be incremented by one for the next address. Exit the edit mode by pressing another mode key.

NOTE

If the data displayed is not to be changed, enter a carriage return (CR) to increment to the next data location or a slash (/) to decrement to the previous data location.

SELECT COMMANDS. Select Commands can be entered with SRC by selecting either (S) (E) (space) (select hex code) (CR), or by entering directly the first two characters of the select command followed by a carriage return (CR). Entering (S) (E) (CR) will display a menu of all select codes. Refer to table 3-15 for a listing of the format codes.

Use one of the following examples which illustrate the two methods to enter a select code.

- The following example shows how to enter the Clear All RAM select function using the hex code:

Enter: SE(SP)A4(CR).

NOTE

Always enter space bar between "SE" and the hex code.

29B Displays

CLEAR ALL RAM **

The terminal displays a prompt (>).

- The following example shows how to enter the Clear All RAM select mode using the first two characters of the select command.

Enter: CL(CR).

29B Displays

CLEAR ALL RAM **

The terminal displays a prompt (>).

Table 3-14. Select Commands

HEX CODE	CODE	TITLE
A1	SW	Swap Nibbles
A2	FI	Fill RAM
A3	IN	Invert RAM
A4	CL	Clear All RAM
A5	SP	Split RAM
A6	SH	Shuffle RAM
B0	DE	Device Size
B1	SU	Sumcheck RAM
B2	SY	System Config
B3	FO	Format Number
B4	NO	Nonblank Fail
B9	DI	Display Test
C1	CA	Calibration
D7	LE	Leader Output
D8	SI	Size Record
D9	NU	Null Count
F0	PR	Program Count
F1	RE	Remote Mode
F2	29	29B TRC
F3	LO	Lock Data On
F4	NI	Nibble Mode
F5	BI	Binary Base
F6	OC	Octal Base
F7	HE	Hex Base
F8	BY	Byte/Nib Mode
F9	TI	Timeout Off
FA	CH	Char Output
FB	EN	Enable Port
FC	RE	Remote On Off

SECTION 4

MAINTENANCE

4.1 OVERVIEW

WARNING

The instructions contained in this section are for qualified service personnel only. Do not attempt to perform them unless you are qualified to do so.

CAUTION

When servicing Data I/O equipment beyond that described in this manual, we recommend that you send your unit to your local service center. If you decide to do the repair, be aware that many parts in Data I/O equipment have been qualified by vendor and may not be interchangeable with the same type of part manufactured elsewhere. Refer to the back of this manual for service office locations.

The support material in this section has been provided to help you keep your 29B in good operating condition. General maintenance practices are discussed in subsection 4.2 while basic troubleshooting procedures are described in subsection 4.4. For those 29B users who prefer to do their own calibration, power supply calibration procedures are provided in subsection 4.3. To complete the calibration cycle, perform a calibration of your Pak or module, with the 29B, by following the calibration instructions in your Pak (or module) manual. If you wish, you can also send your programmer and Pak to your nearest Data I/O Service Center for a complete calibration.

4.2 MAINTENANCE

Regular maintenance of the 29B consists of cleaning and inspecting the unit. To gain access inside the programmer, follow the procedures for cover removal in subsection 4.2.1. Also, remove the protective shield from inside the programmer's front panel opening by following the instructions in subsection 4.2.2.

4.2.1 COVER REMOVAL

To remove your 29B front panel cover, follow these procedures:

WARNING

To avoid electrical shock, disconnect the power cord before removing cover, panel or protective shield.

1. Turn the programmer power off and remove the power cord.
2. Remove any programming Pak (or module) from the programmer (see section 2.3.1).

3. Turn the programmer upside down and remove the four screws at the corners of the base.
4. Return the programmer to its upright position.
5. Remove the cover by lifting it straight up.

4.2.2 REMOVING THE CONTROLLER BOARD PROTECTIVE SHIELD

A protective shield guards the interior of the programmer from dust and damage and protects the operator from shock. To remove it, do the following:

1. Remove the Pak (see subsection 2.3.1).
2. Remove the screw from the middle of the shield.
3. With your thumbs, push back and pull-up on the shield clearing it from the connector.
4. Lift the back edge out, pull the plate up slightly and turn it to the left until it clears the opening on the programmer's front panel.

4.2.3 INSTALLING THE PROTECTIVE SHIELD

1. Tilt the shield to one side and insert into Pak opening.
2. Lower the shield into the programmer's front panel opening.
3. Fit the plate over the Pak connector making sure the cutout for the dipswitch is on the right side.
4. Push down on the shield until it fastens into place. Secure it with the screw removed earlier.

4.2.4 CLEANING

Clean the exterior of the unit with a mild detergent on a damp cloth.

CAUTION

Do not use caustic or abrasive agents; these will damage the 29B.

Clean the fan filter (located on the top of the unit) every three months. Remove the filter and clean it with running water to rinse out accumulated dust. Dry it thoroughly before reinstalling. Press it back into its recessed location on the top of the programmer.

4.2.5 INSPECTION

You can help prevent malfunctions by periodically inspecting your 29B. Check cable connections, card seating, component mounting, etc., for proper seating, wear, and visible damage.

If you find heat-damaged components, be particularly careful to find and correct the cause of the overheating to prevent further damage.

4.3 CALIBRATION

WARNING

To avoid electric shock always test for voltage before touching when working inside the equipment.

Calibration of the 29B consists of checking and adjusting the power supplies. This will normally be performed as part of the calibration of a programming Pak (or module), as proper operation of the module depends on the programmer's power supplies.

This manual explains how to calibrate the programmer's power supplies. The programming Pak's (or module's) instruction manual provides the procedures for calibrating the Pak (or module), and explains how to carry out these instructions on the 29B by providing specific key sequences and steps for entering and executing the calibration mode. Figure 4-1 demonstrates the calibration procedure.

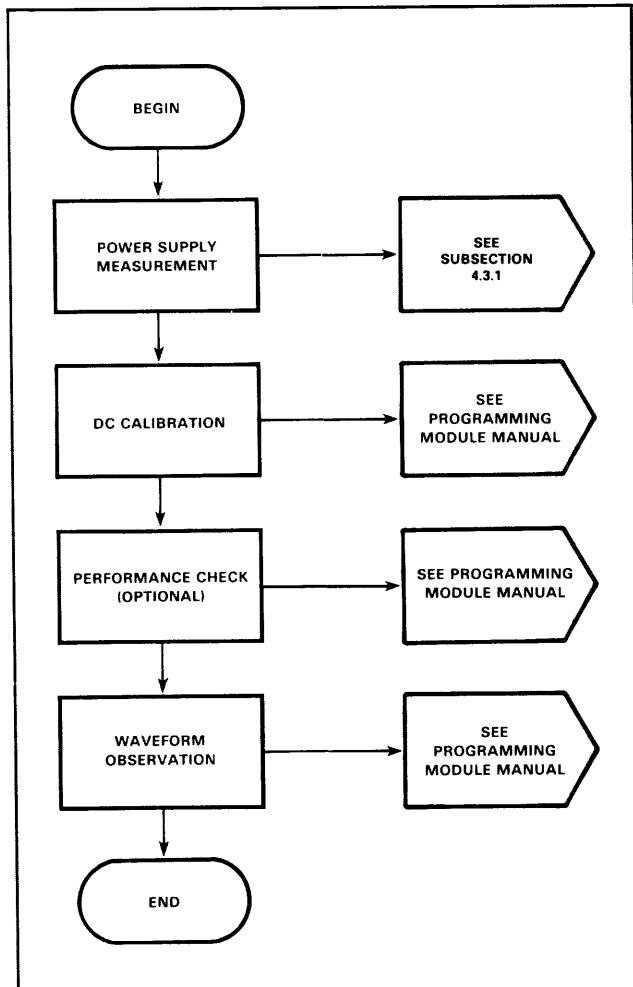


Figure 4-1. Calibration

4.3.1 POWER SUPPLY CALIBRATION

Proper operation of your programming Pak depends on the programmer's power supplies being properly adjusted. Therefore, the programmer power supplies must be calibrated first.

Table 4-1 lists the voltage ranges for the power supplies. Test points for these supplies are located both on the controller board (see figure 4-2) and on the calibration extender. The power supply voltages can be measured in either location.

Table 4-1. Calibration Voltage Ranges

POWER SUPPLY	VOLTAGE RANGE			TEST POINT	ADJUST
	Min.	Nom.	Max.		
+5 V	5.08	5.11	5.15	TP1	R42
+24 V	23.5	24.0	24.5	**	R23
+48 V	49.5	49.7	49.8	**	R16
+PROG V*	4.80	4.90	5.11	Below J7	*
-9 V	-9.20	-9.00	-8.80	**	R26
-5 V	-5.25	-5.00	-4.75	**	Non-adjustable
+12 V	11.4	12.0	12.6	To right of U39	Non-adjustable

* If adjustment is required, ground the right side of R59 on the Controller. Use R55 to adjust PROG V to $+21.60 \pm 0.20$ V.
 ** These test points are printed circuit feed-throughs to the left of J7.

Required Equipment

The following equipment is required for calibrating the 29B's power supplies.

- Digital Voltmeter
- Non-conductive alignment tool, or 1/8" flat-blade screwdriver
- Jumper wire approximately 12 inches long
- IC removal tool or small screwdriver

Set-Up

The following procedures describe calibration set-up. Test points are shown in figure 4-2.

1. Turn the programmer power off and remove the power cord.
2. Remove the programming Pak (or module) from the programmer.

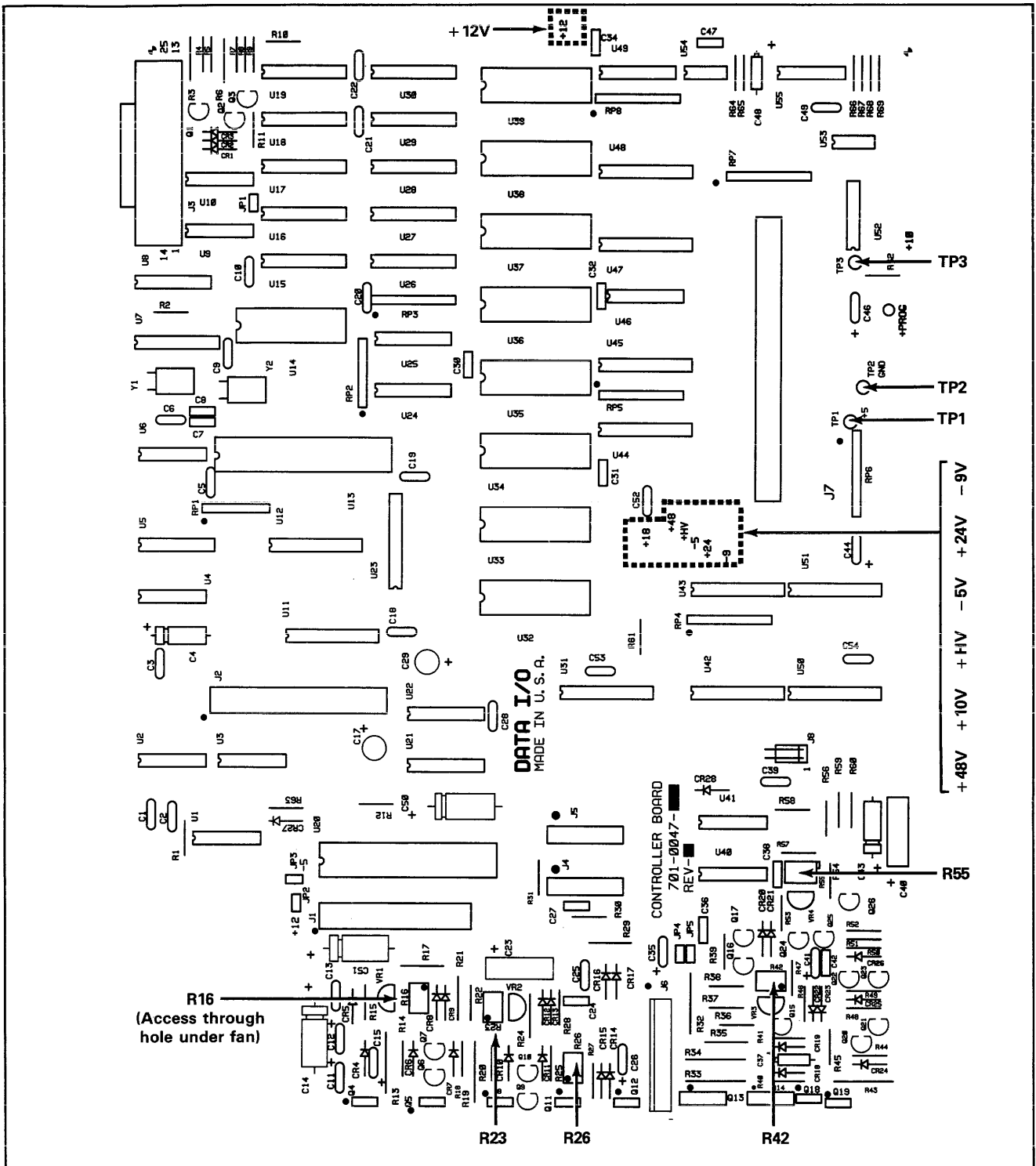


Figure 4-2. Power Supply Test and Adjustment Points

3. Remove the protective shield from inside the programmer. See subsection 4.2.2.
4. Connect your DVM to ground at TP2. See figure 4-2.
5. Ground TP3 by connecting a jumper from TP3 to TP2.
6. Turn the programmer power on.
7. Measure the supplies at the test points shown in figure 4-2. Results should be within the ranges listed in table 4-1.

Adjustment Procedures

If all adjustments are within the specified range, no adjustment will be necessary. If the voltages fall outside the range, adjust them as follows. Adjustment points are shown in figure 4-2.

1. Turn the programmer power off. See section 3.3.
2. Unplug the power cord. See section 2.4.
3. Remove any calibration equipment that has been installed.
4. Remove the top cover (section 4.2.1).
5. Remove the controller board protective shield (subsection 4.2.2).
6. Remove the front panel assembly as shown in figure 4-3b. To do this:
 - a. Remove the four screws from each corner of the panel.
 - b. Remove center brace screw.
7. Disconnect the front panel cable.
8. Ground TP3 by connecting a jumper from TP3 to TP2.
9. Connect power cord and apply power.

CAUTION

Extreme care is required to avoid short-circuiting discrete components while making measurements and adjustments.

10. Adjust the supply (see figure 4-2 for potentiometer location) so that it corresponds to the correct voltage listed in table 4-1. If supply cannot be adjusted within the specified range, refer to subsection 4.4, Troubleshooting.

4.3.2 THE PERFORMANCE CHECK/DC CALIBRATION SET-UP

Some programming Pak manuals contain a "performance check" subsection in the calibration procedure. The set-up procedures below, are also the same procedures you would follow in setting up your programmer for calibration. Your programming Pak (or module) manual will contain complete details on how to do a performance check or set your programmer and Pak up for calibration.

1. Turn the programmer power off. See subsection 3.3.
2. Insert the appropriate calibration equipment into the 29B. As an example, figure 4-4 shows the 29B set-up for calibration with a calibration extender and a Pak installed.

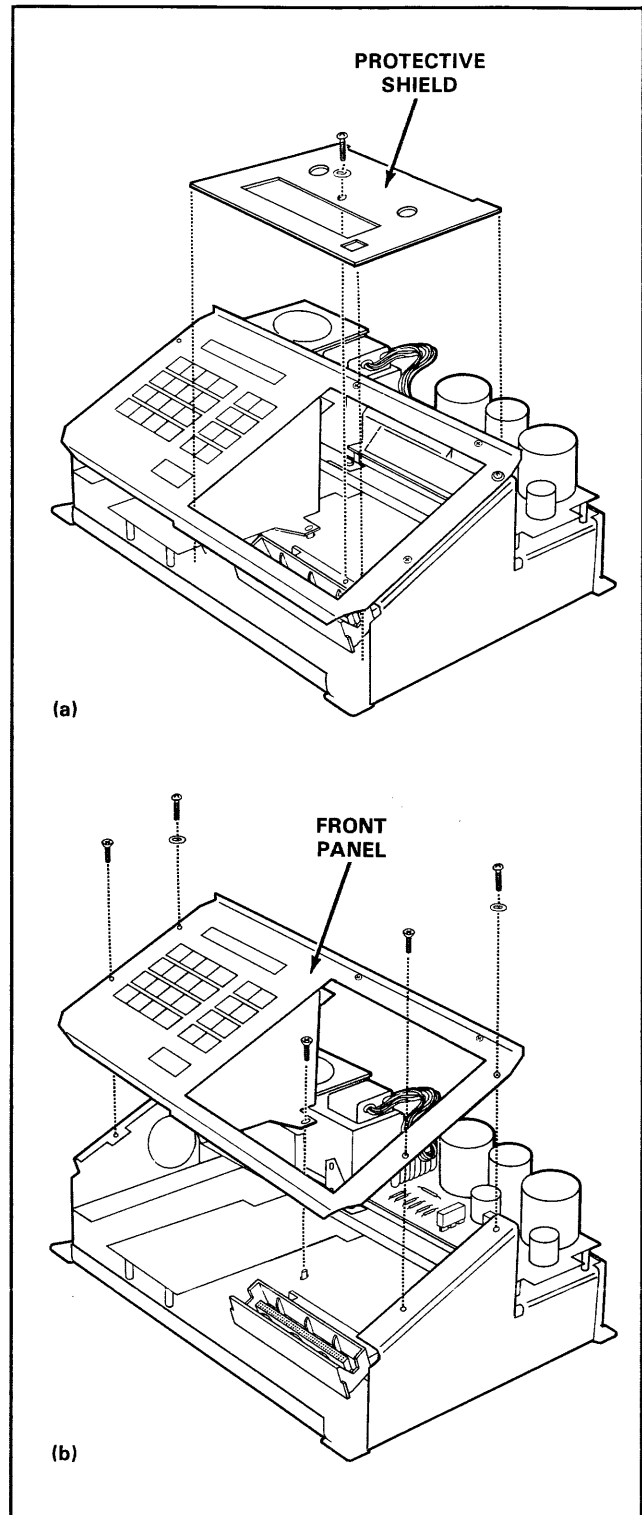


Figure 4-3. Front Panel and Protective Shield Removal

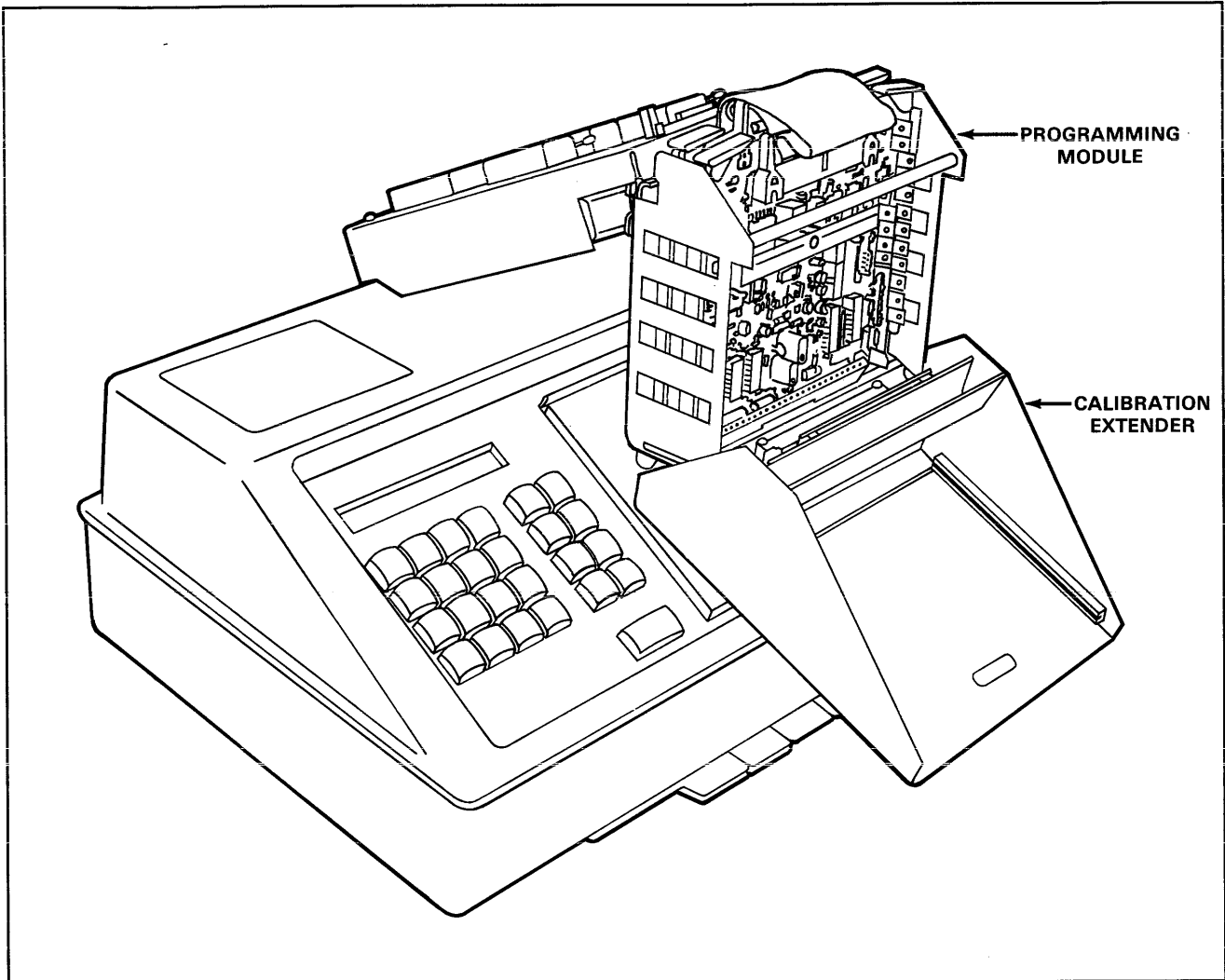


Figure 4-4. 29B with Programming Module Prepared for Calibration

3. Turn the power back on. See subsection 3.2.2.

CAUTION

Remove all devices from the programming Pak (or module) before putting the 29B in the calibration mode. DC voltages applied to the socket during calibration will damage any device in the socket.

4. Enter the calibration mode using select function C1.
- a. Press **SELECT**.
 - b. Enter **C1**.
 - c. Press **START**.

5. The 29B is now ready for step 1 of the Measurement Chart in your Pak (or module) manual. Follow the steps in order.

- To increment one step, press **START**.
- To decrement steps, press **REVIEW**.

After making a measurement, increment to the next step.

6. If any voltages fall outside the specified range, perform a complete DC calibration as outlined in your Pak (or module) manual.
7. To exit the calibration mode, press **COPY**, **VERIFY**, or **EDIT**.

4.3.3 CALIBRATION OF PROGRAMMING Paks AND MODULES

Data I/O manufactures two types of programming units to go with the 29B: programming Paks (which have expanded memory) and standard programming modules. Calibration of standard programming modules requires the Data I/O Universal Calibrator and Calibration Extender. Programming Paks have additional ROM (Read Only Memory), providing software which eliminates the need for the universal calibrator (the calibration extender is still necessary).

This section contains two calibration procedures, one for programming Paks and one for standard programming modules.

Calibration of Programming Paks

The procedure for calibration of programming Paks is as follows:

1. Measure and adjust the 29B's power supplies as necessary. (See subsection 4.3.1).
2. Enter the calibration mode:
 - a. Press **SELECT**.
 - b. Enter **C1**.
 - c. Press **START** twice. The status display (see table 4-2) will appear on the programmer display.

3. Refer to section 4 of your programming Pak instruction manual. This section describes the equipment you will need for calibration. In addition, the measurement chart in this section contains calibration steps for your programming Pak. Step numbers on the programmer's display correspond to those on the Measurement Chart.
 - To increment one step, press **START**. To increment more than one step at a time, enter the desired step number and press **START**.
 - To decrement steps, press **REVIEW**.

Perform the steps in sequence and make adjustments as necessary.

4. For waveform observation, follow the directions in the programming Pak manual. Use select function A2 (see table 3-3) to fill RAM with the correct programming data. Consult the appropriate programming module for the proper RAM data.

NOTE

Always exit the calibration mode before inserting a programmable device.

5. To exit the calibration mode, press **COPY**, **VERIFY**, or **EDIT**.

Table 4-2. Displays and Instructions for Calibrating Standard Programming Modules

DISPLAY	CORRESPONDING INSTRUCTION	OPERATOR ACTION
HEX ADDR 0000		Enter a 4-digit hexadecimal address, then press START .
SET START HI (or SET START LOW)	Set the programmer in START .	Press SELECT until SET START HIGH appears and then press START .
	Set the programmer in STOP .	Press SELECT until SET START LOW appears and then press START .
PGM ONE ADDR		Initiates waveform tests.
(status display) 0000 DFF R00 L F	Monitor the Start/Stop Line.	Select the status display and note the appropriate values.
	Confirm data on the DO bus.	Select the status display and note the appropriate values.
	Load data onto the DI bus.	Select the status display, enter the desired data, and press START .
	Set the programmer in Reverse.	Select the status display and press BACKSPACE .
	Set the programmer in Forward.	Select the status display and press START .

Calibration of Standard Programming Modules

Calibration of standard programming modules (see table 4-2) is performed as follows:

1. Measure and adjust the programmer power supplies as necessary. (See subsection 4.3.1.)
2. Put the programmer into the calibration mode via Select Function **C1**.
 - a. Press **SELECT**.
 - b. Enter **C1**.
 - c. Press **START** twice. The status display (see table 4-2) will appear on the programmer display.

NOTE

Some modules do not allow increment or decrement from certain steps.

NOTE

*Some programming modules generate waveforms only in automatic program mode. **PGM ONE ADDR** is a manual program mode and will not cause such programming modules to output waveforms. For these modules, use the program command instead. To determine if a programming module can be used only in automatic mode, see Section 3 of the programming module operation and maintenance manual.*

3. Refer to the instruction manual for the programming module. Section 4 contains required equipment, calibration instructions, and the calibration charts. To enter step 1, press **START**. To enter any other step, enter the desired step number and press **START**. To increment one step, press **START**. To decrement steps, press **REVIEW**. The programmer displays and calibration chart instructions are given in table 4-2.
4. To exit calibration, press **COPY**, **VERIFY**, or **EDIT**.

4.4 TROUBLESHOOTING

The following troubleshooting information is an aid to understanding malfunctions and locating hardware failures. Subsection 4.4.1 discusses the procedures for establishing the type of trouble occurring in the unit and the steps necessary for further servicing. Subsection 4.4.2 directs the service technician to the portion of the circuitry suspected of failure. These procedures do not isolate the fault to the component level, but the information in this section, along with normal troubleshooting and service techniques, should lead to the solution of most hardware failures.

Circuit descriptions are provided in Section 5, and Schematics are located in Appendix F.

4.4.1 PRELIMINARY TROUBLESHOOTING

The following paragraphs describe some common problems, with corrective steps following each. After performing each step, determine whether the problem still exists. Figure 4-5 shows the overall troubleshooting procedure.

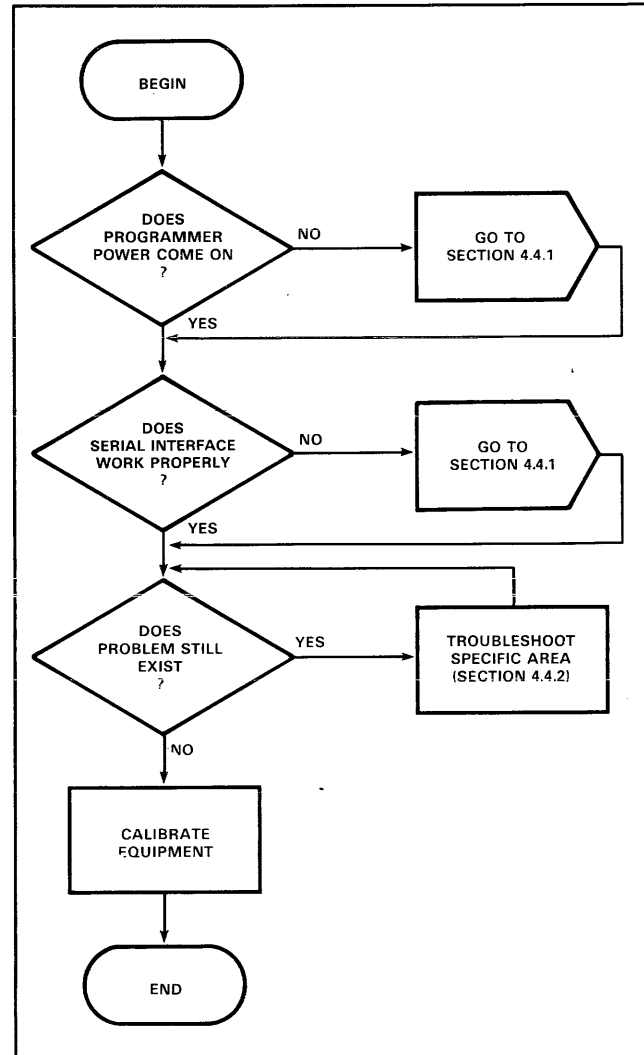


Figure 4-5. Troubleshooting

Erratic or Nonfunctional Programmer

If the programmer does not operate, or operates erratically, check the following:

1. Check that the AC power cord is firmly plugged in and the power switch is on.
2. Check that AC power selection is the same as the line voltage. The voltage at which the programmer will operate is shown on the voltage selector wheel. For proper operation, the line voltage must be within $+5/ - 10$ percent of the voltage shown on the wheel.

3. Check that the programming module is fully seated in the mating connector (J7) of the controller card. During operations with the programming module removed, TP3 must be grounded by a jumper to TP2.
4. Remove any cables attached to the serial interface.
5. Check the power supplies according to the calibration procedure in paragraph 4.2. If this reveals a problem, refer to "power supplies" paragraph 4.4.2.
6. Check the installation of all hardware. Check the orientation and connection of all cables as well as the seating of the PC boards. Check all jumpers.
7. Check the controller as described in paragraph 4.4.2.
8. If steps 1 through 7 do not reveal the problem, contact your Data I/O Service Center.

Serial I/O Failures

If the serial interface does not work properly, check the following:

1. Make sure the programmer and other equipment to which it is connected are set for the same parity, baud rate, and number of stop bits.
2. Check the operation of the peripheral equipment according to the manufacturer's procedures.
3. Troubleshoot the serial interface circuitry as described in paragraph 4.4.2.

4.4.2 TROUBLESHOOTING SPECIFIC AREAS

The following paragraphs discuss specific areas of circuitry. After performing each step, determine whether the problem continues.

Power Supplies

If it has been determined from preliminary troubleshooting that a problem exists in the power supplies, do the following:

1. Turn the programmer off for 30 seconds and then on again. Portions of the power supply employ foldback overcurrent protection. If a supply becomes overloaded, it will remain off even after the overload is eliminated. The protection circuits can be reset by turning the programmer off for 30 seconds and then on again.
2. Refer to table 4-3 to check the voltage for each power supply circuit. Test points are shown in figure 4-2.
3. To replace any components associated with the +5 V, -5V, or +12V supplies, first remove jumpers JP1 through JP5 to protect other circuits from overvoltage. When component replacement is complete, reinsert the jumpers and calibrate the programmer per section 4.3.1.

Table 4-3. Power Supply Voltages

SUPPLY	PIN ON CALIBRATOR OR EXTENDER	VOLTAGES		
		Min.	Nom.	Max.
+5†	+5	5.08	5.11	5.15
+24*	+24	23.5	24.0	24.5
+48*	+48	49.5	49.7	49.8
-9	-9	-9.20	-9.00	-8.80
-5†	JP3	-5.25	-5.00	-4.75
+PROG V*	+PROG	4.80	4.90	5.11
+12†	JP2	11.4	12.0	12.6

* A programming module must be installed or TP3 must be grounded.

† To replace faulty components associated with this supply, first remove jumpers JP1 through JP5. After bringing the supply into range, replace the jumpers and calibrate the programmer.

Controller

If it has been determined from preliminary troubleshooting that a problem exists in the controller, do the following:

1. Check that all socketed devices are seated firmly and all soldered components are intact. Check that jumpers JP1 through JP5 are installed.
2. With an oscilloscope, determine if the $\overline{V_{02}}$ timing signal is present by observing pin 7 of U5 on the controller board. If it is not present, check that the programming module is grounding pin HH (TP3) on J7 (the programming module interface).
3. If steps 1 and 2 do not reveal the problem, contact your Data I/O Service Center.

Serial Interface Circuitry

If it has been determined from preliminary troubleshooting that a problem exists in the serial interface, do the following:

1. Use an oscilloscope to observe the signal on pin 3 of U14 on the controller card. The frequency of the signal at pin 3 should be 16 times the selected baud rate. (Baud rate = $[1/\text{time}]$ divided by 16.)
2. Perform steps a through d. A failure in any step indicates a failure in the ACIA (U14) or the drivers U9 or U10).
 - a. Initiate a copy from the data RAM to the serial port.
 - b. Check the voltages at U14, pins 23 and 24. Both should read 0.0 to 0.5 V.
 - c. Press **START**.
 - d. Use the oscilloscope to observe pin 2 of the serial interface. The 29B should be transmitting data at the selected baud rate. Perform the following procedure (see figure 4-6).
 - (1) Turn 29B power on.
 - (2) Set baud rate to 9600.
 - (3) Using select function A2, enter 00 hex data into all of RAM.
 - (4) Connect oscilloscope probe to pin 2 of serial port connector J3.
 - (5) Copy RAM to port using 0 (default) values for beginning addresses and maximum (default) block size.
 - (6) Observe the waveform. It should be similar to figure 4-6(b).
 - (7) Using select function A2, enter FF hex data.
 - (8) Repeat steps (4) and (5) above.
 - (9) Observe waveform. It should be similar to figure 4-6(a).

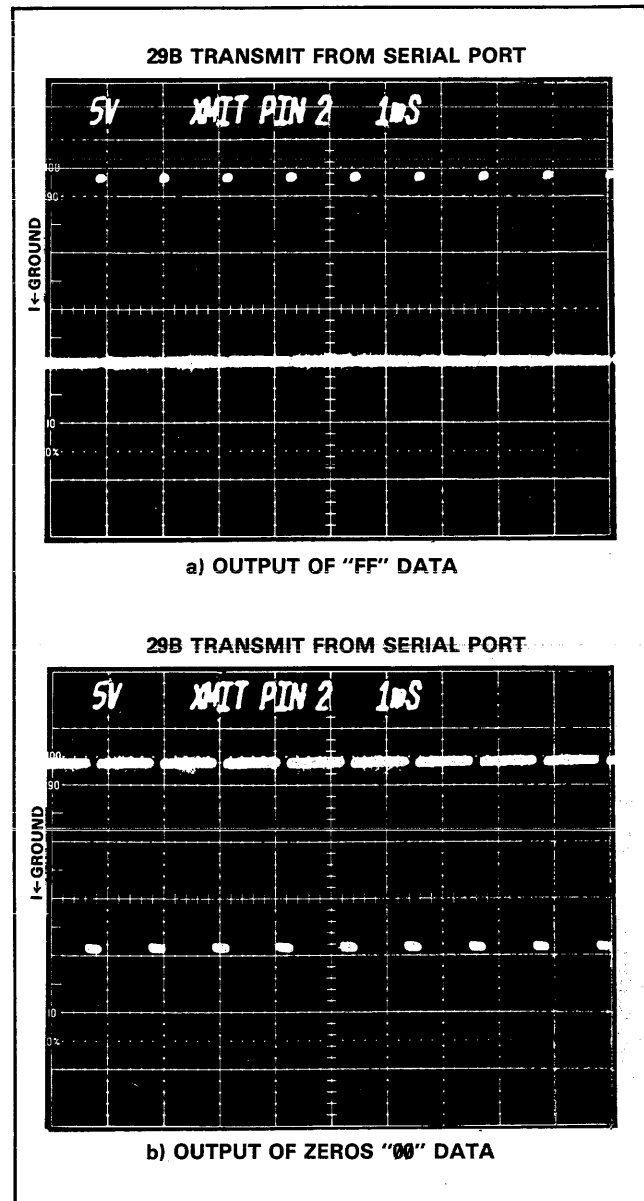


Figure 4-6. Checking Baud Rate on Pin 2

SECTION 5

CIRCUIT DESCRIPTION

5.1 INTRODUCTION

This section describes the 29B Universal Programmer's circuitry. It includes both a general and detailed description and of the programmer's main circuitry functions. Each detailed circuitry function is described in its own subsection along with a block diagram.

This section also discusses the main components of the 29B programmer, address map assignments, and assembly cabling.

5.2 GENERAL CIRCUIT DESCRIPTION

The 29B is a microprocessor-based programmer utilizing bus-style architecture. Figure 5-1 shows the functional blocks of each of the circuits in the 29B. These circuits are described in the following paragraphs.

AC line voltage passes through a voltage selector and filter to the transformer and a cooling fan. The AC is rectified, filtered and regulated for use by all other circuits on the 29B.

A microprocessor unit (MPU) controls all operations for the 29B. MPU circuits are interfaced to a keyboard and display. An RS232 serial port is standard.

The actual programming of devices is performed through a 64-pin connector (J7) which interfaces with programming modules or Paks. Optionally, devices can be programmed by a PROM Handler through a special hardware interface provided for this purpose.

The internal RAM of the 29B provides storage for all programming data information whether from a master device or from an external source (computer, terminal, etc.).

5.3 DETAILED CIRCUIT DESCRIPTION

Figure 5-2 shows an interconnect diagram of the 29B. Each of the circuit areas, power supplies, MPU, keyboard/display, RAM, and interface circuits, are described in the following paragraphs and shown in separate block diagrams.

Figure 5-2 also shows the cabling between assemblies along with associated connector cables and part numbers. The schematic for each board is located in Appendix F.

The component layout of the 29B is shown in figure 5-3.

