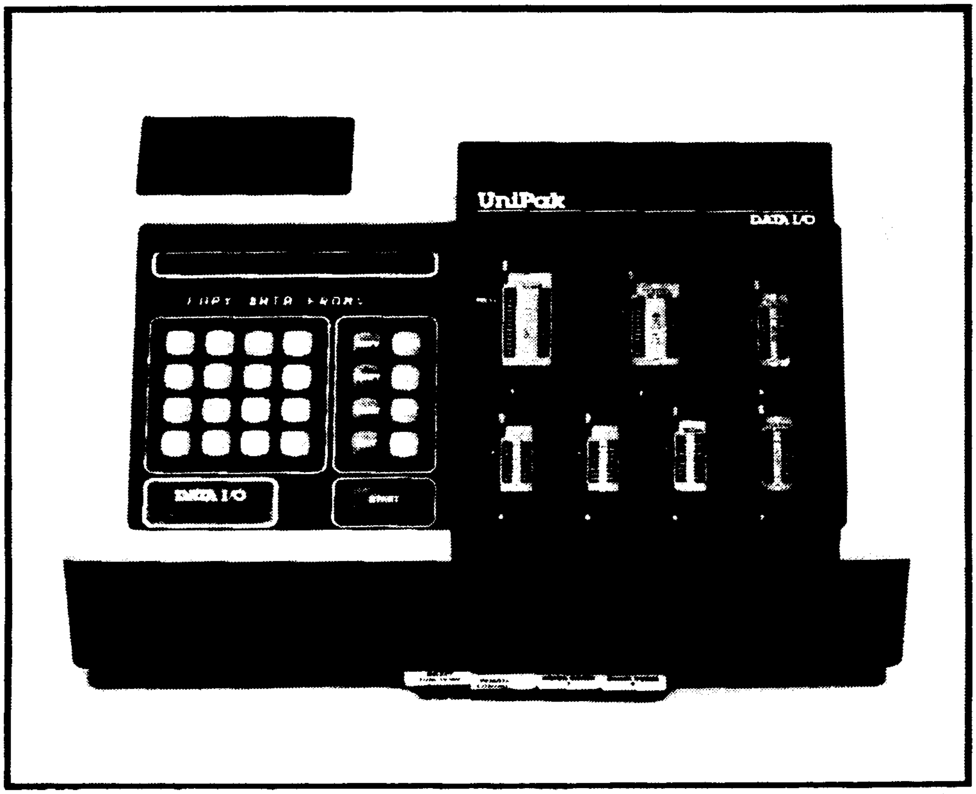


29A UNIVERSAL PROGRAMMER 990-0029

10-990-0029-001 REV E MAY 82

TABLE OF CONTENTS

SECTION 7. CIRCUIT DESCRIPTION

LIST OF FIGURES

LIST OF TABLES

SECTION 1
INTRODUCTION

1.1 GENERAL INFORMATION

The Data I/O 29A Universal Programmer reliably programs semiconductor devices with data received from any one of a number of sources or a combination of sources.

The 29A is compatible with a large line of Data I/O programming modules which allow it to program most commercially available programmable memory and logic devices. Data I/O regularly publishes an up-to-date *Comparison Chart of Programmable Devices,* a ready reference for users to help them keep abreast of programmable devices. It lists the available devices by manufacturer and the Data I/O programming modules used to program them.

Standard features of the 29A include a 16-character alphanumeric display, RS232C serial port, and an 8K x 8 data RAM. A 16K x 8 model is available or an option kit can be used to expand the data RAM.

In addition to the data translation formats available in previous Data I/O programmers, four new formats have been added to allow communication between the 29A and development systems manufactured by Texas Instruments, Motorola, Intel, and Hewlett-Packard. Remote control capability is standard with a streamlined command protocol. An optional computer remote control package, compatible with that used in our System 19s and 17s, is also available.

1.2 SPECIFICATIONS

The following paragraphs describe the major assemblies and power requirements of the 29A. Table 1-1 gives the physical and environmental specifications and Table 1-2 lists the data translation formats available.

1.2.1 MAJOR ASSEMBLIES

The unit is microprocessor-controlled with a 8K x 8-bit data RAM (16K x 8 also available) and RS232C and 20 mA current loop serial input/output (I/O) port. Parallel data transmission and expanded serial transmission are possible when the programmer is connected to the Data I/O Port Multiplier.

Interchangeable programming modules, which are purchased separately, contain all interface electronics and the appropriate control software to program devices. Programming voltages are current-limited and regulated to meet the device manufacturers' specification.

1.2.2 POWER REQUIREMENTS

The following are the 29A's power requirements. Section 2.3 details power connection and inspection.

Operating Voltages. 100, 120, 220, or 240 VAC, within +5% or −10%, grounded, single phase.

Frequency Range. 48 to 62 Hz.

Power Consumption. 35 W nominal.

Circuit Protection. A circuit breaker is an integral part of the POWER ON/OFF switch located on the back of the unit. The power supply is separately fused.

1.2.3 PHYSICAL AND ENVIRONMENTAL SPECIFICATIONS

Table 1-1 lists the physical and environmental specifications of the 29A.

Table 1-1. Physical and Environmental Specifications

Dimensions	38.1 cm x 15.2 cm x 27.3 cm (15" x 6" x 10.8")
Weight	6.4 kg (14.1 lb.)
Operating Temperature Range	5° to 45° C (41° to 113° F)
Storage Temperature Range	−40° to 70° C (−40° to 158° F)
Humidity	Up to 95%, noncondensing.

1.2.4 DATA TRANSLATION FORMATS

Table 1-2 lists the data translation formats available in the 29A.

Table 1-2. Data Translation Formats

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

1.3 PROGRAMMER CONFIGURATION

The standard programmer includes serial I/O capability using over 25 data translation formats and remote control. Table 1-3 lists the hardware options that may be added to or used with the 29A.

Table 1-3. Hardware Options

OPTION	DESCRIPTION	DATA I/O PART NUMBER
16K x 8 RAM Expansion	Enlarges data RAM to 16K x 8.	Contact your Data I/O service representative
Universal Calibrator	Sets up loads to be measured in calibration. Required for calibration of standard programming modules only (see Section 4); used with the Calibration Extender.	910-1071
Calibration Extender	A mechanical assembly required for calibration of all programming modules on the programmer.	910-1521

Many accessories and programming modules are also available for use with your programmer. Contact your Data I/O representative for more information.

1.4 FIELD APPLICATIONS SUPPORT

Data I/O has Field Applications Engineers (FAEs) throughout the world who can provide you with additional information about interfacing Data I/O products with other equipment or answer questions you may have about your equipment.

The locations of the FAEs within the United States are given on the address list at the back of this manual. Call your regional FAE if you have any questions or problems. For international applications support, contact your nearest Data I/O representative.

1.5 CALIBRATION

The need for calibration varies with the amount of use. Generally we suggest calibration whenever programming yields fall below the device manufacturers' recommended minimums or as part of a troubleshooting procedure. Section 4 of this manual describes calibration in detail.

1.6 WARRANTY

Data I/O equipment is warranted against defects in materials and workmanship. The warranty period of 1 year begins when you receive the equipment.

The warranty card inside the back cover 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 Data I/O Service Centers is included inside the back cover of this manual.

1.8 ORDERING

To place an order for equipment, contact your Data I/O sales representative.

Orders for shipment should contain the following information:

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

SECTION 2
INSTALLATION

2.1 INTRODUCTION

The following sections present information necessary for setting up the 29A Universal Programmer. Included are the connection to a power source, power and fuse requirements, and connecting a Data I/O Data Control Unit (DCU) or other peripheral.

2.2 INSPECTION

The 29A was thoroughly tested and inspected before shipment. For trouble-free operation, verify upon receipt that the equipment is in the best possible condition. The unit was carefully packaged before shipment to prevent damage and should arrive free of any defects and in perfect operating condition. All the equipment listed in Table 2-1 should be present. Carefully inspect it for damage that may have occurred during transit. If you note any damage, file a claim with the carrier and notify Data I/O. Check operation only after completing the installation procedures in sections 2.3 through 2.5.

Table 2-1. Required Equipment

DESCRIPTION	DATA I/O PART NUMBER	QTY.
29A Universal Programmer	990-0029-XXX*	1
Power Cord (USA)	401-1577	1
or		
Power Cord (Europe)	416-0010	1
Serial Port Mating Connector	401-3064	1
Serial Port Connector Hood	401-3069	1
Instruction Manual	10-990-0029-001	1
Operator's Guide	12-990-0029	1

* XXX will vary with RAM size and operating voltage

2.3 POWER CONNECTION

The following paragraphs explain the procedures for connecting the 29A to a power source.

2.3.1 CHECKING THE OPERATING VOLTAGE

CAUTION

Do not operate the unit at voltages outside the selected range or you will damage the unit.

The operating voltage of the unit has been factory set according to the customer's specification. The unit will operate when the line voltage is within +5% or −10% of that marked on the sticker attached to the rear panel.

2.3.2 GROUNDING THE UNIT

WARNING

Failure to ground the programmer may create a shock hazard. Do not defeat the 3-wire power cord ground by using extension cords or adapters. Read the following paragraphs carefully.

The power cord contains 3 conductors, color coded as shown in Table 2-2. When the cord is connected to a 3-wire AC power system, the ground connector grounds the programmer's chassis, eliminating shock hazards. Do not use anything, such as a 2-conductor extension cord, that would break contact between the unit and an earth ground.

Table 2-2. Conductor Colors by Country/Continent

	U.S.A.	EUROPE	U.K.	JAPAN
Line	Black	Blue	Brown	Red
Neutral	White	Black	Blue	White
Ground	Green & Yellow	Green & Yellow	Green & Yellow	Black

2.3.3 REMOVING AND INSTALLING THE PROTECTIVE SHIELD

A protective shield guards the interior of the programmer from dust and damage and protects the operator from shock. To remove it, first remove the programming module (see paragraph 2.4.1), then pull up on the two snap-lock connectors. 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. For installation, reverse the procedure.

2.3.4 FUSE ACCESS

There are no user-accessible fuses. An in-line circuit breaker acts as the power on/off switch. Other fuse protection is on the Filter Board. If any of the fuses on the Filter Board burn out, service will be necessary. Contact your Data I/O Service Center.

2.4 PROGRAMMING MODULE INSTALLATION

Install programming modules and socket adapters in the programmer according to paragraphs 2.4.1 and 2.4.2.

2.4.1 THE PROGRAMMING MODULE

A programming module can be installed when the programmer power is either on or off. This feature allows RAM data to remain intact during module changes. Check the Data I/O *Comparison Chart* to select the appropriate module.

To install the module, slide it into the opening, hooking the flange of the module under the back edge of the programmer opening and lower it into position. Press gently on the front edge to ensure a good connection. The audible alarm will stop when good contact is made. Figure 2-1 illustrates this procedure. To remove the module, reverse the installation procedure.

2.4.2 THE SOCKET ADAPTER

If the programming module requires a separate socket adapter, install it in the socket receptacle on the front panel of the programming module as shown in Figure 2-1. See the *Comparison Chart* to select the appropriate socket adapter.

2.5 SERIAL INTERFACE

An RS232C and 20 mA current loop serial interface is used to connect the 29A to computer systems and other peripherals.

2.5.1 CABLING

To connect the 29A to other instruments, you must use the RS232C or 20 mA current loop specifications given in Table 2-3. Figure 2-2 shows sample interconnections in the serial interface. When using the standard terminal remote control, use a half duplex 3-wire hook-up without handshaking.