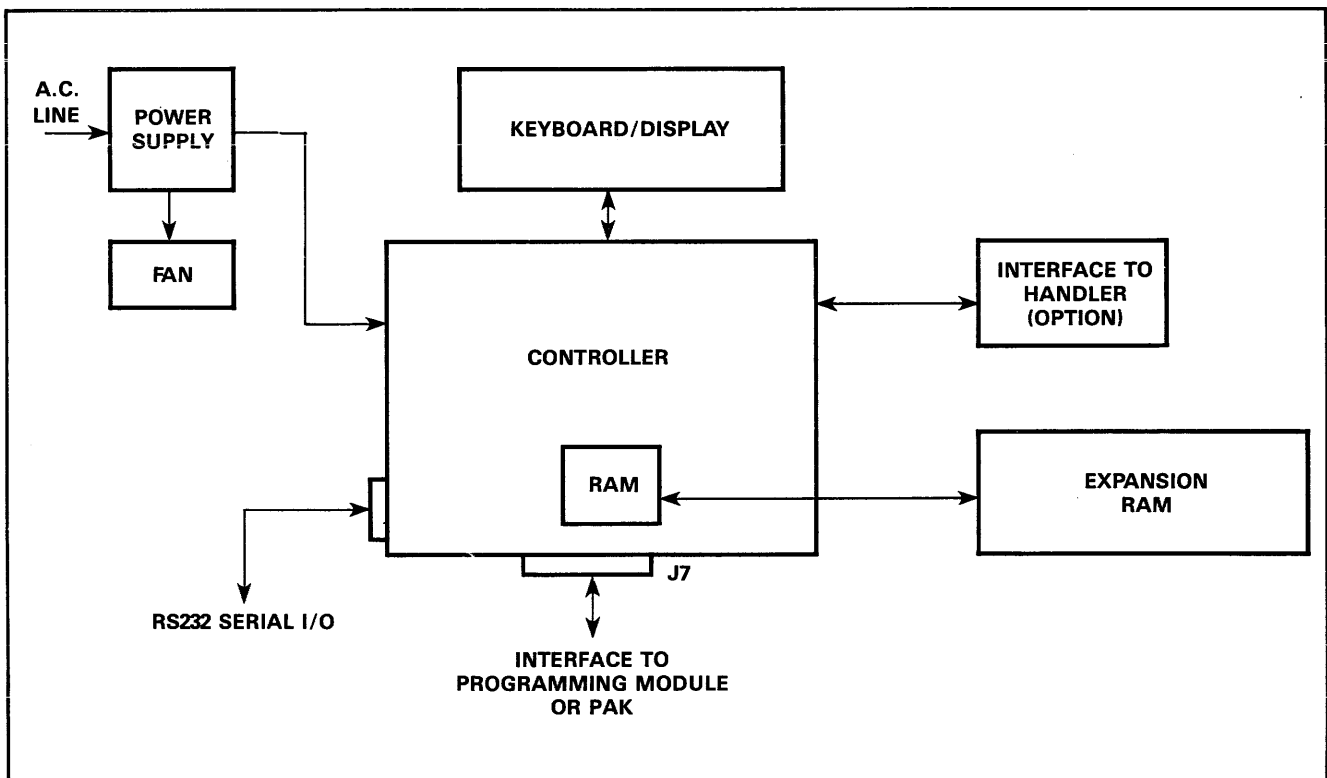


Figure 5-1. Functional Diagram of 29B Universal Programmer

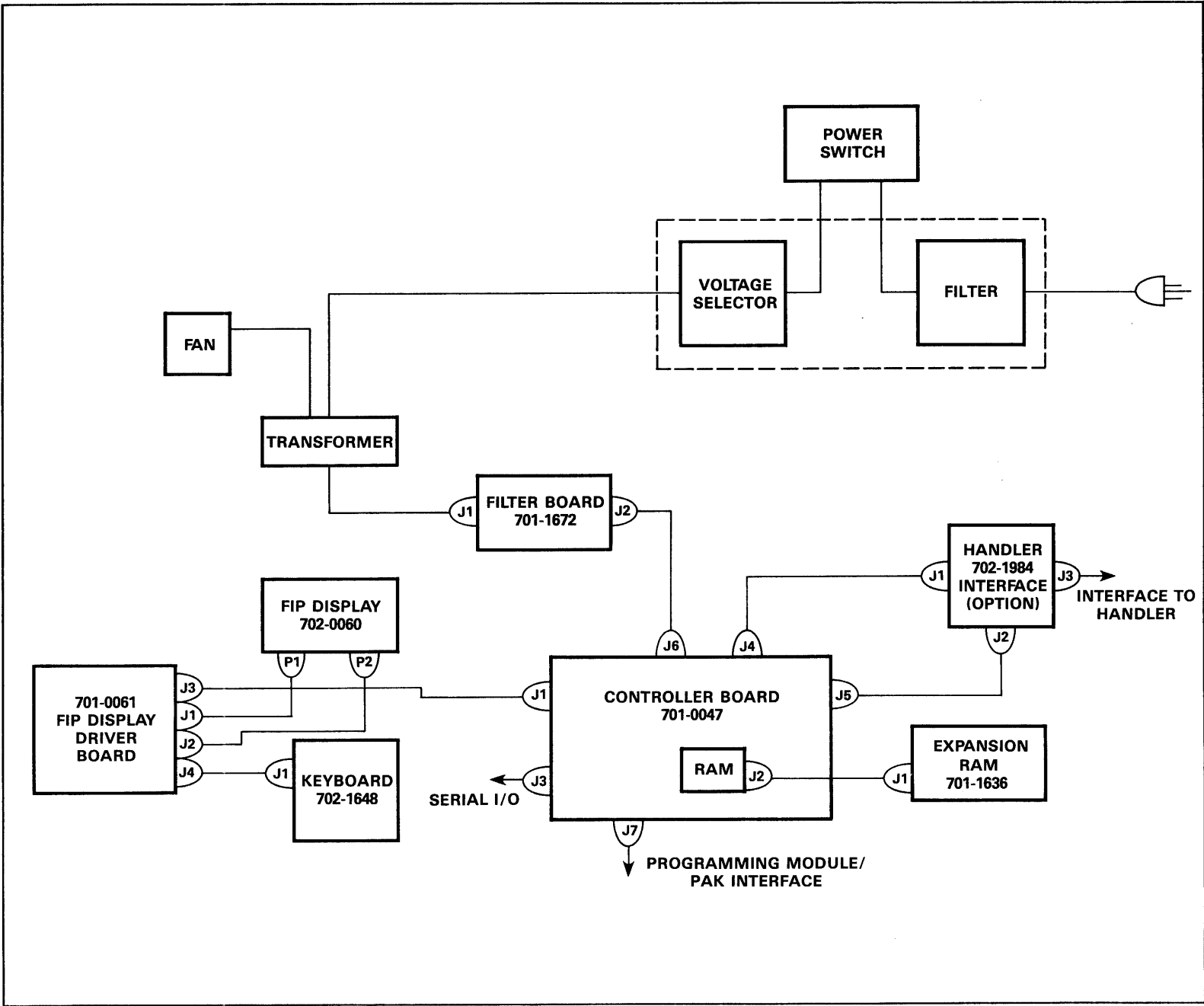


Figure 5-2. Interconnect Diagram for 29B Universal Programmer

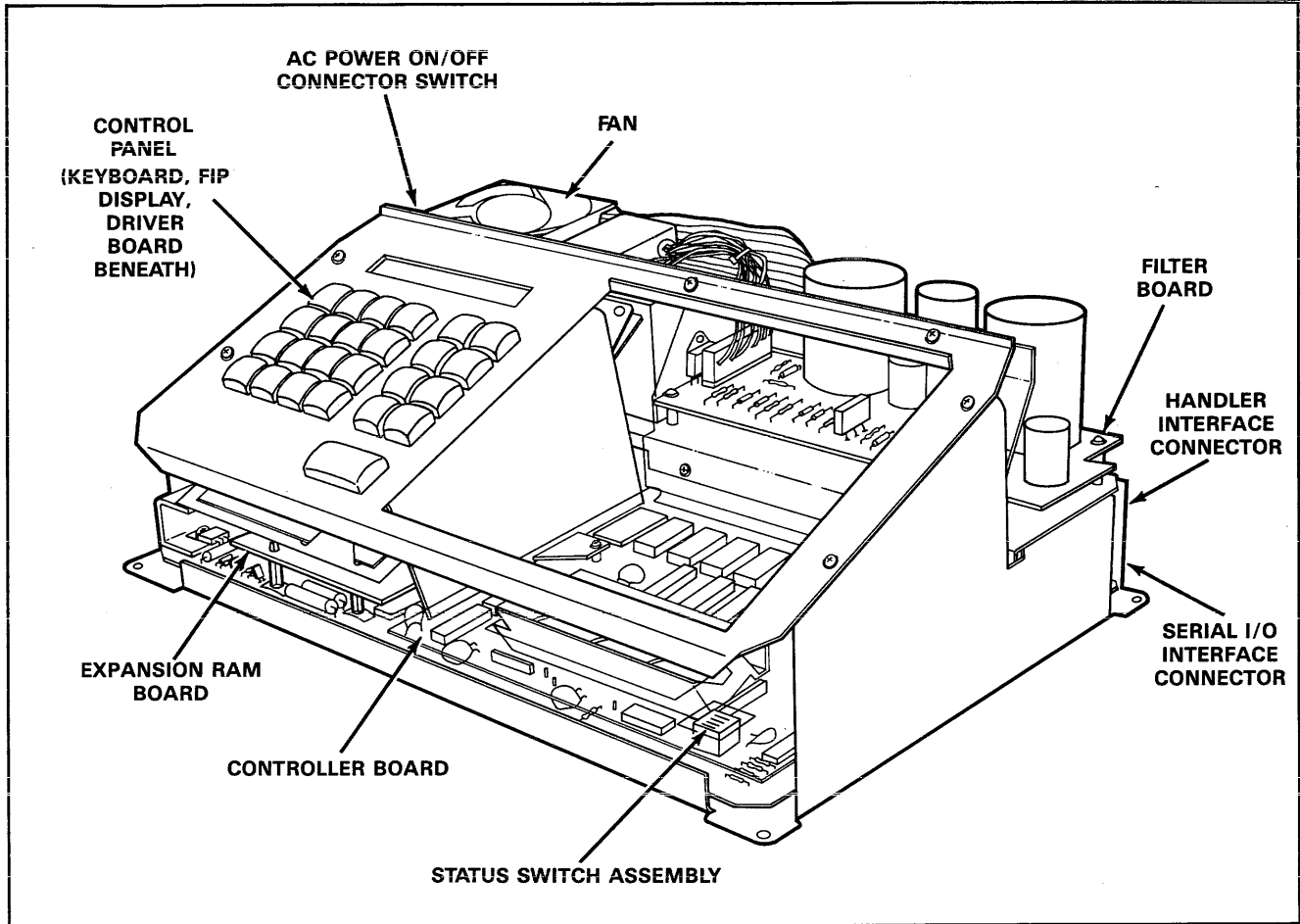


Figure 5-3. Circuit Board Locations

5.3.1 POWER SUPPLIES

Figure 5-4 is a block diagram of the power supply circuitry.

Voltage Selector/Filter

The voltage selector/filter unit provides circuit protection, filtering and provides the ability to tap different voltages from the transformer.

Transformer

The power transformer has multiple primary windings to allow the use of various input line voltages. The secondary develops appropriate voltages for the rectifiers and filters.

Filter Board

Four rectifier and capacitor filter networks on the filter board provide the dc voltages for the various regulators. PNP transistors Q1 and Q2 turn off to isolate +40 and +HV on high voltage shutdown.

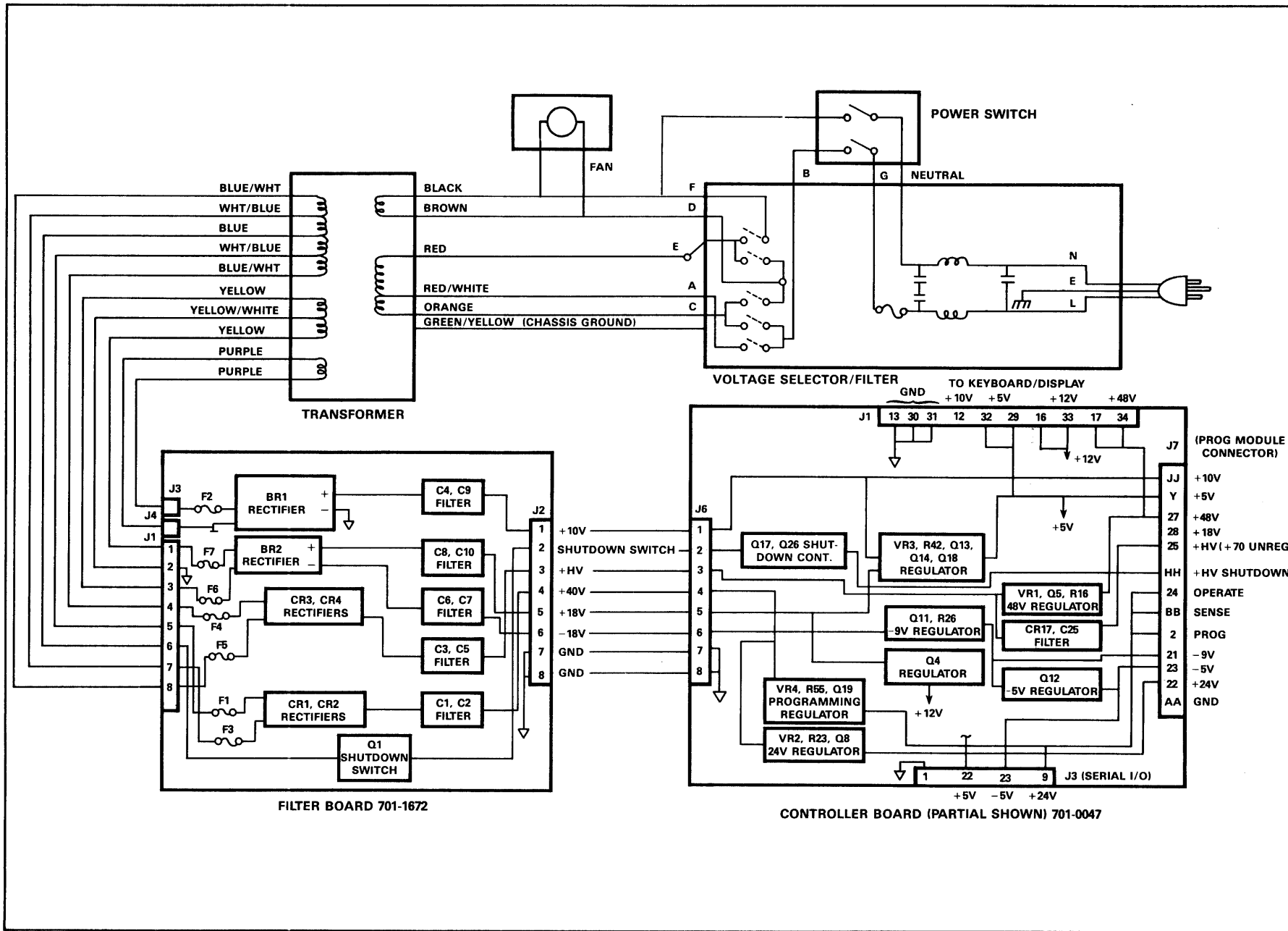
Voltage Regulator

The voltage regulator circuits are located on the controller printed circuit board.

The +5-volt regulator consists of a TL430 shunt regulator (VR3) driving an MJE 240 (Q18) which in turn drives two TIP 35A pass transistors (Q13 and Q14). Feedback is provided to the TL430 by R42, the voltage adjusting potentiometer. Foldback current limiting is achieved by sensing both output current and output voltage. If an overcurrent exists, Q15 senses the increased voltage across R32 and reduces the base drive to the pass transistor, which drops the output voltage. When the output voltage goes below the CR18 Zener diode reference, the base current flows through CR19 to Q15, further dropping the output voltage. To reset the regulator from its foldback condition, input power must be removed long enough to discharge C37. If an over-voltage condition exists, CR20 will begin to conduct, causing Q16 to pull base current from Q18.

The 24-volt supply and the 48-volt supply work on the same principle as the 5-volt supply. The difference is that a current source, rather than an emitter follower, is used to supply base current to the pass transistors.

Figure 5-4. Power Supply Circuitry



The programmable supply (+PROG) uses a Darlington differential pair, Q24-Q25 and Q22-Q23, working into a current source, Q21. Pass transistor Q19 follows the current source voltage. A TL430 shunt regulator (VR4) provides a 5V reference for the plus input of the differential pair (Q24). Two feedback nodes are connected to the sense and and operate lines. By connecting various resistors to these lines, the output voltage can be "programmed" to any level between 5V and 40V. Foldback current limiting is provided in the same manner as in the +5V supply. The +12V, -9V, and -5V supplies use standard monolithic regulators.

The shutdown control signals the card to turn off +HV and +40V unregulated supplies when a programming module is removed. This in turn shuts down the +48V, +24V, +PROG and +70V supplies.

Fuses F1 through F7 on the filter board, in conjunction with the crowbar zeners on all supply outputs, protect the system electronics from an overcurrent condition on the supply lines. An overvoltage condition could occur from failure of power supply components.

5.3.2 KEYBOARD/DISPLAY

The keyboard/display consists of a hex keyboard, four mode keys, three source/destination keys, REVIEW key, START key and a 16-character, 14-segment display. See figure 5-5.

Display operation starts with four scan lines, RS0-RS3, from the keyboard interface IC on the controller. These scan lines continuously count in binary and are sent to the display driver board for decoding. Two demultiplexers (U3 and U4) decode the scan lines for use by the display grid drivers (U1 and U2). Data (six bits) associated with each scan count passes to a character generator PROM (U11) whose four-bit output is demultiplexed into 16 lines by a decoder (U12) and four quad flip-flops (U7 through U10). These outputs are used by the display anode drivers (U5 and U6).

Keyboard operation starts with three continuously counting scan lines from the controller which are demultiplexed and used to scan the keyboard. When a key is pressed, the signal passes back to the controller on one of the five return lines.

5.3.3 CONTROLLER

The controller PCB is shown in block diagram form in Figure 5-6. Each block will be discussed in this section.

The microprocessor drives the bus through the address buffers and data buffer. Control signals are also generated by the processor and sent through the buffers to the bus.

The bus consists of a 16-bit address bus, 8-bit data bus, power supply lines, and several control lines. All communications between portions of the circuitry are handled in the same manner over this bus. The timing of a write cycle is shown in figure 5-7, and the timing of a read cycle is shown in figure 5-8.

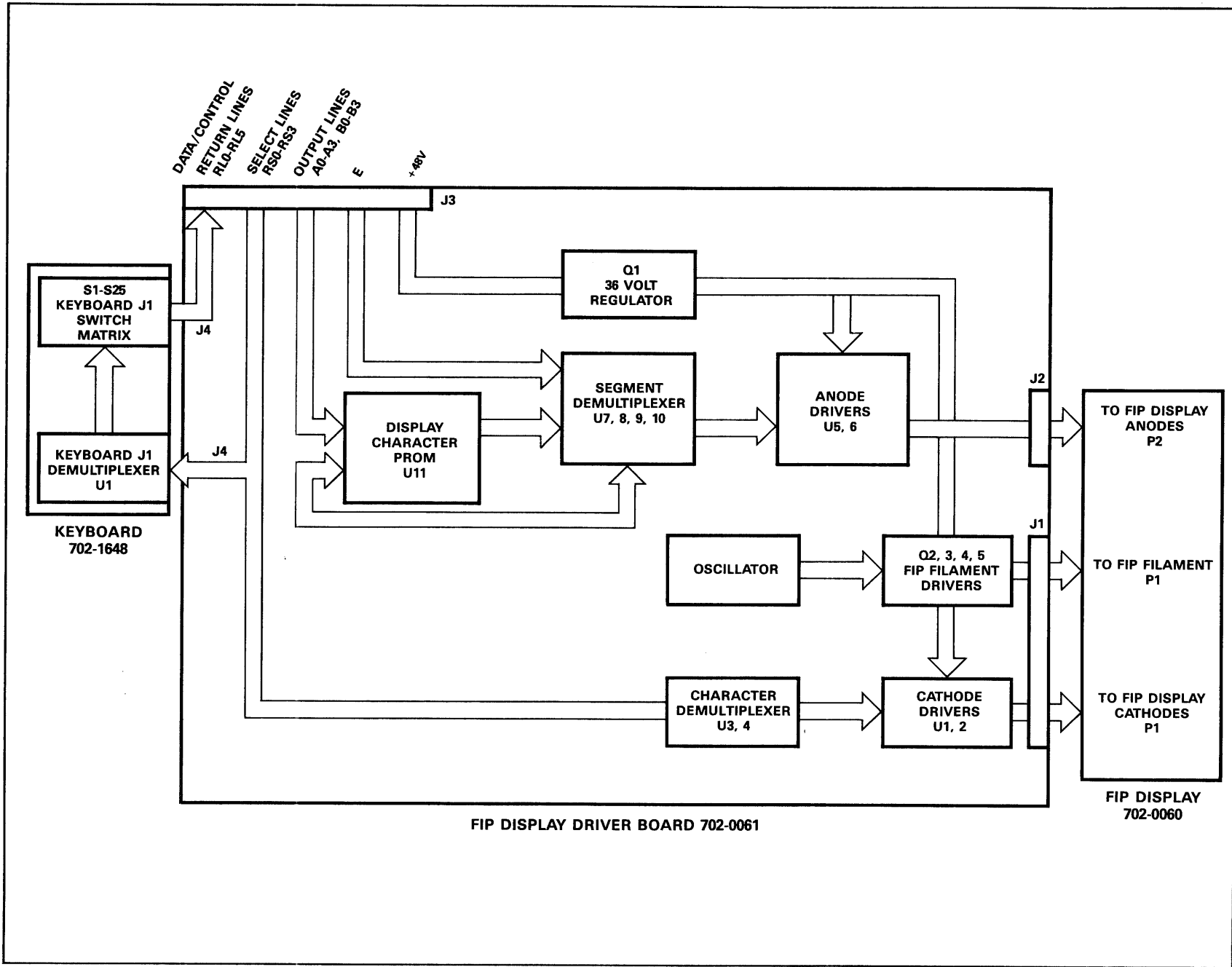
The buffered bus is available at the programming pak/module interface (J7).

Table 5-1 is the address map of the controller and shows the location, in hexadecimal, of each decoded function of the controller.

Table 5-1. Address Map

ADDRESS RANGE	FUNCTION
0000-07FF	Internal Scratch RAM (Controller U18,19,29,30)
E400-E7FF	Future Scratch RAM
0800-0FFF	Future Program Memory (½ U38)
1000-1FFF	Program Memory (U32)
2000-5FFF	Data Ram (16K pages from dynamic RAM board)
6000-9FFF	Expanded memory from programming modules
A000-DFFF	Program Memory (U33-U36)
E000-E0FF	Interface Control Register (U31)
E100-E1FF	I/O Status Switches (U52)
E200-E3FF	I/O Area:
E200	High Order Address Register (U51)
E201	Low Order Address Register (U42)
E202	Data Gate/Data Register (U44,47)
E203	Status Register/Status Gate (U45,46)
E204/E205	Kbd/Display Interface (U20)
E206/E207	Serial E/O (U14)
E208-E2FF	Not to be Used (Overwrites E200-E207)
E300/E301	Dynamic RAM Board - Bank & Page Register
E302-E3FF	Unassigned
E800-EFFF	Future Program Memory (½ U38)
F000-FFFF	Program Memory (U37)

Figure 5-5. Block Diagram FIP Display, Driver, and Keyboard



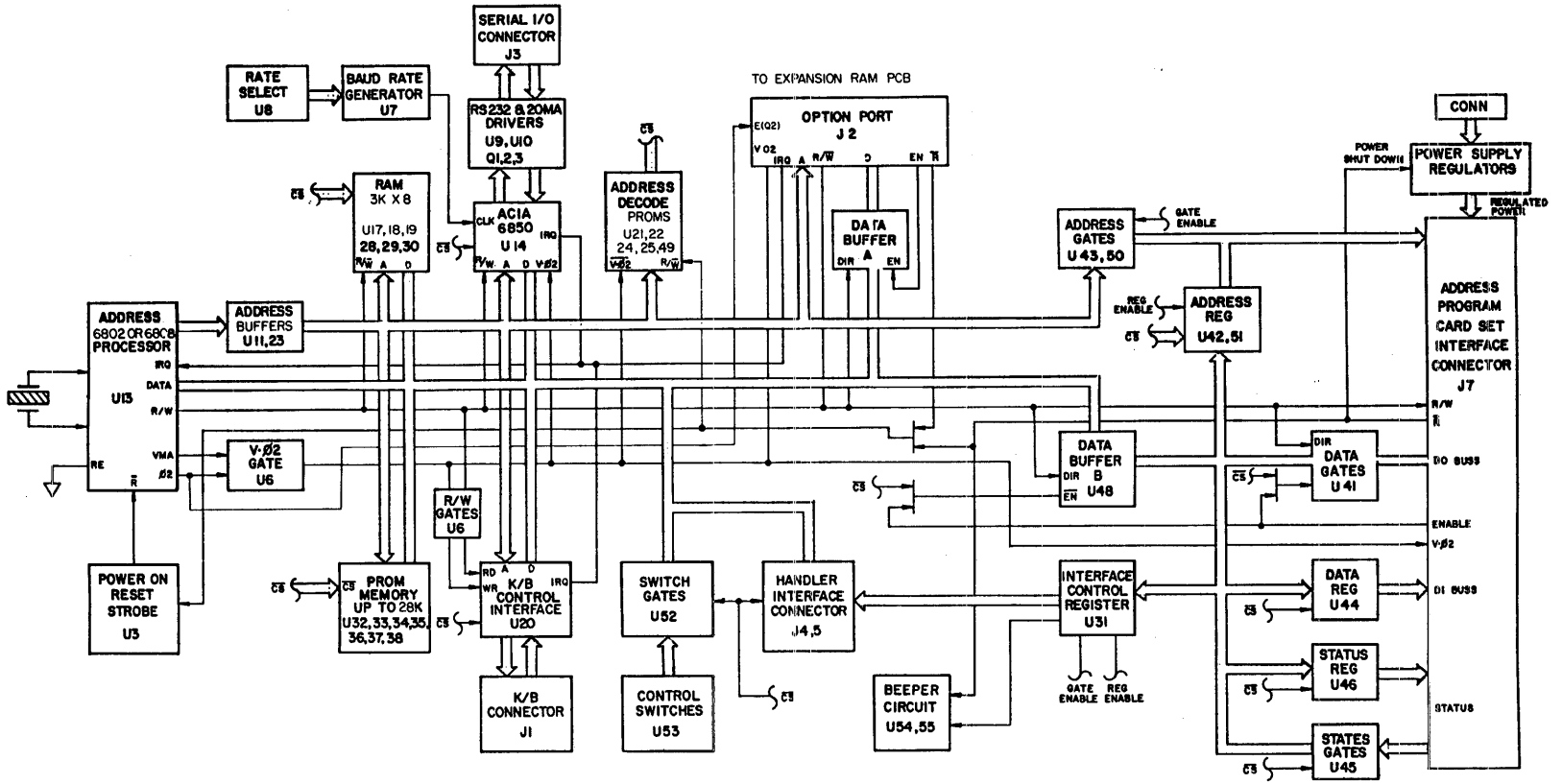


Figure 5-6. Block Diagram, Controller

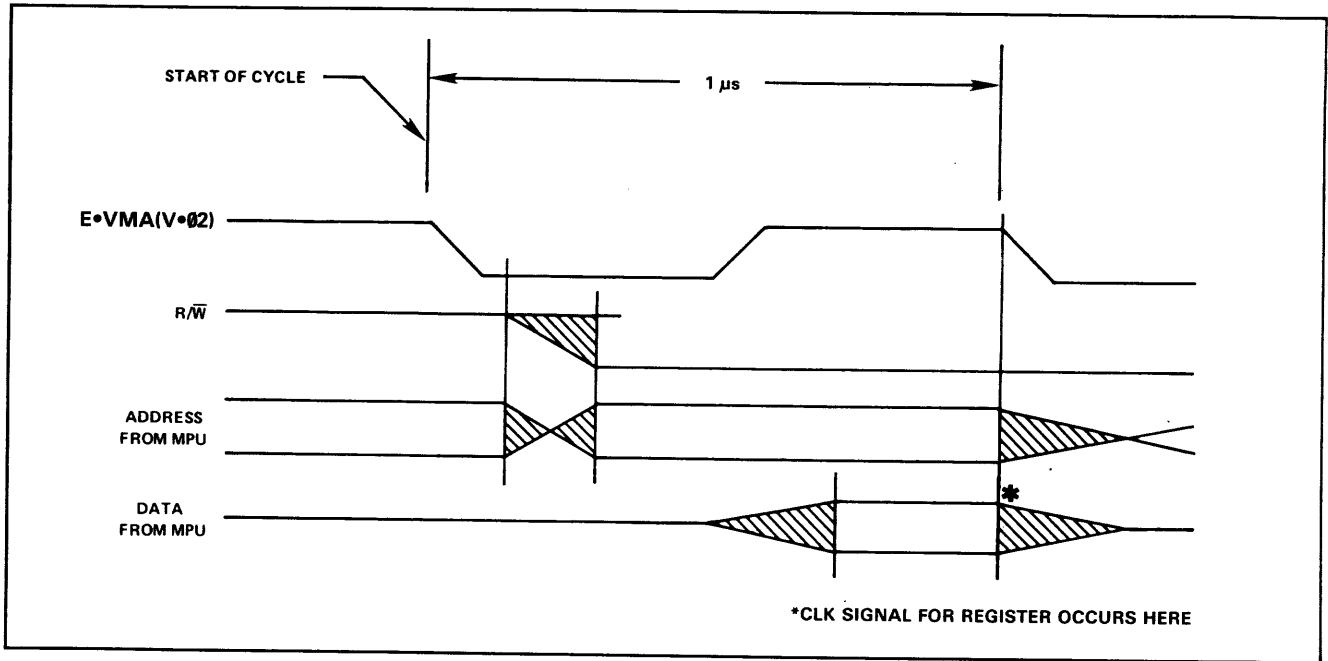


Figure 5-7. Write Timing

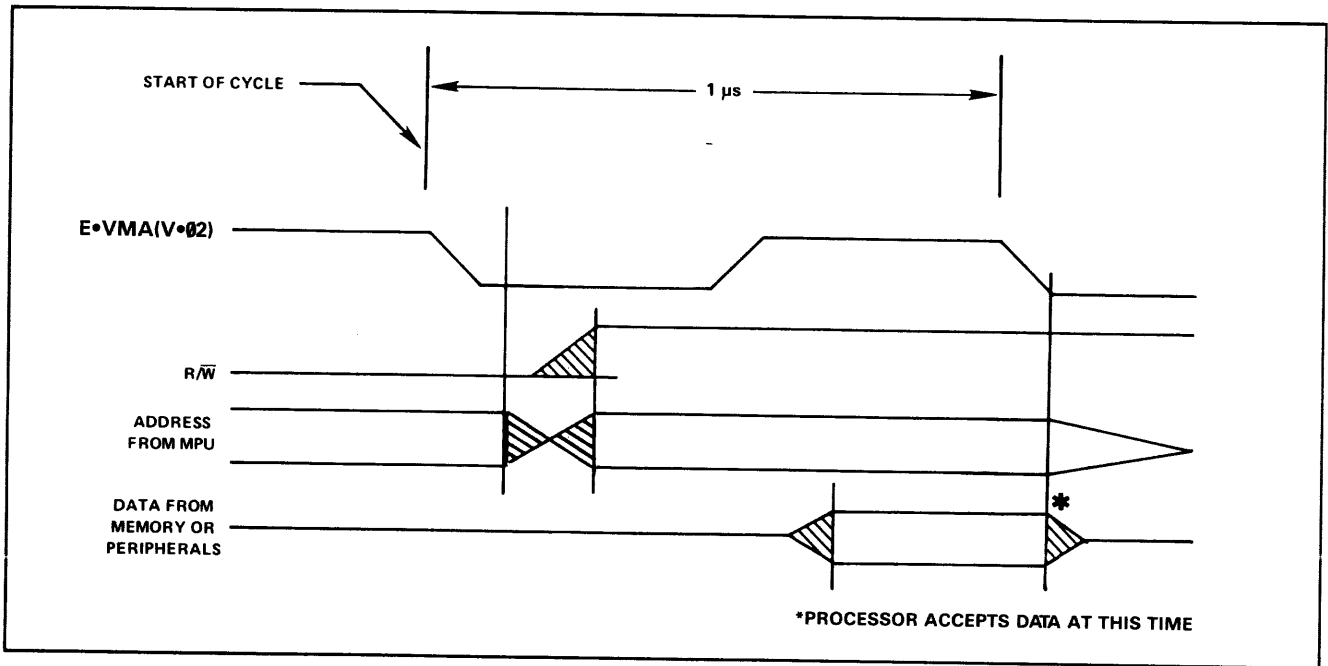


Figure 5-8. Read Timing

The decode PROMs provide selection of the various gates, registers and other devices connected to the data bus. Refer to the memory map of table 5-1. V_{cc2} is connected to the chip select of each decode PROM to provide the correct timing for writing to registers or reading gates or memory. R/\bar{W} is used for EPROM timing.

The programming module interface is provided by address registers, status gates, data register, data gates and control register.

When a programming module is removed, the microprocessor is held reset by a high on the line HH of J7. When the programming module is installed, line HH of J7 is grounded, removing the reset after a short delay. This feature allows programming modules to be changed with the power on in order to preserve RAM data.

Additional flexibility of the programming module interface can be gained with software control of the interface control register. The programming module interface can be set up so that the microprocessor bus is buffered and directly available at the port. This is accomplished by disabling the address register outputs, enabling the address gates in the outward direction, and connecting the data gate directly to

the R/W line. The data gate is enabled at the appropriate address by decoding done externally to the port over the Data Gate Enable line.

The serial interface is controlled by a 6850 Asynchronous Communications Interface Adapter (ACIA, U14) and appropriate software. The timing signal for the ACIA is provided by the baud rate generator U7. The baud rate is selected by rate select switch U8. The status switch provides for selecting parity and stop bits. The ACIA occupies two addresses (table 5-1) and uses IRQ to interface with the microprocessor.

The on-board program memory occupies up to 28K bytes of PROM U32-U38 decoded in 4K segments.

Temporary data storage on the controller board consists of 2K bytes of RAM (U17, U28) decoded in 1K segments.

Expansion RAM Board

The Expansion RAM Board (figure 5-9) allows the expansion of the base unit memory by up to 64K bytes. The RAM is in 16K byte banks with four banks per page. Future expansion of the board RAM will permit 128K of memory.

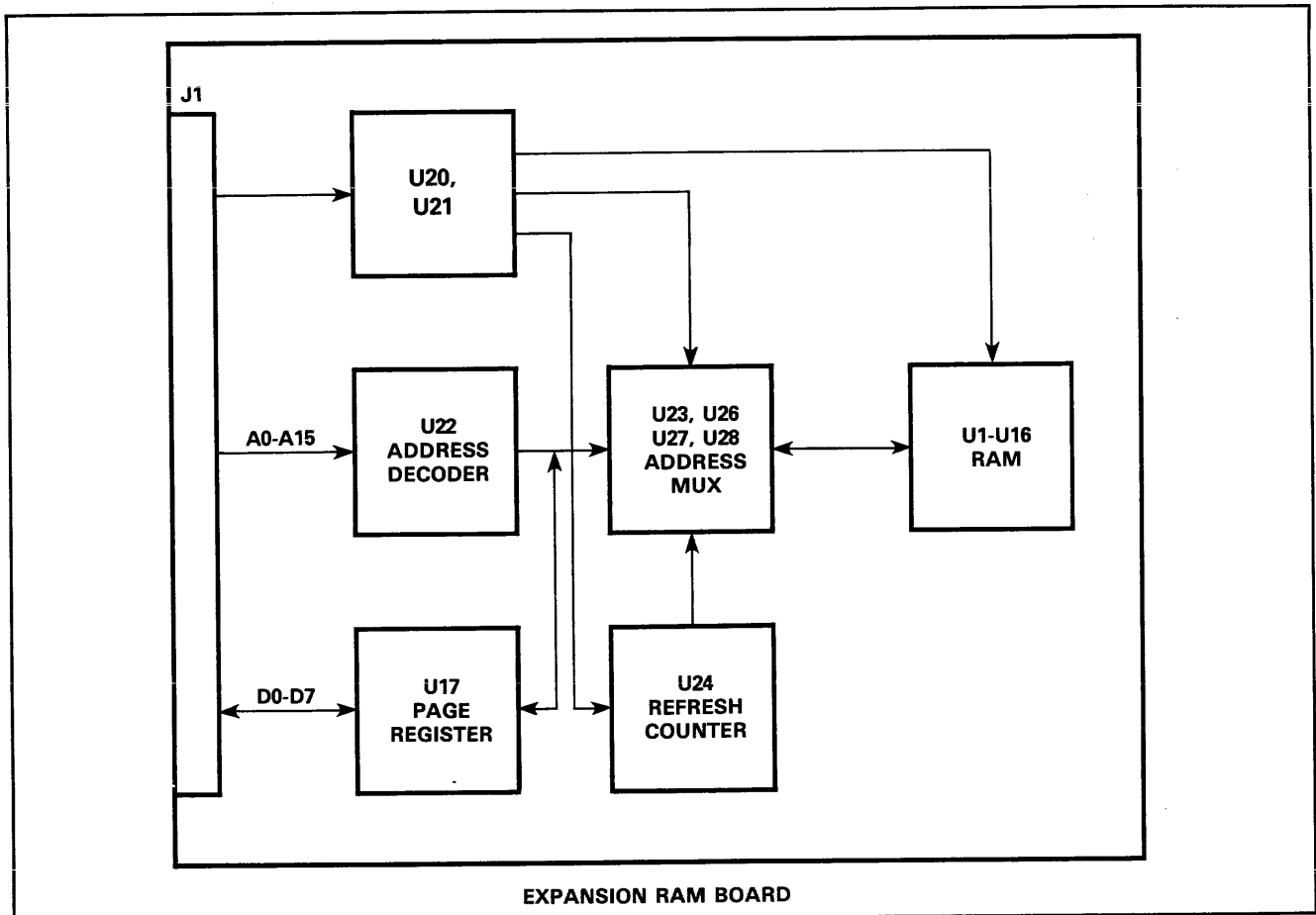


Figure 5-9. Expansion RAM Board Block Diagram

Timing is generated using system timing in conjunction with a tapped digital delay line (U20), with a PAL (U21) as the combinational logic.

Address decoding is performed by PAL U22. U17 serves as the bank and page register, with D6 and D7 at address E300 controlling the bank select, and D0 at address E301 controlling the page select.

Address multiplexing is accomplished via U23, U26, U27 and U28. Multiplex timing is generated by the delay line/PAL timing generator.

The board uses invisible refresh for the RAMs. During the first half of each 'E' cycle, the RAMs are refreshed. Refresh timing is generated by the delay line/PAL timing generator. The refresh address is generated by U24.

64K or 128K (future) configuration is selected by the programming of PAL, U22.

The D-RAM array is addressed at 2000 through 5FFF, hexadecimal.

Handler Interface Board (Optional)

The handler interface board (Figure 5-10) is comprised of a 3-to-8 decoder (U1) which receives binning information from the controller register U31 and outputs this information to the handler interface connector at the back panel of the 29B. Also at this interface a strobe signal is used to notify the handler when the binning information is stable.

Only one of four possible binning signals are sent at any one time: pass bin, illegal bit bin, program error bin, or verify error bin.

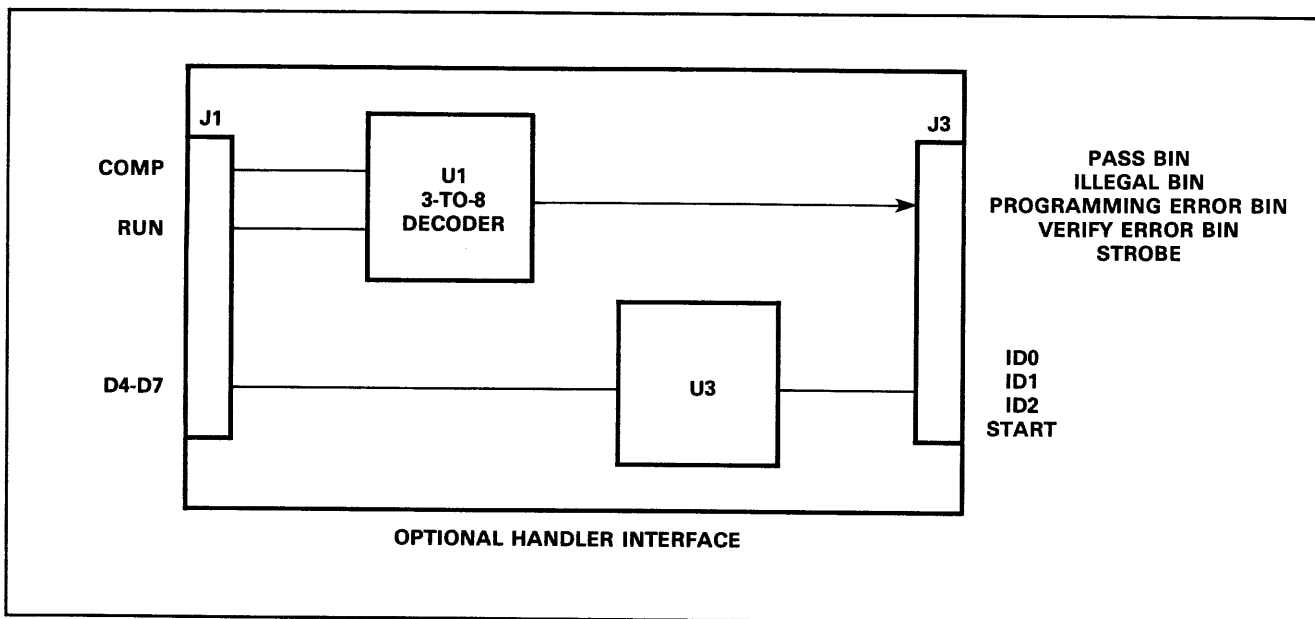


Figure 5-10. Optional Handler Interface, Block Diagram

APPENDIX A

DATA TRANSLATION FORMATS

A.1 INTRODUCTION

This appendix defines the data translation formats available for the 29B. The 29B is capable of interfacing with all RS232C serial equipment employing a data translation format described in this appendix.

Each data translation format is assigned a 2-digit code which the operator enters into the programmer (from the keyboard or, in remote control, through the serial port) to send or receive data in that format. In addition to the data translation format code, there is a 1-digit instrument control code which specifies control characters to be transmitted to, or received from, peripheral instruments.

In several cases, the 29B's standard display symbols will be shortened to accommodate large address fields used with some translation formats. These are:

- Copy Ram to Port
 FORMAT: HP 64000 Absolute (Format #89)
 DISPLAY: RAM> PORvZZZZZZ

Address
Offset

- Copy RAM to Port
 FORMAT: Motorola Exormax (Format #87)
 DISPLAY: RAM> PORvZZZZZZ

Address
Offset

- Copy Port to RAM
 FORMAT: HP 64000 Absolute (Format #89)
 DISPLAY: POvZZZZZZ/YYYY

Address
Offset Size Block

- Copy Port to RAM
 FORMAT: Motorola Exormax (Format #87)
 DISPLAY: POv—ZZZZZ/YYYY

Address
Offset Size Block

- Pressing REVIEW to review I/O parameters.
 FORMAT: HP 64000 Absolute* and Motorola Exormax
 DISPLAY: XXXX/YYYY>ZZZZZ

RAM Begin
Offset Size Block Address

A.2 DATA VERIFICATION

For data verification the 29B calculates a sum-check of all data sent to or from the programmer. At the end of a successful input operation, the programmer will display the sum-check of all data transferred. It will also compare any received sum-check fields with its own calculation. If the two agree, the programmer will display the sum-check; a mismatch will produce an error message. Output data is always followed by a sum-check field which may be printed on disk or tape for use in subsequent input operations.

* In the HP format, ZZZZZZ represents the six least significant digits in the 8-digit address field. If either of the two most significant digits in the field is not zero, the display will show PORT instead of the address. To view the address, reinitiate the key sequence for the input or output operation.

A.3 CODES

Each format is assigned a 2-digit data translation format code which the operator enters to instruct the programmer as to which format to use. In addition to this code, a 1-digit instrument control code may be used to specify control characters for peripheral equipment. The codes must be formatted as shown in figure A-1. If no codes are entered into the programmer, the current default values will be in effect.

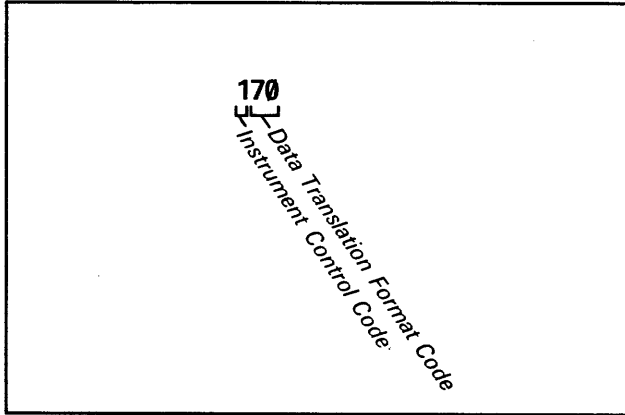


Figure A-1. Formatting the Instrument Control Code and Data Translation Format Code

See table A-1 for a definition of instrument control codes and table A-2 for a definition of data translation format codes.

Table A-1. Instrument Control Codes

CONTROL CODE	PROGRAMMER ACTION
0	Sends data immediately and continuously until acknowledging a "reader off" code. It will then stop sending data until receiving a "reader on" code. Sending no control codes results in normal, uninterrupted transmission.
1	Sends "reader on" (ASCII DC1/Hex 11) when ready to receive data, and "reader off" (ASCII DC3/Hex 13) when all data is received. Also send "punch on" (ASCII DC2/Hex 12) before sending data, and "punch off" (ASCII DC4/Hex 14) after sending data.
2	Sends data after acknowledging a "reader on" (ASCII DC1/Hex 11), and stops sending data after acknowledging a "reader off" (ASCII DC3/Hex 13).

Table A-2. Data Translation Formats

FORMAT	CODE
Binary	10
DEC Binary	11
ASCII-BNPF	01 (05)*
ASCII-BHLF	02 (06)*
ASCII-B10F	03 (07)*
5-Level BNPF	08 (09)*
Spectrum	12 (13)*
ASCII-Octal (Space)	30 (35) +
ASCII-Octal (Percent)	31 (36) +
ASCII-Octal (Apostrophe)	32
ASCII-Octal SMS	37
ASCII-Hex (Space)	50 (55) +
ASCII-Hex (Percent)	51 (56) +
ASCII-Hex (Apostrophe)	52
ASCII-Hex SMS	57
ASCII-Hex (Comma)	53 (58) +
RCA Cosmac	70
Fairchild Fairbug	80
MOS Technology	81
Motorola Exorciser	82
Intel Intellec 8/MDS	83
Signetics Absolute Object	85
Tektronix Hexadecimal	86
Motorola Exormax	87
Intel MCS-86 Hexadecimal Object	88
Hewlett-Packard 64000 Absolute	89
Texas Instruments SDSMAC	90

* For transmission of data without start codes, these alternate data translation format codes are used.
+ For transmission of data with the SOH (CTRL A) start code, these alternate data translation format codes are used.

A.4 LEADER AND NULL OUTPUT

A leader is a string of characters that is attached to the beginning and end of a data file. It is used to separate different files from one another and allows extra room which may be necessary for loading and unloading the data medium to or from equipment. For the 29B, the leader is sent at the beginning and end of a data output operation. With one exception, this leader will always be comprised of carriage return [CR], a line feed [LF], and 50 nulls in succession.

Null count is the number of null characters in a string of characters between each record or line within a file. What actually comprises a data record depends upon the format that is being used. Records and lines can basically be thought of as separations of data within a file.

Null count is a parameter which can be defined by the 29B user for use with printers with a slow carriage return response time. The number of nulls can be set to any value from zero to 254 decimal (FE hexadecimal). With one exception, the string of characters actually sent between each and every record or line of the file includes a carriage return [CR], a line feed [LF], and the number of nulls defined by the null count.

The exception referred to above for the leader and the null count occurs when the user defines the null count equal to the value of "FF" hexadecimal (or 255 decimal). In this case, the leader is made up of solitary carriage return (no line feed and no nulls). Also, the string separating the records of the file is a carriage return (no line feeds and no nulls).

Parity for the beginning and end leader is the same as the parity for the data within the file. The same is true for the carriage return [CR], line feed [LF] and nulls separating the records or lines of the file. They have the same parity as the data. However, it should be noted that binary formats (10, 11 and 89) do not have parity for their data; therefore, the leader, the carriage return, line feed, and nulls separating the records for these files have no parity.

A.5 TRANSLATION FORMATS

This section gives information on the translation formats available for input and output by the 29B.

A.5.1 BINARY TRANSFER, CODE 10

Data transfer in the Binary format consists of a stream of 8-bit data words preceded by a byte count and followed by a sum-check. The Binary format does not have addresses.

A paper tape generated by a programmer will contain a 5-byte, arrow-shaped header followed by a null and a 4-nibble byte count. The start code, an 8-bit rubout, follows the byte count. The end of data is signalled by two nulls and a 2-byte sum-check of the data field. Refer to figure A-2.

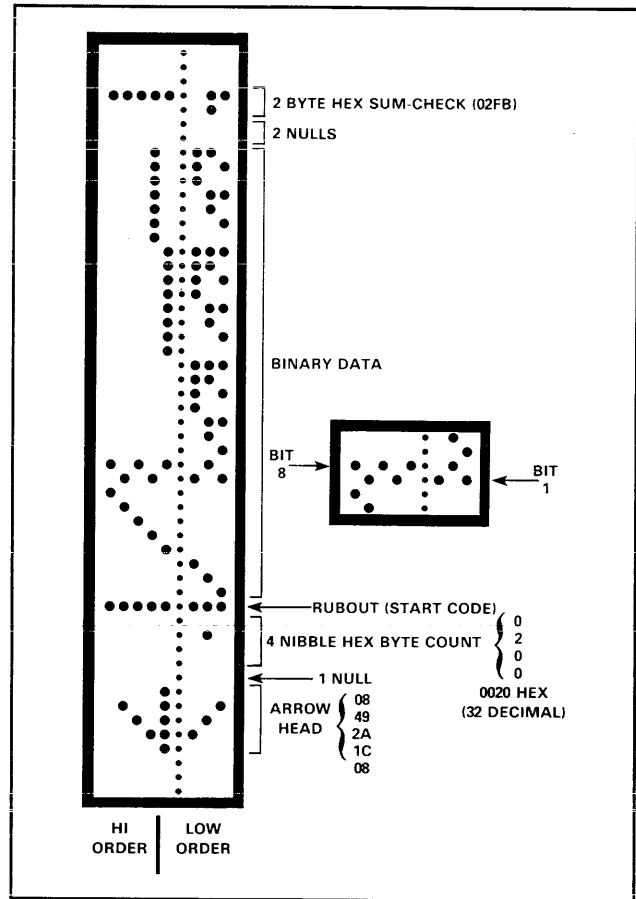


Figure A-2. Input or Output Binary Tape

The programmer stores incoming binary data upon receipt of the start character. Data is stored in RAM starting at the first RAM address and ending at the last incoming data byte. Transmission may be aborted by pressing any mode key.

A.5.2 DEC BINARY FORMAT, CODE 11

Data transmission in the DEC Binary format is a stream of 8-bit data words with no control characters except the start code. The start code is one null preceded by at least one rubout. A tape output from the programmer will contain 32 rubouts in the leader. The DEC Binary format does not have addresses.

A.5.3 ASCII BINARY FORMAT, CODES 01, 02 AND 03 (OR 05, 06, AND 07)

In these formats, bytes are recorded in ASCII codes with binary digits represented by N's and P's, L's and H's, or 1's and 0's, respectively. See figure A-3. The ASCII Binary formats do not have addresses.

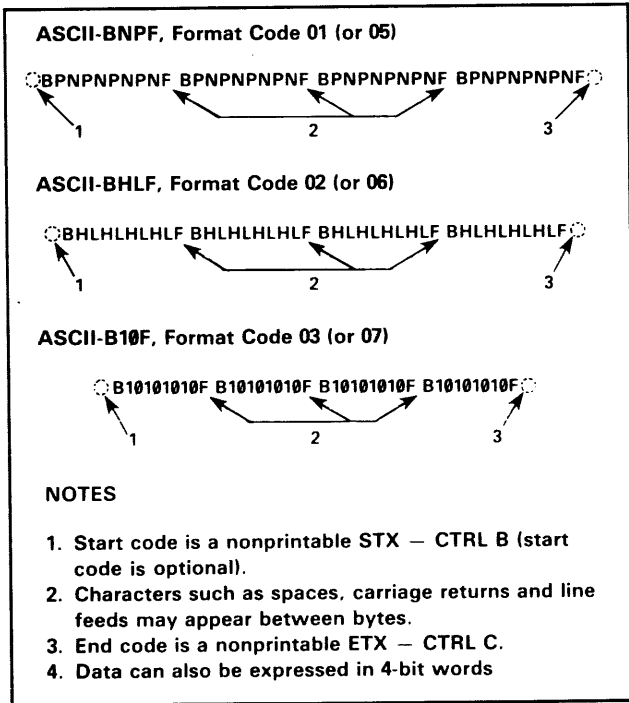


Figure A-3. ASCII Binary Formats

Figure A-3 shows four data bytes coded in each of the three ASCII Binary formats. Incoming bytes are stored in RAM sequentially starting at the first RAM address. Bytes are sandwiched between "B" and "F" characters and are normally separated by spaces. Data can also be expressed in 4-bit words. Any other characters, such as carriage returns or line feeds, may be inserted between an "F" and the next "B". The start codes are a nonprintable STX, control B (or hex 02), and the end code is a nonprintable ETX, control C (or a hex 03).

NOTE

Data without a start code may be input to or output from the programmer by use of alternate data translation format codes. These are: ASCII-BNPF, 05; ASCII-BHLF, 06; ASCII-B10F, 07.

A single data byte can be aborted if the programmer receives an E character between B and F characters. Data will continue to be stored in sequential RAM addresses. The entire data transfer can be aborted by pressing any mode key (COPY, VERIFY, SELECT, EDIT).

Data is output in 4-byte lines with a space between bytes.

A.5.4 THE 5-LEVEL BNPF FORMAT, CODES 08 OR 09

Except for the start and end codes, the same character set and specifications are used for the ASCII-BNPF and 5-level BNPF formats.

Data for input to the programmer is punched on 5-hole Telex paper tapes to be read by an ASCII-based reader that has an adjustable tape guide. The reader reads the tape as it would an 8-level tape, recording the 5 holes that are on the tape as 5 bits of data. The 3 most significant bits are recorded as if they were holes on an 8-level tape. The programmer's software converts the resulting 8-bit codes into valid data for entry in RAM.

The start code for the format is a left parenthesis, ("Figs K" on a Telex machine), and the end code is a right parenthesis, ("Figs L" on a Telex machine). The 5-level BNPF format does not have addresses.

NOTE

Data without a start code may be input to or output from the programmer by use of the alternate data translation format code, 09.

A.5.5 SPECTRUM FORMAT, CODES 12 OR 13

In this format, bytes are recorded in ASCII codes with binary digits represented by 1's and 0's. Each byte is preceded by an address.

Figure A-4 shows 2 data bytes coded in the Spectrum format. Bytes are sandwiched between the space and carriage-return characters and are normally separated by line feeds. The start code is a nonprintable STX, control B (or hex 02), and the end code is a nonprintable ETX, control C (or hex 03).

NOTE

Data without a start code may be input to or output from the programmer by use of the alternate data translation format code, 13.

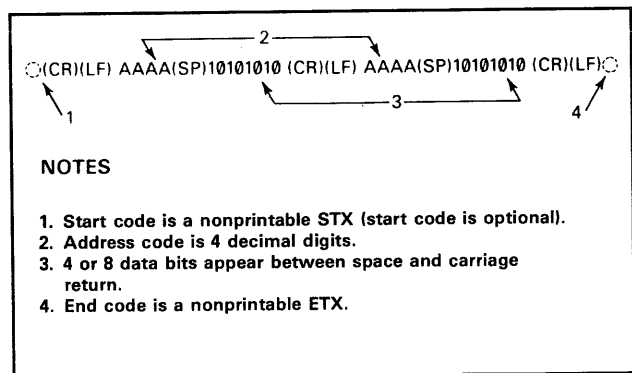


Figure A-4. Spectrum Format

A single data byte can be aborted if the programmer receives an "E" character between a space and a carriage return. Data will continue to be stored in sequential RAM addresses. The entire data transfer can be aborted by pressing any mode key (COPY, VERIFY, SELECT or EDIT).

Data output to a printer will have one address and one byte of data on each line. The programmer first sends an STX (optionally), then the data, and finally an ETX.

A.5.6 ASCII OCTAL AND HEX FORMATS, CODES 30-37 AND 50-58

Each of these formats has a start and end code, and similar address and sum-check specifications. Figure A-5

illustrates 4 data bytes coded in each of the 9 ASCII-Octal and Hex formats. Data in these formats is organized in sequential bytes separated by the execute character (space, percent, apostrophe, or comma). Characters immediately preceding the execute character are interpreted as data. ASCII-Hex and Octal formats can express 8-bit data, by 2 or 3 octal, or 1 or 2 hex characters. Line feeds, carriage returns and other characters may be included in the data stream as long as a data byte directly precedes each execute character.

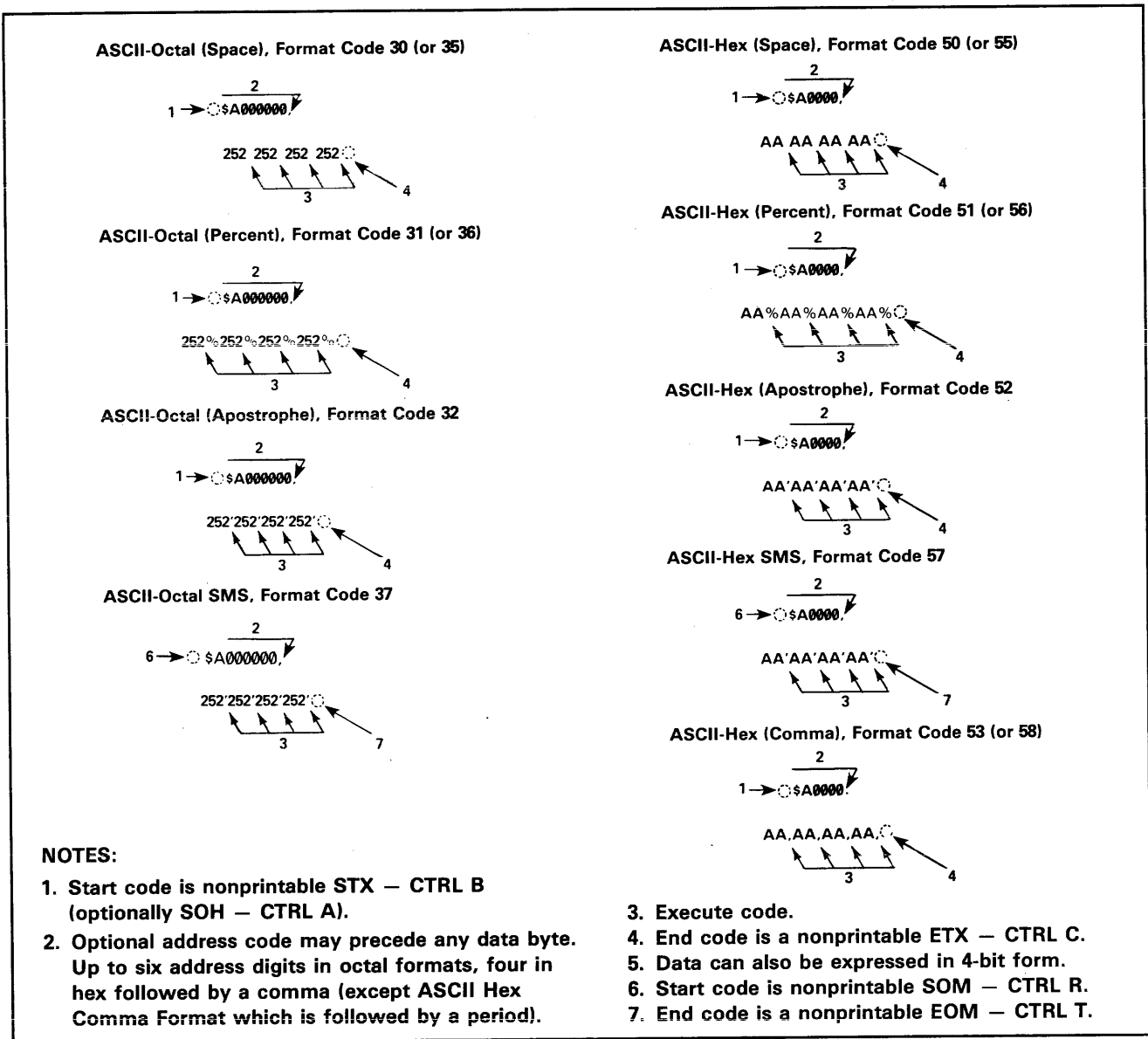


Figure A-5. ASCII-Octal and Hex Formats

Although each data byte has an address, most are implied. Data bytes are addressed sequentially unless an explicit address is included in the data stream. This address is preceded by a "\$" and an "A", must contain 2 to 4 hex or 3 to 6 octal characters, and must be followed by a comma, except for the ASCII-Hex (Comma) format, which uses a period. The programmer skips to the new address to store the next data byte; succeeding bytes are again stored sequentially. See figure A-6.

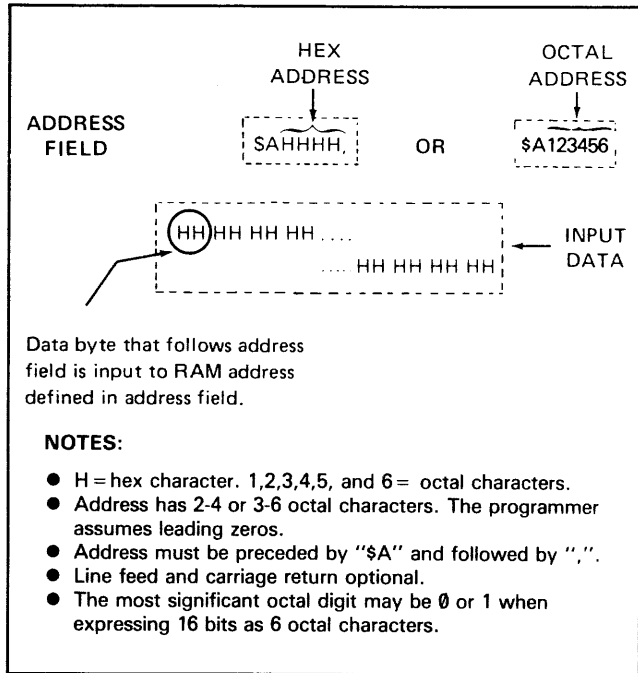


Figure A-6. Optional Address Field in ASCII-Octal and Hex Formats

Each format has an end code, which terminates input operations. However, if a new start code follows within 16 characters of an end code, input will continue uninterrupted.

After receiving the final end code following an input operation, the programmer calculates a sum-check of all incoming data. Optionally, a sum-check can also be entered in the input data stream. The programmer compares this sum-check with its own calculated sum-check. If they match, the programmer will display the sum-check; if not, a sum-check error will be displayed. Specifications for the optional sum-check are given in figure A-7.

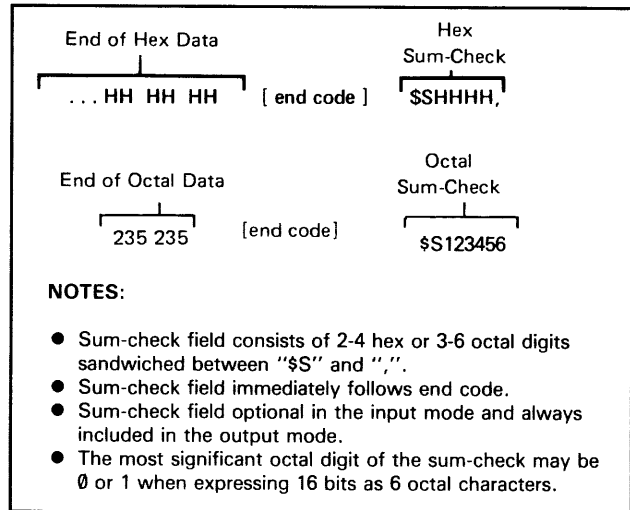


Figure A-7. Syntax of the Sum-Check Field in I/O Operations

Output is begun by invoking an output-to-port operation. The programmer divides the output data into 8-line blocks.

Data transmission is begun with the start code, a nonprintable STX, optionally SOH.* Data blocks follow, each one prefaced by an address for the first data byte in the block. The end of transmission is signalled by the end code, a nonprintable ETX. Directly following the end code is a sum-check of the transferred data.

* ASCII-Octal SMS and ASCII-Hex SMS use SOM (CTRL R) as a start code and EOM (CTRL T) as an end code.

Address specification is required for only the first data byte in the transfer. An address must have 1 to 4 hex characters and be followed by a space. The programmer records the next hex character after the space as the start of the first data byte. (A carriage return must follow the space if the start code ?M is used.) Succeeding bytes are recorded sequentially.

Each data record is followed by a comma if the next record is not preceded by an address, or by a semicolon if it starts with an address. Records consist of data bytes expressed as two hexadecimal characters and followed by either a comma or semicolon, and a carriage return. Any characters received between a comma or semicolon and a carriage return will be ignored by the programmer.

The carriage-return character is significant to this format because it can signal either the continuation or the end of data flow; if the carriage return is preceded by a comma or semicolon, more data must follow; the absence of a comma or semicolon before the carriage return indicates the end of transmission.

Output data records are followed by either a comma or a semicolon and a carriage return. The Start-of-File records are expressed exactly as for input.

• FAIRCHILD FAIRBUG, CODE 80

In the Fairbug format, input and output requirements are identical; both have 8-byte records and identical control characters. Figure A-9 simulates a Fairbug data file. A file begins with a 5-character prefix and ends with a one-character suffix. The Start-of-File character is an "S", followed by the address of the first data byte. Each data byte is represented by 2 hexadecimal characters.

NOTE

Address specification is optional in this format; a record with no address directly follows the previous record.

A 1-digit hexadecimal checksum follows the data in each data record. The checksum represents, in hexadecimal notation, the sum of the binary equivalents of the 16 digits in the record; the half carry from the fourth bit is ignored.

The programmer ignores any character (except for address characters) between a checksum and the start character of the next data record. These spaces can be used for any comments.

The last record consists of an asterisk only, which indicates the end of data transmission.

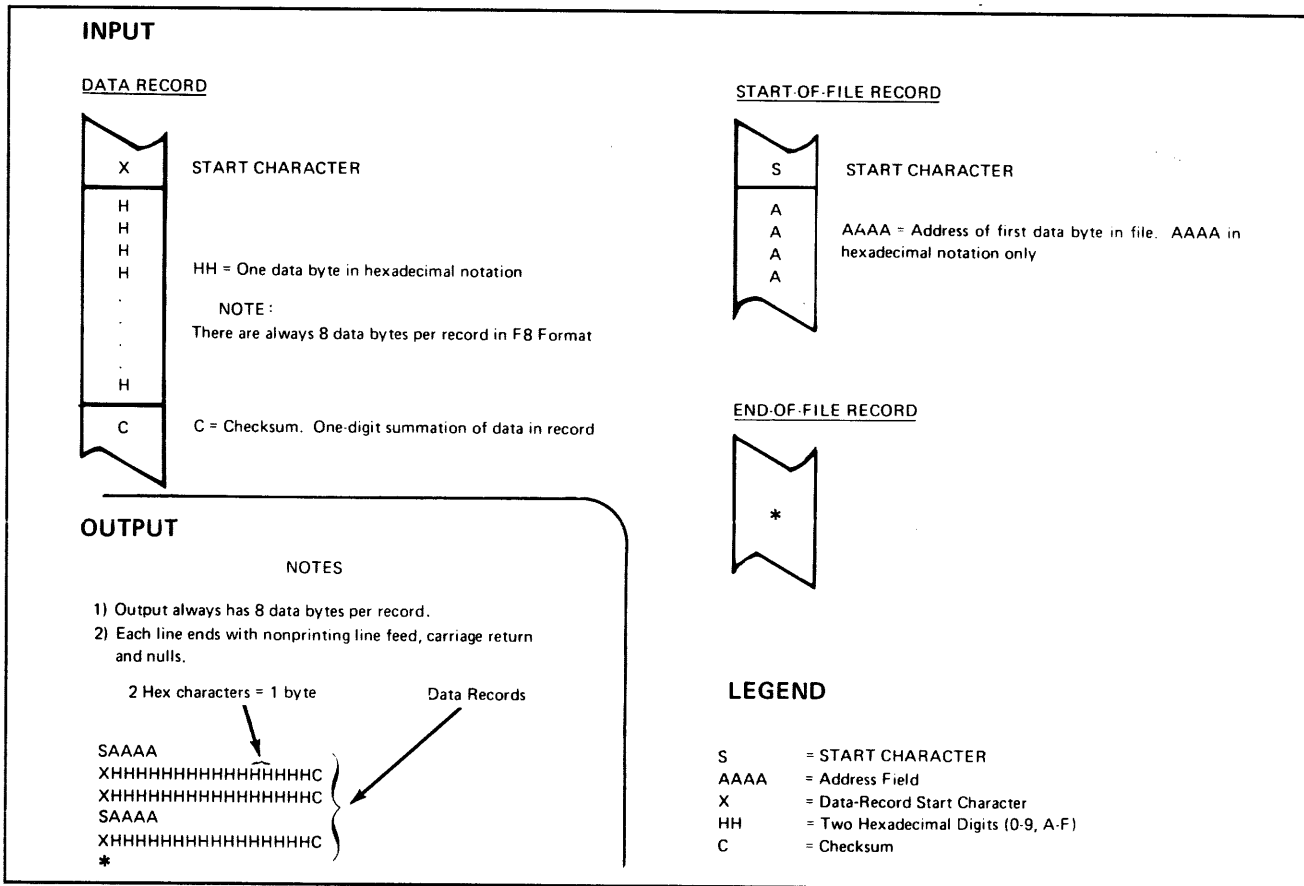


Figure A-9. Specifications for Fairchild Fairbug Data Files

• **MOS TECHNOLOGY FORMAT, CODE 81**

The data in each record is sandwiched between a 7-character prefix and a 4-character suffix. The number of data bytes in each record must be indicated by the byte count in the prefix. The input file can be divided into records of various length.

Figure A-10 simulates a series of valid data records. Each data record begins with a semicolon. The programmer will

ignore all characters received prior to the first semicolon. All other characters in a valid record must be valid hex digits (0-9, A-F). A 2-digit byte count follows the start character. The byte count, expressed in hexadecimal digits, must equal the number of data bytes in the record. The next 4 digits make up the address of the first data byte in the record. Data bytes follow, each represented by 2 hexadecimal digits.

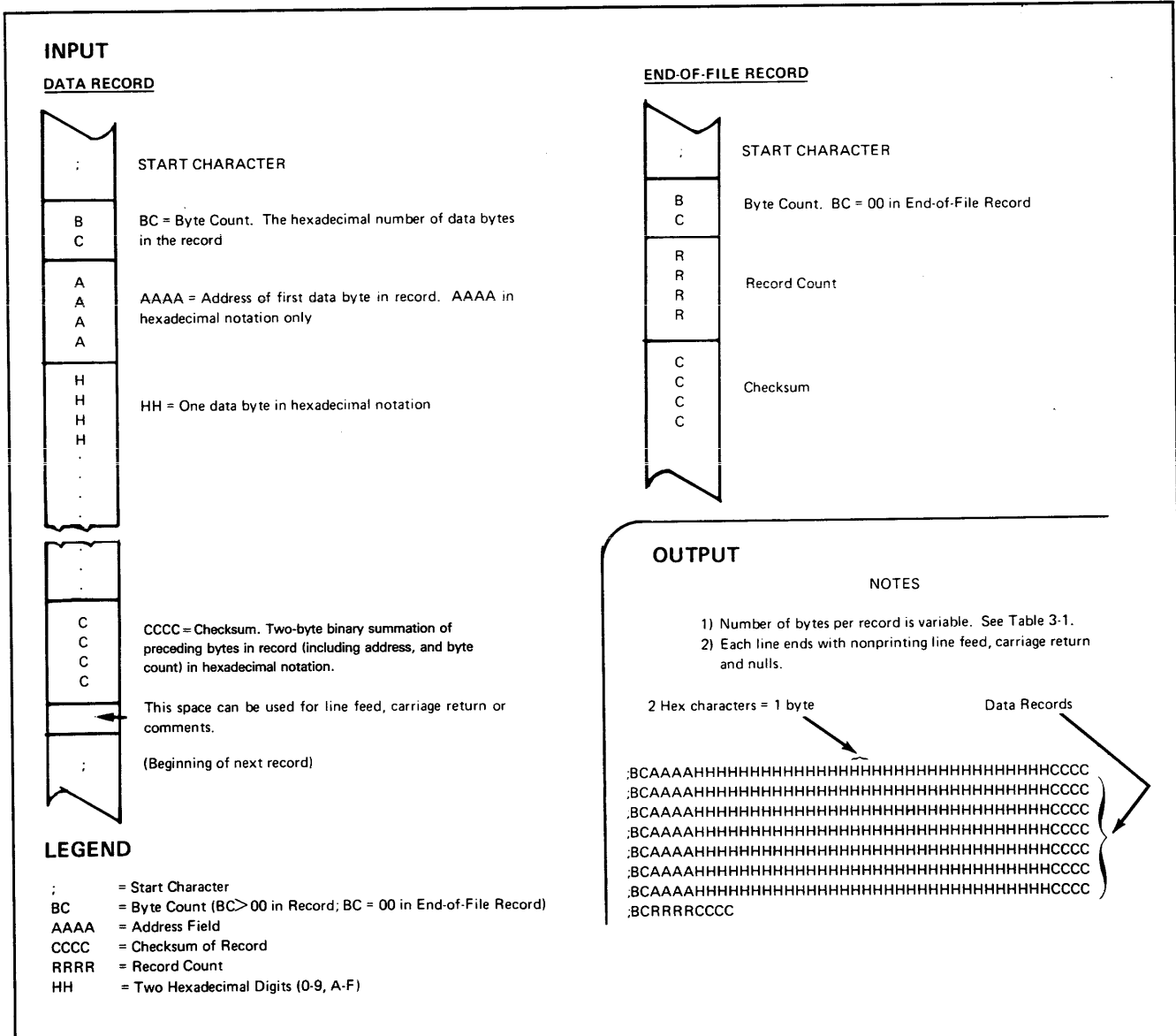


Figure A-10. Specifications for MOS Technology Data Files

• **MOTOROLA EXORCISER FORMAT, CODE 82**

Motorola data files may begin with a sign-on record, which is initiated by the code S0. Valid data records start with an 8-character prefix and end with a 2-character suffix. Figure A-11 demonstrates a series of valid Motorola data records.

Each data record begins with the start characters "S1"; the programmer will ignore all earlier characters. The third

and fourth characters represent the byte count, which expresses the number of data, address and sum-check bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix. Data bytes follow, each represented by 2 hexadecimal characters. The number of data bytes occurring must be three less than the byte count. The suffix is a 2-character checksum.

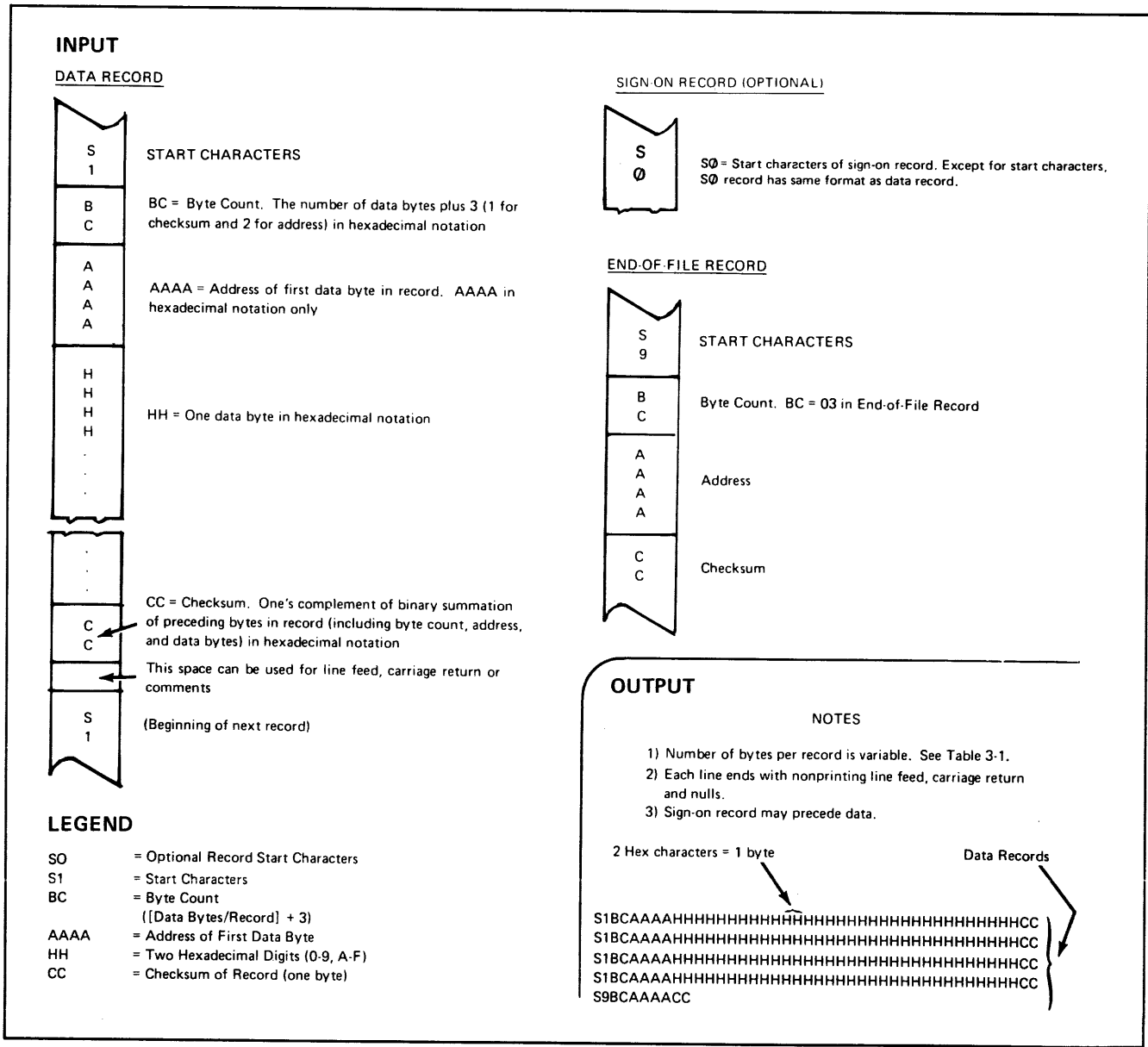


Figure A-11. Specifications for Motorola Data Files

• INTEL INTELLEC 8/MDS FORMAT, CODE 83

Intel data records begin with a 9-character prefix and end with a 2-character suffix. The byte count must equal the number of data bytes in the record.