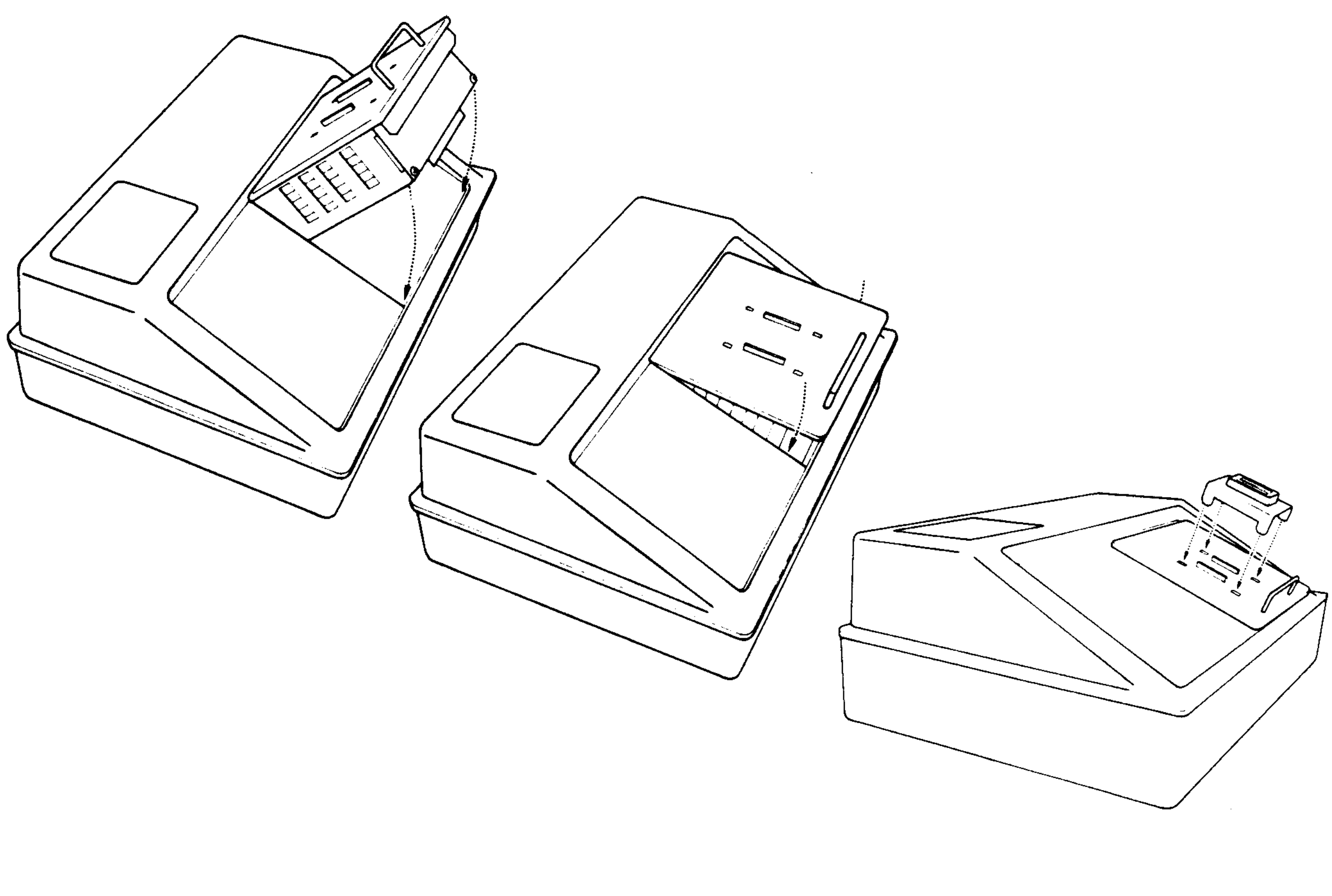
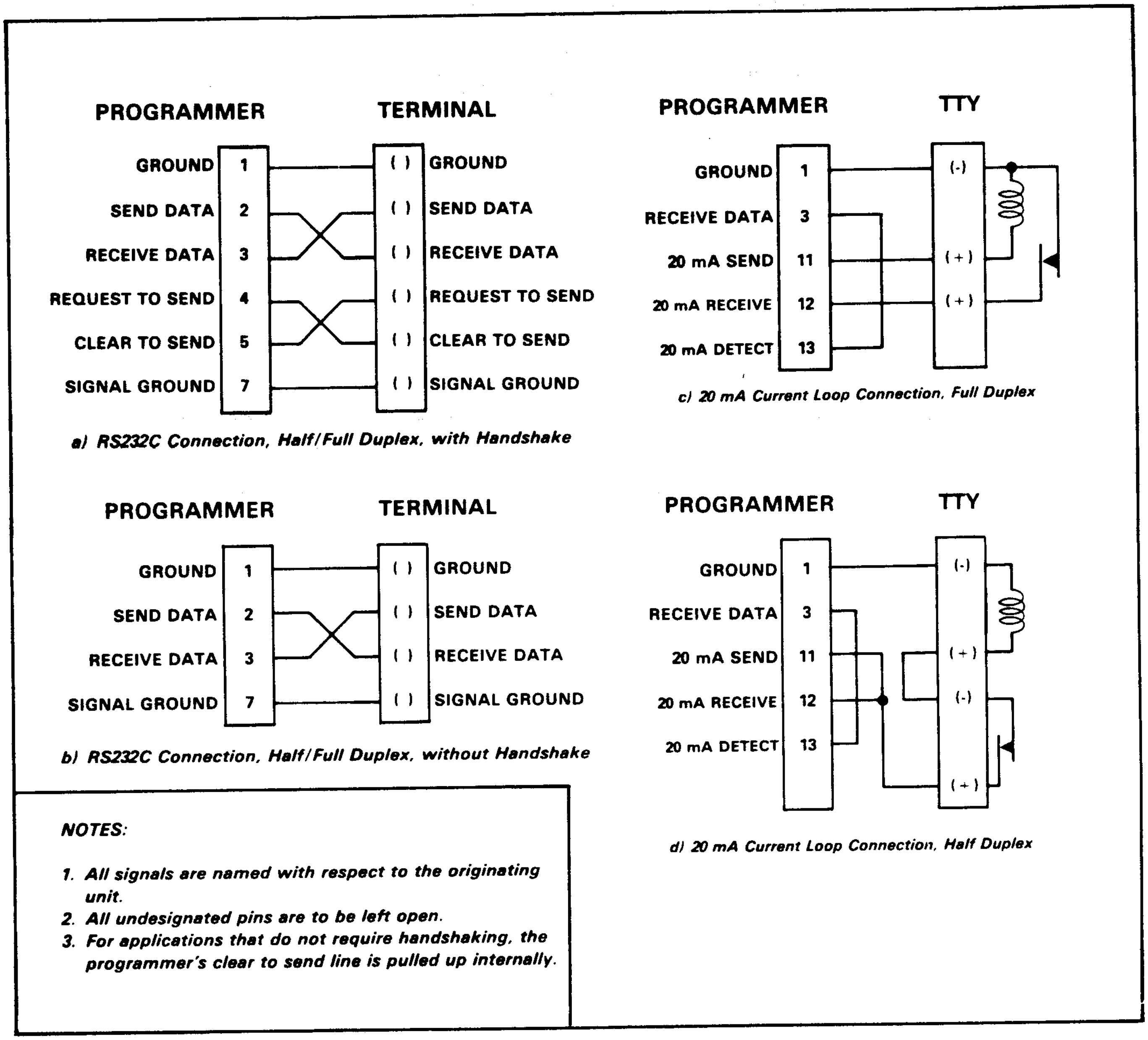


Figure 2-1. Programming Module and Socket Adapter Installation

Table 2-3. Serial Interface Connector Pin Assignment

PIN NO.	SIGNAL MNEMONIC	DESCRIPTION
1	Ground	In the RS232C environment this line is common for the −12V and provides a safety ground connection to the RS232C-compatible terminal. In the TTY environment, this −12 VDC signal line provides the signal return for a TTY terminal.
2	Send Data	Transmits data within RS232C voltage levels (+12 V and −5 V).
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 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*	This line is positive when modem detects a carrier signal. This 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 on 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.

* Pins 5 and 8 have internal pull-ups and need no connection if unused.



a) RS232C Connection, Half/Full Duplex, with Handshake

b) RS232C Connection, Half/Full Duplex, without Handshake

c) 20 mA Current Loop Connection, Full Duplex

d) 20 mA Current Loop Connection, Half Duplex

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.***

Figure 2-2. Sample Interconnection Methods

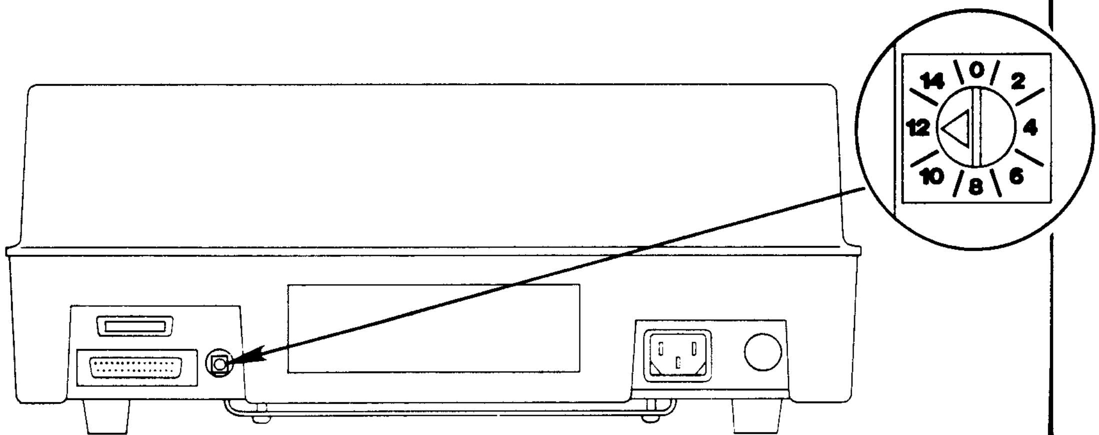
2.5.2 SETTING PARAMETERS

Before the 29A can operate with another system, three parameters must be set—parity, stop bits, and the baud rate.

Whenever the 29A is interfaced with other instruments, the parity, number of stop bits, and baud rate MUST be the same throughout the system.

To set the baud rate, turn the rotary switch, shown in Figure 2-3, to the correct position through the hole in the rear panel. It is easily adjusted with a small screwdriver.

To set parity and stop bits, use the U53 Status Switch assembly on the Controller Board. This switch assembly can be accessed through the cut-out in the protective shield after the programming module has been removed. Figure 2-4 shows the location of the Status Switch assembly and possible settings.