Figure A-12 simulates a series of valid data records. Each

record begins with a colon, which is followed by a 2-character byte count. The 4 digits following the byte count give the address of the first data byte.

Each data byte is represented by 2 hex digits; the number of data bytes in each record must equal the byte count.

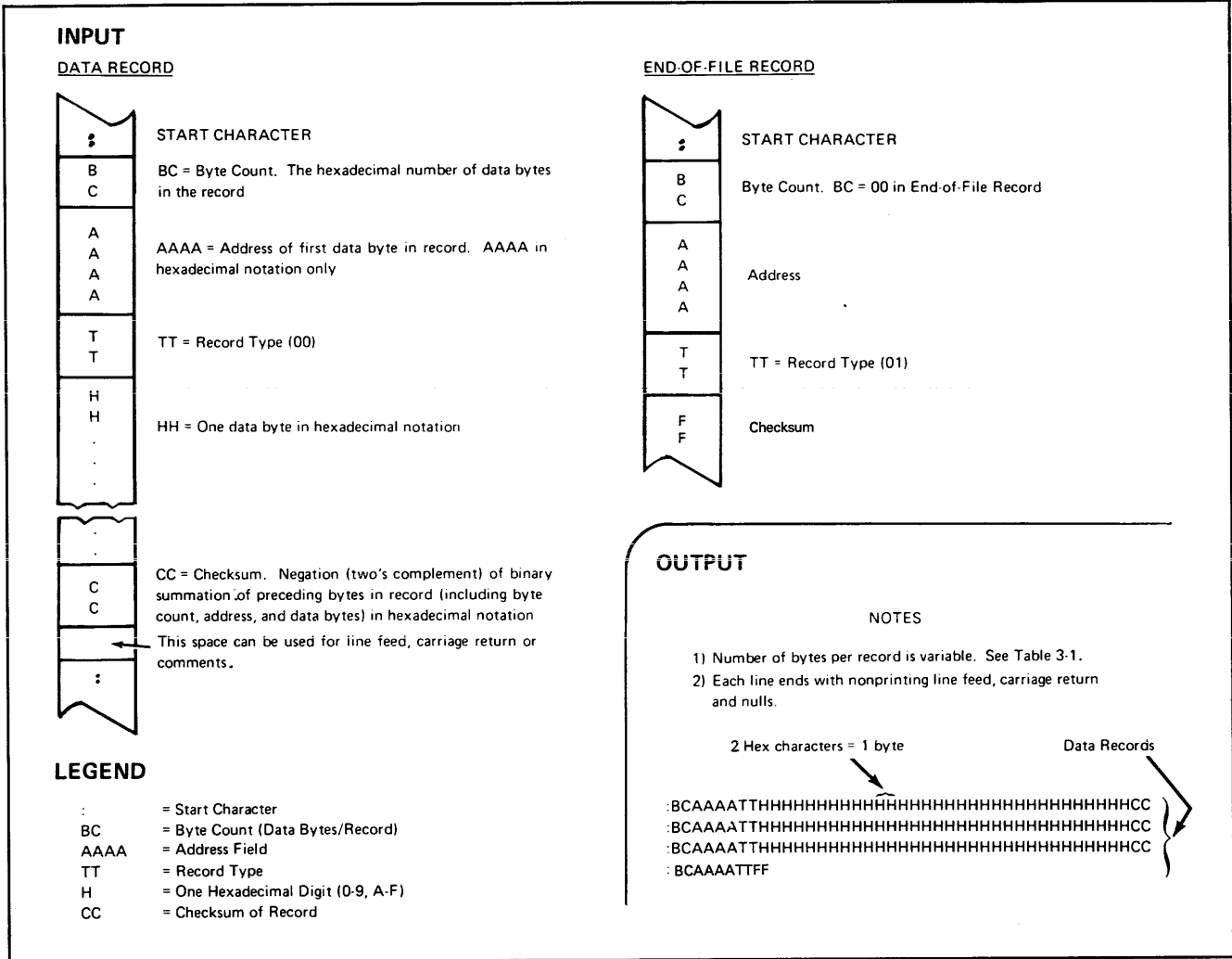


Figure A-12. Specifications for Intel Intellec 8/MDS Data Files

• SIGNETICS ABSOLUTE OBJECT FORMAT, CODE 85

Figure A-13 shows the specifications of Signetics format files. The data in each record is sandwiched between a 9-character prefix and a 2-character suffix.

The start character is a colon. This is followed by the address of the first data byte, the byte count, and a 2-digit

address check. Data is represented by pairs of hexadecimal characters. The byte count must equal the number of data bytes in the record. The suffix is a 2-character data check, calculated using the same operations described in figure A-13 for the address check.

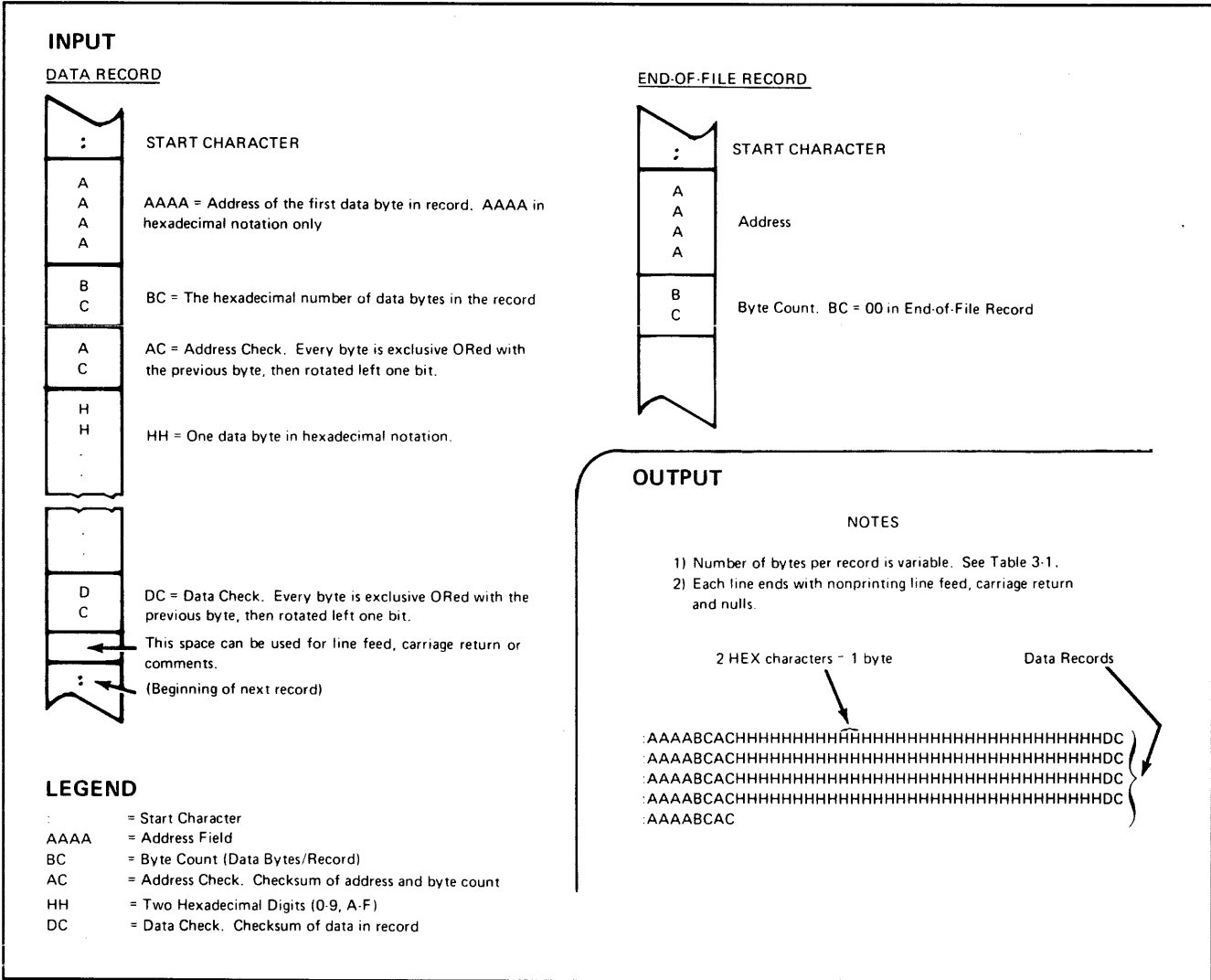


Figure A-13. Specifications for Signetics Absolute Object Data Files

• **TEKTRONIX HEXADECIMAL FORMAT, CODE 86**

Figure A-14 illustrates a valid Tektronix data file. The data in each record is sandwiched between the start character (a slash) and a 2-character sum-check. Following the start character, the next 4 characters of the prefix express the address of the first data byte. The address is followed by a byte count, which represents the number of data bytes in the record, and by a sum-check of the address and byte count. Data bytes follow, represented by pairs of hexadecimal characters and succeeded by a sum-check of

the data bytes. The End-of-File record consists only of control characters used to signal the end of transmission and a byte count and sum-check for verification.

Data is output from the programmer starting at the first RAM address and continuing until the number of bytes in the specified block have been transmitted. The programmer divides output data into records prefaced by a start character and an address field for the first byte in the record.

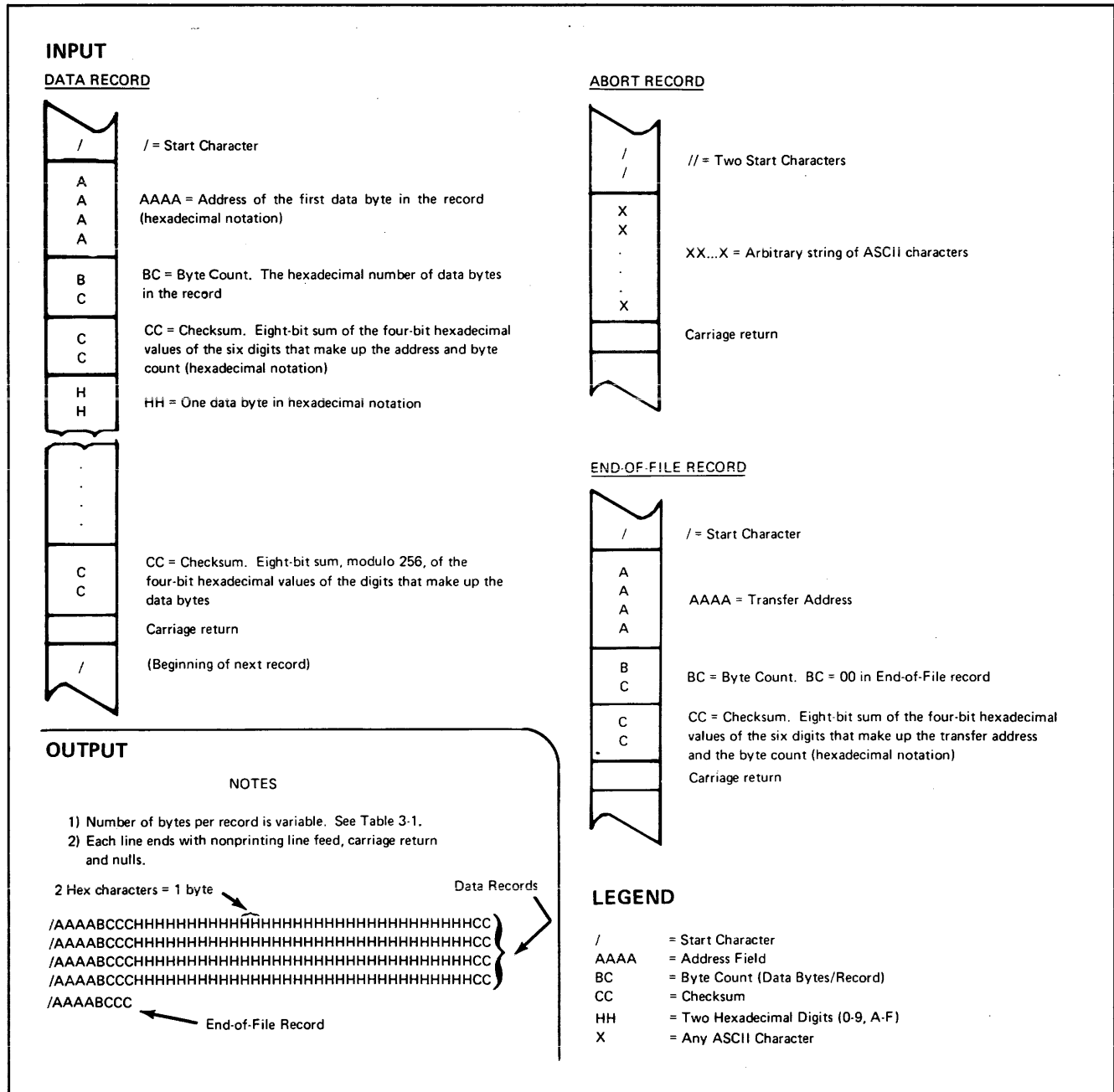


Figure A-14. Specifications for Tektronix Hexadecimal Data Files

• **MOTOROLA EXORMAX FORMAT, CODE 87**

Motorola data files may begin with a sign-on record, initiated by the code S0. Data records start with an 8- or 10-character prefix and end with a 2-character suffix. Figure A-15 demonstrates a series of Motorola Exormax data records.

Each data record begins with the start characters S1 or S2—S1 if the following address field has 4 characters, S2 if it has

6 characters. The third and fourth characters represent the byte count, which expresses the number of data, address and checksum bytes in the record. The address of the first data byte in the record is expressed by the last 4 characters of the prefix (6 characters for addresses above hex FFFF). Data bytes follow, each represented by two hexadecimal characters. The number of data bytes occurring must be 3 or 4 less than the byte count. The suffix is a 2-character checksum.

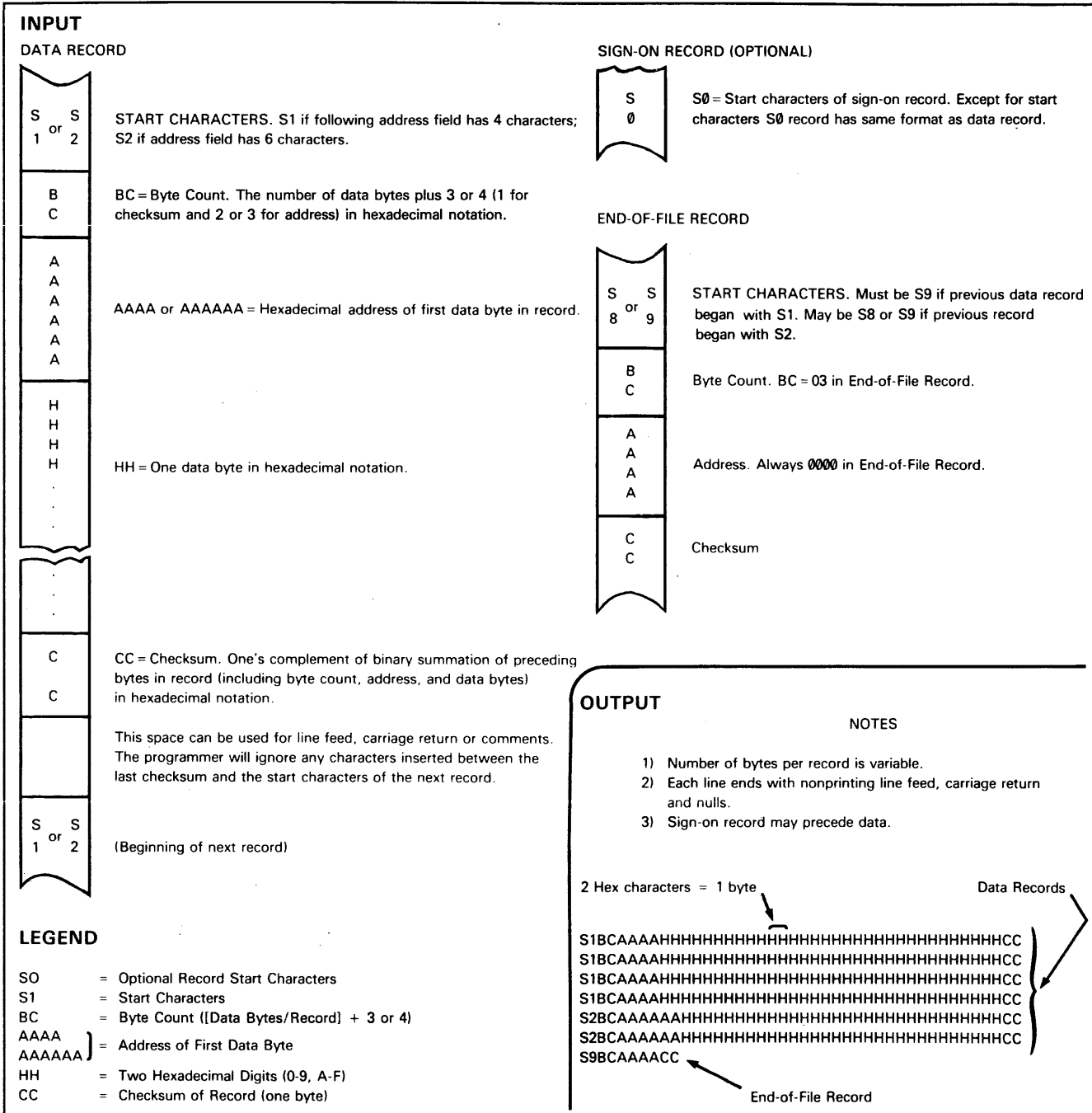


Figure A-15. Specifications for Motorola Exormax Data Files

• INTEL MCS-86 HEXADECIMAL OBJECT, CODE 88

The Intel 16-bit Hexadecimal Object file record format has a 9 character (4-field) prefix that defines the start of record, byte count, load address, and record type and a 2-character checksum suffix. Figure A-16 illustrates the four types of records used with this format.

The four record types are:

- 00 = data record
- 01 = end record (signals end of file)
- 02 = extended address record (added to the offset to determine the absolute destination address)
- 03 = start record (ignored during input and not sent during output by Data I/O translator firmware)

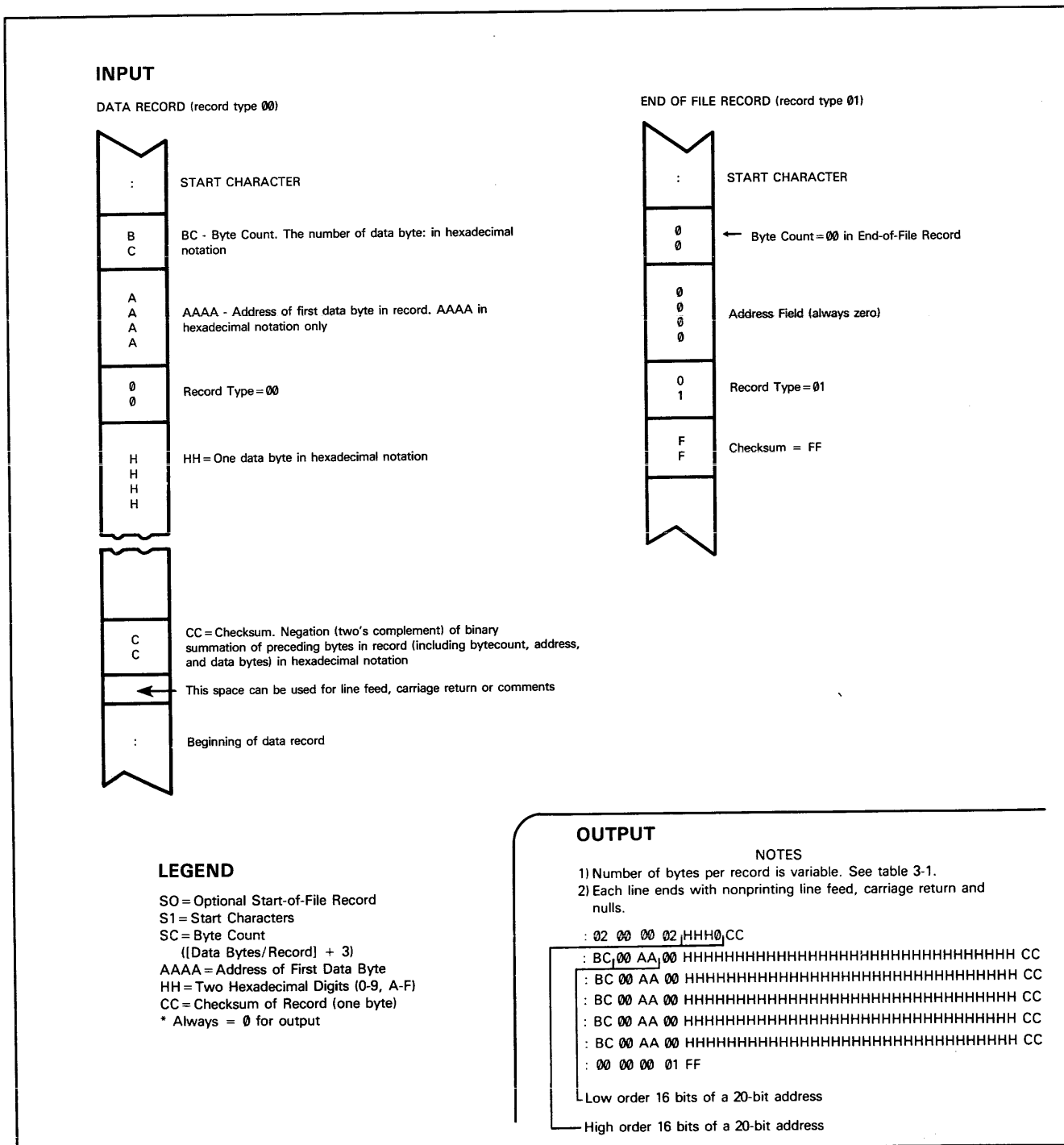
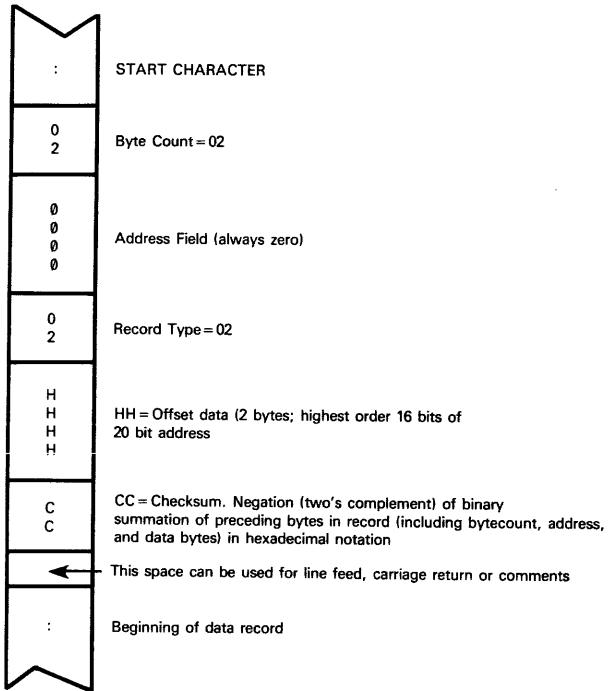


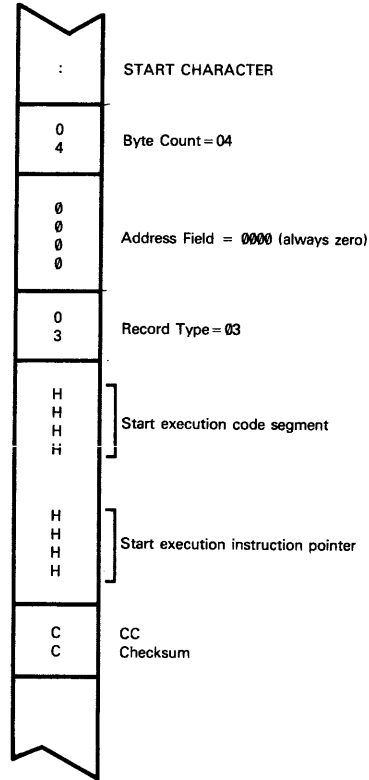
Figure A-16. Specifications for Intel MCS-86 Hexadecimal Object (Code 88) Data Files
-Continued on Next Page-

INPUT

EXTENDED ADDRESS RECORD (record type 02)



START ADDRESS RECORD (record type 03)
(not used by Data I/O firmware)



LEGEND

SO = Optional Start-of-File Record
 S1 = Start Characters
 SC = Byte Count
 ((Data Bytes/Record) + 3)
 AAAA = Address of First Data Byte
 HH = Two Hexadecimal Digits (0-9, A-F)
 CC = Checksum of Record (one byte)
 * Always = 0 for output

Figure A-16. (Continued)

Record type 02, the extended address record, defines bits 4 to 19 of the segment base address. It can appear randomly anywhere within the object file and in any order; i.e., it can be defined such that the data bytes at high addresses are sent before the bytes at lower addresses. Figure A-17 illustrates how the extended address is used to determine a byte address.

Example of How to Calculate Address

Problem: Find address for the first data byte for the following file.

```
:02 0000 02 1230 BA
:10 0045 00 55AA FF.....BC
```

Solution:

Step 1: Find the record address for the byte. The first data byte is 55. Its record address is 0045 from above.

Step 2: Find the base address (offset). The base address is 1230 from above.

Step 3: Shift the base address one place to the left and then add to the record address.

Base Address	1230	(upper 16 bits)
+ Record Address	0045	(lower 16 bits)
Address for first data byte =		12345 (20-bit address)

Figure A-17. Calculating an Address Using the Intel MCS-86 (Code 88) Extended Address Board

NOTE

Always specify the address offset when using this format, even when the offset is zero.

During output translation, the firmware will force the record size to 16 (decimal) if the record size is specified greater than 16. There is no such limitation for record sizes specified less than 16.

• HEWLETT-PACKARD 64000 ABSOLUTE FORMAT, CODE 89

Hewlett-Packard Absolute is a binary format with control and data-checking characters. See figure A-18.

Data files begin with a Start-of-File record including the data bus width, data width base, transfer address, and a checksum of the bytes in the record.

Data records follow the Start-of-File record. Each begins with 2 byte counts: the first expresses the number of 16-bit words in the record not including the checksum and itself; the second expresses the number of 8-bit data bytes in the record. Next comes a 32-bit address, which describes the storage location of the following data byte. Data bytes follow; after the last data byte comes a checksum of every byte in the record except the first byte.

The End-of-File record consists only of a byte count, which is always zero.

• TEXAS INSTRUMENTS SDSMAC FORMAT, CODE 90

Data files in the SDSMAC format consist of a Start-of-File record, data records, and an End-of-File record. See figure A-19.

Each record is composed of a series of small fields, each initiated by a tag character. The programmer recognizes and acknowledges the following tag characters:

- 0 - always followed by a file header.
- 7 - always followed by a checksum which the programmer acknowledges.
- 8 - always followed by a checksum which the programmer ignores.
- 9 - always followed by a load address.
- B - always followed by 4 data characters.
- F - denotes the end of a data record.

The Start-of-File record begins with a tag character and a 12-character file header. Next come interspersed address fields and data fields (each with tag characters). If any data fields appear before the first address field in the file, the first of those data fields is assigned to address 0000. Address fields may be expressed for any data byte, but none are required. The record ends with a checksum field initiated by the tag character 7 or 8, a 4-character checksum, and the tag character F.

Data records follow the same format as the Start-of-File record but do not contain a file header.

The End-of-File record consists of a colon (:) only. The output translator sends a control S after the colon.

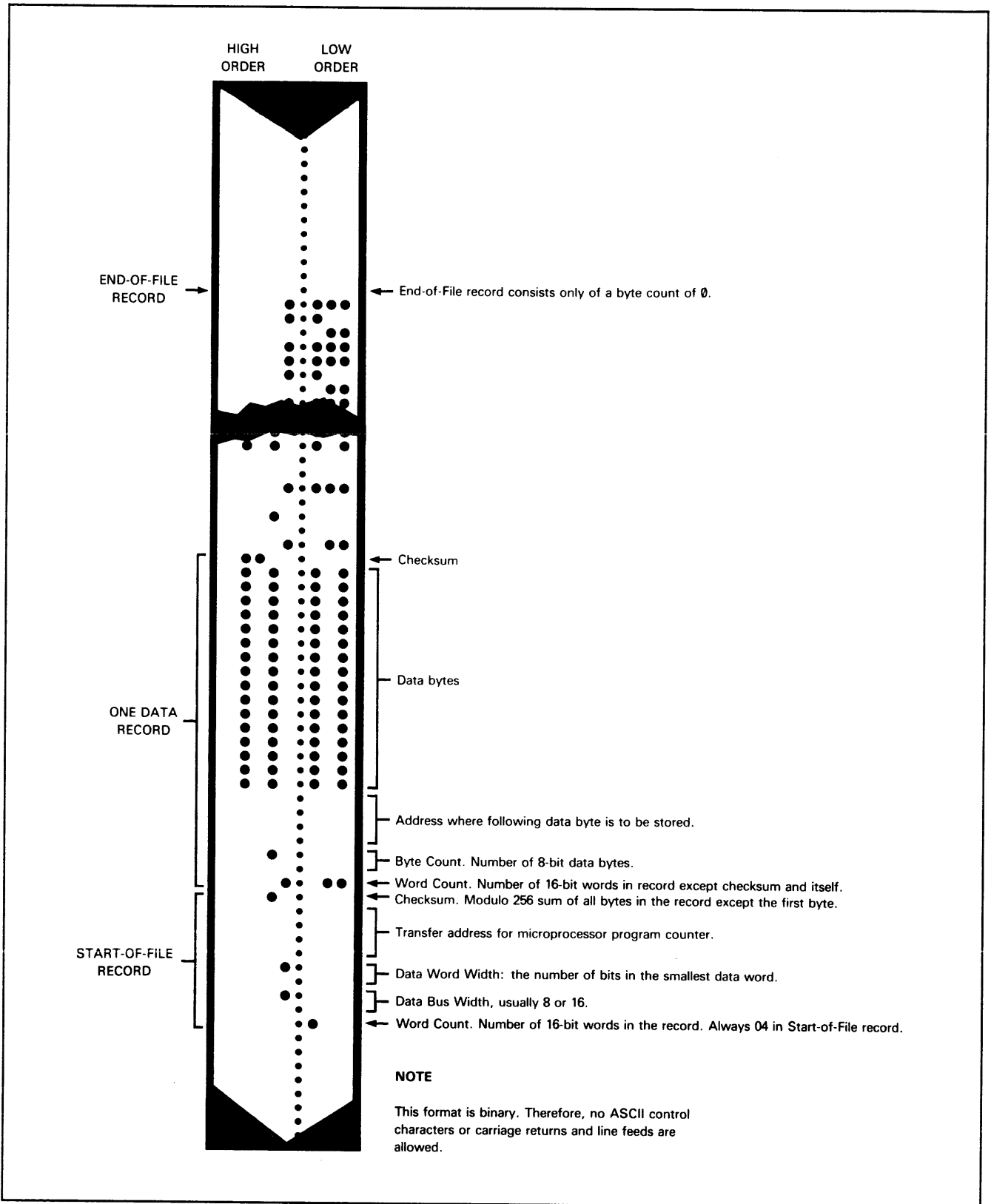


Figure A-18. Specifications for Hewlett Packard Absolute Format Data Files

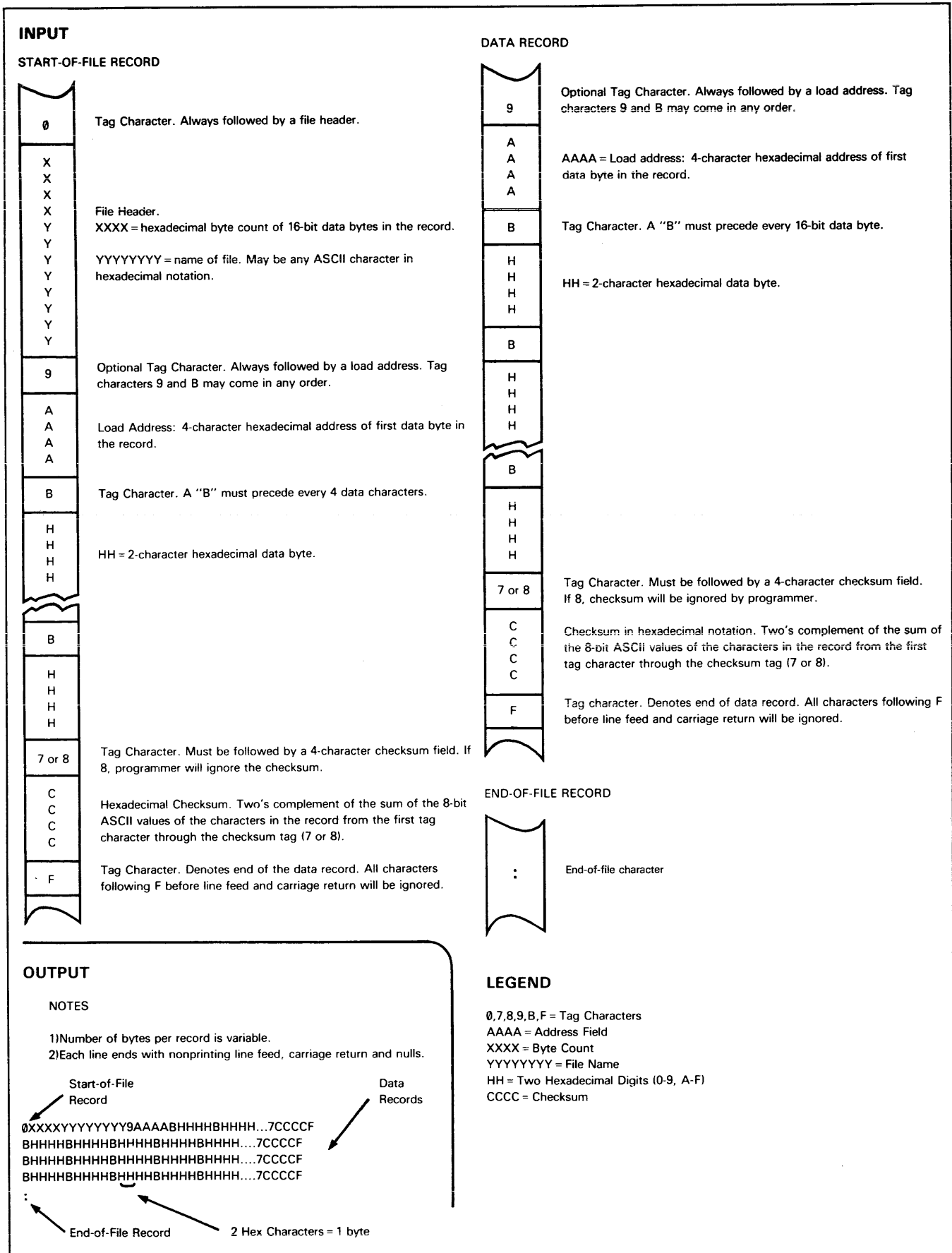


Figure A-19. Specifications for Texas Instruments SDSMAC Data Files

APPENDIX B

TERMINAL REMOTE CONTROL (TRC)

B.1 INTRODUCTION

This section summarizes the commands available for data transfer, editing, or review via Terminal Remote Control (TRC) for the 29B Programmer.

Figure B-1 shows the link between the 29B programmer and the remote system. When in Terminal Remote Control, all data manipulations are controlled remotely, either by a computer, manual data entry, or by a data transferring device, such as a tape reader. Existing RAM data is edited by any of the editing commands or by the various commands invoked while the programmer is in the Memory Mode. Edited RAM data can then be programmed into a device.

Data blocks transferred between the programmer and a device are in binary. Data blocks transferred between the remote system and the programmer are coded in a format specified by the operator (see the appendix A for description of the various data translation formats).

The programmer, under control of the terminal, executes the commands entered at the terminal and returns the prompt character (>) to the terminal upon successful execution of a command. When in the memory mode the current RAM address location and the data at that location are displayed.

B.2 SYSTEM VARIATIONS

The TRC system is not recommended for newly designed computer controlled programming systems; whereas, the CRC remote system is recommended (see subsection 3.9.1). However, since TRC has been widely used in the past with the System 19, it has been adapted to operate with the 29B. The few differences in operation between the System 19 and the Model 29B TRC are listed below; however, these should not require any significant changes in existing TRC programming system software and drivers.

1. All memory address related commands will take longer with 29B because it allows for up to 64k addressing which requires more software execution because of page addressing. These commands include all the major commands: "PG" (Program), "LD" (Load), "VF" (Verify), "DI" (Data Input), "DC" (Data Compare), and "DO" (Data Output) as well as less significant commands such as "SC" (Sumcheck), "CA" (Clear All RAM), "CM" (Complement Memory), "SN" (Swap Nibbles), and "SD" (Split RAM Data).
2. Memory mode commands which manipulate or search all 64k bytes of memory will require at least four times more time to perform their operation. These commands include: I (Insert Data), D (Delete Data), U (Fill Unprogrammed Data), S (Search Data), B (Substitute Data).

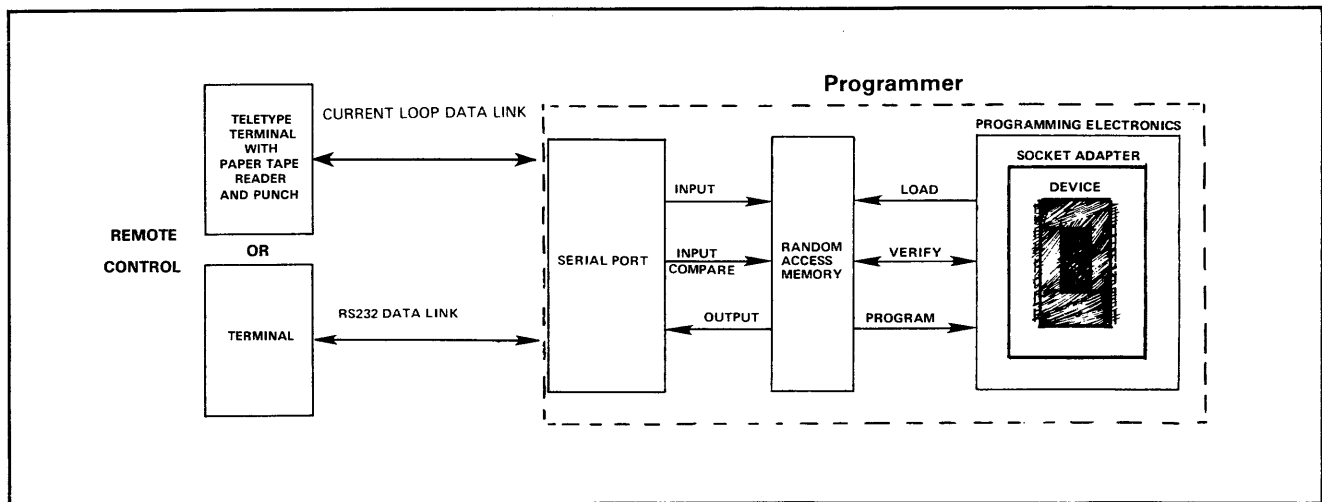


Figure B-1. System Configuration Between Programmer and Terminal Remote Control

3. The 29B message displayed on the terminal when first entering TRC has one extra line-feed preceding the text. Otherwise these messages are the same.
4. For the Model 29B after the end of an input or input compare data translation there must be a pause of incoming characters for 1/2 second before the prompt is returned. All incoming characters after the end of data translation and before the returning prompt are purged (thrown away).
5. The System 19 sends out an extra carriage return, linefeed, and null count after the STX (start character) for formats 01 through 09, 12, 13 whereas the model 29B does not.
6. The Model 29B sends out an extra carriage return, linefeed, and null count after the leader for formats 10 and 11 whereas the System 19 does not.
7. During input or input compare translation the System 19 beeps each time an error occurs whereas the Model 29B does not.
8. For the Model 29B a BREAK or ESCAPE key depressed during input or input compare translation will abort the operation just as with System 19, however, the Model 29B will not return a prompt until data stops coming in.
9. The "SM" (Memory Size) command with the Model 29B will show only the lower 16 bits of a 17-bit ending address if the "VM" (Virtual Memory Offset) command has been defined to be greater than zero. If the address offset is equal to zero (its default value) then there is no problem since the ending address can be shown with 16 bits.
10. Generally the quantity of errors given in response to "QB" (Input Compare Errors), "QC" (Buffer-Overflow Errors), and "QD" (Error Status) commands will be the same for both the System 19 and the Model 29B, however for some translators the quantity may be different.