POSITION	BAUD RATE
0	50
1	75
2	110
3	134.5
4	150
5	300
6	600
7	1200
8	1800
9	2000
10 or A	2400
11 or B	3600
12 or C	4800
13 or D	7200
14 or E	9600
15 or F	19,200

Figure 2-3. Baud Rate Switch

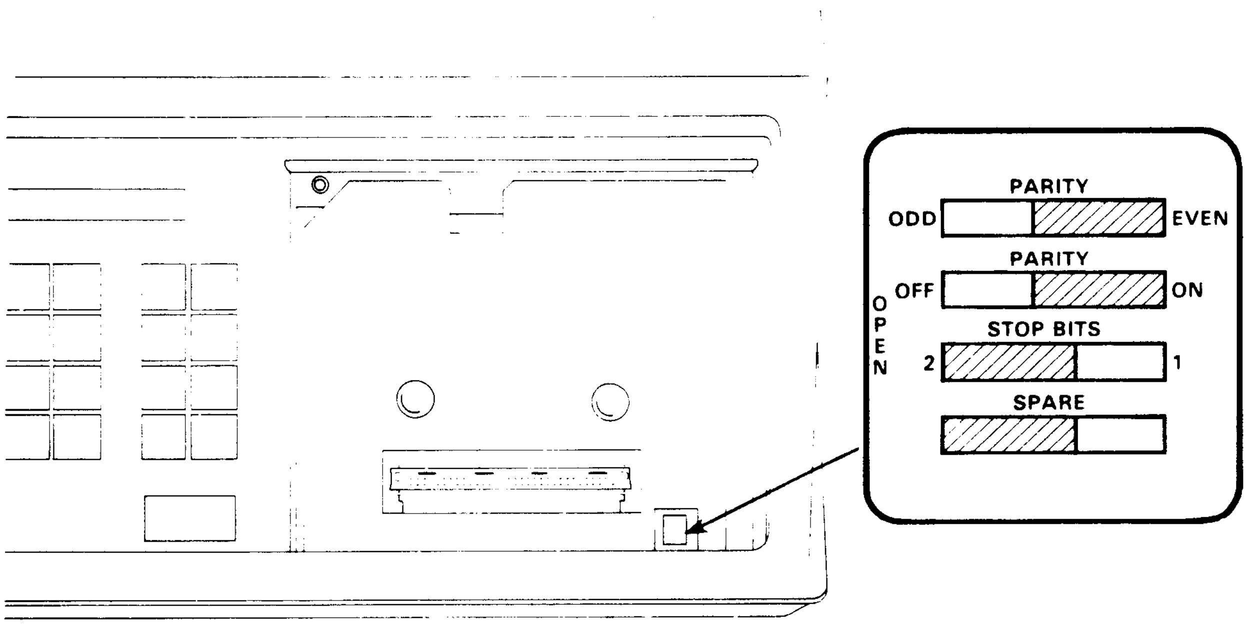


Figure 2-4. Status Switch Settings

2.6 REPACKING FOR SHIPMENT

If a problem arises and you must ship the instrument to Data I/O for service or repair, attach a tag to it describing the work required and identifying the owner. In correspondence, identify the unit by the serial number, model number and name.

If you use the original shipping container, place the instrument in the container with the appropriate packing material and seal the container well with strong tape. If you use some other container, be certain that it is a heavy carton. Wrap the instrument with heavy paper or plastic. Use appropriate packing material and seal the carton with strong tape. Mark the container "DELICATE INSTRUMENT" or "FRAGILE."

SECTION 3
OPERATION

3.1 GENERAL DESCRIPTION

All data transfers or verifications occur between the programmer's internal RAM and the serial port or the device in the programming module. The following defines the operations performed:

- Moving data from a source (device, RAM, or port) to a destination (device, RAM, or port) is a **Copy**.
- Comparing data between a source and a destination is a **Verify** operation.
- Changing data at single addresses within the programmer RAM is an **Edit** operation.
- RAM data can be manipulated and parameters set using **Select Function.**

3.1.1 CONTROL AND INDICATORS

Figure 3-1 and Table 3-1 show and describe the controls and indicators of the 29A.

3.1.2 PRELIMINARY INFORMATION ON USING THE OPERATION PROCEDURES IN THIS MANUAL

Some basic assumptions are made in the operating procedures in the manual. Because of the many

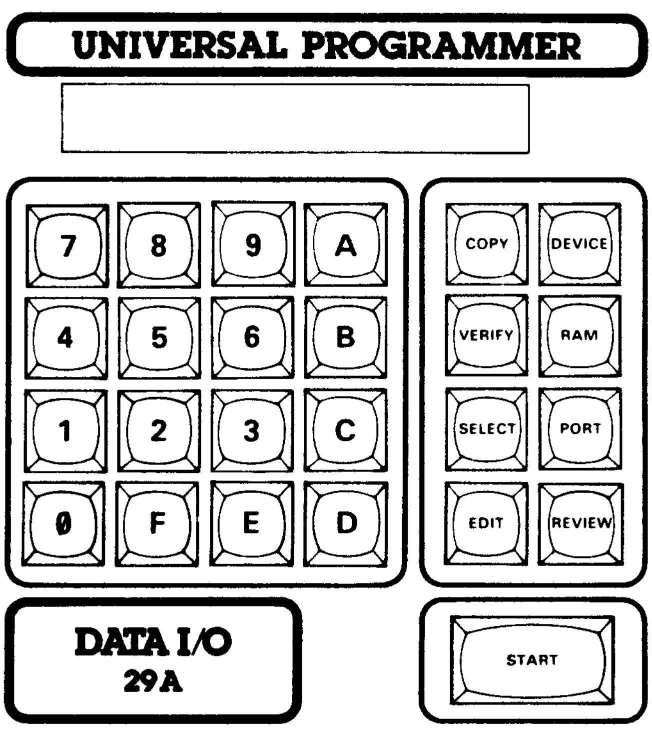


Figure 3-1. Front Panel

Table 3-1. Controls and Indicators

CONTROL/ INDICATOR	FUNCTION
Power ON/OFF Switch	Located on the rear panel; applies or removes AC power.
Hex Keyboard	Sends hex digits to memory and display.
Display	Displays the current status of the programmer.
START Key	Commands the programmer to execute the operations selected and sends entered hex values and instructions to memory.
REVIEW Key	Gives the programmer backwards "stepping" capability through addresses and calibration steps and provides a delete function when entering parameters.
DEVICE RAM PORT	Source/destination keys: see section 3.2 and 3.4.
COPY VERIFY	Function keys: see sections 3.2 and 3.4.
SELECT Key	Prepares the programmer to accept codes for Select Functions. See sections 3.2 and 3.5.
EDIT	Allows access to the programmer RAM data. See sections 3.2 and 3.4.

programming modules available for use with the 29A, the procedures given are standardized. You should consult the appropriate programming module manual for any variations on the procedures given here.

The operating procedures are described in sections 3.4 and 3.5. They are divided according to whether they are accomplished directly from the programmer keyboard or only via the SELECT key. Section 3.6 describes how these operations are accomplished when in Remote Control.

The procedures assume that no errors occur. The programmer signals errors with 3 consecutive beeps (except in Remote Control). It will also beep once when an incorrect key is pressed. When an error occurs, the display will show the error message and a numeric code. These are described and corrective measures given in section 3.7.

The programmer is informed of the device word length and width by the socket adapter (when using a standard programming module), or, with an expanded memory programming module, the Family and Pinout Codes, or characterizer.

3.2 OPERATION OVERVIEW

The 29A uses a "source/destination" method of syntax. The three source/destination keys (PORT, RAM, and DEVICE) are used in conjunction with the COPY and VERIFY function keys. The operator specifies a function, a source of data, and the destination for that data. The operation is initiated by pressing START.

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

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

3.2.1 COPY

The COPY key is used in conjunction with the source/destination keys to copy data from one medium to another or one RAM location to another. There are five basic Copy operations:

- A data transfer from a device to the programmer RAM is a **Load** operation.
- A data transfer from the serial port to the programmer RAM is an **Input** operation.
- A data transfer from the programmer RAM to a device is a **Program** operation.
- A data transfer from the programmer RAM to the serial port is an **Output** operation.
- A data transfer that moves a block of RAM data to another location within RAM is a **Block Move.**

LOAD. A Load operation consists of taking programming data from a master device and transferring it to the programmer RAM. When the data transfer is complete, the 29A calculates the sum-check[1] of the loaded data and displays it.

INPUT. In an Input operation, data from the serial port is transferred to the programmer RAM. When completed, the programmer calculates 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.

[1] *A sum-check is an automatic routine performed by the programmed to ensure accuracy in data transfers. It is further explained in the Glossary.*

PROGRAM. A Program operation duplicates the data in the programmer RAM into a device. Programming is automatic starting with an illegal-bit test and a blank check[2] to insure the device can be programmed. Then the data is transferred one byte at a time to the programming module. The programming module programs that byte into the device, then sends the programmer a signal to send the next byte. This continues until all data is programmed into the device. Nonblank parts can be programmed with additional data by pressing START after the nonblank error is displayed. After programming is completed, the data in the device is automatically compared with the source data to insure correct programming.

OUTPUT. In an Output operation data from the programmer RAM is transferred to the serial port.

BLOCK MOVE. A Block Move is accomplished with the COPY key. In a Block Move, the blocks of data within RAM can be rearranged. Figure 3-2 illustrates what can be accomplished with a Block Move.

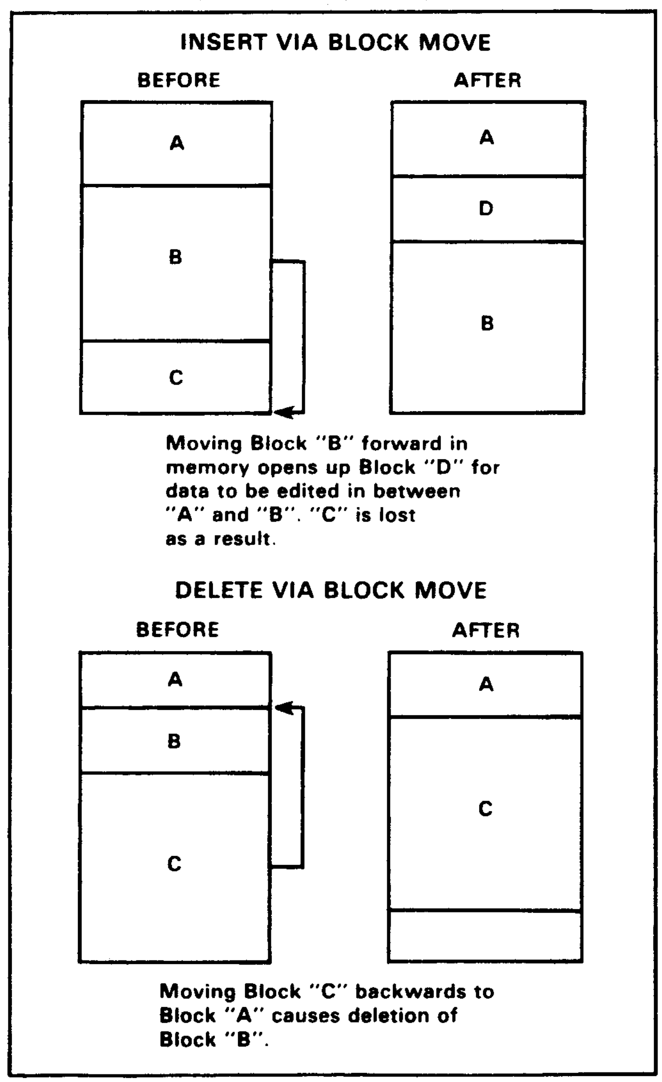


Figure 3-2. Block Move Capabilities

[2] *Illegal-bit test and blank check are defined in the Glossary.*

3.2.2 VERIFY

A Verify is a byte-by-byte comparison of the data in two locations, one referred to as the "source" and one as the "destination".

In a Verify Device operation, data from the device (the destination) is compared byte-by-byte twice with the data in RAM (the source). On the first pass (first comparison), parametric checking is done by lowering Vcc within the manufacturer's specified range. On the second pass, Vcc is raised high. (This procedure may vary depending on the programming module used.)

In an Input Verify operation, incoming data from the serial port is compared byte-by-byte with the data in RAM.

3.2.3 SELECT FUNCTIONS

The SELECT key gives you access to additional functions that are used for changing the default values of parameters, RAM data manipulations, and accessing certain, less frequently used operations. See section 3.5 for a list of Select Functions, descriptions, and key sequences.

3.2.4 EDIT

An Edit operation allows you to view and change data at specific RAM addresses. Select Functions F5, F6, and F7 can be used to select either a binary, octal, or hexadecimal number base. (The default is hexadecimal.) Data may be viewed or entered from the keyboard in binary, octal, or hexadecimal notation. Addresses are incremented one by one with the START key and decremented one by one with the REVIEW key. It is also possible to jump to any selected address by pressing the EDIT key, entering the new address, and pressing START.

3.3 PREOPERATIONAL PROCEDURES

CAUTION

Follow the procedures in the section carefully to avoid damaging the programmer or devices installed in it.

3.3.1 POWER-UP

The ON/OFF switch of the 29A is located on the rear panel. When you turn on the programmer it will show *SELF TEST-OK* after running through a self-test procedure that checks the programmer's scratch RAM, firmware, and size and operation of the data RAM. When this test is complete the programmer is ready for operation.

3.3.2 SETTING PARAMETERS

Default values are present in the programmer for certain parameters associated with operational procedures. When the programmer is powered up, these default values will remain in effect unless you change them. Once you do change them, the new value will remain until you turn the power off or enter other values. When a parameter is displayed it will be either the default or the last value you entered.

Many commonly changed parameters, such as the Block Size or Begin RAM Address, you can change while keying in the operation. Other less commonly changed parameters are changed via Select Functions. See section 3.5 for complete descriptions and key sequences for these. Those that can be directly changed during the operation are noted on the flowcharts for each operation.

3.3.3 FAMILY AND PINOUT CODES

Some Data I/O programming modules such as the UniPak or MOSPak require you to key in Family and Pinout Codes that identify the device you are working with. When this is necessary the 29A display (or terminal display) will prompt you with FXXPYY. XX and YY denote any default value that may exist or a previously entered code. This step is pointed out in the flowcharts accompanying each operation. To enter the codes simply enter the two-digit Family Code and the two-digit Pinout Code and press START. The codes you entered will now remain in effect for all operations until you enter new ones or the 29A is turned off.

3.3.4 OPERATION NOTES

- Whenever the ∧ character is in the display, a value can be entered or changed.
- While some operations are in progress, a rotating action symbol (denoted by ⌧) will appear on the display.
- An operation can be aborted at any step in its progress by pressing one of the four mode keys (COPY, VERIFY, SELECT, or EDIT). If a function is in progress when one of these keys is pressed, it will stop and the display will show *FUNCTION ABORT.*
- When doing Select Functions, two asterisks in the far right of the display is the completion symbol, signifying that the previously entered Select Function has been completed.

3.4 KEYBOARD OPERATIONS

Table 3-2 shows those operations accomplished from the programmer keyboard in conjunction with the COPY and VERIFY keys. Other operations are covered in section 3.4.4, Edit; section 3.5, Select Functions; and section 3.6, Remote Control.

Each keyboard operation is accomplished by a flowchart giving the key sequences and displays for the operation.

NOTE

In the flowcharts for each operation it is presumed that the original, power-up defaults are in effect rather than a previously entered value.

3.4.1 THE SOURCE/DESTINATION METHOD OF SYNTAX

The source/destination syntax method is used in conjunction with the function keys — DEVICE, RAM, and PORT. The COPY and VERIFY keys tell the programmer whether the data is to be copied from one medium to another or verified against other data to ensure they are identical.

The generalized key sequence for the COPY and VERIFY keys is:

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

Table 3-2. COPY and VERIFY Keyboard Operations

COPY KEY			
SOURCE \ DESTINATION	**DEVICE** (blank device)	**RAM** (programmer data RAM)	**PORT** (peripheral)
DEVICE (master device)		LOAD FROM DEVICE See Figure 3-3	
RAM (programmer data RAM)	PROGRAM DEVICE See Figure 3-5	BLOCK MOVE See Figure 3-7	OUTPUT TO PORT See Figure 3-6
PORT (peripheral)		INPUT FROM PORT See Figure 3-4	

VERIFY KEY			
SOURCE \ DESTINATION	**DEVICE** (previously programmed device)	**RAM** (programmer data RAM)	**PORT** (peripheral)
DEVICE (master device)			
RAM (programmer data RAM)	VERIFY DEVICE See Figure 3-8		INPUT VERIFY See Figure 3-9
PORT (peripheral)			

XXXX, YYYY, and ZZZZ are parameters associated with the source/destination keys. XXXX is the beginning source address, YYYY is the block size, and ZZZZ is the beginning destination address. Table 3-3 gives the usage, definition, and default values for these parameters. If the default values are correct, it is not necessary to enter them in the key sequences.

3.4.2 COPY OPERATIONS

The following operations use the COPY key. Copy operations can be repeated after they have been keyed in by pressing START. These are noted in the flowcharts for each operation.

LOAD FROM DEVICE. This operation loads the programmer data RAM with programming data from a master device. The generalized syntax is:

[COPY][DEVICE]XXXX/YYYY[RAM]ZZZZ[START]

Begin Device Address
Block Size
Begin RAM Address

Figure 3-3 shows the key sequence in detail.

Table 3-3. Address Parameters

KEY	DENOTED BY	ADDRESS PARAMETER	DEFINITION AND DEFAULT
DEVICE	XXXX or ZZZZ	Begin Device Address	First device address from which data is output or to which data is input. The default is 0. To return to the default value press the REVIEW key or enter 0.
RAM	XXXX or ZZZZ	Begin RAM Address	First data RAM address from which data is output or to which data is input. The default is 0. To return to the default value press the REVIEW key or enter 0.
PORT	XXXX or ZZZZ	Address Offset	When addresses larger than RAM are to be dealt with, the address offset is subtracted from all addresses on input and added on output. The result is added to the Begin RAM Address. The default is 0 on output and the first incoming address on input. If you change the Address Offset and want to return to the default enter FFFF. This will return you to the default.
DEVICE RAM PORT	YYYY	Block Size	The number of bytes to be transferred. The default value is the device size in device related operations. In port related operations it is from the first address specified to the end of RAM. To return to the default value, press REVIEW or enter 0.

* *Whenever defaults are in effect **ADDR** is displayed for address parameters and **SIZE** is displayed for Block Size.*

INPUT FROM PORT. This operation takes data coming in the serial port and transfers it to the programmer's data RAM. You may wish to specify the data translation format, word width, record size, and number of nulls before initiating this operation. This is accomplished with Select Functions detailed in section 3.5. The generalized syntax is:

[COPY][PORT]XXXX/YYYY[RAM]ZZZZ[START]

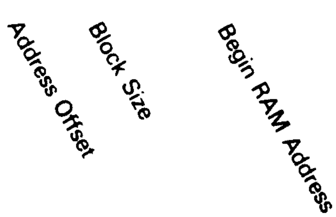


Figure 3-4 gives the detailed key sequences for this operation.

PROGRAM A DEVICE. This operation takes data from the programmer's data RAM and programs it into a device. Programming is preceded by an illegal-bit test and blank check and followed by an automatic parametric verify. The generalized syntax for this operation is:

[COPY][RAM]XXXX/YYYY[DEVICE]ZZZZ[START]

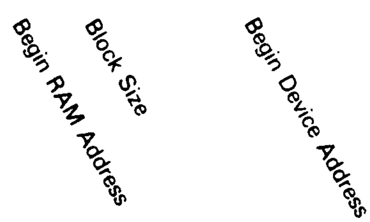


Figure 3-5 illustrates the procedure in detail.

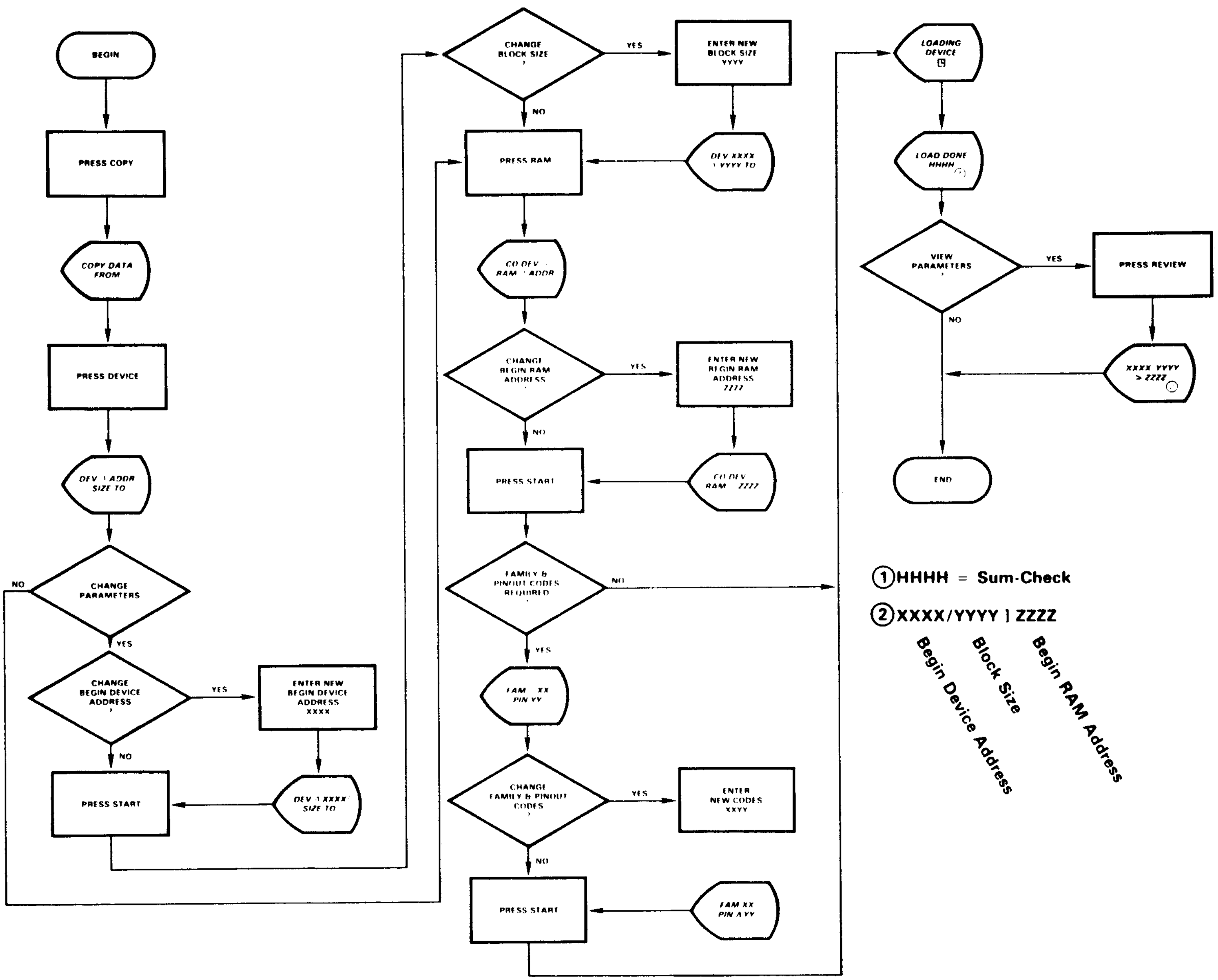
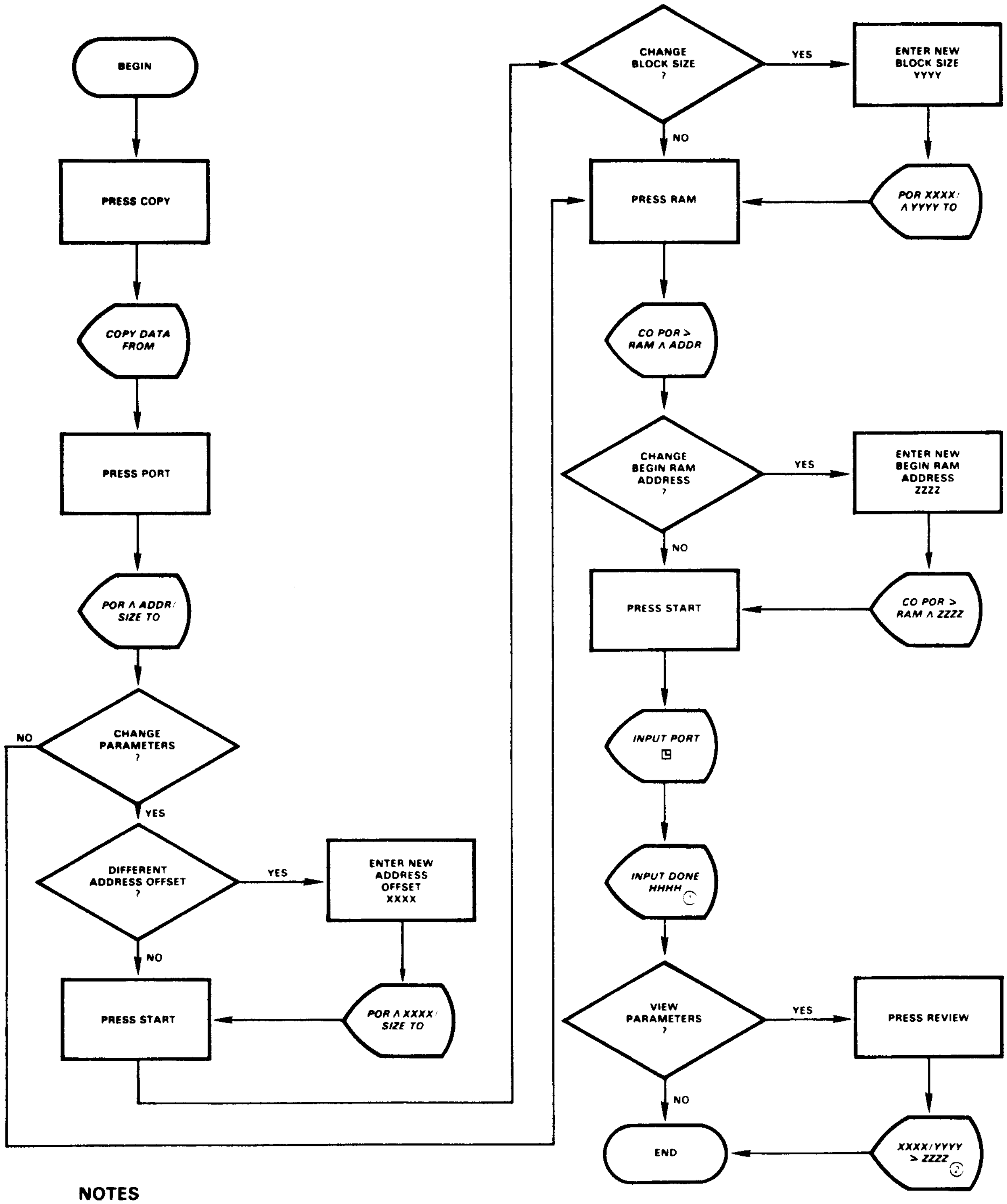