B.3 COMMAND GROUPS

The commands available in remote control are grouped according to their functions. Tables B-1 through B-6 summarize the commands available in remote-control operation of the programmer, as well as programmer responses to these commands. Syntax for the commands in the summary tables is shown in section B.4.

Control-Key Commands

The Control-Key Commands summarized in table B-1 are used to execute or suspend a command, or to display the last command.

Table B-1. Control-Key Commands.

COMMAND	NAME	DESCRIPTION
ESC/BREAK		Abort a command. The BREAK key is used to abort a binary data transfer. Note: When using during the execution of the "DI" or "DC" command may not abort the command if a data stream continues to come into the programmer from the remote terminal or computer.
SPACE		Re-executes the D, I and C commands. Note: With the 'C' and 'I' command the data parameter must be entered each time.
LINE FEED		Displays the last command executed. Note: Does not function for memory mode commands.
RETURN [CR]		Executes a command or re-executes a previous command. The letters [CR] are used in this appendix to represent the return, or carriage return command. Note: Does not function for memory mode commands.

Utility Commands

The Utility Commands summarized in table B-2 set some general operating parameters required in many operations.

Device Commands

The Device Commands summarized in table B-3 execute the basic operations used in programming: Load, Program, Verify, etc.

Table B-2. Utility Commands

COMMAND	NAME	DESCRIPTION
KF [AD]	Keyboard Format	Sets number based (A/D) for address and data entry. Default Value - Hexadecimal.
CN	Configuration Number	Programmer outputs the software configuration identification number.
NM	Nibble Mode	Selects a 4-bit word size to override 8-bit programming modules.
BM	Byte Mode	Nullifies the Nibble Mode command.
SM	Size Memory	Terminal displays beginning and ending RAM addresses. The address-offset value is the beginning address (see "VM"). The ending RAM address is the beginning address plus the available RAM size.
SF [SC]	Select External Function	Allows entry of Select Codes (SC) carried in software on some programming modules.
SC	Sumcheck	Programmer sends sumcheck of all RAM data to terminal. Note: If data in RAM has been recently changed the operator should wait three seconds before issuing this command to get valid results.
CC	No Operation	
ZP	Exit Terminal Remote Control	Control returns to programmer keypad.

Table B-3 Device Commands

COMMAND	NAME	DESCRIPTION
CD	Clear Device Counter	Programmer clears the count of consecutive program operations.
SP	Device Size	Terminal displays number of words (bytes) and word size of device for which programmer is set up.
FP	Display Family Pinout	Multiple device type Paks only; e.g. the UniPak™. Terminal displays current device family and pinout codes.
FY [FF/PP]	Select Family Pinout	Multiple device type Paks only; e.g. the UniPak™. Selects device family and pinout codes.
PO [PP]	Select Pinout	Multiple device type Paks only; e.g. the UniPak™. Selects device pinout code.
LD [RAM BEG/BLK SIZ/DEV BEG]	Load	Device data is loaded into RAM.
PG [RAM BEG/BLK SIZ/DEV BEG].	Program	RAM data is programmed into device.
VF [RAM BEG/BLK SIZ/DEV BEG]	Verify	Device data is verified against RAM data.

I/O Commands

The I/O commands summarized in table B-4 set up the programmer to transmit or receive data through the serial

port. This includes setting a translation format, parity, address control, and other considerations fundamental to I/O data transfers.

Table B-4 I/O Commands

COMMAND	NAME	DESCRIPTION
OP	Odd Parity	Programmer sets odd parity for data transfer. Default Value - programmer's parity- switch setting.
EP	Even Parity	Programmer sets even parity for data transfer. Default Value - programmer's parity-switch setting.
NP	No Parity	Programmer sets no parity for data transfer. Default Value - programmer's parity-switch setting.
DP	Default Parity	Programmer returns to default parity value for data transfer, which is the programmer's parity-switch setting.
ND [NN]	Null Define	Programmer sets number of nulls (NN) to be output after a line of data. Default Value - one null.
RS [RR]	Record Size	Programmer sets number of bytes (RR) per output record in formats with variable record lengths. Default Value - 16 bytes (8 in Fairchild Fairbug).
OS	Select One Stop Bit	Programmer sets one stop bit for serial data transfers. Default Value - programmer's stop-bit switch.
TS	Select Stop Bits	Two Programmer sets two stop bits for serial data transfers. Default Value - programmer's stop-bit switch.
DS	Default Stop Bits	Programmer returns to default value for stop bits, which is the programmer's stop-bit switch.
FM [FC]	Translation Format	Programmer sets the format code (FC) for data transfer. Default Value - format 50, ASCII-Hex (Space).
VM [AAAA] For 16-bit Format I/O: VM [AAAA/AAAA]	Virtual Memory Offset	RAM beginning is redefined from 0 to value (AAAA). Note: During data input translation to the programmer input data is offset by (AAAA).
DT	Disable Timeout	Disables the 25-second I/O timeout. Restored only by power-on.
DI [RAM BEG/ BLK SIZE]	Data Input	Programmer accepts data input from terminal to RAM.
DO [RAM BEG/ BLK SIZ]	Data Output	Programmer sends data from RAM to terminal.
DC [RAM BEG/ BLK SIZ]	Data Compare	Programmer compares data from external source with RAM data.
QA	I/O Parameters	Terminal display 4 sets of characters: the address offset, the translation format, the number of nulls and the record size currently in effect.
QB	Input Compare	Programmer outputs the number of misverify errors that occurred in an Input operation.
QC	Buffer-Overflow	Programmer outputs the number of input-buffer-overflow errors.
QD	Error Status	Programmer outputs three figures: the number of missing or invalid characters, the number of sumcheck errors (or missing F characters in ASCII Binary formats) and the number of parity errors.

Page missing from original document

Page missing from original document

B.4 SYNTAX

Commands must be keyed in according to the rules of syntax listed below. The correct (or typical) syntax for each command is illustrated by means of the following symbols:

[] -	Symbol used to enclose a command input by the operator.
() -	Symbol used to enclose a response output by the 29B programmer.
RAM BEG -	RAM beginning address.
BLK SIZ -	Address block size, or number of bytes.
DEV BEG -	Device beginning address.
DATA -	Valid data nibble, byte.

NOTE

These symbols are indicators only. They are not entered by the operator as part of the commands or output by the programmer as part of the responses.

In a command, a parameter may be omitted if the default value is desired. Defaults for specific commands are listed in paragraphs B.8.2 through B.8.6.

For commands that require entering of address or data, this information must be entered using the numerical base in effect, either octal, binary, decimal, or hexadecimal. The base is defined by the KF command and the default base is hexadecimal. When data is being transferred via the serial I/O port, it must conform to translation code format defined by the FM command.

B.5 PROGRAMMER RESPONSES TO TERMINAL DATA ENTRY

The Model 29B Programmer responds to commands entered at the terminal in one of three ways:

1. If the command has been entered correctly, the programmer will execute the command, send a reply to the terminal and return the prompt character (>).
2. If the operator enters an incorrect Command Code, the programmer will return a question mark (?).
3. If the operator attempts an illegal operation, the programmer will either reject an entered command and emit a beep, or display "***ERROR" and the error number. Errors codes are listed in Appendix E of this manual.

B.6 SERIAL I/O INTERFACE

This information is provided in subsection 2.5 of this manual. However, it should be restated that Terminal Remote Control is full duplex. Thus, for a terminal or computer to be properly interfaced it must be set to full duplex.

B.7 ACCESS TO TERMINAL REMOTE CONTROL

B.7.1 ENTRY

To place the Model 29B Programmer in terminal remote control, enter Select Code F2 on the programmer's keyboard. The terminal will display the Data I/O title block:
Data I/O Corp. terminal remote control
copyright 1979 955-1903 Rev. F.

Use the following procedure to enter terminal remote control:

ENTER TERMINAL REMOTE CONTROL Command

Operation: ENTER TERMINAL REMOTE CONTROL (from front panel keyboard)
Code: F2
Syntax: [SELECT] [F] [2] [START] [START]
Explanation: Entered by the operator on the 29B front panel keyboard to place the programmer under terminal remote control.

B.7.2 EXIT

The operator may use either the terminal or the 29B keyboard to exit from terminal remote control. Use the applicable of the two following procedures to exit terminal remote control:

EXIT REMOTE CONTROL (from terminal) Command

Operation: EXIT REMOTE CONTROL (from terminal)
Code: ZP
Syntax: (>) [Z] [P] [CR]
Explanation: This command causes the terminal to surrender control to the programmer. When exiting remote control via the ZP command, all operating parameters except the virtual memory offset will be retained. Upon re-entering terminal remote control, the terminal will display a prompt character ">" instead of the Data I/O title block. The title block will display if the programmer has subsequently been turned off.

EXIT REMOTE CONTROL (from keyboard) Command

Operation: EXIT REMOTE CONTROL (from front panel keyboard)
Code: Press any mode key (COPY, VERIFY, SELECT, or EDIT)
Syntax: (>) [any mode (blue) key]
Explanation: Pressing a mode key on the programmer (any blue key) returns control to the front panel. All parameters will be retained, and data displayed at the terminal before depressing a mode key will remain unchanged upon re-entering remote control.

B.8 COMMAND SUMMARY

The entire Command summary, with commands detailed individually, is described in paragraphs B.8.1 - B.8.6. Tables B-1 through B-6 summarize the commands.

NOTE

The programmer must send the prompt character (>) to the terminal before any command can be executed. The prompt character is sent to the terminal:

1. on entering remote control
2. after a previous command has been successfully executed
3. after either BREAK or ESCAPE has halted a command
4. after an error message has been printed at the terminal.

B.8.1 CONTROL-KEY COMMANDS

The following are Control-Key Commands:

ESCAPE (or BREAK) Command

Operation: ESCAPE and BREAK
Code: ESC or BREAK
Syntax: [ESC] or [BREAK]
Explanation: ESC or BREAK suspend the present command at any stage of execution. ESCAPE may be used at any time except when inputting binary data. Only the BREAK key is used to suspend a binary-input operation. Note: During input data translation an abort will not result in an immediate prompt for a new command until the input data stops coming in for a period of 1/2 second.

SPACEBAR Command

Operation: SPACEBAR
Code: SPACE
Syntax: [SPACEBAR]
Explanation: In the Memory Mode, SPACE is used to partially re-execute the following commands: Delete (D), Insert (I) and Change (C). Note: A carriage return and any data parameters required by these command must still be entered before the command is executed.

LINE FEED Command

Operation: LINE FEED
Code: LINE FEED
Explanation: LINE FEED causes the terminal to display the last-executed command (but not to execute it).

NOTE

The following commands are not re-executed by a return or redisplayed by a line-feed: DC, DI, FM, FY, KF, MM, MO, ND, PO, RS, SF, ZP, and all the memory mode commands.

RETURN Command

Operation: RETURN
Code: RETURN
Syntax: [CR]
Explanation: RETURN executes commands that are not executed immediately on entry of the command. For commands that allow re-execution, RETURN is used to re-execute the previous command. This eliminates the need to retype lengthy commands.

NOTE

The following commands are not reexecuted by a return or redisplayed by a line feed: DC, DI, FM, FY, KF, MM, MO, ND, PO, RS, SF, ZP, and all the memory mode commands.

B.8.2 UTILITY COMMANDS

The following are Utility Commands:

KEYBOARD FORMAT Command

Operation: KEYBOARD FORMAT
Code: KF
Syntax: (>) [K] [F] [Address Base] [/] [Data Base]
[CR]

Explanation: The KF command sets the address and data bases the operator will use in communicating addresses and data. These number bases may differ; for example, hexadecimal addresses and binary data may be displayed together. Default Value - hexadecimal.

Bases: 0 = hexadecimal
1 = decimal
2 = octal
3 = binary

CONFIGURATION NUMBER Command

Operation: CONFIGURATION NUMBER
Code: CN
Syntax: (>) [C] [N] [CR]

Explanation: This command displays the configuration number which identifies the software resident in the 29B.

NIBBLE MODE Command

Operation: NIBBLE MODE
Code: NM
Syntax: (>) [N] [M] [CR]

Explanation: This command sets a 4-bit word size to override 8-bit programming Paks. Default Value - programming electronics word size.

NOTE

The word size in effect does not change upon entering or exiting remote control.

BYTE MODE Command

Operation: BYTE MODE
Code: BM
Syntax: (>) [B] [M] [CR]

Explanation: This command nullifies the Nibble Mode command, allowing the programming Pak word size to take effect.

SIZE MEMORY Command

Operation: SIZE MEMORY
Code: SM
Syntax: (>) [S] [M] [CR]

Explanation: This command causes the terminal to display the beginning and ending addresses in RAM, added to the address offset in effect. The beginning address value is the address-offset. The ending RAM address is the beginning address plus the available RAM size.

NOTE

For the 29B version with 64k bytes RAM: If the virtual address offset (see the "VM" command) is greater than zero the ending address displayed will not show the most significant bit(s). This is because only the 16 lowest order binary bits of the address are shown on the display. The operator must add the most significant bits which will be equal to "1" binary to derive the ending address.

SELECT EXTERNAL FUNCTION Command

Operation: SELECT EXTERNAL FUNCTION
Code: SF
Syntax: (>) [S] [F] [Select Code] [CR]

Explanation: Some Select Codes are associated with the programming Pak, and can only be accessed when that Pak is installed in the programmer. To access those Select Code functions in terminal remote control, the Select Code is entered after SF.

SUMCHECK Command

Operation: SUMCHECK
Code: SC
Syntax: (>) [S] [C] [CR]

Explanation: This command instructs the programmer to compute the sumcheck of all RAM data and to display it at the terminal. For example, the terminal will display "SUM=06FE."

NOTE

If data in the RAM has been recently changed the operator should wait at least three seconds before issuing this command to insure valid results.

NO OPERATION Command

Operation: NO OPERATION
Code: CC
Syntax: (>) [C] [C] [CR]

Explanation: No operation is performed by the 29B. The command may be used in the operating program when no action is required. A prompt character ">" is displayed.

B.8.3 DEVICE COMMANDS

The following are Device Commands:

CLEAR DEVICE COUNTER Command

Operation: CLEAR DEVICE COUNTER

Code: CD

Syntax: (>) [C] [D] [CR]

Explanation: The CD command clears the count of the number of program operations accumulated since power-on, since the last F0 command was entered at the programmer keyboard, or since the last CD command.

DEVICE SIZE Command

Operation: DEVICE SIZE

Code: SP

Syntax: (>) [C] [D] [CR]

Explanation: This command causes the terminal to print the number of words and number of bits per word of the device which the programmer is configured to program. Addresses are in the specified keyboard format. In hex format, for example, a 1024 x 8 matrix would be displayed as 3FF/8.

DISPLAY FAMILY & PINOUT Command

Operation: DISPLAY FAMILY & PINOUT

Code: FP

Syntax: (>) [F] [P] [CR]

Explanation: With a Data I/O multi-device pak installed in the programmer this command is used to display the device Family and Pinout Codes. These codes are found in the Pak's manual.

SELECT FAMILY & PINOUT Command

Operation: SELECT FAMILY & PINOUT

Code: FY

Syntax: [F] [Y] [Family Code] [/] [Pinout Code] [CR]

Explanation: With a Data I/O multi-device pak installed in the programmer this command is used to select the device Family and Pinout Codes. These codes are found in the Pak's manual.

SELECT PINOUT Command

Operation: SELECT PINOUT

Code: PO

Syntax: (>) [P] [O] [Pinout Code] [CR]

Explanation: With a Data I/O multi-device pak installed in the programmer, this command can be used to change the device Pinout Code when the correct Family Code is already selected. These codes are found in the Pak's manual.

LOAD Command

Operation: LOAD (Device to RAM)

Code: LD

Syntax: (>) [C] [D] [Begin RAM Address] [/] [Number of Bytes] [/] [Begin Device Address] [CR]

Explanation: This command causes device data to be loaded into programmer RAM. If address limits are specified, device data starting at the specified device address is loaded into RAM, starting at the Begin RAM address and continuing until the specified number of bytes has been sent. If address limits are not specified the entire device data will be loaded into RAM. For 4-bit devices, device data will be loaded to the low-order nibble in RAM. Upon successful loading of data from a part the sumcheck of the data loaded is displayed.

PROGRAM Command

Operation: PROGRAM (RAM to Device)

Code: PG

Syntax: (>) [P] [G] [Begin RAM Address] [/] [Number of Bytes] [/] [Begin Device Address] [CR]

Explanation: This command causes an illegal-bit check and a blank check of the device before RAM data is programmed into a device. If address limits are specified, RAM data between those addresses is programmed, starting at the Begin Device Address. If address limits are not specified, the entire device will be programmed. For 4-bit devices, the low-order nibble in RAM will be programmed into a device. The Swap Nibbles command may be used previous to the "PG" command to move data from the high-order nibble of RAM to the low-order nibble of RAM, so that it can be subsequently programmed by the "PG" command.

If the device to be programmed is non-blank the 29B will display on the CRT terminal "DEVICE NOT BLANK. S=SKIP." If the operator still wishes to program the part he responds by typing "S". If not, he will abort the command by typing "ESC" key on keyboard.

If the device cannot be programmed because a bit or more is already programmed where it should not have been, then an illegal-bit error exists. The 29B will display on the CRT terminal "ILLEGAL BIT" and a prompt ">" to signify the command was aborted and that the 29B is ready for another command.

Upon successful programming of a part the programmer outputs a count of the number of programming operations completed. To reset this count see the "CD" command.

Pressing the RETURN key repeats the Program operation.

VERIFY Command

Operation: VERIFY (Device vs. RAM)
Code: VF
Syntax: (>) [V] [F] [Begin RAM Address] [/]
[Number of Bytes] [/] [Begin Device Address]
[CR]

Explanation: This command compares data in a device with data in RAM. If address limits are specified, verification occurs only within the specified number of bytes, starting at the specified RAM and device addresses. If address limits are not specified the entire device data is verified against the RAM data.

For 4-bit devices, device data will be verified against the low-order nibble of RAM. After a successful verification, the programmer returns a sumcheck for the data verified and displays "VERIFY OK" and a prompt character.

If the device data fails to verify against the RAM data the following message is displayed on the terminal CRT screen: "PASS 1 VERIFY ERRORS = NN," where NN is a number denoting the quantity of errors. There could also be a "PASS 2 VERIFY ERRORS = NN" if errors were only detected on the second of the two verify passes.

Also the address, RAM data and device data for each mis-verify location is displayed on the terminal's CRT screen. At the last mis-verify location, the sumcheck for the RAM data is displayed along with the message "VERIFY ERROR."

B.8.4 I/O COMMANDS

The following are I/O Commands:

ODD PARITY Command

Operation: ODD PARITY
Code: OP
Syntax: (>) [O] [P] [CR]
Explanation: The programmer sets odd parity for output data and inspects incoming data for odd parity. Default Value—programmer's parity-switch setting.

EVEN PARITY Command

Operation: EVEN PARITY
Code: EP
Syntax: (>) [E] [P] [CR]
Explanation: The programmer sets even parity for output data and inspects incoming data for even parity. Default Value—programmer's parity-switch setting.

NO PARITY Command

Operation: NO PARITY
Code: NP
Syntax: (>) [N] [P] [CR]
Explanation: This command instructs the programmer not to check input data for parity, and not to set parity for output data. Default Value—programmer's parity-switch setting.

DEFAULT PARITY Command

Operation: DEFAULT PARITY
Code: DP
Syntax: (>) [D] [P] [CR]
Explanation: This command returns the programmer to the default parity value, which is the programmer's parity-switch setting.

NULL DEFINE Command

Operation: NULL DEFINE
Code: ND
Syntax: (>) [N] [D] [Decimal Number of Nulls] [CR]
Explanation: This command allows choice of sufficient nulls for the terminal to recover at the end of a line of output data. Any number from 0 up to 254 may be entered. Default Value—1 null. A carriage return and a line feed always precedes the nulls for null counts values from 0 to 254. However, a null count of 255 (FF hexadecimal) is used to specify a single carriage return without a linefeed and without any nulls.

NOTE

The number of output nulls in effect does not change upon entering or exiting remote control.

RECORD SIZE Command

Operation: RECORD SIZE
Code: RS
Syntax: (>) [R] [S] [Decimal Number of Bytes] [CR]
Explanation: This command sets the number of bytes per output record for those formats which allow variable record lengths. Any number up to 255 may be entered. Default Value—16 bytes per record (8 bytes per record in Fairchild Fairbug).

NOTES

1. *The record size in effect does not change upon entering or exiting remote control.*
2. *When you change from one translation format to another, the record size in effect remains the same, unless you change to Fairchild Fairbug, which has mandatory 8-byte records.*

SELECT ONE STOP BIT Command

Operation: SELECT ONE STOP BIT
Code: OS
Syntax: (>) [0] [S] [CR]
Explanation: This command sets one stop bit for serial data transfers. Default Value - programmer's stop-bit-switch setting.

SELECT TWO STOP BITS Command

Operation: SELECT TWO STOP BITS
Code: TS
Syntax: (>) [T] [S] [CR]
Explanation: This command sets two stop bits for serial data transfers. Default Value - programmer's stop-bit-switch setting.

DEFAULT STOP BITS Command

Operation: DEFAULT STOP BITS
Code: DS
Syntax: (>) [D] [S] [CR]
Explanation: This command returns the programmer to the stop-bit default value, which is the programmer's stop-bit-switch setting.

TRANSLATION FORMAT Command

Operation: TRANSLATION FORMAT
Code: FM
Syntax: (>) [F] [M] [Control-Format Code] [CR]
Explanation: This command selects the translation format for serial I/O data transfer. The one digit Instrument Control code and the two-digit translation-format codes are listed in Appendix A. Default Value - ASCII-HEX (Space), format code 50.

NOTE

The translation format in effect does not change upon entering or exiting remote control.

VIRTUAL MEMORY OFFSET Command

Operation: VIRTUAL MEMORY OFFSET
Code: VM
Syntax: (>) [V] [M] [Offset] [CR]
Explanation: The offset value redefines all programmer RAM addresses. Thus, RAM Address 0 becomes equal to the offset, address 1 becomes the offset plus 1, address 2 becomes the offset plus 2, etc. These new addresses are used in the Memory Mode, and are also used in place of an address offset in I/O data transfers.

Thus for data translation using terminal remote control VM has two meanings. First it defines the address value of the lowest RAM location or the RAM offset. Second it defines (and is used in place of) the I/O address offset (equivalent to the port address for keyboard operations). However, the I/O address offset is undefined and assumes a value equal to zero for those formats which don't send address information as part of data translation. Therefore, there is effectively no I/O address offset regardless of the value

defined by "VM" for formats such as format 10 and 11 which do not send address information (see Appendix A). However, even for data translation using these formats the "VM" command will continue to define the address value of the RAM locations.

The VM can be set greater than 64K for I/O translations of 16-bit formats with 5, 6, and 8 character address fields by entering a slash between the high-order and four low-order characters of the offset value.

NOTE

The 29B version with 64K of RAM is not designed to manipulate any virtual address defined to be greater than 64K (FFFF). Thus if the lowest address location is defined by the "VM" command to be greater than zero then its highest virtual address should logically be greater than 64K. However, the 29B version with 64K of RAM can not display an address field greater than 64K, will instead actually display only the 16 least significant bits of the address field. Moreover an operation which would require data to be addressed which is greater than 64K will result in an error #56 (RAM range error).

DISABLE TIMEOUT Command

Operation: DISABLE TIMEOUT
Code: DT
Syntax: (>) [D] [T] [CR]

Explanation: This command disables I/O timeout. The I/O timeout is a time interval (25 seconds) used during data translation (either DI, DC, or DO commands) during which the 29B will wait for data coming in (for DI and DC commands) or for the output device not to be busy (for the DO command). The timeout will be restored only when the programmer is switched ON again.

DATA INPUT Command

Operation: DATA INPUT
Code: DI
Syntax: (>) [D] [I] [Begin RAM Address] [/] [Number of Bytes] [CR]

Explanation: Data originating at a terminal or computer is translated (using the selected translation format, see "FM" command) into binary data which is then stored in RAM at the specified address limits. After the input operation is complete, the terminal displays the sumcheck of all data translated.

NOTE

If Binary (format 10) without an arrow-shaped leader, DEC Binary (format 11), or ASCII Binary data (formats 01 through 09, 12, 13) with no end code is input from a paper tape reader, the "Number of Bytes," must be specified, corresponding to the length of the tape starting at the first data character.

NOTE

Using a "BREAK" or "ESC" key during execution of the above command may not abort the command if a stream of data continues to be output to the programmer.

DATA OUTPUT Command

Operation: DATA OUTPUT
Code: DO
Syntax: (>) [D] [O] [Beginning RAM Address] [/] [Number of Bytes] [CR]

Explanation: This command causes the programmer to output RAM data between the specified addresses to the terminal or computer in the data format previously specified in the FM command. With a 4-bit device selected, RAM data from the low-order nibble will be output. If output is through a tape punch, the tape will contain a leader of fifty null characters, data in the specified format, then fifty more null characters. There is one exception: If the null count (defined by the "ND" command) is set to a value of FF (hexadecimal) then no nulls are sent at the beginning or the ending of data translation. After the output operation is complete, the terminal displays the sumcheck of all data transferred.

NOTE

The BREAK or ESCAPE key can be used to abort a binary data output operation.

DATA COMPARE Command

Operation: DATA COMPARE
 Code: DC
 Syntax: (>) [D] [C] [Begin RAM Address] [/]
 [Number of Bytes] [CR]

Explanation: This command is the same as Data Input (DI) except that remote data is not transferred - it is compared with RAM data. The programmer sends to the terminal the sumcheck of all data compared. Any failure causes an error message to be sent to the terminal. The QB command can be used to determine the number of non-compare errors.

NOTE

If Binary without an arrow-shaped leader, DEC Binary or ASCII Binary data with no end code is input from a paper tape reader, the "Number of Bytes," must be specified corresponding to the length of the tape starting at the first data character.

NOTE

Using a "BREAK" or "ESC" key during execution of the above command may not abort the command if a stream of data continues to be output to the programmer.

I/O PARAMETERS Command

Operation: I/O PARAMETERS
 Code: QA
 Syntax: (>) [Q] [A] [CR]
 Explanation: On this command, the programmer outputs 4 sets of characters: the address offset, the translation format, the number of nulls and the output-record size currently in effect. The display is as follows:

AAAA FF NNN RRR

Where the above symbols are defined as follows:

AAAA or AAAAAA represents the address offset in effect, represented by 4-6 digits depending on the keyboard format in effect.

FF represents the translation format in effect.

NNN represents the decimal number of output nulls in effect.

RRR represents the decimal number of bytes per output recorded in effect.

INPUT COMPARE ERRORS Command

Operation: INPUT COMPARE ERRORS
 Code: QB
 Syntax: (>) [Q] [B] [CR]
 Explanation: On this command, the programmer outputs the number of misverify errors that occurred during an Input ("DI" or "DC") operation.

BUFFER-OVERFLOW ERRORS Command

Operation: BUFFER-OVERFLOW ERRORS
 Code: QC
 Syntax: (>) [Q] [C] [CR]
 Explanation: On this command, the programmer outputs the number of characters received during an input "DI" or "DC" operation after the programmer instructed the terminal to stop sending data.

I/O ERROR STATUS Command

Operation: I/O ERROR STATUS
 Code: QD
 Syntax: (>) [Q] [D] [CR]
 Explanation: On this command, the programmer outputs three sets of characters which give the total number of each of these three types of errors that occurred during an input ("DI" or "DC") operation. The display is as follows:

XX YY ZZ

Where the above symbols are defined as follows:

- XX represents the decimal number of either:
1. non-valid characters (Formats 01-03, 5-9, 12-13).
 2. missing data characters (Formats 30-37 and 50-58).
 3. non-hex characters (Formats 80-88, 90).

YY represents the decimal number of sumcheck errors.

ZZ represents the decimal number of parity errors.

B.8.5 EDITING COMMANDS

The following are Editing Commands:

COMPLEMENT MEMORY Command

Operation: COMPLEMENT MEMORY
Code: CM
Syntax: (>) [C] [M] [Beginning Address] [/] [Number of Bytes] [CR]
Explanation: This command causes the programmer to complement RAM contents at the specified addresses; that is, all one bits are turned into zeros, and all zero bits are turned into ones. If no addresses are specified, all RAM data is complemented.

CLEAR ALL RAM Command

Operation: CLEAR ALL RAM
Code: CA
Syntax: (>) [C] [A] [CR]
Explanation: This command clears all RAM memory to zeros, whether the programmer is set for 4-bit or 8-bit operation.

SWAP NIBBLES Command

Operation: SWAP NIBBLES
Code: SN
Syntax: (>) [S] [N] [CR]
Explanation: This command exchanges high-order and low-order halves of every word in RAM.

SPLIT RAM DATA Command

Operation: SPLIT RAM DATA
Code: SD
Syntax: (>) [S] [D] [Center Point] [CR]
Explanation: For 16-bit microprocessor data; complement of Shuffle Ram Data (below). After you serially input a block of 16-bit data to RAM (each sequential pair of 8-bit bytes representing a 16-bit word), the SD command "splits" the 16-bit block into two adjacent 8-bit blocks, separated by the specified center point. The split stores the even-numbered 8-bit bytes of each byte pair in sequence from address 0 to the center point; odd-numbered bytes are stored in sequence at addresses beginning at the center point. The reorganized data occupies the same original block in RAM.

You can then program each block of data into an 8-bit device, and the two devices can be addressed in parallel (while in use) to deliver 16-bit words to the processor.

Typically, the center point will equal the number of words in the 8-bit device to be programmed. In any event, it must meet two requirements:

1. It must be a power of 2.
2. It must be less than or equal to half the size of the resident RAM.

Center-point default value—RAM midpoint.

SHUFFLE RAM DATA Command

Operation: SHUFFLE RAM DATA
Code: SH
Syntax: (>) [S] [H] [Center Point] [CR]
Explanation: For 16-bit microprocessor data. Complement of Split RAM Data, this command merges into one 16-bit block the two adjacent 8-bit blocks of data which met at the specified center point address. Two 8-bit devices are first loaded next to each other in RAM, beginning at address 0, to create the two blocks, which are then merged for serial transfer. The center-point must be a power of 2 between 0 and RAM midpoint. Center-point default value—RAM midpoint.

RAM-RAM BLOCK MOVE Command

Operation: RAM-RAM BLOCK MOVE
Code: BL
Syntax: (>) [B] [C] [RAM Source Address] [/] [Number of Bytes] [/] [RAM Destination Address] [CR]
Explanation: This command instructs the programmer to move the specified number of bytes beginning at the RAM Source Address to a new RAM location beginning at the RAM Destination Address.

ENTER MEMORY MODE (at address) Command

Operation: ENTER MEMORY MODE (At Specified Address)

Code: MM

Syntax: (>) [M] [M] [Valid Address] [CR]

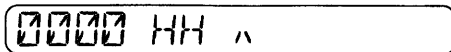
Explanation: The terminal enters the Memory Mode at the specified "valid address." A valid address is defined as an address within the word limits of virtual memory expressed in the number base previously selected with the KF (Keyboard Format) command. If an address is not given prior to RETURN, the starting address of virtual memory is assumed default value equals 0. The carriage return causes a display of the address and the data at that address. Also the cursor stops just past the data field awaiting further memory mode commands. See the following example:

EXAMPLE: Entering memory mode using the default value for virtual memory of 0.

keystrokes:



display:



Where 0000 is the address

HH is the hexadecimal value of the data

— is the cursor used as a prompt for a Memory Mode Command (see section B.8.6).

ENTER MEMORY MODE (at last address) Command

Operation: ENTER MEMORY MODE (At Last-Displayed Address)

Code: MO

Syntax: (>) [M] [O] [CR]

Explanation: The terminal enters the Memory Mode at the last-displayed address. Otherwise, this command is the same as the "MM" command.

B.8.6 MEMORY MODE COMMANDS

In the Memory Mode, RAM data is displayed and manipulated. The printer or CRT displays two fields - address and data - and the print head or CRT cursor indexes to the right of the data field awaiting Memory Mode commands. The addresses displayed are virtual addresses, in other words, the beginning of RAM is defined by whatever value was issued by the "VM" command. Addresses below this value are undefined.

To enter the memory mode use either the "MM" or "MO" command (see subsection B.8.5). To exit the memory mode use the memory mode "E" command.

The following are memory mode commands:

DISPLAY NEXT ADDRESS Command

Operation: DISPLAY NEXT ADDRESS

Code: N

Syntax: (Address) (Data) [N]

Explanation: Each N keystroke displays the next higher address and the data at that address.

DISPLAY LAST ADDRESS Command

Operation: DISPLAY LAST ADDRESS

Code: L

Syntax: (Address) (Data) [L]

Explanation: Each L keystroke displays the next lower address and the data at that address.

JUMP ADDRESS Command

Operation: JUMP ADDRESS

Code: J

Syntax: (Address) (Data) [J] [Valid Address] [CR]

Explanation: This command allows the operator to jump from the currently displayed address to the specified valid address. The default address is the Virtual Memory Offset.

CHANGE DATA Command

Operation: CHANGE DATA
 Code: C
 Syntax: (Address) (Data) [C] [Valid Data] [CR] or [SPACE]

Explanation: This command changes the displayed data to the specified valid data. The RETURN key causes the command to execute once. To execute the change command more than once use the SPACE BAR instead of the RETURN key. This is the same as pressing RETURN and "C" again; thus, valid data must be entered each time.

EXAMPLE: Change DD to EE at address 004.

BEFORE			AFTER	
Ad- dress	Data	Key- strokes	Address	Data
000	AA		000	AA
001	BB		001	BB
002	CC		002	CC
003	DD		003	DD
004	DD	C EE [CR]	004	EE

INSERT DATA Command

Operation: INSERT DATA
 Code: I
 Syntax: (Address) (Data) [I] [Valid Data] [CR] or [SPACE]

Explanation: This command places valid data into the displayed address and moves the old data and all subsequent data to the next higher address. The RETURN key causes the command to execute once. To execute an Insert command more than once use the SPACE BAR instead of the RETURN key. This is the same as hitting RETURN and "I" again. Thus, valid data must be entered each time.

EXAMPLE: Insert EE into address 002.

BEFORE			AFTER	
Ad- dress	Data	Key- strokes	Address	Data
000	AA		000	AA
001	BB		001	BB
002	CC	I EE CR	002	EE
003	DD		003	CC
			004	DD

DELETE DATA Command

Operation: DELETE DATA
 Code: D
 Syntax: (Address) (Data) [D] [CR] or [SPACE]

Explanation: This command causes data at the displayed address to be removed from memory, and all following data is moved down one location (toward lower addresses). The last RAM memory location is loaded with zero. The RETURN key causes the command to execute once. To execute more than once use the SPACE BAR instead of the RETURN key. This is the same as entering RETURN and "D" again.

EXAMPLE: Delete EE into address 002.

BEFORE			AFTER	
Ad- dress	Data	Key- strokes	Address	Data
000	AA		000	AA
001	BB		001	BB
002	EE	D CR	002	CC
003	CC		003	DD
			004	DD

REPEAT DATA Command

Operation: REPEAT DATA
 Code: R
 Syntax: (Address) (Data) [R] [Valid Data] [Number of Locations] [CR]

Explanation: This command places valid data into the specified number of memory locations beginning at the displayed address.

EXAMPLE: Repeat EE three times starting at address 002.

BEFORE			AFTER	
Ad- dress	Data	Key- strokes	Address	Data
000	11		000	11
001	22		001	22
002	33	R EE / 3 CR	002	EE
003	44		003	EE
004	55		004	EE
005	66		005	66
006	77		006	77

FILL DATA Command

Operation: FILL DATA
Code: F
Syntax: (Address) (Data) [F] [Valid Data] [CR]
Explanation: This command causes valid data to be placed in each RAM address from the displayed address to the end of RAM. The terminal then exits the memory mode and displays the prompt character (>).

FILL UNPROGRAMMED DATA Command

Operation: FILL UNPROGRAMMED DATA
Code: U
Syntax: (Address) (Data) [U] [CR]
Explanation: This command fills memory from the displayed address to the end of RAM with no-program information. That is, if the programmer is configured to program VOH devices, remaining memory is filled with logical "0" (low state). Likewise, if the programmer is set to program VOL devices, remaining memory is filled with logical "1" (high state). The terminal then exits the Memory Mode and displays the prompt character (>).

SEARCH DATA Command

Operation: SEARCH DATA
Code: S
Syntax: (Address) (Data) [S] [Valid Data] [/] [Beginning Address] [/] [Number of Bytes] [CR]
Explanation: This command displays all locations containing the specified valid data. Entry of the beginning address and number of bytes is optional. If addresses are entered, the display is bounded by those address limits. If no addresses are specified, all RAM locations containing the entered valid data are displayed.

SUBSTITUTE DATA Command

Operation: SUBSTITUTE DATA
Code: B
Syntax: (Address) (Data) [B] [Valid Data 1] [/] [Valid Data 2] [/] [Beginning Address] [/] [Number of Bytes] [CR]
Explanation: This command causes "Valid Data 2" to replace "Valid Data 1" within the specified address limits. If no addresses are specified, substitution occurs in all of RAM.

LIST DATA Command

Operation: LIST
Code: T
Syntax: (Address) (Data) [T] [Number of Locations]
Explanation: This command causes the terminal to list the specified number of RAM addresses and their data starting with the displayed address.

ESCAPE MEMORY MODE Command

Operation: ESCAPE MEMORY MODE
Code: E
Syntax: (Address) (Data) [E]
Explanation: This command causes exit from the Memory Mode. The terminal displays a prompt character (>) and will then recognize two-character commands. The ESC key can also be used for this purpose, but if ESC is pressed while processing is occurring, data in scratch pad buffers will be lost. Therefore, it is recommended that E always be used to escape the Memory Mode.