Figure 3-3. Load from Device



NOTES

①HHHH = Sum-Check

②XXXX/YYYY] ZZZZ

- XXXX: Begin Device Address
- YYYY: Block Size
- ZZZZ: Begin RAM Address

Figure 3-4. Input from Port

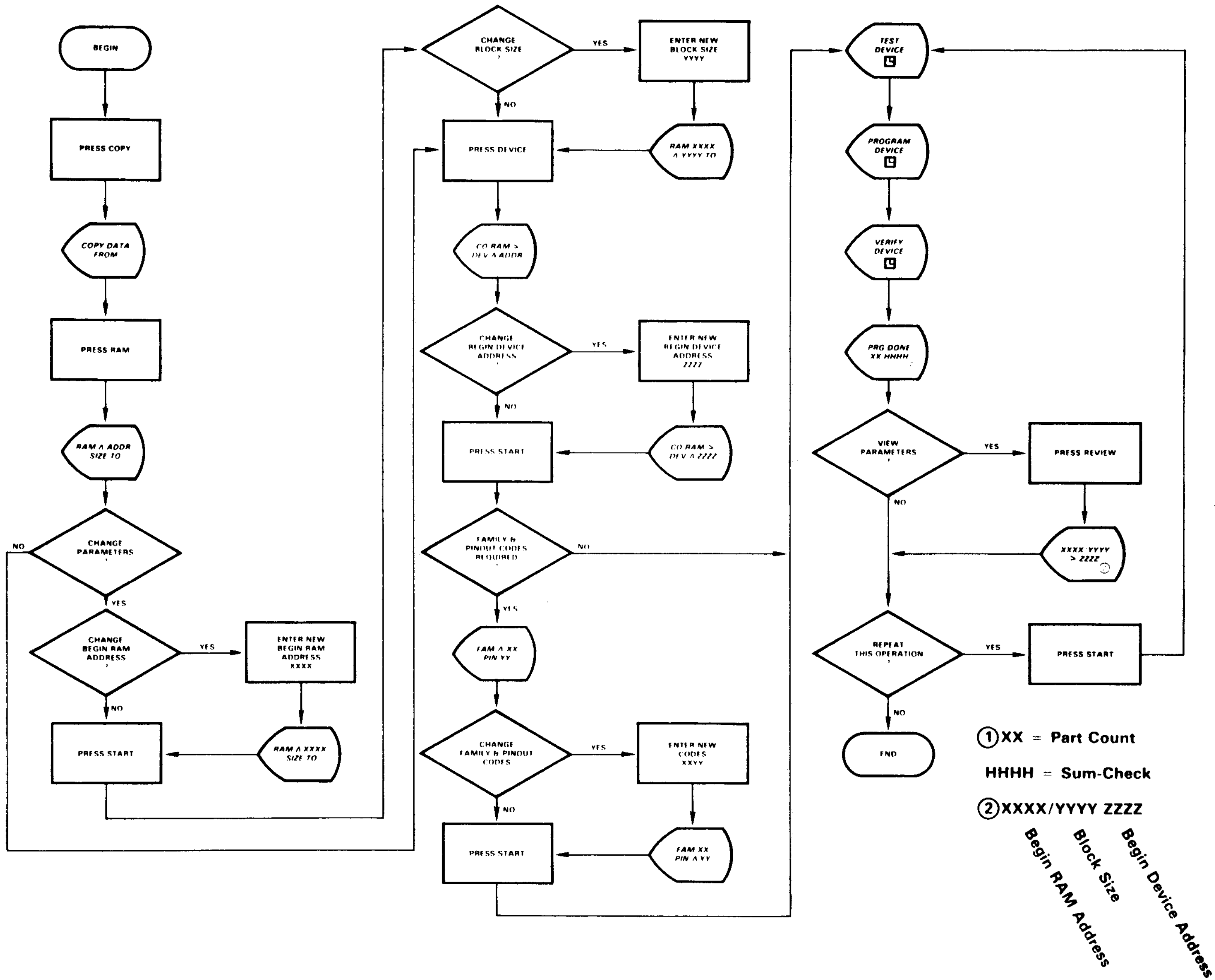


Figure 3-5. Program Device

OUTPUT TO PORT. This operation takes data from the programmer data RAM and outputs it to the serial port. The generalized syntax for this operation is:

[COPY][RAM]XXXX/YYYY[PORT]ZZZZ[START]

- XXXX: Begin RAM Address
- YYYY: Block Size
- ZZZZ: Address Offset

Figure 3-6 illustrates this procedure in detail.

BLOCK MOVE. Blocks of RAM data can be relocated in a RAM-to-RAM Block Move operation. The generalized syntax for this operation is:

[COPY][RAM]XXXX/YYYY[RAM]ZZZZ[START]

- XXXX: Begin RAM Address
- YYYY: Block Size
- ZZZZ: Destination RAM Address

Figure 3-7 illustrates this procedure in detail.

3.4.3 VERIFY OPERATIONS

The following operations use the VERIFY key as the function.

VERIFY DEVICE. This operation verifies a previously programmed device against the data in RAM to ensure identity. (The programmer data RAM must first be loaded with the correct data.) The generalized syntax is:

[VERIFY][RAM]XXXX/YYYY[DEVICE]ZZZZ[START]

- XXXX: Begin RAM Address
- YYYY: Block Size
- ZZZZ: Begin Device Address

Figure 3-8 illustrates the procedure in detail.

If a device fails a Verify operation the programmer will display:

V1 0000 D00 R00

- V1: verify pass 1 or 2
- 0000: address of verify fail
- D00: device data at that address
- R00: RAM data at that address

Pressing START while this is displayed will let you skip that address and go on to the next address to mis-verify. This allows you to check each mis-verified address.

INPUT VERIFY. This operation verifies data incoming from the serial port against the data in the programmer's data RAM. (The programmer's data RAM must be loaded with the correct data.) The generalized syntax is:

[VERIFY][RAM]XXXX/YYYY[PORT]ZZZZ[START]

- XXXX: Begin RAM Address
- YYYY: Block Size
- ZZZZ: Address Offset

Figure 3-9 illustrates this procedure in detail.

3.4.4 EDIT OPERATIONS

The EDIT key can be used to view or change data at individual RAM addresses. Follow the procedure outlined in Figure 3-10.

When viewing data at a specific address, it can be displayed or entered in either hexadecimal, octal, or binary notation. The desired base should be selected prior to initiating the Edit operation, via Select Functions F5, F6, or F7. The default is hexadecimal.

Edit operations take into account any previously entered device parameter for the Begin Device Address.

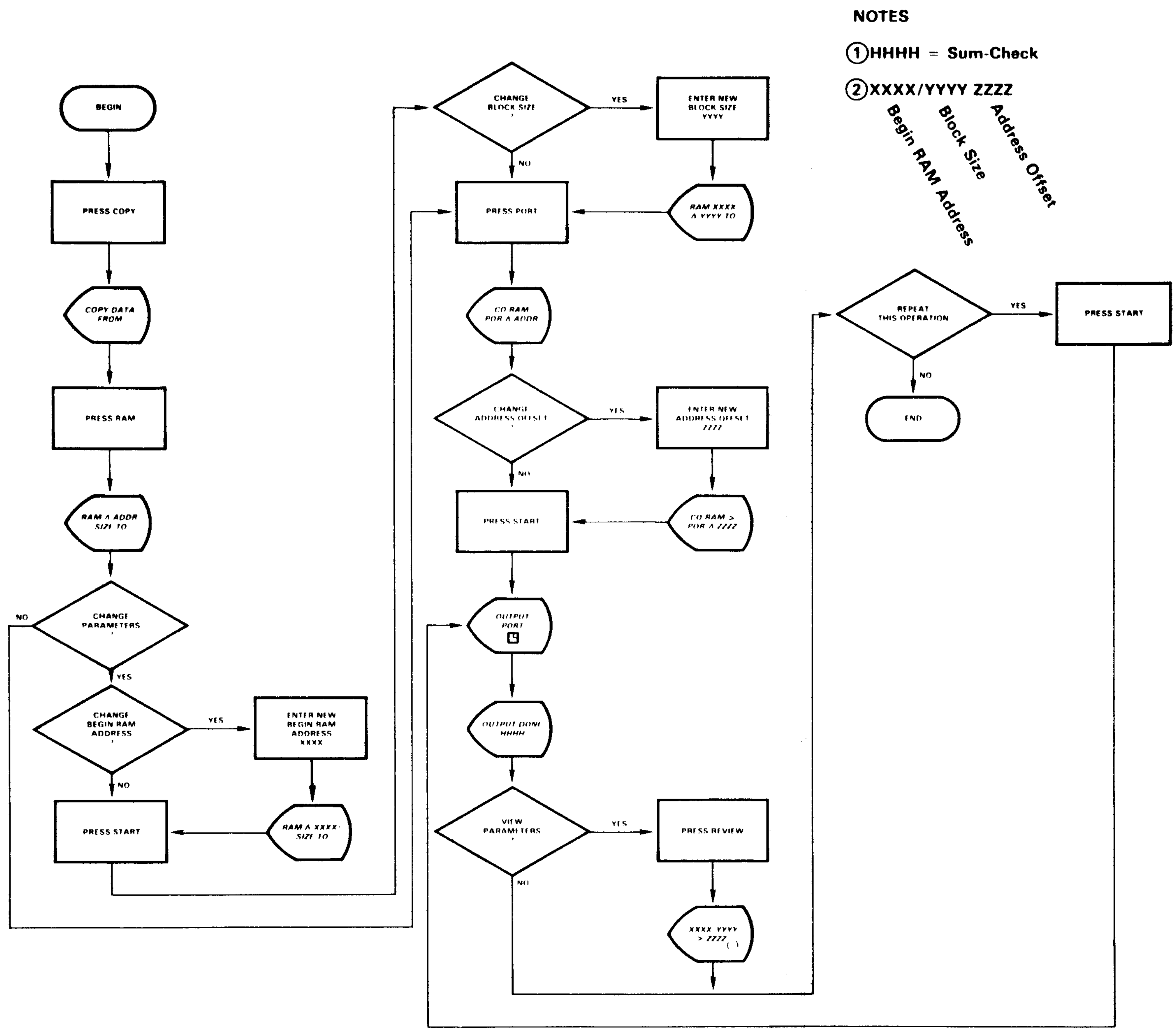


Figure 3-6. Output to Port

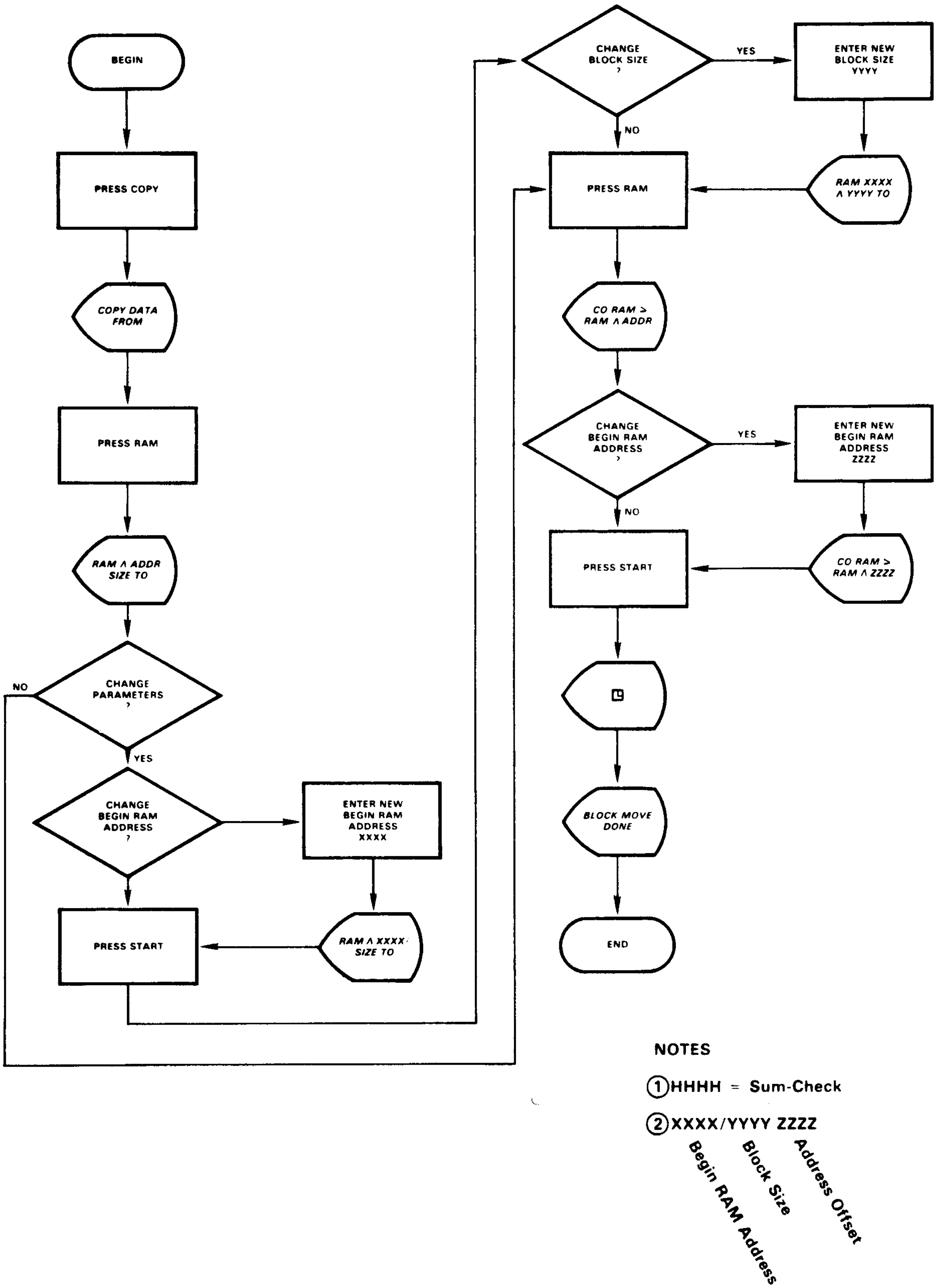
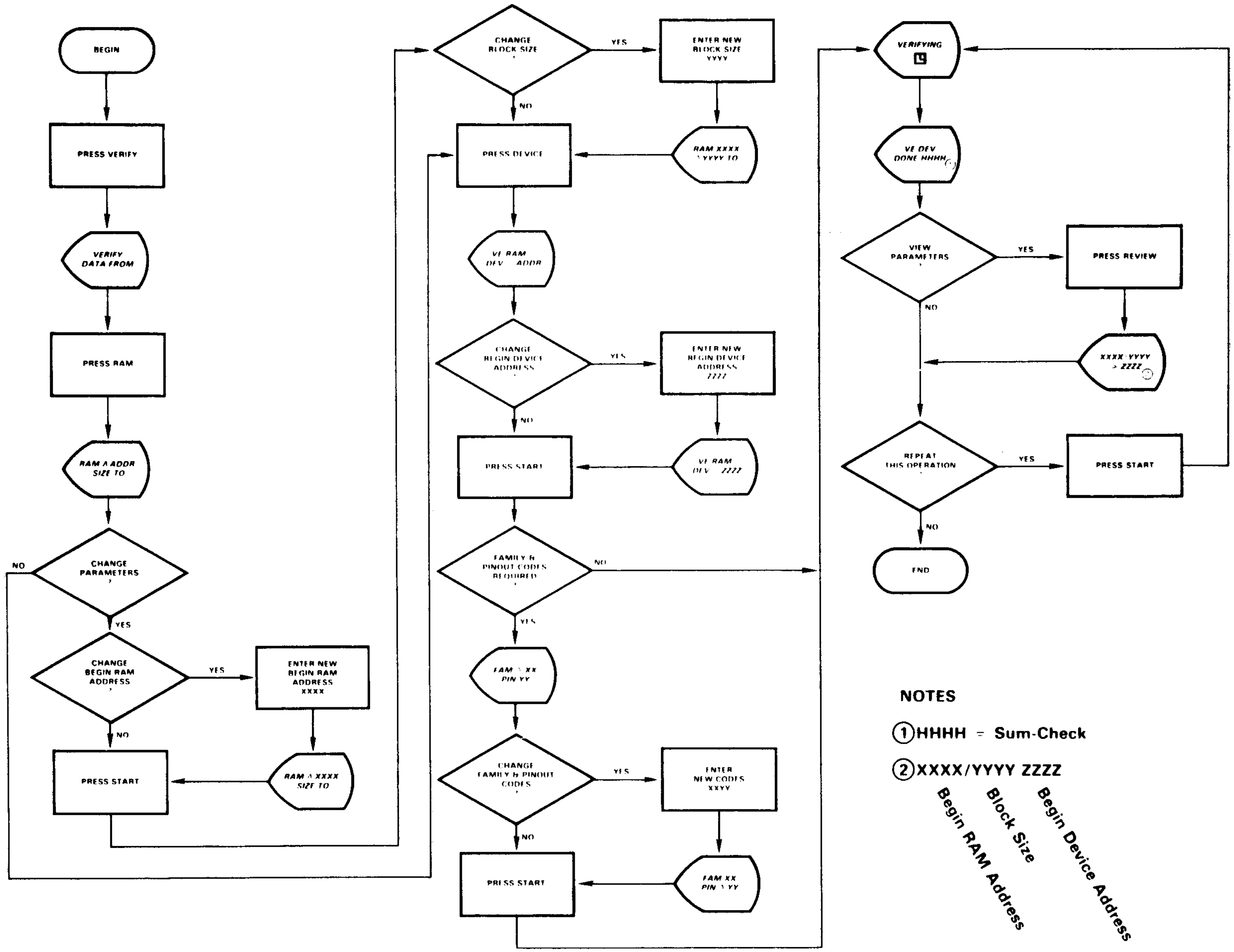


Figure 3-7. Block Move

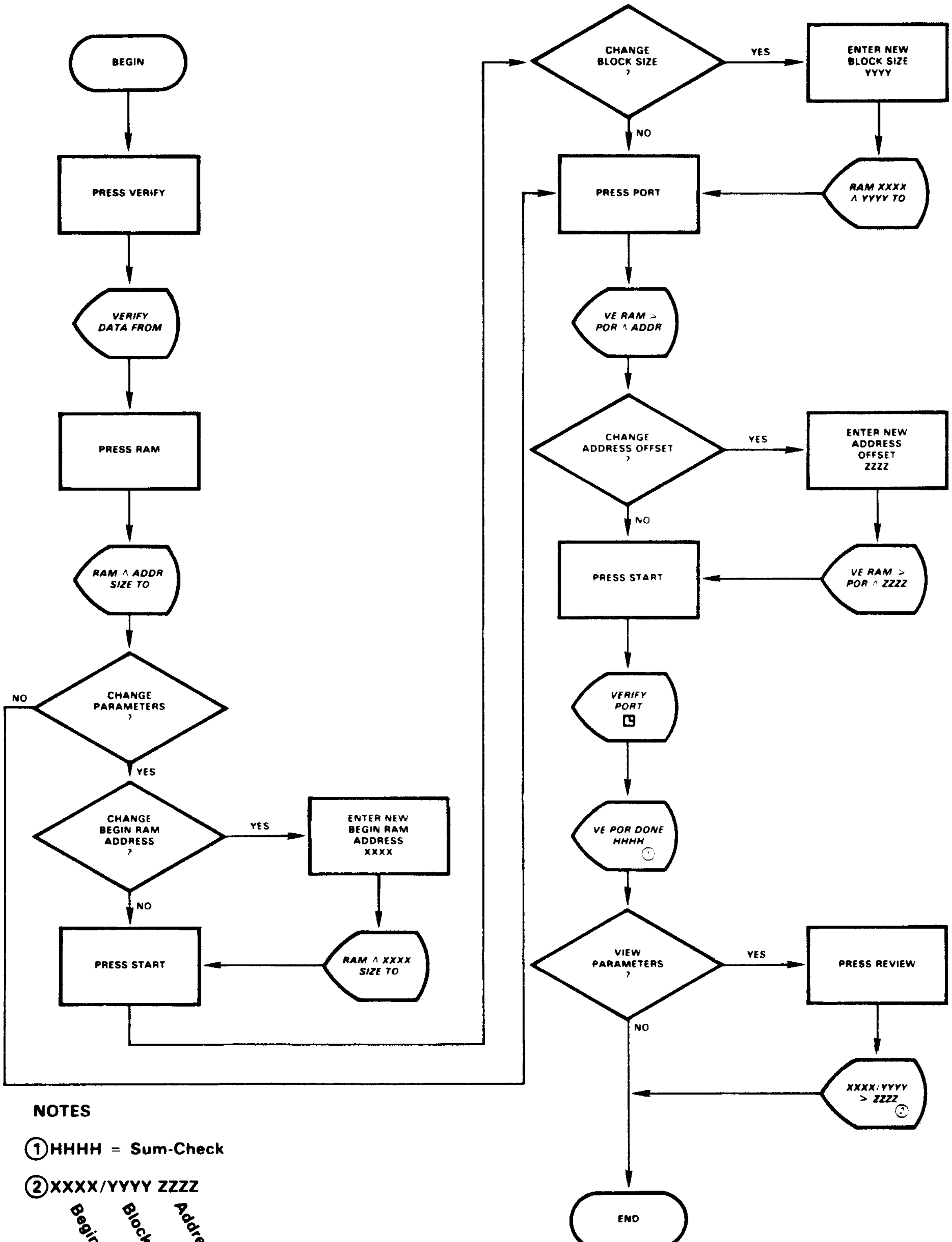


NOTES

① HHHH = Sum-Check

② XXXX/YYYY ZZZZ — Begin RAM Address / Block Size / Begin Device Address

Figure 3-8. Verify Device



NOTES

(1) HHHH = Sum-Check

(2) XXXX/YYYY ZZZZ — XXXX: Begin RAM Address; YYYY: Block Size; ZZZZ: Address Offset