APPENDIX C

29B/HANDLER INTERFACE

C.1 INTRODUCTION

This appendix describes how to operate the 29B with the Handler UniPak™ and a commercially available handler. Covered are components, setup, and basic operation of the system. For more detailed information, consult your handler and Handler UniPak™ manuals.

C.2 COMPATIBILITY

The Handler UniPak™ will currently interface with both the Delta and Micro Component Technology (MCT) handlers. An Exatron handler is also compatible with the 29B, but does not require the Handler UniPak™ or the handler port. Instead, the Exatron handler interfaces with the 29B directly via computer remote control (CRC) and with the standard programming Paks. (Exatron supplies the interface package with the handler.) A third type of handler interface, also not requiring the Handler UniPak™, interfaces via the 29B's handler port and standard programming Paks (see section C.3).

C.3 29B HANDLER PORT INFORMATION

The following information is provided for 29B users who are making their own interface. Figure C-1 shows the port pin numbers; table C-1 defines the function of each. Figure

C-2 illustrates the handler control timing. To prevent device oscillation or other errors, it is recommended that the cable connecting the handler device contacts to the programming Pak socket be shielded and minimal length.

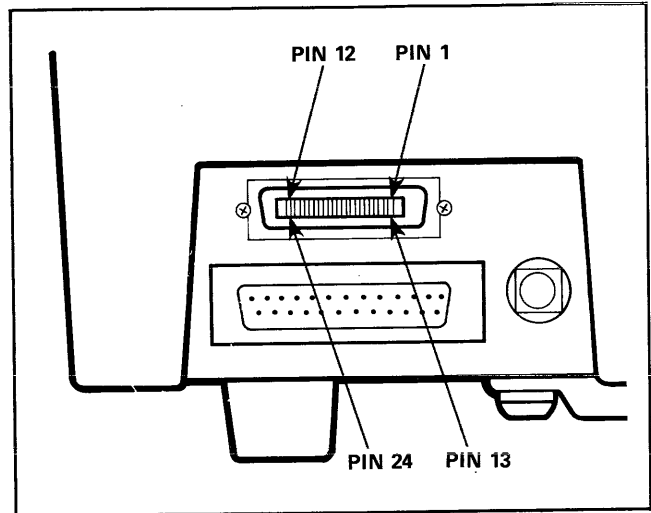


Figure C-1. 29B Handler Port Pin Assignment

Table C-1. Handler Interface Connector Pin Assignment

Pin No.	Signal Mnemonic	Description
1, 2	Ground	
3, 4, 5	+5V	Connected to the programmer's +5V logic supply. Available for external use if required (200mA max.).
6, 7, 8	Ground	
9	Illegal Bit Bin	Signal lines for selecting the appropriate handler bin. These open-collector outputs go to ground for 2 milliseconds when selected by the programmer.
10	Pass Bin	
11	Strobe	Signal used for handlers that need a strobe as well as a binning signal. This open-collector output goes to ground coincidentally with any of the four binning signals.
12-16	Ground	
17	Start	This input is internally pulled to +5V and should be grounded to inform the programmer of when a device is in place in the handler. This signal should not go high again until 1 millisecond after the previous binning signal goes high.
18, 19, 20	ID1, ID0, ID2	These inputs inform the programmer what is connected to the port. If pins 18, 19, 20 are all high, the 29B identifies the port as being not connected. If pin 19 is low, and 18 and 20 are high, the port is identified as being connected to a handler.
21	Programming Error Bin	Signal lines for selecting the appropriate handler bin. These open-collector outputs go to ground for 2 milliseconds when selected by the programmer.
22	Verify Error Bin	
23	Spare Bin	For future use.
24	Ground	

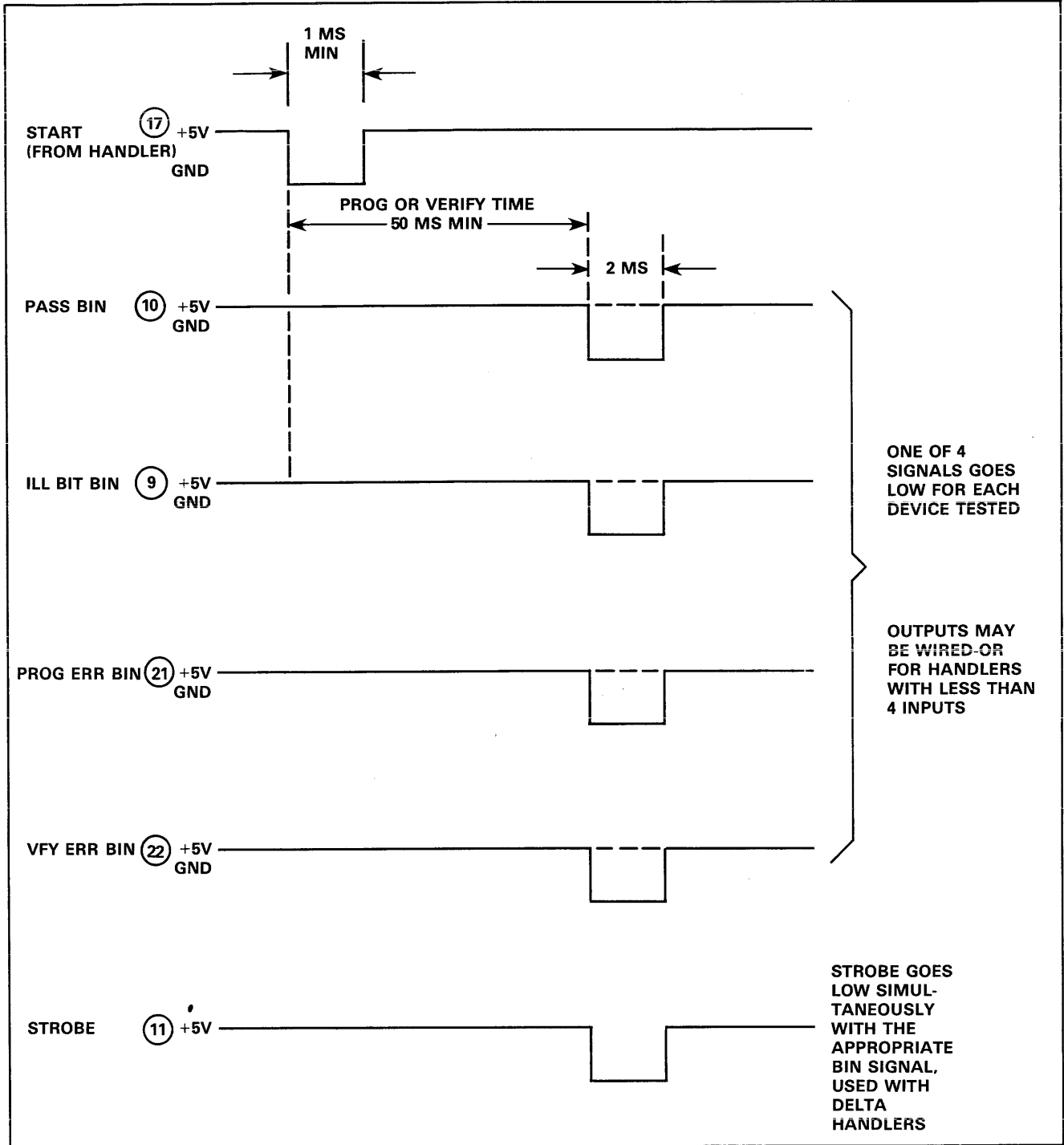


Figure C-2. Handler Control Signal Timing Diagram

C.4 SYSTEM COMPONENTS

As shown in figure C-3, the 29B/handler system consists of 1) the 29B Programmer, 2) the Handler UniPak™, 3) a handler compatible with the 29B, 4) a performance board, which configures the system to the particular device to be programmed, and 5) a terminal (optional). Also included in the system are the various cables that transmit data between components of the system.

C.5 PERFORMANCE BOARD SELECTION

The Performance Board configures the 29B/handler system to the particular device the user wishes to program. For all but 16-pin devices, select the board that has the same number of pins as the device you want to program.

For the 16-pin devices, there are two performance boards: one for 4-bit output devices and one for 8-bit output devices.

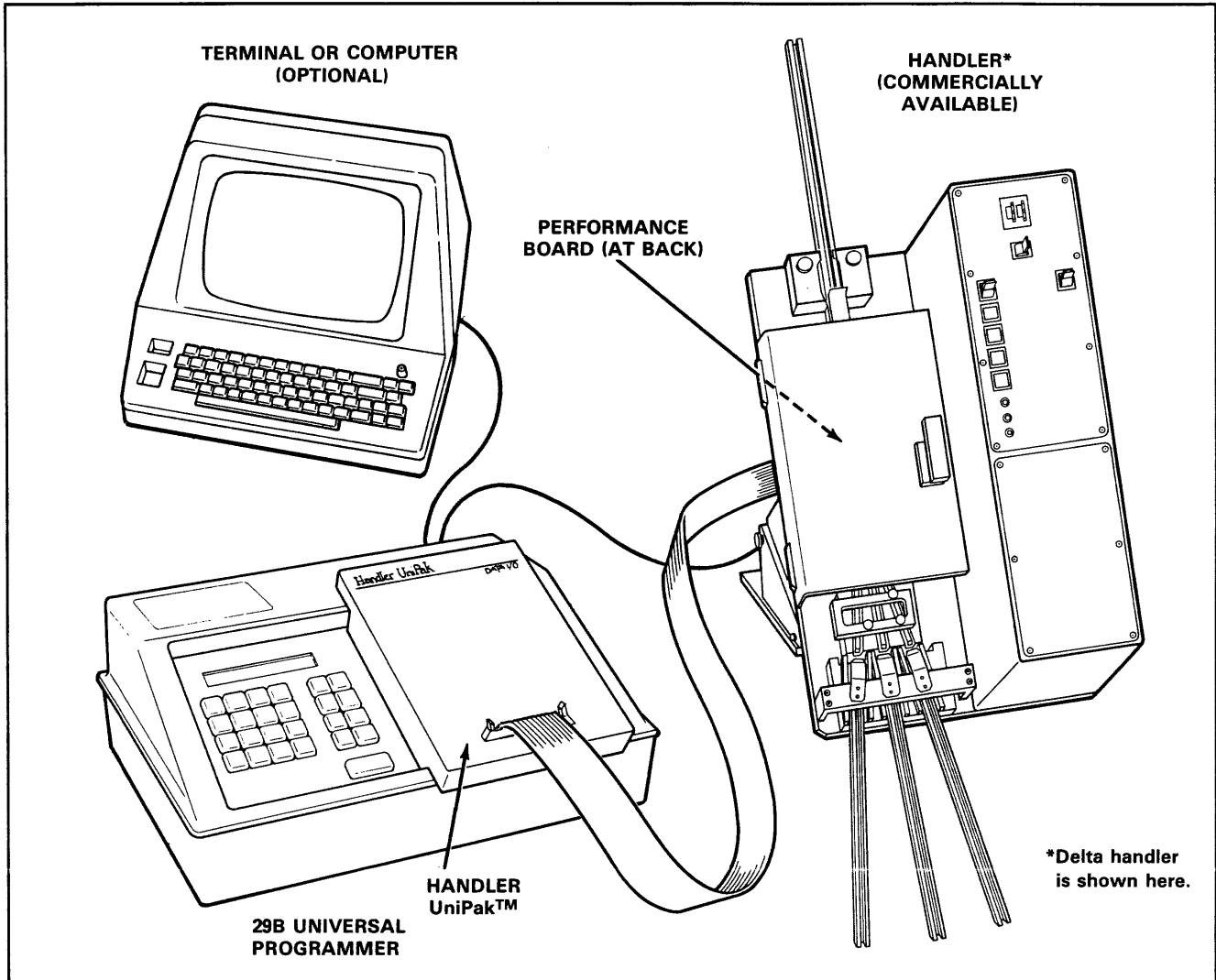


Figure C-3. The 29B/Handler System

C.6 MCT HANDLER SETUP

NOTE

The Delta handler does not require that any of its circuit boards be altered. This section refers only to the MCT handler.

Setting up the MCT handler for use with the 29B requires two procedures: 1) Altering the Opto-Interface Board and 2) setting the binning switches. These procedures are described in the MCT handler interface kit manual, (10-950-0102-001), and in section 1 of the MCT handler manual.

C.7 SYSTEM SETUP

This section of the appendix describes how to set up the 29B with a handler. Although the Delta and MCT handlers do not look the same, system setup is similar. Following is a list of the procedures which set up the components of the system for operation:

- Installing the Handler UniPak™ in the 29B (see section 2 of this manual for procedure)
- Connecting the Handler UniPak™ to the performance board (section C.7.1)
- Installing the performance board to the handler (section C.7.2)
- Connecting the 29B control cable to the handler (section C.7.3)
- Connecting a computer or terminal to the 29B (section C.7.4).

To set up the 29B/handler system, refer to figures C-4

through C-8 and follow the procedures given in the following subsections.

C.7.1 CONNECTING THE HANDLER UNIPAK™ TO THE PERFORMANCE BOARD

The Handler UniPak™ connects to the performance board via a 50-pin ribbon cable. The cable/performance board functions much like a socket adapter, sending the programming signals to the handler.

To connect the cable, proceed as follows:

1. Ensure that the connector tabs on the top of the Handler UniPak™ are snapped out (see figure C-4).
2. With the colored sidestrip (indicating pin 1) of the cable on the left, insert the 50-pin ribbon cable into the Handler UniPak™'s connector.
3. Ensure that the connector tabs on the Handler UniPak™ are now snapped in.

NOTE

Pin 1 is marked on both the ribbon cable and the connector. On the Handler UniPak™'s connector, the pin designator is on the underside and to the left (see figure C-2).

4. Ensure that the connector tabs on the performance board are snapped out (see figure C-4).
5. Insert the 50-pin ribbon cable into the connector on the performance board. Ensure that the colored sidestrip (indicating pin 1) of the cable is on the same side as pin 1 indication on the performance board. Ensure that the connector tabs are now snapped back in.

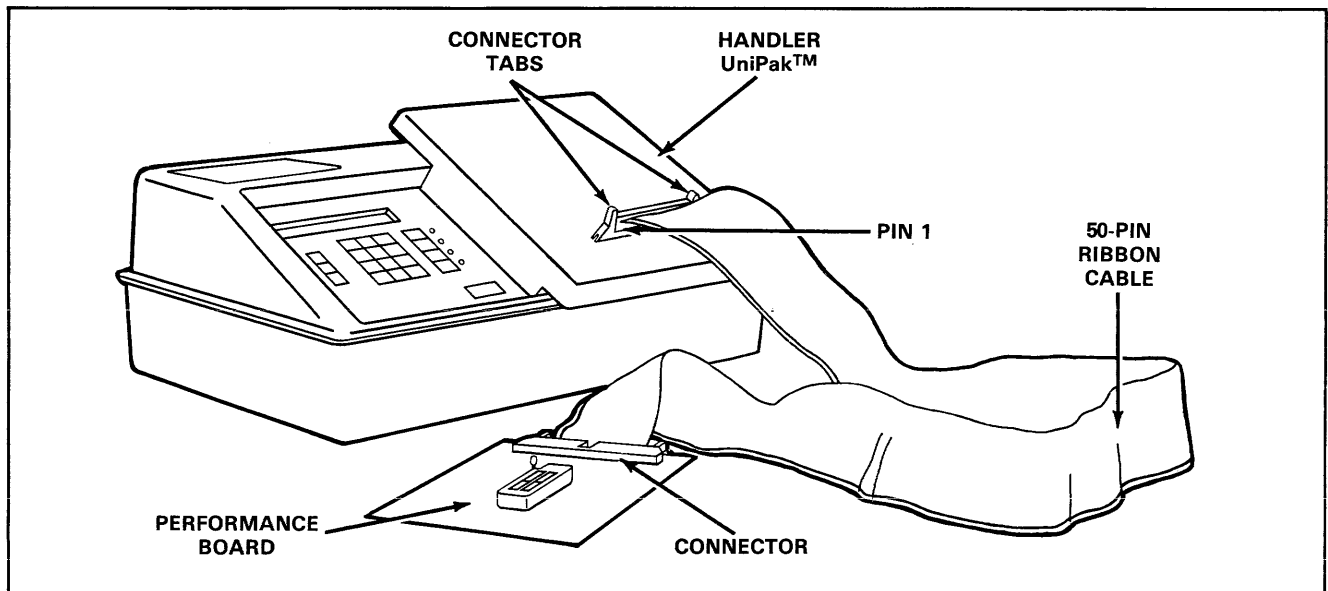


Figure C-4. Connecting the Handler UniPak™ and Performance Board

C.7.2 INSTALLING THE PERFORMANCE BOARD

To install the performance board in the handler, fit the pins on the handler into the holes on the performance board. Four standoffs provide a secure physical attachment. See figures C-5 (MCT handler) and C-6 (Delta handler) for an illustration of the attachment procedure.

C.7.3 CONNECTING THE 29B TO THE HANDLER

The handler connects to the 29B via a digital control interface cable. This cable provides handler control, including binning commands. See figures C-7 (MCT handler) and C-8 (Delta handler) for an illustration of the procedure.

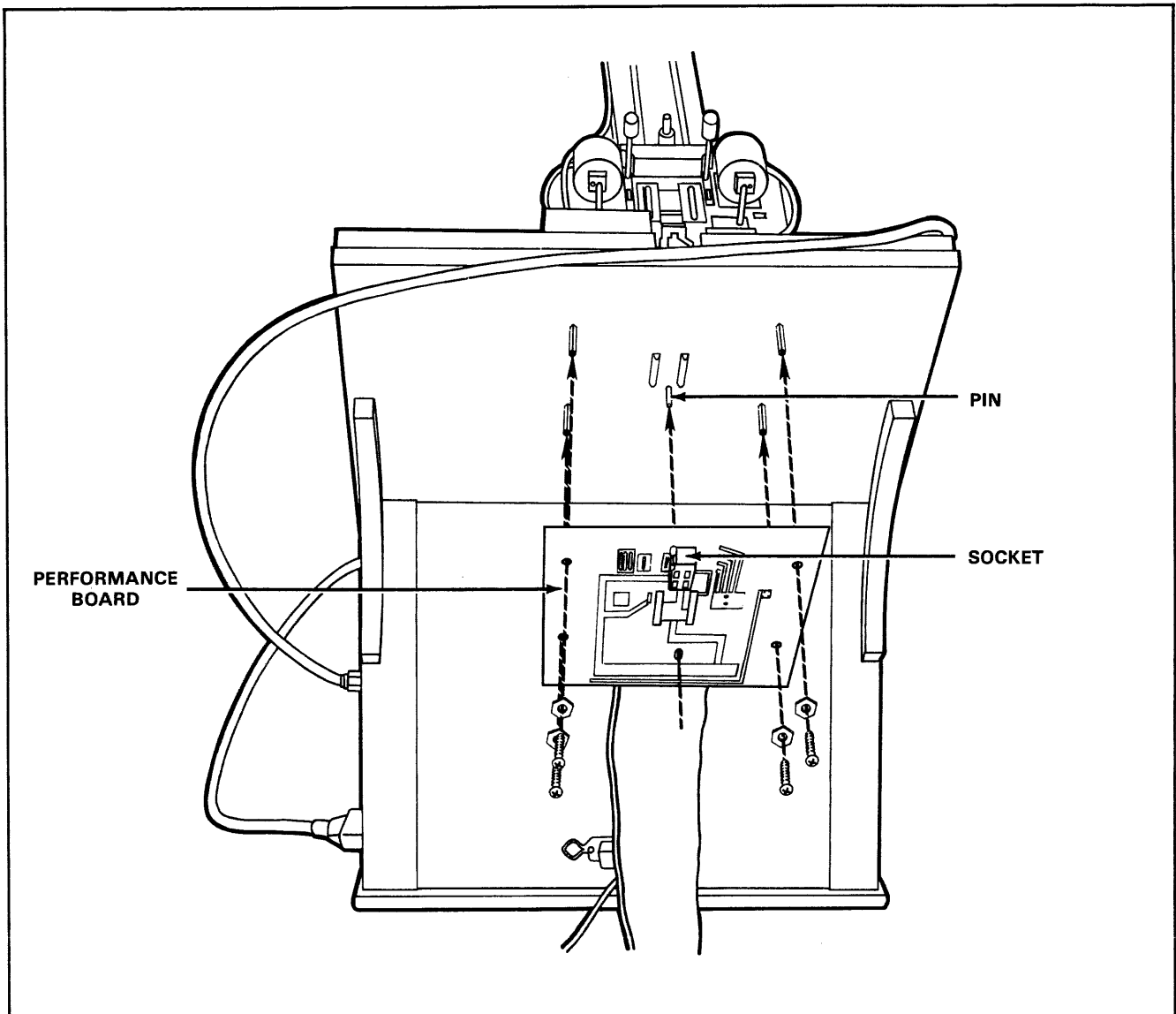


Figure C-5. Installing the Performance Board-MCT Handler

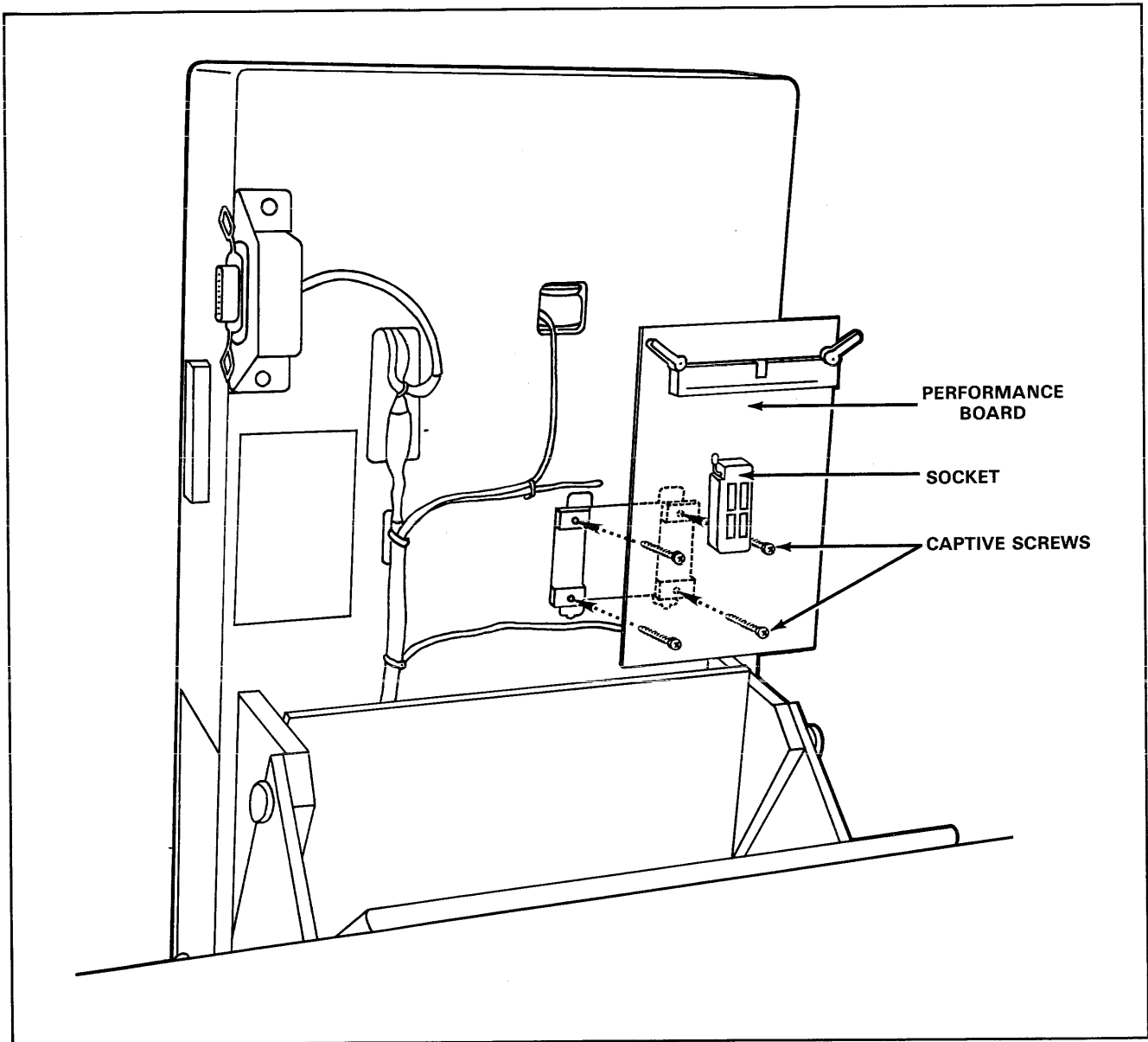


Figure C-6. Installing the Performance Board—Delta Handler

C.7.4 CONNECTING A COMPUTER OR TERMINAL TO THE 29B

Commands can be issued to the 29B via a computer or terminal using Computer Remote Control (CRC). To operate the 29B via computer, the user will have to provide special interface software. If you wish to perform operations with the system using a computer or terminal, connect an RS232C transmission cable between the 29B and the computer or terminal. See section 2.5 of this manual for cable installation details, and section 3.9.1 for more information on CRC.

C.8 POWER UP

CAUTION

Voltage transients can damage a device. Be sure the performance board and the handler are free of devices before you apply power to the system or before you install or remove the Handler UniPak™.

To power up the system, follow the procedures given in this manual (sections 2.4 and 3.2.2) and the handler's manuals.

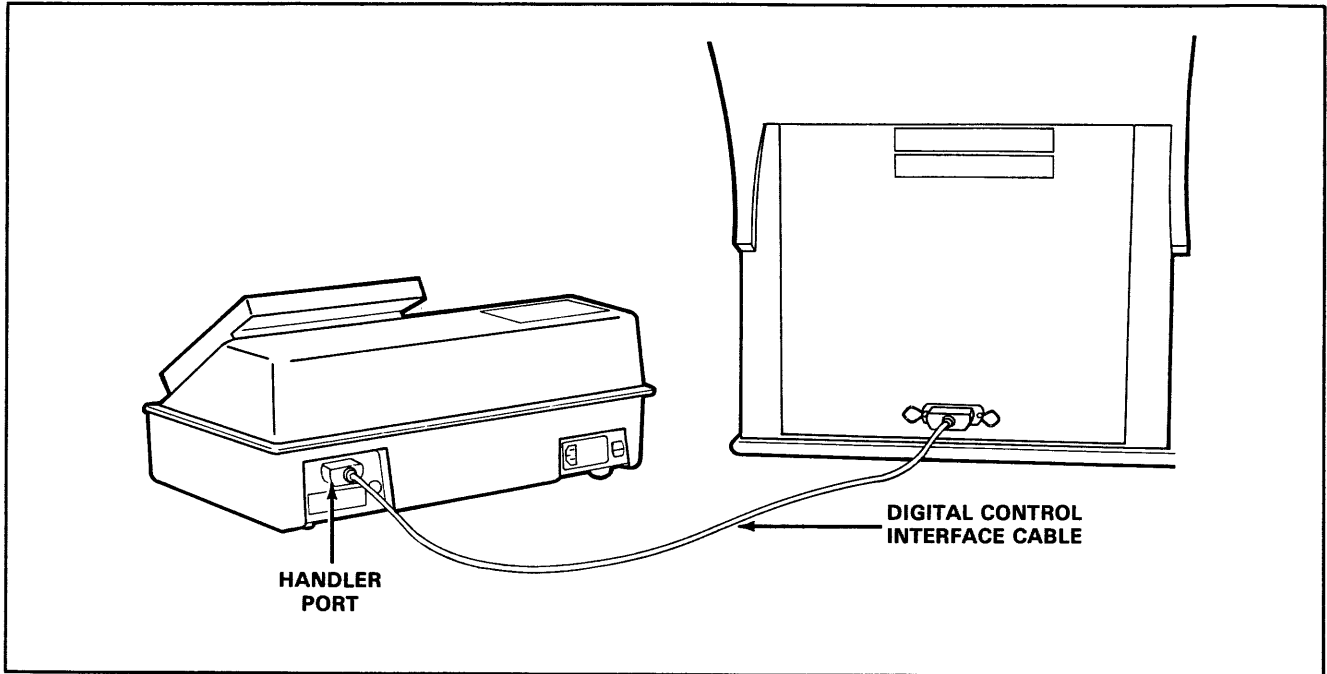


Figure C-7. Handler/Programmer Connection—MCT Handler

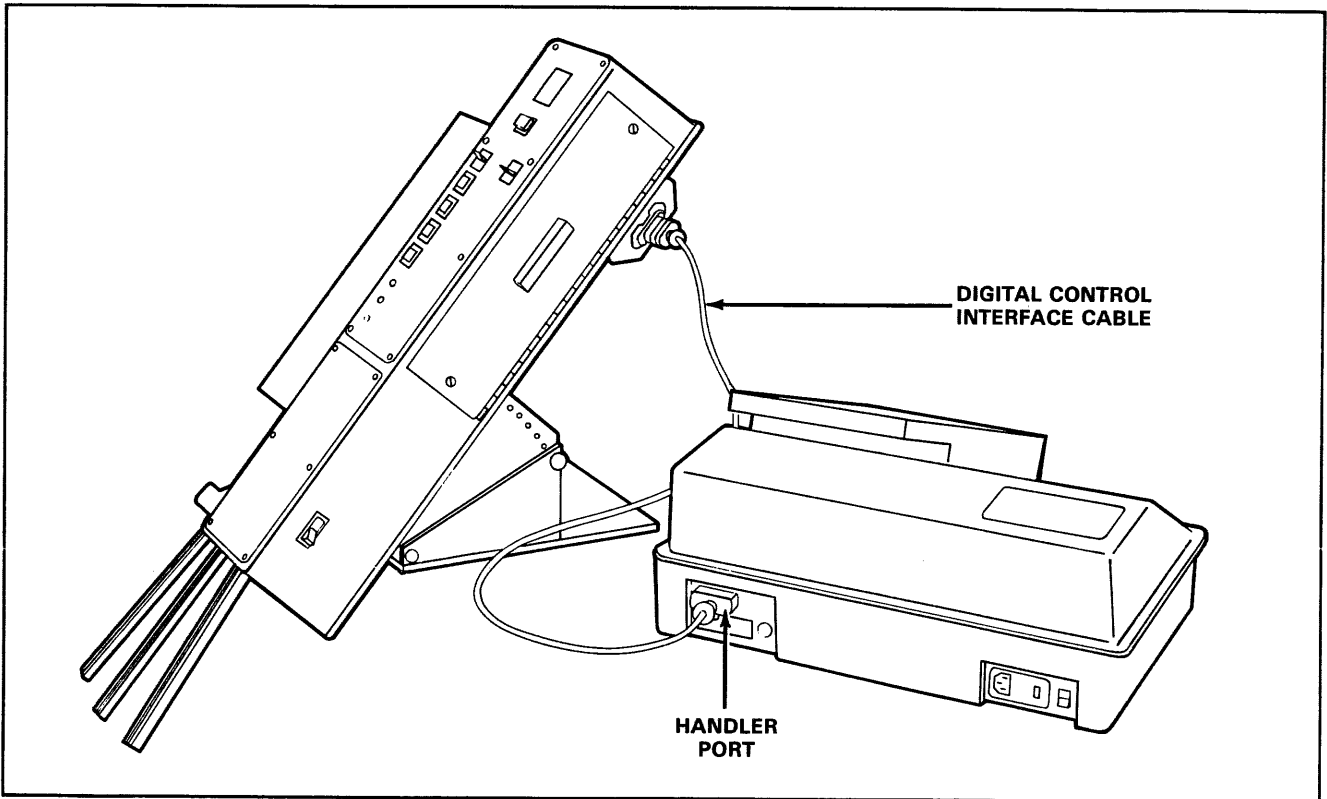


Figure C-8. Handler/Programmer Connection—Delta Handler

C.9 BASIC DATA TRANSFER OPERATIONS

The basic operations that can be accomplished with the 29B/handler system are:

- load the 29B's RAM with master device data (section C.9.5)
- program devices with the 29B's RAM data (section C.9.6)
- verify the 29B's RAM data against device data (section C.9.7)

These operations may be performed on the handler from either the 29B's keyboard or via CRC. This section describes operational setup and basic data transfer operations that are unique to the 29B/handler system. A more detailed discussion of these data transfer operations can be found in section 3 of this manual.

C.9.1 ENTERING CRC MODE (Optional)

If you are going to use a computer or a terminal to perform operations with the system, enter select code F1 on the 29B to enter CRC mode.

To exit CRC, send the "Z" return command, or press any mode key on the 29B's keyboard. See section 3.9.1 of this manual for a more detailed discussion of the CRC format.

There are two remote handler commands specific to computer remote control. A description of these commands is given in table C-2.

Table C-2. Handler-Specific Commands (CRC Mode)

COMMAND	DESCRIPTION
%	This command puts the programmer into a waiting state until the handler indicates a start condition or the operator presses the START key on the 29B. Once the start condition has been received by the 29B, it will output a prompt ">" symbol to the serial port and wait for another command.
N!	This command instructs the handler to distribute programmed and verified devices to the number "n" bin, where "n" is an integer from 1 to 4.

C.9.2 ENTERING FAMILY AND PINOUT CODES

Before you can perform any device-related operations with the Handler UniPak™, you must key in the family and pinout codes of the devices you wish to program. The codes may be entered from either the 29B's keyboard or the terminal. Check the device table in section 1 of the Handler UniPak™ manual and locate the two-digit family and two-digit pinout codes of the device you wish to program. To enter the family and pinout codes into the system, follow the procedures given in section 3.4.1 of this manual. After a family and pinout code has been entered, it will remain in effect until the user changes it or power to the 29B is shut off.

C.9.3 DEVICE INSERTION

Devices may be placed into the 29B/handler system at two locations: 1) the performance board, and 2) the top of the handler.

If you want to load the 29B's RAM with a master device's data, or if you want to verify the 29B's RAM data with master device data, insert a device into the performance board's device socket. See section 3.4.2 of this manual for device insertion procedure.

Insert devices at the top of the handler if you want to program a device(s) or verify preprogrammed device(s). When inserting devices into the Delta handler, insert pin 1 (notched end) of the device first. When using the MCT handler, insert pin 1 last. Devices inserted backward will cause a backward device error to display on the 29B keyboard.

C.9.4 BINNING CONTROL

Front Panel Control

When programming or verifying devices from the front panel, binning controls are defined as follows:

Bin 1: Device passes

Bin 2: Device fails—illegal bit

Bin 3: Device fails—failure during programming

Bin 4: Device fails—failed to verify after programming

NOTE

The Delta handler sends both program and verify failures to bin 3.

CRC Mode

When operating the handler via CRC, information regarding the status of an operation (verify or program) can be obtained from the 29B. Based on this status information, the user can then bin the device(s) using the "!" command. For example, to send a device to bin 2, you would enter 2![CR] from the terminal's keyboard (where [CR] denotes a carriage return). In general, you would send a binning command after each device has been programmed or has failed an operation. See section 3.9 of this manual for more information on CRC.

C.9.5 PREVENTING NONBLANK DEVICES FROM BEING PROGRAMMED (Select Code B4)

To prevent the system from programming nonblank devices, enter the "nonblank fail" select code B4 from the 29B's keyboard. This code, when enabled, will prevent you from programming over parts that already contain data. Parts that are nonblank will go to bin 2, and an illegal bit error message will display on the 29B's keyboard.

NOTE

When six devices in a row fail for the same reason, the system will shut down, and the sixth device will remain in the handler. It may then be binned either manually or via CRC.

Once the "nonblank fail" code is set, the only way to disable the code is to shut off the 29B's power. See section 3.8.5 of this manual for the key sequence that enables this function.

C.9.6 LOAD RAM WITH MASTER DEVICE DATA

Front Panel Control

To load the 29B's RAM with data from a master device, follow the procedure given in section 3.6.1 of this manual, with one exception:

Step 6 of the procedure is device insertion. For this step, insert the master device into the performance board's socket.

CRC Mode

To load the 29B's RAM with data from a master device, proceed as follows:

1. Insert the master device into the performance board's socket (see section 3.4.2 for insertion discussion).
2. Issue the family and pinout codes for the device using the "@" command if they are not already entered. See section 3.9.1 for details.
3. Issue the Load command: "L[CR]".
4. The 29B will respond with one of three characters: ">", "F" or "?". See section 3.9.1 for details on CRC.

C.9.7 PROGRAM DEVICE(S) WITH 29B RAM DATA

Front Panel Control

To program device(s) with data from the 29B's RAM, proceed as follows:

1. Set up the handler for the type of devices to be programmed (set device width, binning controls, etc.).
2. Insert the device(s) into the top of the handler. See the handler manual for details on device insertion.
3. Start the handler.
4. Perform the "Program Device with RAM Data" procedure given in section 3.6.3 of this manual, with one exception: Step 6 calls for device insertion. Ignore this step, since the devices to be programmed are already in the handler.

NOTE

Devices will be automatically programmed or failed as long as the handler continues to feed them. The programmed or failed devices will be automatically binned as described in section C.9.4.

CRC Mode

To program device(s) with data from the 29B's RAM while in CRC mode, proceed as follows:

1. Set up the handler for the type of devices to be programmed (set device width, binning controls, etc.).
2. Insert the device(s) into the top of the handler. See the handler manual for details on device insertion.
3. Start the handler.
4. Issue the family and pinout codes for the device, (using the "@" command) if they are not already entered (see section 3.9.1 for details).
5. Issue the "T" command for an illegal-bit check and give the "B" command for blank check. Issue binning signals in case of error using the "!" command.
6. Issue the Program command: "P[CR]".
7. The 29B programmer will respond with one of three characters: ">", "F" or "?". See section 3.9.1 for details on CRC.
8. Issue a binning command (1!, 2!, 3! or 4!) to bin the device.
9. If the "%" function has been enabled, wait for the 29B to issue a ">" prompt. If this function is not being used, wait until the next device is ready.
10. Repeat steps 5 through 9 until all devices have been programmed.

C.9.8 VERIFY DEVICE'S DATA WITH 29B RAM DATA

Front Panel Control

To verify device(s) with data from the 29B's RAM, proceed as follows:

1. Set up the handler for the type of devices to be verified (set device width, binning controls, etc.).
2. Insert the device(s) into the top of the handler. See the handler manual for details on device insertion.
3. Start the handler.
4. Perform the "Verify Device Data with 29B RAM Data" procedure given in section 3.6.6 of this manual, with one exception: Step 6 calls for device insertion. Ignore this step, since the devices to be verified are already in the handler.

NOTE

Devices will be automatically verified or failed as long as the handler continues to feed them. The verified or failed devices will be automatically binned as described in section C.9.4.

CRC Mode

To verify device(s) with data from the 29B's RAM while in CRC mode, proceed as follows:

1. Set up the handler for the type of devices to be verified (set device width, binning controls, etc.).

2. Insert the device(s) into the top of the handler. See the handler manual for details on device insertion.
3. Start the handler.
4. Issue the family and pinout codes for the device (using the "@" command) if they are not already entered (see section 3.9.1 for details).
5. Issue the Verify command: "V[CR]".
6. The 29B programmer will respond with one of three characters: ">", "F" or "?". See section 3.9.1 for details on CRC.
7. Issue a binning command (1!, 2!, 3! or 4!) to bin the device.
8. If the "%" function has been enabled, wait for the 29B to issue a ">" prompt. If this function is not being used, wait until the next device is ready.
9. Repeat steps 5 through 8 until all devices have been verified.