Figure 3-9. Input Verify

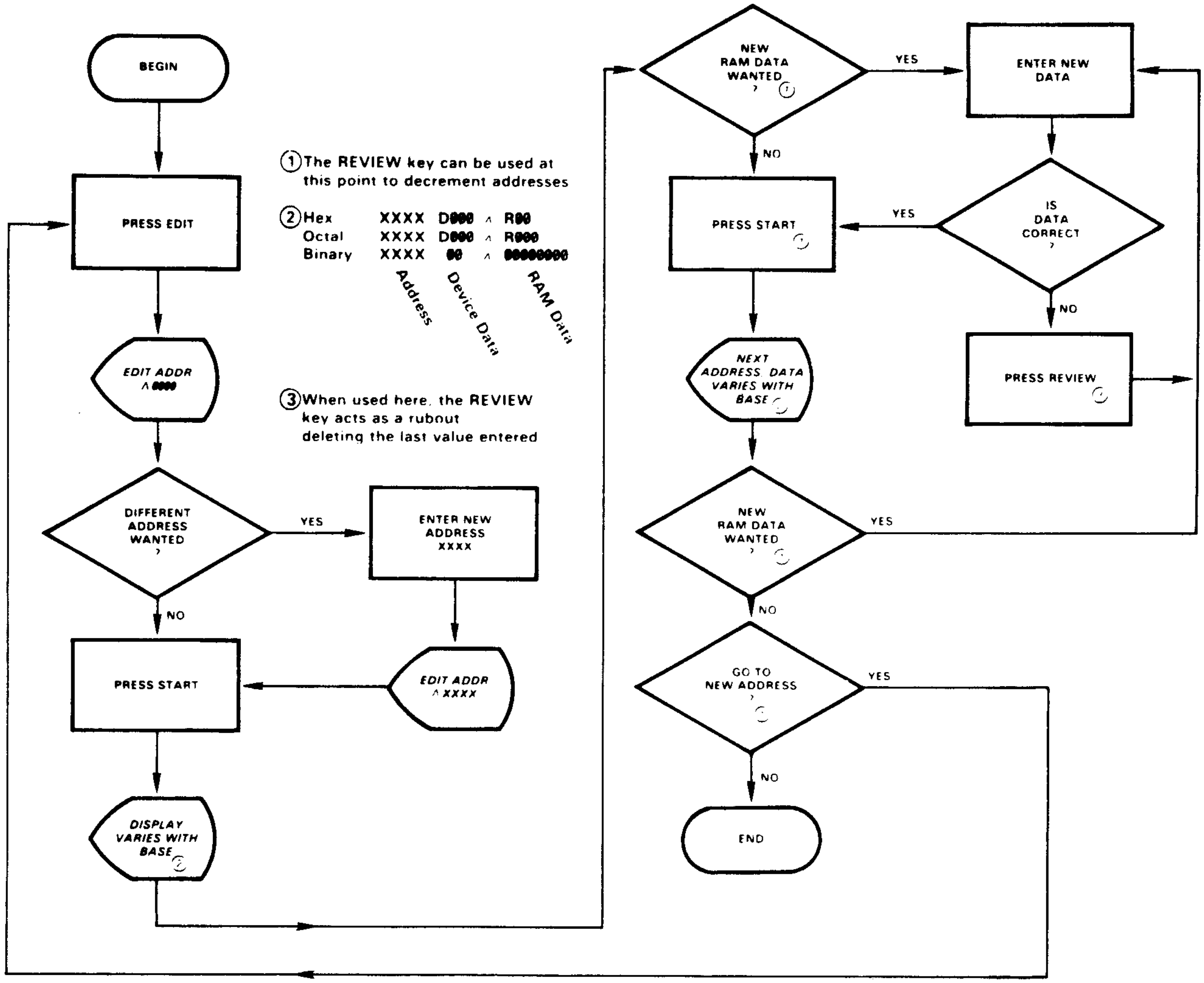


Figure 3-10. Edit

3.5 SELECT FUNCTIONS

Many of the 29A's operations are initiated by hexadecimal Select Functions. The SELECT key is used to access these operations. Pressing SELECT informs the programmer that a Select Function command is to follow.

3.5.1 ACCESSING SELECT FUNCTIONS

There are three ways to access the Select Functions: direct entry, stepping, or scrolling.

In direct entry, press SELECT. The 29A will display *SELECT CODE* > . Enter the hex code for the desired function or data translation format and press START. The display will prompt you if any additional entries are required.

Repeated depressions of the SELECT key allow you to step through the complete menu of Select Functions. When the desired one is displayed, press START. To step backwards through the Select Functions press the REVIEW key. The functions are displayed in hexadecimal order.

The 29A will also scroll through the whole Select Function menu automatically. This is done by pressing SELECT then START. Each function is momentarily displayed before the programmer moves on to the next function. When the desired function is displayed, you can stop the menu by pressing any key. To back up, press REVIEW. To initiate the function press START.

3.5.2 DESCRIPTIONS AND KEY SEQUENCES

Table 3-4 lists the Select Functions, descriptions, and key sequences. The key sequences given are for direct entry. When stepping or scrolling, simply skip the first two or three steps as shown in Table 3-4.

Some programming modules also have Select Functions or Select Codes that can be used when the module is installed in the 29A. These are described in the programming module manuals.

Table 3-4. Select Functions

CODE	MENU DISPLAY	DESCRIPTION	ENTRY KEY SEQUENCE	ENTRY DISPLAY
DATA MANIPULATION COMMANDS				
A1	*SWAP NIBBLES A1*	Exchanges high and low order nibbles of every byte.	1. Press SELECT 2. Enter A1. 3. Press START. * 4. Press START to exchange high or low order nibbles of every byte.	 *SWAP NIBBLES A1* *SWAP NIBBLES* **
A2	*FILL RAM A2*	Fills 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, it will fill only the lower order nibble of RAM: otherwise it will default to the word width.	1. Press SELECT. 2. Enter A2. * 3. Press START. 4. Enter the hex data (XX). 5. Press START.	 *FILL RAM 00 A2* *FILL RAM XX A2* *FILL RAM* **
A3	*INVERT RAM A3*	Performs the ones complement of 4 or 8 bits of each word as determined by the word size in effect.	1. Press SELECT. 2. Enter A3. 3. Press START. * 4. Press START to complement all of RAM.	 *INVERT RAM A3* *INVERT RAM* **
A4	*CLEAR ALL RAM A4*	Clears all of RAM to zeros.	1. Press SELECT. 2. Enter A4. 3. Press START. * 4. Press START.	 *CLEAR ALL RAM A4* *CLEAR ALL RAM* **
A5	*SPLIT RAM A5*	Splits 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 midpoint (XXXX).	1. Press SELECT. 2. Enter A5. * 3. Press START. 4. Enter the center point (YYYY). 5. Press START to split RAM.	 *SPLIT RAM ∧ XXXX* *SPLIT RAM ∧ YYYY* *SPLIT RAM* **
A6	*SHUFFLE RAM A6*	Shuffles 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 RAM midpoint (XXXX).	1. Press SELECT. 2. Enter A6. * 3. Press START. 4. Enter the center point (YYYY) if the default is not correct. 5. Press START *to shuffle RAM.*	 *SHUFFLE RAM XXXX* *SHUFFLE RAM YYYY* *SHUFFLE RAM* **
UTILITY AND INQUIRY COMMANDS				
B0	*DEVICE SIZE B0*	Displays the device word size and word width.	1. Press SELECT. 2. Enter B0. * 3. Press START.	 *DEV SZ XXXX Y* ** (device size)

* Entry point when scrolling or stepping through the menu.

Table 3-4. Select Functions

			ENTRY	
CODE	MENU DISPLAY	DESCRIPTION	KEY SEQUENCE	DISPLAY
B1	*SUMCHECK RAM B1*	Displays the RAM sum-check.	1. Press SELECT. 2. Enter B1. * 3. Press START.	 *SUMCHECK XXXX* **
B2	*SYSTEM CONFIG B2*	Displays the software configuration number.	1. Press SELECT. 2. Enter B2. * 3. Press START.	 *CONFIG XXXX* **
B3	*FORMAT NUMBER B3*	Displays the data translation format in effect and allows you to change it. The default value is ASCII-Hex (Space), (050 or the previous entry,)	1. To see the data translation format in effect: a. Press SELECT. b. Enter B3. * c. Press START. d. If a new format is desired, enter the format code (XXX). e. Press START. + f. Press START. 2. To enter the format code directly: a. Press SELECT. b. Enter the format number (XXXX). c. Press START. + d. Press START.	 *HEX SPCE STX* *FORMAT NO XXX* *FORMAT NAME* *FORMAT NO XXX* ** *SELECT CODEV* *FORMAT NO XXX* *Format Name* *FORMAT NO XXX* **
B9	*DISPLAY TEST B9*	Lights all display segments for 5 seconds.	1. Press SELECT. 2. Enter B9. 3. Press START. * 4. Press START to test.	 *DISPLAY TEST B9* *DISPLAY TEST XX*
C1	*CALIBRATION C1*	Puts the programmer in the calibration mode. This function is inhibited in remote operation. See Section 4, Calibration,	1. Press SELECT. 2. Enter C1. * 3. Press START.	 *CAL STEP ^Ø1* *(with expanded memory programming module installed)* *or* *HEX ADDR ^ØØØØ* *(with standard programming module installed)*
FØ	*PROGRAM COUNT FØ*	Displays the count of devices programmed since power-up or last reset.	1. Press SELECT 2. Enter FØ. * 3. Press START. 4. Press START again if you want to reset the parts count to ØØ.	 *PROGRAM COUNT XX* *PROGRAM COUNT* **
F1	*REMOTE MODE F1*	Puts the programmer in the Optional Computer Remote Control (only in models equipped with this option).	1. Press SELECT. 2. Enter F1. * 3. Press START. 4. Press START to enter the remote mode.	 *REMOTE MODE F1* *REMOTE MODE* **

* Entry point when scrolling or stepping through the menu.

\+ Pressing SELECT instead of START at this point will step you through the format menu.