C.10 OTHER COMMANDS

Commands other than the Load, Program and Verify mentioned in this appendix operate in the same way for handler operation as they do for non-handler operation. See section 3 of this manual for key sequences and terminal commands for other operations.

APPENDIX D

REFERENCE MATERIAL

Glossary

Abbreviations

Cross-Reference Chart of Number Bases

ASCII and IEEE Code Chart

ASCII Control Characters

Glossary

address field. A set of characters used with some data translation format, which defines the address of the next data byte.

address offset. A value subtracted from addresses during input translation and then added to addresses during output translation.

begin device address. The first device address from which or to which data is being transferred.

begin RAM address. The first address of the programmer data RAM from which or to which data is to be transferred.

blank check. A test performed by a programmer to detect the presence of any programmed bits. A device with no programmed bits is "blank."

block size. The hexadecimal number of bytes to be transferred in a data transfer.

bootstrap. The basic software routine which performs initial power-up machine checks and prepares the machine to receive and respond to operating system instructions.

configuration number. A 4-digit hex number that identifies the software revision level of the programmer.

data translation format. Form in which the translator software accepts input data and transmits output data.

default value. The value the unit uses for a parameter unless the operator specifies another value.

device. Any PROM, PAL, EPROM, EEPROM, or programmable logic part.

end code. Character in a data translation format which signals the completion of a data transfer.

error code. A code which signals specific errors to the operator.

Family and Pinout Codes. Two-digit codes used by some Data I/O programming modules to identify programming variables including pinout, address limit and programming algorithms.

generic family. Devices of different memory size developed by a manufacturer that require the same programming voltages, currents, and timing relationships. They can be programmed by the same programming module.

handshaking. The required sequence of signals for communication between two units. The I/O bus protocol for a unit defines its handshaking requirements. This is especially true for asynchronous I/O systems in which each signal requires a response to complete an I/O operation.

illegal-bit test. A test performed by a programmer to detect the presence of any programmed bits of incorrect polarity (illegal bits).

mode. A software routine in a machine, characterized by a specific automatic sequence of steps.

nibble. One half of an 8-bit byte.

programming module. Generic term for Data I/O Programming Pak, Program Card Set, UniPak, Gang Module, FPLF Pak, MOS Pak, IFL Pak, and other programming electronics.

record size. The number of bytes printed on a line of a teletype or other printer; or the number of bytes printed on a paper tape before another address field is printed.

scratch pad memory. The internal RAM memory used by the processor for performing calculations.

select function. A two-digit hex number used to specify data translation formats, serial interface operations, or certain RAM data manipulations.

start code. Character in a data translation format which signals the beginning of a data transfer.

Glossary (Continued)

sum-check. A summation of bits calculated according to the rules of simple addition and usually expressed as a four-digit hex number; any carry from the most significant bit or digit is discarded. A sum-check is used to verify the integrity of data transfers.

waveforms (programmable). The graphical representation of the timing and magnitude of programming pulses. If the programming waveforms are not kept within tolerance, programming yield is jeopardized.

word limit. The highest address in a device. For example, the word limit of a 1Kx8 device is 1K (or hex 3FF).
Synonymous with address limit.

word width. The number of bits in a byte or word (4 or 8).

HEX DATA	BINARY DATA
84	1000100
C1	1100001
62	01100010
24	00100100
01CB	0000 0001 1100 1011
↑	↑
Sum-check in hexadecimal notation	Sixteen-bit binary sum-check

Sample Sum-check Calculation

Abbreviations

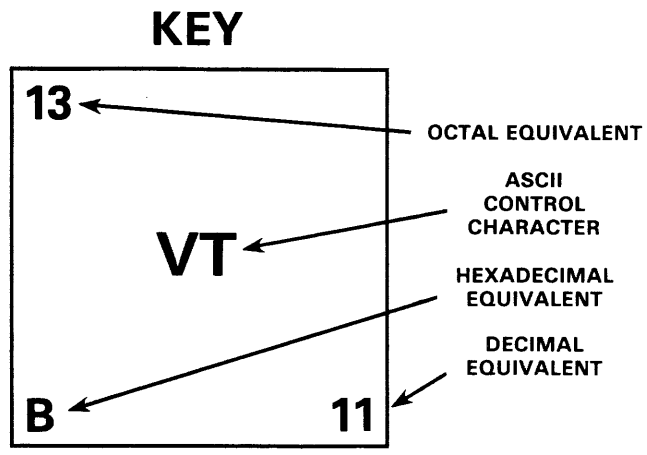
The following is a list of abbreviations commonly used in Data I/O Instruction manuals.

A ₄ - address line 4	J - jack or connector
ADDR - address	JP - jumper
BC - bin count, number of bins	K - relay
BR - bridge rectifier	LIM - limit
C - capacitor	LSB - Least Significant Bit
\overline{CE} - Chip Enable	LSD - Least Significant Digit
Clk - clock	MSB - Most Significant Bit
Clk. Inh. - clock inhibit	MSD - Most Significant Digit
Cntl. - control	NMI - Non-Maskable Interrupt
Cont. - control	NO CONT SECT - No Contiguous Sector
CR - diode	Oper. - operate
CTS - Clear To Send	PA ₁₅ - programmer address line 15
D ₅ - data line 5	PAK - programming module
DAC - Digital to Analog Converter	PCS - Program Card Set
DC - division count, number of partitions	PD ₆ - programmer data line 6
DCD - Data Carrier Detect	PN - Part Number
DCU - Data I/O Data Control Unit	PR/ \overline{OE} - Preset/Output Enable
DI ₂ - data input 2	Prog - Program
DIR - Directory	Prog. Pulse - Program Pulse
DO ₂ - data output 2	PROM - Programmable Read Only Memory
DS - display	Q - transistor
DSR - Data Set Ready	R - resistor
DTR - Data Terminal Ready	RAM - Random Access Memory
DVM - digital voltmeter	Read Inh. - Read Inhibit
Emul - emulate	Rec. - receive
ERR - error	Rev. - reverse
ESC - escape	RP - resistor pack
F - fuse	RST - Reset
FC - Translation Format Code	R/ \overline{W} - Read/Write
FFPP - Family Code (FF) and Pinout Code (PP)	RX - Receive Data
FIP - Fluorescent Indicator Panel	S - switch
FPGA - Field Programmable Gate Array	TOR - Turn On Reset
FPLA - Field Programmable Logic Array	TX - Transmit Data
FPLS - Field Programmable Logic Sequencer	U - integrated circuit device
FPRP - Field Programmable ROM Patch	$\overline{V_{02}}$ - the "AND"-ing of the Valid Memory Address line and the phase 2 line
FRME - frame	Ver. A - Verify pass A
Fwd. - Forward	Ver. B - Verify pass B
Gnd. - Ground	VFY - verify
HLT. - Halt	VMA - Valid Memory Address
HV - high voltage	VR - Voltage Regulator
ID ₄ - identification line 4	V _{REF} - Voltage Reference
IFL - Integrated Fused Logic	W/L - Word Limit
I/O - Input/Output	Write Inh. - Write Inhibit
\overline{IRQ} - Interrupt Request	

Cross-Reference Chart of Number Bases

Binary	Octal	Hexadecimal	Decimal	Standard Abbreviation
0000	0	0	0	
0001	1	1	1	
0010	2	2	2	
0011	3	3	3	
0100	4	4	4	
0101	5	5	5	
0110	6	6	6	
0111	7	7	7	
1000	10	8	8	
1001	11	9	9	
1010	12	A	10	
1011	13	B	11	
1100	14	C	12	
1101	15	D	13	
1110	16	E	14	
1111	17	F	15	
0001 0000	20	10	16	
0010 0000	40	20	32	
0100 0000	100	40	64	
1000 0000	200	80	128	
0001 0000 0000	400	100	256	
0010 0000 0000	1000	200	512	
0100 0000 0000	2000	400	1,024	1K
1000 0000 0000	4000	800	2,048	2K
1100 0000 0000	6000	C00	3,072	3K
0001 0000 0000 0000	10000	1000	4,096	4K
0001 0100 0000 0000	12000	1400	5,120	5K
0001 1000 0000 0000	14000	1800	6,144	6K
0001 1100 0000 0000	16000	1C00	7,168	7K
0010 0000 0000 0000	20000	2000	8,192	8K
0010 0100 0000 0000	22000	2400	9,216	9K
0010 1000 0000 0000	24000	2800	10,240	10K
0100 0000 0000 0000	40000	4000	16,384	16K
1000 0000 0000 0000	100000	8000	32,768	32K
0001 0000 0000 0000 0000	200000	10000	65,536	64K

ASCII & IEEE Code Chart



**ASCII & IEEE Code Chart
(Continued)**

7 6 5 BITS 4 3 2 1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS & SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL	20 DLE	40 SP	60 0	100 @	120 P	140 1	160 p
0 0 0 1	1 SOH	21 DC1	41 !	61 1	101 A	121 Q	141 a	161 q
0 0 1 0	2 STX	22 DC2	42 "	62 2	102 B	122 R	142 b	162 r
0 0 1 1	3 ETX	23 DC3	43 #	63 3	103 C	123 S	143 c	163 s
0 1 0 0	4 EOT	24 DC4	44 \$	64 4	104 D	124 T	144 d	164 t
0 1 0 1	5 ENQ	25 NAK	45 %	65 5	105 E	125 U	145 e	165 u
0 1 1 0	6 ACK	26 SYN	46 &	66 6	106 F	126 V	146 f	166 v
0 1 1 1	7 BEL	27 ETB	47 '	67 7	107 G	127 W	147 g	167 w
1 0 0 0	10 BS	30 CAN	50 (70 8	110 H	130 X	150 h	170 x
1 0 0 1	11 HT	31 EM	51)	71 9	111 I	131 Y	151 i	171 y
1 0 1 0	12 LF	32 SUB	52 *	72 :	112 J	132 Z	152 j	172 z
1 0 1 1	13 VT	33 ESC	53 +	73 ;	113 K	133 [153 k	173 }
1 1 0 0	14 FF	34 FS	54 ,	74 v	114 L	134 \	154 l	174
1 1 0 1	15 CR	35 GS	55 .	75 =	115 M	135]	155 m	175 }
1 1 1 0	16 SO	36 RS	56 .	76 >	116 N	136 ^	156 n	176 ~
1 1 1 1	17 SI	37 US	57 /	77 ?	117 O	137 _	157 o	177 Rubout
	Addressed Commands	Universal Commands	Listen Addresses		Talk Addresses		Secondary Addresses or Commands	

ASCII Control Characters

ACK	acknowledge
BEL	bell
BS	backspace
CAN	cancel
CR	carriage return
DC1	playback on, CNTL Q, X-ON
DC2	record on, CNTL R, PUNCH-ON, SOM
DC3	playback off, CNTL S, X-OFF
DC4	record off, CNTL T, PUNCH-OFF, EOM
DEL	delete, rubout
DLE	data link escape
EM	end of medium
ENQ	enquiry
EOT	end of transmission
ESC	escape
ETB	end of transmission block
ETX	end of text
FF	form feed
FS	file separator
GS	group separator
HT	horizontal tabulation
LF	line feed
NAK	negative acknowledge
NUL	null
RS	record separator
SI	shift in
SO	shift out
SOH	start of heading
STX	start of text
SUB	substitute
SYN	synchronous idle
US	unit separator
VT	vertical tab

APPENDIX E
ERROR CODES

Table E-1. Error Codes

DISPLAY	DESCRIPTION	CORRECTIVE ACTION
<i>SORC/DEST ERR</i> 15	Illegal source/destination key sequence was entered.	Check key sequence and re-enter.
<i>COMMAND ERROR</i> 17	Illegal key sequence while in standard Remote Control.	Check key sequence and re-enter.
<i>ILL RAM PAGE</i> 18	The operation would require a 16K RAM page change, which the present PAK software does not support.	Either modify Begin RAM or block size so that the first and last bytes of the operation reside in the same 16K range (0-3FFF, 4000-7FFF, 8000-BFFF, C000-FFFF), or contact your local Data I/O Service Center for a PAK software update.
<i>NONBLANK</i> 20	Device failed the blank test.	Press START to override this error and program the device.
<i>ILLEGAL BIT</i> 21	Not possible to program the device due to already programmed locations of incorrect polarity.	Erase the device if possible or discard it.
<i>PROGRAM FAIL</i> 22	The program electronics were unable to program the device.	Either the device is bad or the programming module is inoperative or out of calibration.
<i>VERIFY FAIL 1</i> 23	The device data was incorrect on the first pass of the automatic verify sequence during device programming.	This error indicates that the device failed the low voltage verify; the data in the part is not the same as the RAM data.
<i>VERIFY FAIL 2</i> 24	The device data was incorrect on the second pass of automatic verify sequence during programming.	This error indicates that the device failed the high voltage verify; data in the part is not the same as the RAM data.
<i>NO PROG PAK</i> 25	A device-related operation was attempted without any programming module installed.	Install the appropriate programming module.
<i>PROG PAK RST</i> 26	The programming electronics will not start operation due to a reset condition.	Usually an overcurrent caused by an incorrectly inserted or bad device.
<i>RAM EXCEEDED</i> 27	There is insufficient RAM. The total allotment of RAM resident is less than the word limit or block size, or the begin address is set too high.	Program smaller parts or specific lower beginning address. If enough RAM is installed, it may be faulty
<i>ERRORS</i> 30-39	These are pak-related errors.	Refer to your programming pak manual.
<i>FRAME ERR</i> 41	The serial interface detected a start bit but the stop bit was incorrectly positioned.	Check the baud rate and stop bit switches or use handshake.
<i>OVERRUN ERR</i> 42	The serial interface received characters when the programmer was unable to service them.	Check the baud rate and stop bit switches or use handshake.
<i>FRME+OVR ERR</i> 43	Combination of FRAME ERR 41 and OVERRUN ERR 42.	Check the baud rate and stop bit switches or use handshake.

Table E-1. Error Codes (Continued)

DISPLAY		DESCRIPTION	CORRECTIVE ACTION
<i>I/O TIMEOUT</i>	46	No character (or only nulls and rubouts) were received on serial input for 25 seconds after pressing the START key, or no characters could be transmitted for a period of 25 seconds due to the state of the handshake lines.	Check all connections; then restart operation. I/O timeout can be disabled by select code F9.
<i>FAULTY ACIA</i>	47	ACIA chip may have failed.	Contact your local Data I/O Service Center.
<i>I/O OVERRUN</i>	48	The serial port input buffer received too many characters after the handshake line informed the sending device to stop.	Make sure the handshake lines are hooked up and operative.
<i>ERROR</i>	50*	Incoming data in optional Terminal Remote Control (TRC) mode was not recognized by the programmer. As a result, data was not stored in RAM.	Check all connections of units in the system including serial port cable. Then check to make sure you have the correct data format and data source, and then try again.
<i>I/O VFY FAIL</i>	52	The data from the serial port did not match the data in RAM.	Check and resend the data.
<i>ERROR</i>	56*	This is a Terminal Remote Control (TRC) related error that occurs when a begin RAM address is below the virtual memory offset, or, if the virtual memory offset, plus block size exceeds the 29B data RAM storage area. Also, a block size of zero will cause this error.	Make sure the beginning RAM address is greater or equal to the address offset. Also, decrease the block size so that it conforms to the RAM size and that it is greater than zero.
<i>ERROR</i>	58*	Programmer has not received the family and pinout codes for the device while in TRC.	Define the correct family and pinout codes for the device. In TRC use the "FY" command to define the family and pinout codes.
<i>NO RAM</i>	61	There is no working RAM in the programmer.	Replace faulty RAM or have the programmer serviced by your local Data I/O Service Center.
<i>RAM BIT ERR</i>	62	The highest RAM address in the programmer is not on a 1K boundary.	Replace faulty RAM or have the programmer serviced by your local Data I/O Service Center.
<i>RAM WRITE ERR</i>	63	The programmer is unable to write the intended data in RAM.	Failure of the associated RAM chip; replace the failed chip.
<i>RAM DATA ERR</i>	64	The programmer detected a spurious change in RAM data.	Reload data into RAM. If problem persists, service the programmer or notify your local Data I/O Service Center.
<i>ERROR</i>	65	The sum-check of software residing in the installed programming module is in error.	Contact your local Data I/O Service Center.
<i>IRQ ERR</i>	66	The IRQ line to the processor was held low for no apparent reason.	Ignore. If the error persists, service the programmer.

Table E-1. Error Codes (Continued)

DISPLAY	DESCRIPTION	CORRECTIVE ACTION
67*	Programmer received a non-valid command in Computer Remote Control (CRC).	Use the password to release data lock. (See subsection 3.8.)
<i>DATA LOCKED</i>	68 Data locked via Select Function F3.	
<i>ERRORS</i>	70 thru 79 These are pak-related errors. Refer to your programming pak manual.	Refer to your programming pak manual.
<i>ERROR</i>	80* Incoming data in Terminal Remote Control (TRC) has improper parity, checksum or data characters (missing or invalid). To determine which of these errors apply, use TRC command "QA."	Wrong parity? Check the 29B parity setting. See section 2.5.2. Incorrect checksum? Check the connection of units in the system, data format, and data source, and then try again. Improper (missing or invalid) data characters? Check the connection of all units in the system, data format, and data source, and then try again.
<i>PARITY ERR</i>	81 The incoming data has incorrect parity.	Check the parity switch and try again.
<i>SUMCHK ERR</i>	82 The sum-check field received by the programmer does not agree with its own calculated sum-check. For ASCII Binary formats, this error message indicates a missing F character.	Check all connections of units in the system, data format, and data source, and then try again.
<i>COMPOSITE ERR</i>	83, 85, 86 and 87 A composite error occurs from any combination of errors 81, 82 and 84. These combinations are: Error 83 = errors 81 and 82 Error 85 = errors 81 and 84 Error 86 = errors 82 and 84 Error 87 = errors 81, 82 and 84	
<i>INVALID DATA</i>	84 The programmer received invalid or not enough data characters. Non-data characters (formats 01-03, 5-9 12-13) Non-hex characters (formats 70, 81, 88, 90)	Check the connection of all units in the system, data format and data source, and then try again.
<i>INVALID FORM</i>	90 Non-existent I/O format is selected in Computer Remote Control (CRC).	Enter a legal format code.
<i>I/O FORM ERR</i>	91 The programmer received an invalid character in the address field.	Check the connection of all units in the system data format, and data source, and then try again.
<i>I/O FORM ERR</i>	92 The address check was in error (Signetics Twin and Tektronix Hexadecimal formats only).	Check the connection of units in the system, data format, and data source, and then try again.

Table E-1. Error Codes (Continued)

DISPLAY	DESCRIPTION	CORRECTIVE ACTION
<i>BLOCK MOVE ERR 97</i>	Block Move was attempted outside RAM boundaries.	Redefine parameters.
<i>DEV EXCEEDED 98</i>	Programming data exceeded the last device address.	Redefine parameters.
<i>ERRORS A0 thru A9</i>	These are pak-related errors. Refer to your programming pak manual.	Refer to your programming pak manual.
<i>I/O FORM ERR 93</i>	The number of input records did not equal the Record Count (MOS Technology format only).	Check the connection of all units in the system, data format, and data source, and then try again.)
<i>BAD REC TYPE 94</i>	The record type was in error. (Intel-Intellec 8/MDS, Intel MCS-86 and T.I. SDSMAC formats only.)	Check the connection of all units in the system, data format, and data source, and then try again.
<i>ERROR 96*</i>	Illegal center point for RAM shuffle.	Check parameters and re-enter.

*Remote Control only; will not occur during front panel operation, hence no front panel display.

APPENDIX F

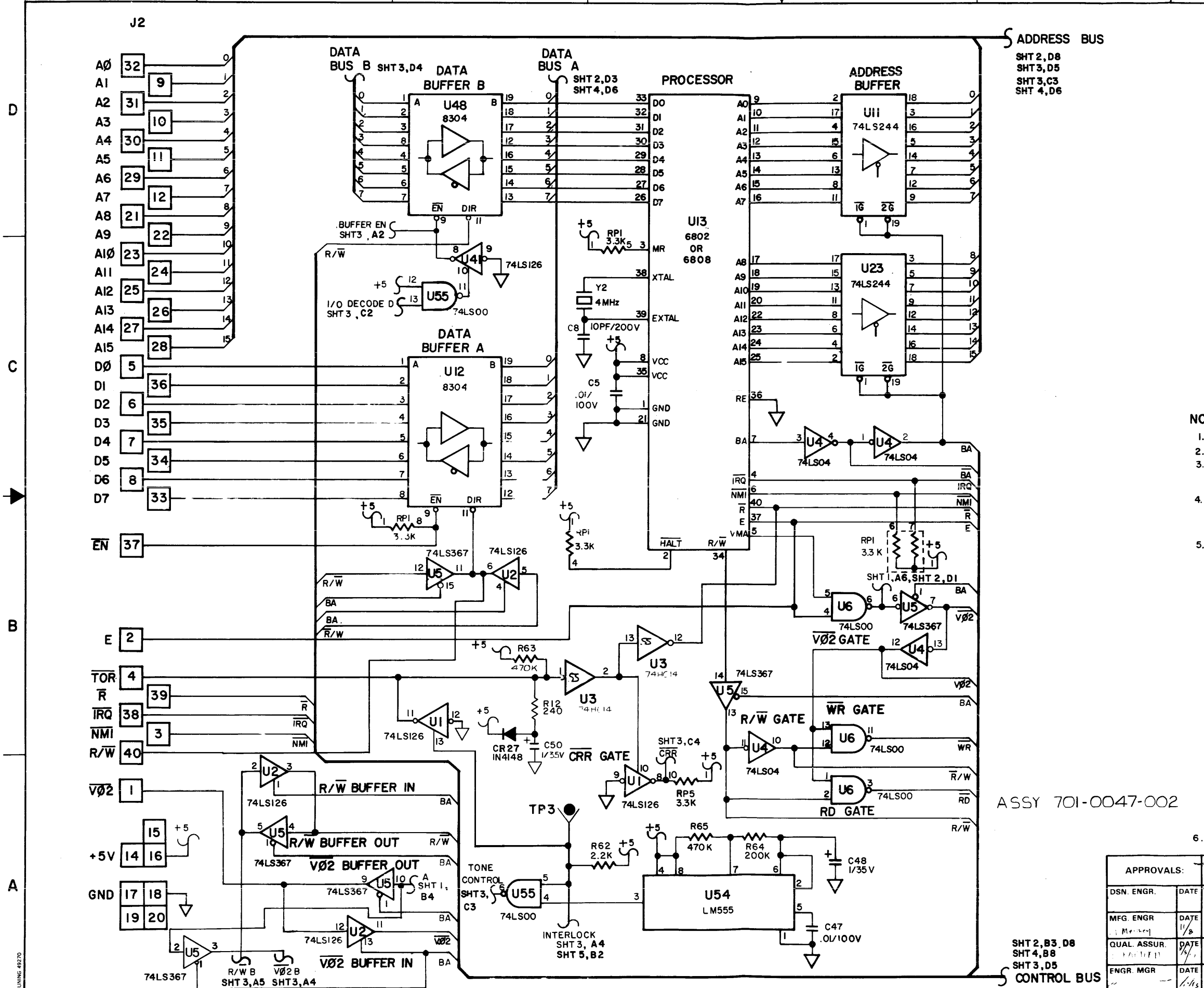
SCHEMATICS

Controller Board	30-701-0047
Front Panel Keyboard	30-702-1648
Front Panel Display Driver	30-702-0061
FIP Display Board	30-702-0060
128K Memory Expansion Board	30-701-1636
Handler Interface Board	30-702-1984 (optional)
Filter Board	30-701-1672

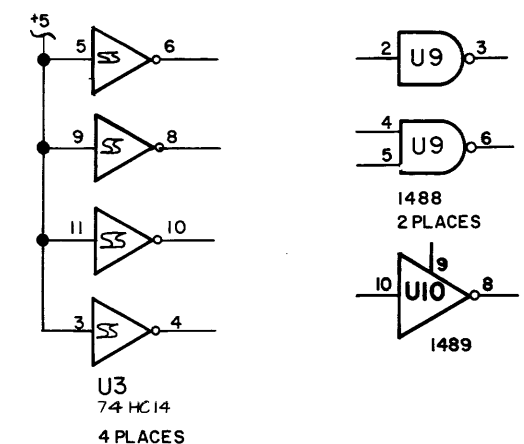
CAUTION

When servicing Data I/O equipment beyond that described in this manual, we recommend that you send your unit to your local service center. If you decide to do the repair be aware that many parts in Data I/O equipment have been qualified by vendor and may not be interchangeable with the same type of part manufactured elsewhere. Refer to the back of this manual for service office locations.

REVISIONS					
LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
A	RELEASE	NL		WT	1/1/83
B	ECP #0006, ADCN #A1	SH		WAS	4/1/84
C	ADCN B1, ECP NO. 0024	NL		WAS	2/1/84



- NOTES: UNLESS OTHERWISE SPECIFIED
1. ALL RESISTORS ARE 1/4W AND IN OHMS, 5%.
 2. ALL CAPACITORS ARE IN MICROFARADS.
 3. LAST REFERENCE DESIGNATOR USED: C54, CR28, J8, JP5, Q26, R69, RP8, TP3, U55, VR4, Y2
 4. REFERENCE DESIGNATORS NOT USED: C7, C16, C33, C45
 5. UNUSED GATES:



ASSY 701-0047-002

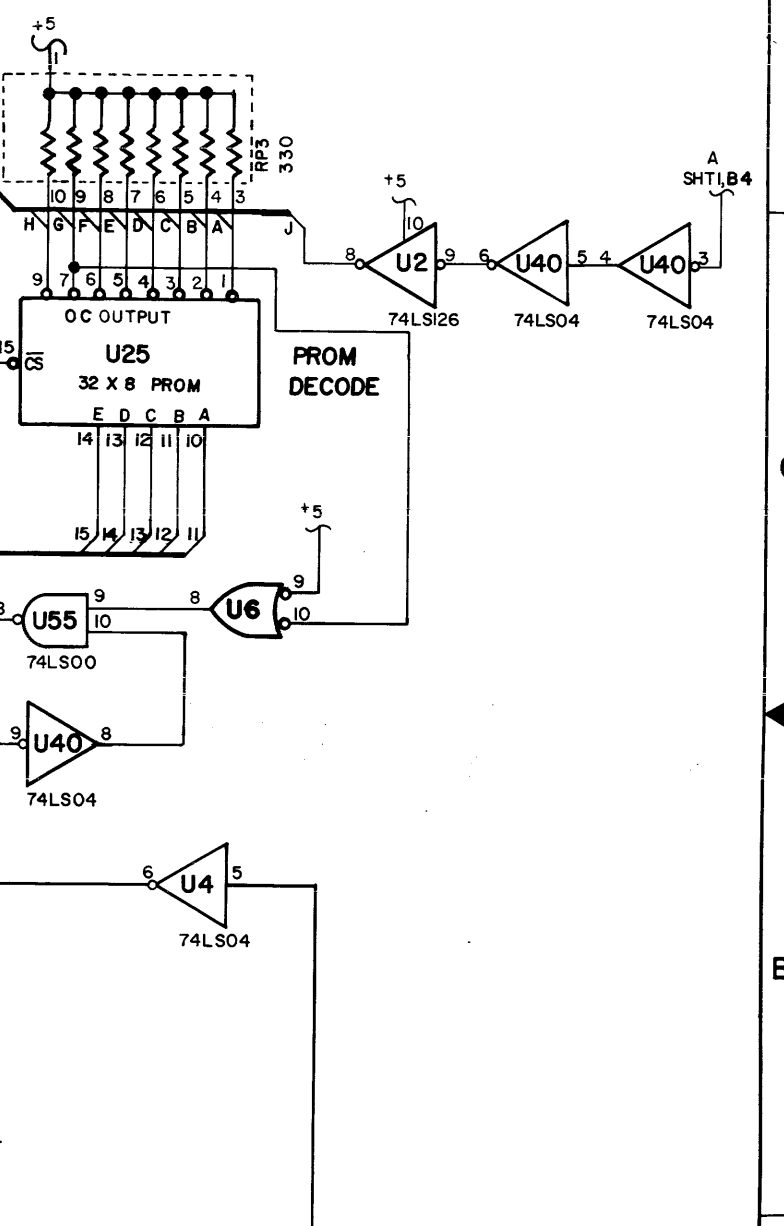
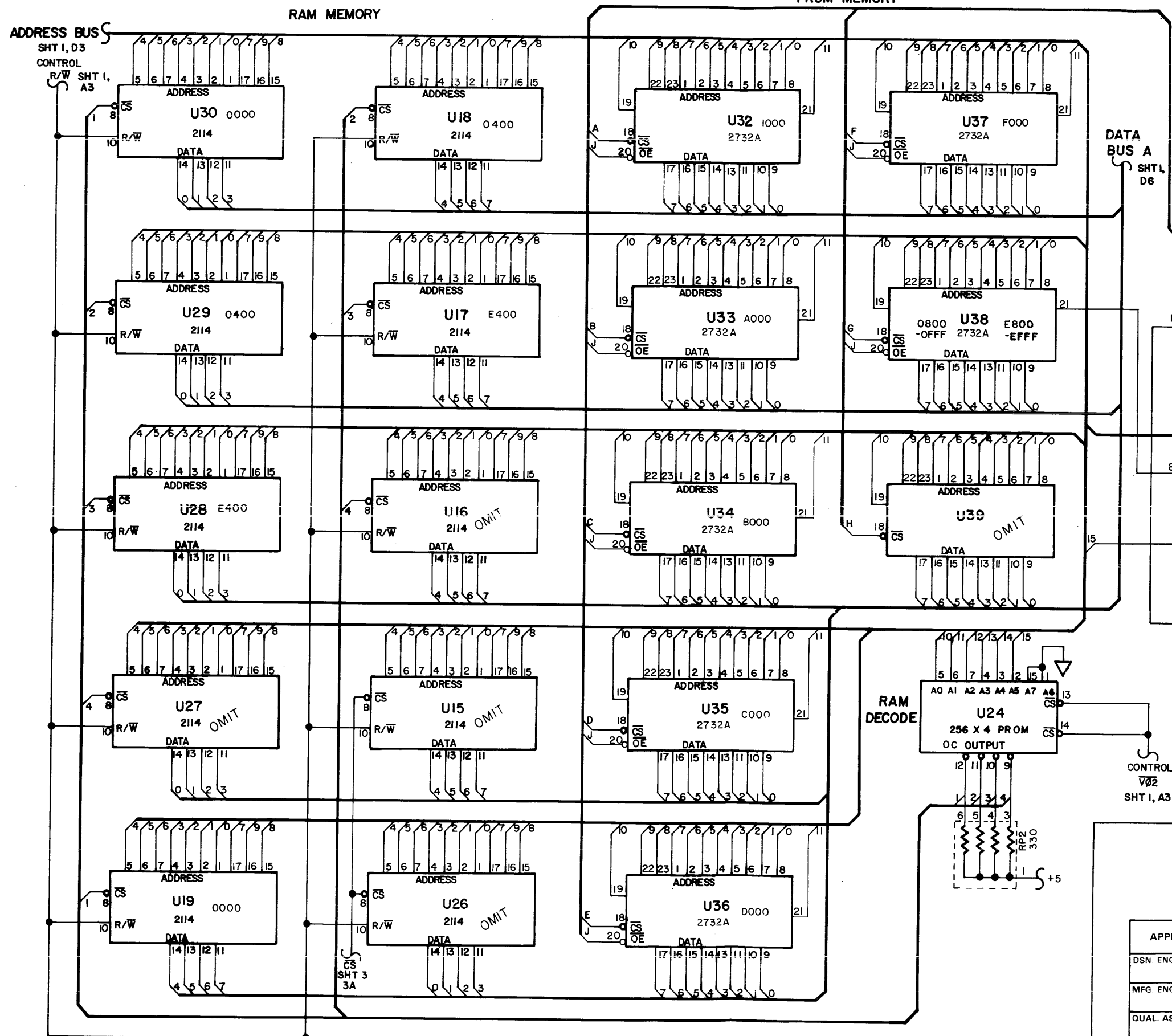
6. POWER SUPPLY CHART ON SHEET 4.

APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		TOLERANCES, UNLESS OTHERWISE SPECIFIED:		TITLE	
DSN. ENGR.	DATE	XX	±	XXX	±	DATA I/O	
MFG. ENGR	DATE	XXX	±	ANGULAR		SCHEMATIC DIAGRAM	
QUAL. ASSUR.	DATE	DRAWN BY:		DATE	SIZE	CODE IDENT.	DRAWING NO.
ENGR. MGR	DATE	CHECKED BY:		DATE	D	54193	30-701-0047
		SCALE				SHEET 1 OF 5	

SHT 2, B3, D8
SHT 4, B8
SHT 3, D5
SHT 4, D6
CONTROL BUS

8 7 6 5 4 3 2

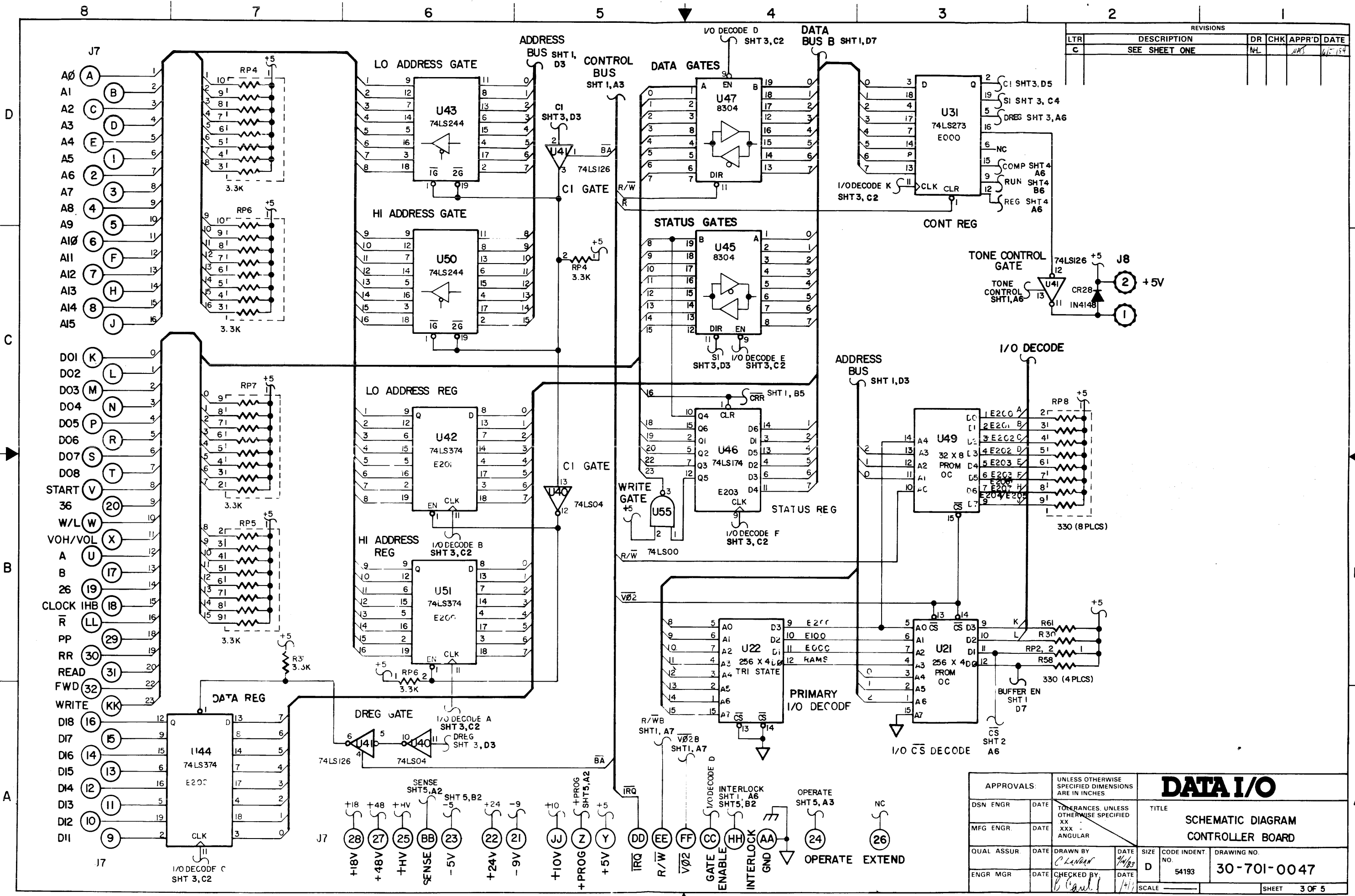
REVISIONS					
LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
C	SEE SHEET ONE	HL			6/1/83



APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES.		DATA I/O	
DSN ENGR.	DATE	TOLERANCES, UNLESS OTHERWISE SPECIFIED: XX ± XXX ± ANGULAR		TITLE SCHEMATIC DIAGRAM CONTROLLER BOARD	
MFG. ENGR.	DATE	DRAWN BY: <i>C. Langan</i>		DATE 4/83	SIZE D
QUAL. ASSUR.	DATE	CHECKED BY: <i>[Signature]</i>		DATE 1/11	CODE INDET. NO. 54193
ENGR. MGR.	DATE				DRAWING NO. 30-701-0047
SCALE				SHEET 2 OF 5	

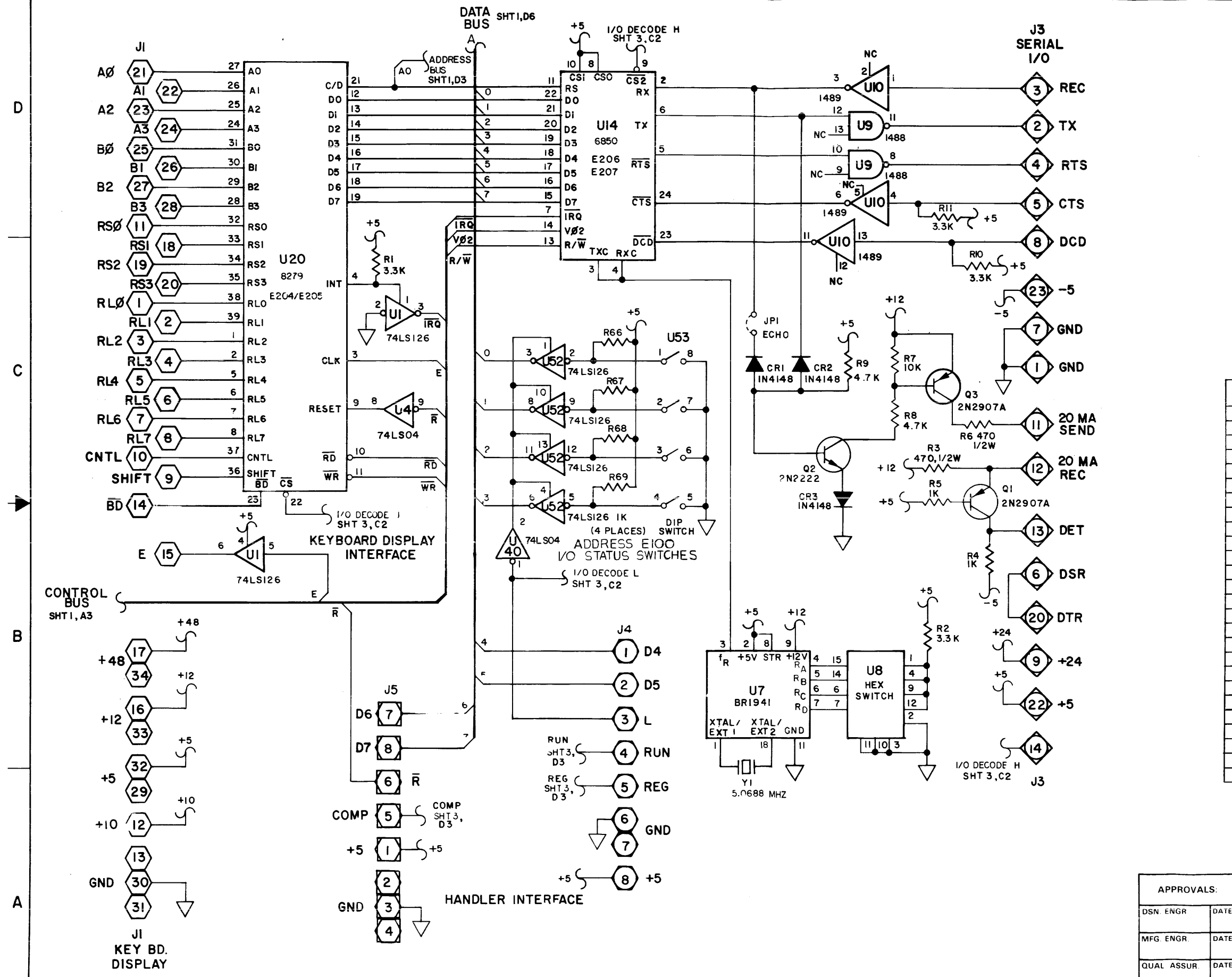
BRUNING 48270

REVISIONS				
LTR	DESCRIPTION	DR	CHK	APPR'D DATE
C	SEE SHEET ONE	ML	JMS	1/15/84



APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		DATA I/O	
DSN ENGR	DATE	TOLERANCES UNLESS OTHERWISE SPECIFIED XX - XXX - ANGULAR		TITLE	
MFG ENGR	DATE			SCHEMATIC DIAGRAM CONTROLLER BOARD	
QUAL ASSUR	DATE	DRAWN BY C. LANGAN	DATE 7/1/83	SIZE D	CODE IDENT. NO. 54193
ENGR MGR	DATE	CHECKED BY D. GARDNER	DATE 1/15/84	DRAWING NO. 30-701-0047	
SCALE				SHEET 3 OF 5	

REVISIONS					
LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
C	SEE SHEET ONE	NL		WAS	6/27/84

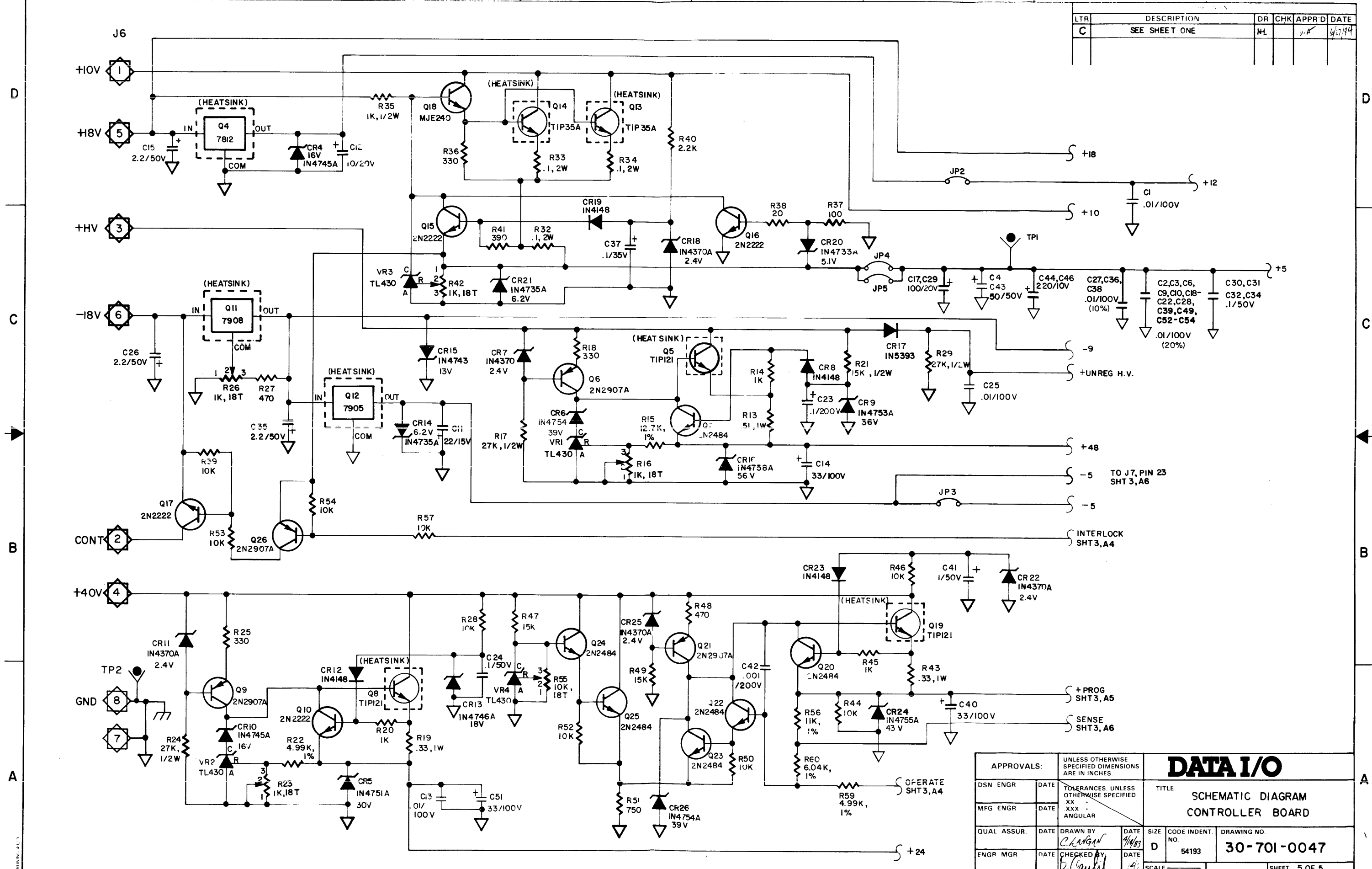


REF DES	GND	+5	+12	-5
U1-U4	7	14		
U5	8	16		
U6	7	14		
U9	7		14	1
U10	7	14		
U11, U12	10	20		
U14	1	12		
U17-U19	9	18		
U20	20	40		
U21, U22	8	16		
U23	10	20		
U24, U25	8	16		
U28-U30	9	18		
U31	10	20		
U32-U38	12	24		
U40, U41	7	14		
U42-U45	10	20		
U46	8	16		
U47, U48	10	20		
U49	8	16		
U50, U51	10	20		
U52	7	14		
U55	7	14		

APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES.		DATA I/O	
DSN ENGR	DATE	TOLERANCES, UNLESS OTHERWISE SPECIFIED .XX : .XXX : ANGULAR			
MFG ENGR	DATE	DRAWN BY <i>C. Longue</i>		DATE <i>1/83</i>	TITLE SCHEMATIC DIAGRAM CONTROLLER BOARD
QUAL ASSUR	DATE	CHECKED BY <i>Paul Antel</i>		DATE <i>1/83</i>	
ENGR MGR	DATE	SIZE D	CODE INDET. NO. 54193	DRAWING NO. 30-701-0047	SCALE NONE SHEET 4 OF 5

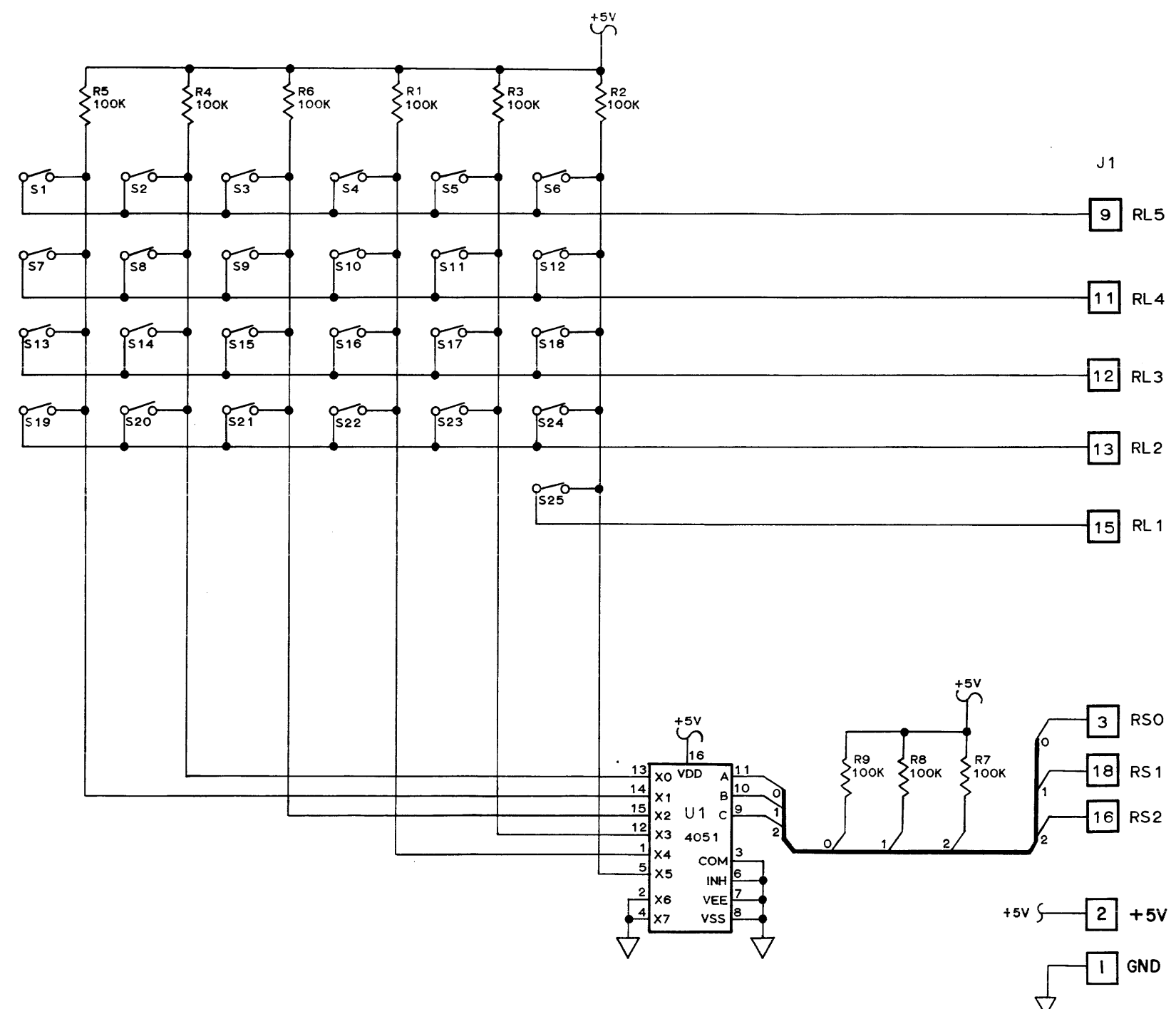
8 7 6 5 4 3 2 1

LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
C	SEE SHEET ONE	NL		V.F.	4/1/94



APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		DATA I/O		
DSN ENGR	DATE	TOLERANCES UNLESS OTHERWISE SPECIFIED		TITLE		
MFG ENGR	DATE	XX XXX - ANGULAR		SCHEMATIC DIAGRAM CONTROLLER BOARD		
QUAL ASSUR.	DATE	DRAWN BY	DATE	SIZE	CODE INCHENT NO	DRAWING NO.
ENGR MGR	DATE	CHECKED BY	DATE	D	54193	30-701-0047
				SCALE	SHEET 5 OF 5	

REVISIONS					
LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
A	RELEASE	NW	MM		7-8-81

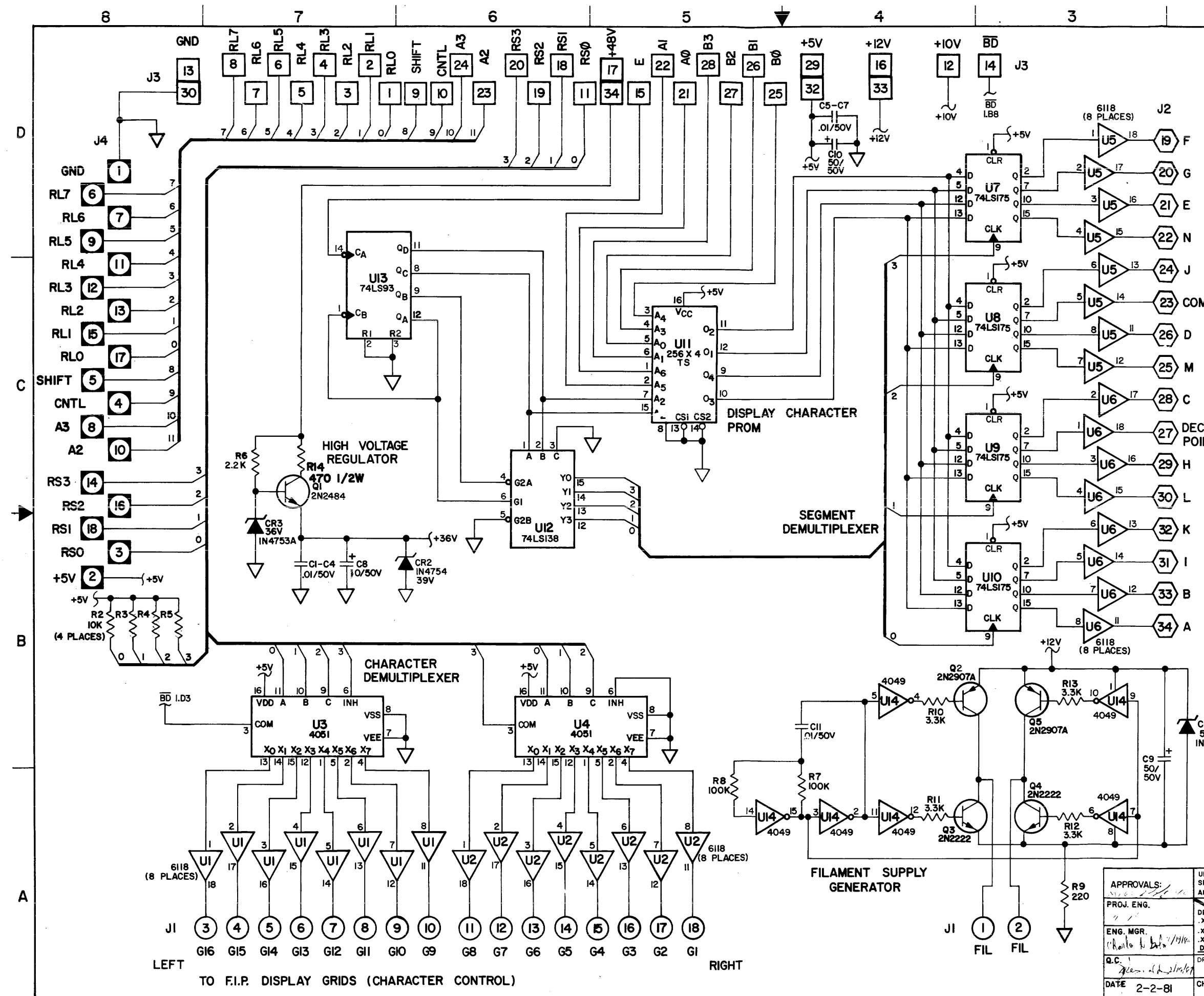


NOTES: UNLESS OTHERWISE SPECIFIED
 1. ALL RESISTORS ARE 1/4W AND IN OHMS, 5%
 2. LAST REFERENCE DESIGNATOR USED: R9, J1, S25, U1
 3. 702-1648-001 SHOWN

APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES.		DATA I/O ISAQUAH, WASH.	
DSN ENGR. <i>[Signature]</i>	DATE 7/8/81	TOLERANCES, UNLESS OTHERWISE SPECIFIED: XX ± XXX ± ANGULAR			
MFG ENGR. <i>[Signature]</i>	DATE 7/8/81	DRAWN BY: <i>[Signature]</i>	DATE 7/8/81	SIZE D	CODE INDENT. NO. 54193
ENGR. MGR. <i>[Signature]</i>	DATE 7/8/81	CHECKED BY: <i>[Signature]</i>	DATE 7/8/81	SCALE	DRAWING NO. 30-702-1648
					SHEET 1 OF 1

BRUNING 49270

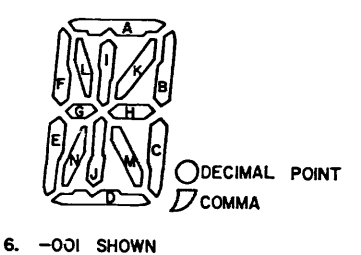
REVISIONS					
ZONE	LTR	DESCRIPTION	CM	PE	DATE
A		RELEASE			2-2-81
B		ECN 4096			4-14-81
C		ECN 4135			5-8-81
D		ECN 4304			8-3-81



TO F.I.P. DISPLAY ANODES (CHARACTER SEGMENTS)

- NOTES: UNLESS OTHERWISE SPECIFIED
1. ALL RESISTORS ARE 1/4W AND IN OHMS, 5%
 2. ALL CAPACITORS ARE IN MICROFARADS.
 3. LAST REFERENCE DESIGNATOR USED:

	NOT SHOWN		
	+5V	+36V	GND
U1, 2, 5, 6	10		9
U7-U10	16		8
U12	16		8
U13	5		10



APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES	DATA I/O ISSAQUAH WASH		
PROJ. ENG.					
ENG. MGR.		TOLERANCES	TITLE SCHEMATIC DIAGRAM FIP DISPLAY DRIVER		
		DECIMALS ANGULAR			
Q.C.		.X ±	SIZE	CODE IDENT NO	DRAWING NO
DATE	2-2-81	.XX ±	D	54193	30-702-0061
		.XXX ±	SCALE	NONE	SHEET 1 OF 1
		DO NOT SCALE DRAWING			

8

7

6

5

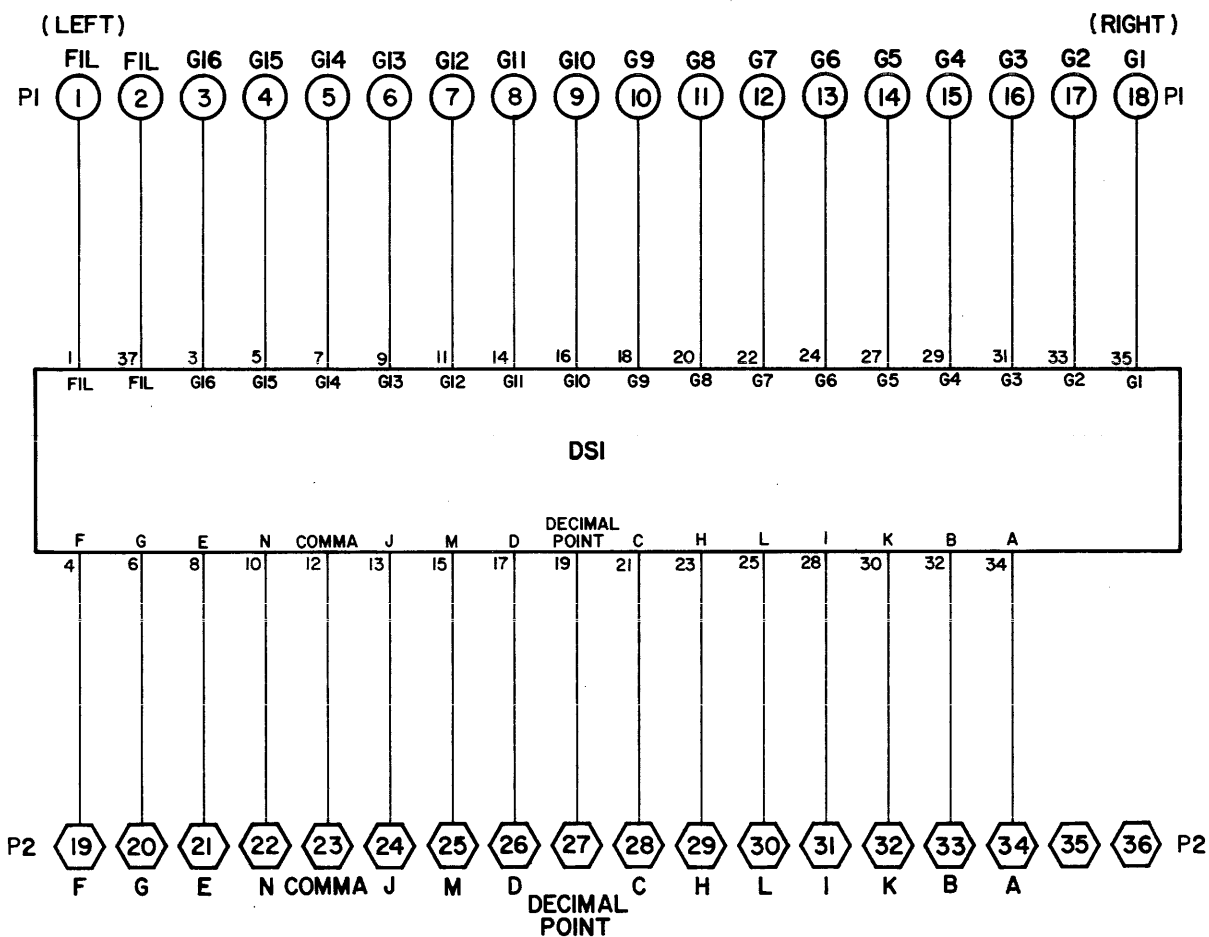
4

3

2

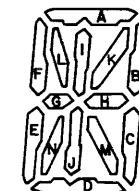
1

REVISIONS					
LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
A	RELEASE				2-81



NOTES:

1. DISPLAY SEGMENTS 311-2000

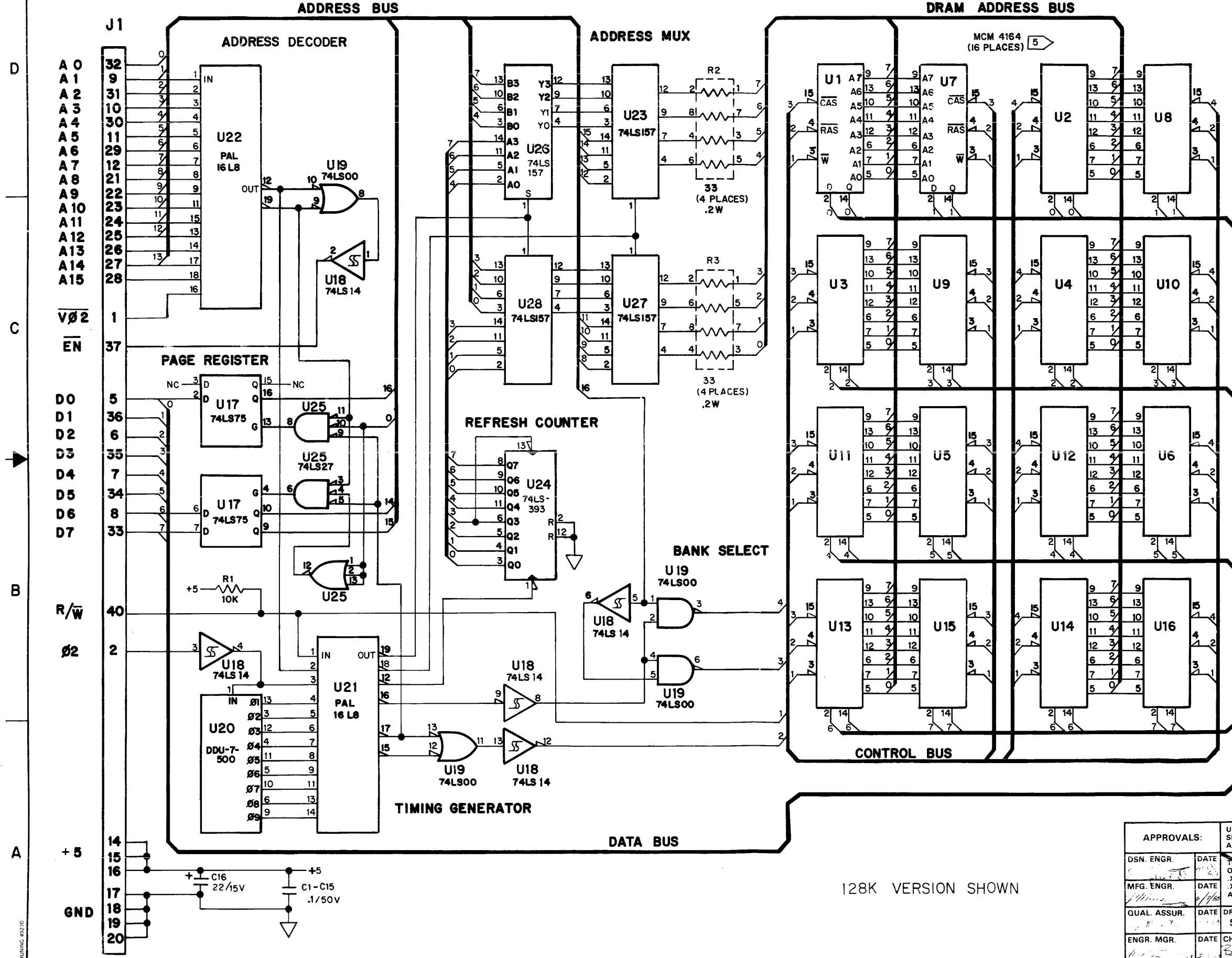


2 -001 SHOWN

APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES.		DATA I/O ISSAQUAH, WASH.	
DSN. ENGR.	DATE	TOLERANCES, UNLESS OTHERWISE SPECIFIED:		TITLE	
MFG. ENGR.	DATE	.XX ±		SCHEMATIC DIAGRAM	
QUAL. ASSUR.	DATE	.XXX ±		FIP DISPLAY	
ENGR. MGR.	DATE	ANGULAR		DRAWING NO.	
DRAWN BY:		DATE	SIZE	CODE INDENT. NO.	DRAWING NO.
B. Valdivia		21/81	D	54193	30-702-0060
CHECKED BY:		DATE	SCALE	SHEET 1 OF 1	

8 7 6 5 4 3 2 1

REVISIONS				
LTR	DESCRIPTION	DR	CHK	APPR'D DATE
A	RELEASE	SH		
B	ECN 4796	AK		
C	ECN 4954	SH		
D	ECN 5034	SH		



- NOTES: UNLESS OTHERWISE SPECIFIED:**
- ALL RESISTORS ARE 1/4W AND IN OHMS, 5%.
 - ALL CAPACITORS ARE IN MICROFARADS.
 - LAST REFERENCE DESIGNATOR USED: C16, J1, R3, U28.
 - POWER AND GROUND:

REF. DES.	+5	GND
U1 - U16	8	16
U18 - U20, U25	14	7
U17	5	12
U21, 22	20	10
U23, U26 - U28	16	8, 15
U24	14	7

5. PIN 1 ON ALL MCM 4164'S NOT CONNECTED.
- 6 UNUSED GATES;
-

128K VERSION SHOWN

APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES.		DATA I/O	
DSN. ENGR.	DATE	TOLERANCES, UNLESS OTHERWISE SPECIFIED:	TITLE		
MFG. ENGR.	DATE	.XX ±	SCHEMATIC DIAGRAM,		
QUAL. ASSUR.	DATE	.XXX ±	128K EXPANSION RAM BOARD		
ENGR. MGR.	DATE	ANGULAR	DRAWN BY:	DATE	DRAWING NO.
			S. HUMBLE	11/82	30-701-1636
			CHECKED BY:	DATE	CODE INCHENT. NO.
			Ben Cantel	11/83	54193
					SCALE NONE
					SHEET 1 OF 1

DRAWING 80276

8

7

6

5

4

3

2

1

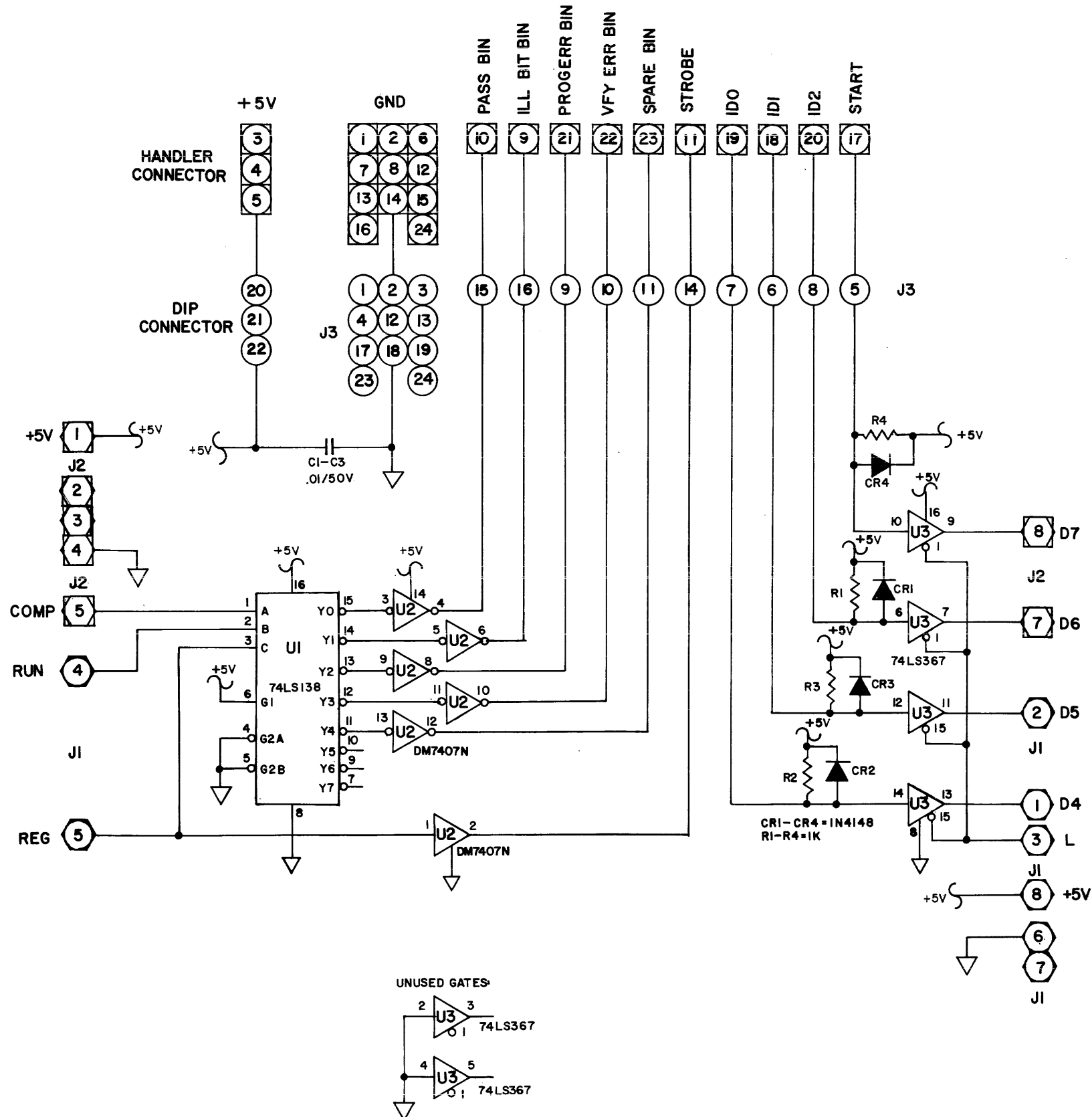
REVISIONS					
LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
A	RELEASE	S.H.	MB	K.S.	4-13-81

D

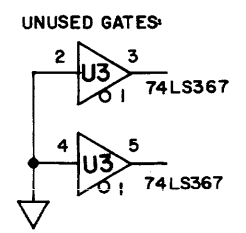
C

B

A



- NOTES: UNLESS OTHERWISE SPECIFIED
1. ALL RESISTORS ARE 1/4W AND IN OHMS, 5%.
 2. ALL CAPACITORS ARE IN MICROFARADS.
 3. LAST REFERENCE DESIGNATOR USED:
J3, U3, CR4, R4, C3
 4. -001 SHOWN



APPROVALS:		UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES.		TITLE	
DSN. ENGR. <i>Paul S. ...</i>	DATE 4/1/81	TOLERANCES, UNLESS OTHERWISE SPECIFIED: .XX ± .XXX ± ANGULAR		SCHEMATIC DIAGRAM HANDLER INTERFACE	
MFG. ENGR. <i>H. J. ...</i>	DATE 4/1/81	DRAWN BY: SUENEE H.	DATE 4/1/81	SIZE D	CODE INVENT. NO. 64193
ENGR. MGR. <i>Charles H. ...</i>	DATE 4/1/81	CHECKED BY: MATHIE ...	DATE 4-81	DRAWING NO. 30-702-1984	
SCALE NONE				SHEET 1 OF 1	

BRUNING 49270

4

3

2

1

D

D

C

C

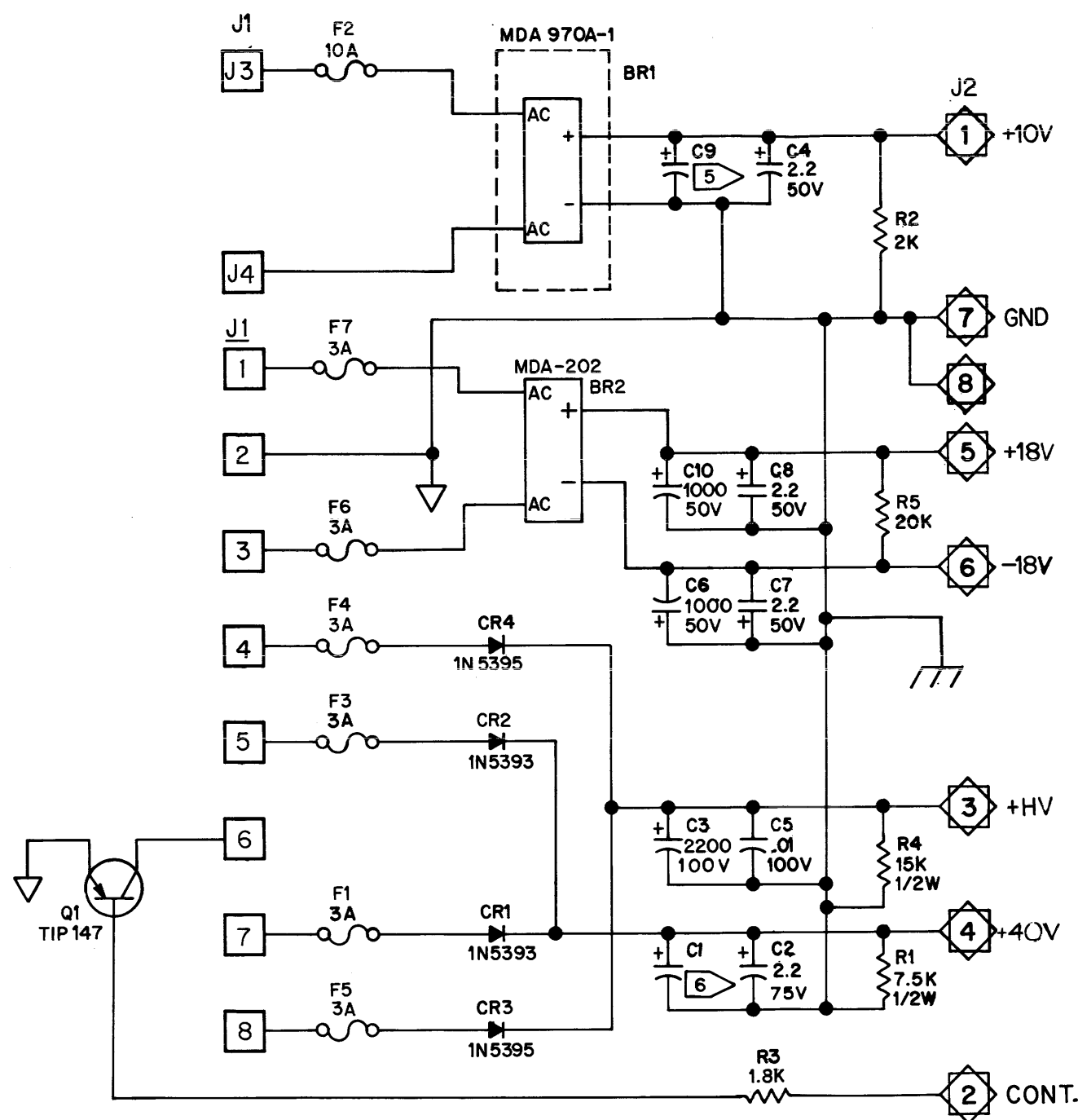
B

B

A

A

REVISIONS				
LTR	DESCRIPTION	CK	PE	APP DATE
A	RELEASE	6/23		
B	ADCN A1, ECP NO. 0001	6/24	WT	
C	ADCN B1, ECP NO. 0024	6/24	WT	6/24



NOTES: UNLESS OTHERWISE SPECIFIED

- 1. ALL RESISTORS 1/4W AND IN OHMS, ±5%.
- 2. ALL CAPACITORS ARE IN MICROFARADS.
- 3. LAST REF. DES. USED:

J2 CR4
BR2 C10
F7 R5

- 5 C9A: 15000/20V (-009,-010).
C9B:(QTY 2) 6800/25V IN PARALLEL(-011,-012).
- 6 C1A: 4700/80V (-010,-012).
C1B: 5200/75V (-009-011).

THIS DOCUMENT CONTAINS INFORMATION CONSIDERED PROPRIETARY, AND SHALL NOT BE REPRODUCED WHOLLY OR IN PART, NOR DISCLOSED TO OTHERS WITHOUT THE SPECIFIC WRITTEN PERMISSION OF DATA I/O CORP.

ASSY 701-1672-009

TOLERANCES (EXCEPT AS NOTED)		DATA I/O	
DECIMAL ±	ANGULAR ±	TITLE SCHEMATIC DIAGRAM, FILTER BOARD	
		DRAWN BY: JUNE 21, 83 BARRY VALDIVIA	
APPROVED BY:	SIZE C	CODE IDENT. NO. 54193	DRAWING NO. 30-701-1672
	SCALE NONE	SHEET 1 OF 1	