Table 3-4. Select Functions

CODE	MENU DISPLAY	DESCRIPTION	ENTRY KEY SEQUENCE	DISPLAY
F3	*LOCK DATA ON*	*F3* Sets the data lock on. This protects the data in RAM for a series of identical programming operations. While the data lock is in effect, keys used to manipulate data are disabled. The only operations possible are: • Copy operations that move data from RAM to the port or device • Verify operations (except Input Verify) • Abort operation in progress • Abort operation in progress • Release data lock	1. To engage data lock: a. Press SELECT. b. Enter F3. * c. Press START. 2. To release data lock: a. Press SELECT. b. Press PORT. c. Press REVIEW. The data lock is now inactive.	 *LOCK DATA ON* ** *PASSWORD?* *SELECT CODE≥*
F4	*NIBBLE MODE*	*F4* Selects a 4-bit word size to override 8-bit programming electronics for I/O transfers.	1. Press SELECT. 2. Enter F4. 3. Press START. * 4. Press START.	 *NIBBLE MODE F4* *NIBBLE MODE* **
F5	*BINARY BASE*	*F5* Sets the number base for Edit operations to binary.	1. Press SELECT. 2. Enter F5. 3. Press START. * 4. Press START to enable binary base for Edit operations.	 *BINARY BASE F5* *BINARY BASE* **
F6	*OCTAL BASE*	*F6* Sets the number base for Edit operations to octal.	1. Press SELECT. 2. Enter F5. 3. Press START. * 4. Press START to enable octal base for Edit operations.	 *OCTAL BASE F6* *OCTAL BASE* **
F7	*HEX BASE*	*F7* Resets the number base for Edit operations to hex. This is the default base.	1. Press SELECT. 2. Enter F7. 3. Press START. * 4. Press START to enable hex base.	 *HEX BASE F7* *HEX BASE* **
F8	*BYTE/NIB MODE*	*F8* Nullifies F4; allows word size of the programming module to take effect.	1. Press START. 2. Enter F8. 3. Press START. * 4. Press START to establish the word size of programming electronics.	 *BYTE/NIB MODE F8* *BYTE/NIB MODE* **

SERIAL I/O COMMANDS

CODE	MENU DISPLAY	DESCRIPTION	KEY SEQUENCE	DISPLAY
D7	*LEADER OUTPUT*	*D7* Sends 50 nulls from the serial port.	1. Press SELECT. 2. Enter D7. 3. Press START. * 4. Press START to output a 50-null leader.	 *LEADER OUTPUT D7* *LEADER OUTPUT* **

* Entry point when scrolling or stepping through the menu.

Table 3-4. Select Functions

CODE	MENU DISPLAY		DESCRIPTION	ENTRY KEY SEQUENCE	ENTRY DISPLAY	
D8	*SIZE RECORD*	*D8*	Changes the number of bytes per data record on the serial output. The value entered must be in hex notation.	1. Press SELECT.		
				2. Enter D8.		
				3. Press START. The record size in effect (XX) will be displayed.	*SIZE RECvXX*	*D8*
				4. If a new value is desired enter it (YY).	*SIZE REC YY*	*D8*
				5. Press START.	*SIZE RECORD*	**
D9	*NULL COUNT*	*D9*	Sets up to 254 nulls (FE in hexadecimal) following each data record on output. Selecting 255 (FF) sends no nulls and no line feed.	1. Press SELECT.		
				2. Enter D9.		
				*3. Press START. The number of nulls in effect will be displayed.	*NULL COUNT*	*Ø1 D9*
				4. If a different value is wanted, enter the hexadecimal value (XX).		
				5. Press START to enable the new null count.	*NULL COUNT*	**
F9	*TIMEOUT OFF*	*F9*	Disables the standard 25-second I/O timeout.	1. Press SELECT.		
				2. Enter F9.		
				3. Press START.	*TIMEOUT OFF*	*F9*
				*4. Press START to disable the timeout.	*TIMEOUT OFF*	**
				5. The timeout can only be re-enabled by turning the power off and back on again.		
FA	*CHAR OUTPUT*	*FA*	After this code is entered, enter the hex code for an ASCII character (see Appendix C). The character is transmitted to the port each time you press START. This function is inhibited in remote control.	1. Press SELECT.		
				2. Enter FA.		
				3. Press START.	*CHAR OUTPUT*	*ØØ*
				*4. Enter hex value of ASCII character.	*CHAR OUTPUT*	*XX*
				5. Press START.	*CHAR OUTPUT*	*XX*
				6. Repeat step 5 to repeat the character. Repeat steps 4 and 5 to select other characters.		
FB	*ENABLE PORT*	*FB*	Enables Remote Control and the Input Interrupt and forces the RTS line high at all times for remote control from peripherals requiring hardware handshake. The default at power-up is RTS low and Remote Control and Interrupt disabled.	1. Press SELECT.		
				2. Enter FB.		
				3. Press START.	*ENABLE PORT*	*FB*
				*4. Press START to set the RTS line high.	*ENABLE PORT*	**
				5. To reset the RTS line low, turn the power off and back on.		
FC	*REMOTE ON OFF*	*FC*	Allows you to turn remote control on and off remotely via hexadecimal codes you select. (Applies to both standard and Optional Computer Remote Control.) See Appendix D for a more detailed description.	1. Press SELECT.		
				2. Enter FC.		
				*3. Press START.	*RMT ON OFF*	*ØØØØ*
				4. Enter on code XX; press START.	*RMT ON OFF*	*XX ØØ*
				5. Enter off code YY.	*RMT ON OFF*	*XXYY*
				6. Press START.	*RMT ON OFF*	**

* Entry point when scrolling or stepping through the menu.

3.6 REMOTE CONTROL

The 29A's standard Remote Control capability allows control of the programmer's operations from a terminal or through computer software.

NOTE

Select Function FB Remote Control. As long as the controlling instrument is properly interfaced (see Section 2, Installation), all operations can be done either on the programmer keyboard or the controlling instrument.

3.6.1 COMMAND PROTOCOL

The syntax for Remote Control is similar to that of keyboard operations, using the source/destination syntax method.

When keying in commands from a terminal, the 29A recognizes the first two characters of each command, as shown in Table 3-5.

Table 3-5. Command Entry in Remote Control

KEYBOARD COMMAND	REMOTE CONTROL COMMAND
COPY	CO_v
VERIFY	VE_v
PORT	PO_v
RAM	RA_v
DEVICE	DE_v
SELECT	SE_v
REVIEW	/
EDIT	ED_v

The space bar (denoted by $_v$) used after the command acts as a delimiter, setting the boundaries for that command. The programmer will not define the characters input until the space bar is used. And since the programmer only recognizes the first two characters, some variation is possible. For example:

$\text{CO}_v\text{DE}_v\text{XXXX}_v\text{YYYY}_v\text{TO}_v\text{RA}_v\text{ZZZZ}$[CR]

$\text{COPY}_v\text{DEVICE}_v\text{XXXX}_v\text{YYYY}_v\text{TO}_v\text{RAM}_v\text{ZZZZ}$[CR]

In the examples above, 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 keyboard.

The carriage return [CR] at the end of the line acts as an execute key. As characters are input to the programmer, they are stored until the [CR] is input, signalling the programmer to execute that line of characters.

NOTE

The word "TO" must be keyed in prior to the destination.

3.6.2 COMMAND ENTRY

There are two methods of command entry, direct or interactive.

In direct command entry, you type in the commands, using the space bar between words, as shown in the examples in paragraph 3.6.1. X, Y, and Z values are optional.

The interactive method streamlines the entries required of the operator. In the interactive method, you key in the function, then press [CR]. The terminal displays prompt you just as the 29A keyboard does with, for example, COPY DATA FROM > if the COPY key is used. The operation occurs just as they do when using the programmer's keyboard except that you key in command rather than pressing keys on the 29A.

When entering data on the terminal, the slash (/) is used in place of the REVIEW key. When pressed, it will delete the previous character or characters. It will not delete anything prior to a space.

3.6.3 INPUTTING PARAMETERS

The parameters required are the same as those given in Table 3-3 for keyboard operations. The values 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 the space bar. Figure 3-11 shows examples of inputting parameters.

WHEN DEFAULT VALUES ARE ALL CORRECT:

```
DIRECT ENTRY
CO DE TO RA CR
INTERACTIVE
CO CR
(COPY DATA FROM > displayed on terminal)
DE CR
(DEV ADDR,SIZE > displayed on terminal)
TO RA 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 DE XXXX YYYY TO RA ZZZZ CR
INTERACTIVE
CO CR
(COPY DATA FROM > displayed on terminal)
DE CR
(DEV ADDR,SIZE > displayed on terminal)
XXXX YYYY TO RA 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 DEV ,XXXX TO RA CR
INTERACTIVE
CO CR
(COPY DATA FROM > displayed on terminal)
DE CR
(DEV ADDR,SIZE > displayed on terminal)
,YYYY TO RA CR
(COPY DEV>RAM ADDR>displayed on terminal) CR
```

Figure 3-11. Inputting Remote Control Parameters

Table 3-6. Remote Control Commands

OPERATION	DIRECT ENTRY	INTERACTIVE ENTRY ENTRY	 TERMINAL DISPLAY
Input from Port	COvPOvXXXXvYYYYvTOvRAvZZZZ + [CR]	CO[CR] PO[CR] XXXX,YYYYvTOvRA[CR] ZZZZ[CR]	COPY DATA FROM > POR ADDR, SIZE > COPY POR > RAM ADDR >
Load from Device	COvDEvXXXXvYYYYvTOvRAvZZZZ + [CR]	CO[CR] DE[CR] XXXX,YYYYvTOvRA[CR] ZZZZ[CR] *	COPY DATA FROM> DEV ADDR, SIZE> COPY DEV > RAM ADDR >
Program Device	COvRAvXXXXvYYYYvTOvDEvZZZZ + [CR]	CO[CR] RA[CR] XXXX,YYYYvTOvDE[CR] ZZZZ[CR] *	COPY DATA FROM > RAM ADDR, SIZE> COPY RAM > DEV ADDR >
Output to Port	COvRAvXXXXvYYYYvTOvPOvZZZZ + [CR]	CO[CR] RA[CR] XXXX,YYYYvTOvPO[CR] ZZZZ[CR]	COPY DATA FROM > RAM ADDR, SIZE> COPY RAM > POR ADDR >
Block Move	COvRAvXXXXvYYYYvTOvRAvZZZZ + [CR]	CO[CR] RA[CR] XXXX,YYYYvTOvRA[CR] ZZZZ[CR]	COPY DATA FROM> RAM ADDR,SIZE> COPY RAM > RAM ADDR >
Verify Device	VEvRAXXXXvYYYYvTOvDEvZZZZ + [CR]	VE[CR] RA[CR] XXXX,YYYYvTOvDE[CR] ZZZZ[CR]	VERIFY DATA FROM > RAM ADDR,SIZE > VE RAM> DEV ADDR>
Input Verify	VEvRAvXXXXvYYYYvTOvPOvZZZZ + [CR]	VE[CR] RA[CR] XXXX,YYYYvTOvPO[CR] ZZZZ[CR]	VERIFY DATA FROM> RAM ADDR,SIZE > VE RAM>POR ADDR>

* If Family and Pinout Codes are necessary, the terminal will prompt for them at this point.

\+ If Family and Pinout Codes are required, enter FFPP after ZZZZ.

3.6.4 COPY AND VERIFY OPERATIONS

Table 3-6 lists the basic Remote Control protocols for Copy and Verify operations. Each is a "worst case" example, showing entry of all parameters. These will not be necessary when the default values are satisfactory.

3.6.5 EDIT OPERATIONS

There are three variations to Remote Control Edit operations.

- To view the last address edited (the default is Ø), key in:

  ED[CR]

- To view a specific address, key in:

  EDvHHHH[CR]

  desired address

- To enter data at a specific address, enter:

ED$_v$HHHH$_v$HH[CR]

desired address   data*

* *The exact number of digits will vary depending on the base specified.*

Just as in keyboard Edit operations, Remote Control Edit operations take into account any previously set device address parameters.

3.6.6 SELECT FUNCTIONS

In Remote Control it is possible to display the whole Select Function menu at one time. This is done by entering:

SE[CR]

Figure 3-12 shows the Select Function menu.

Individual Select Functions can be accessed in two ways.

- Entering the first two letters of the Select Function (as shown in Figure 3-12) followed by a [CR].
- Entering SE$_v$HH[CR] where HH is the hex code for the desired Select Function (when no additional parameters are required), or entering SE$_v$HH$_v$XXXX[CR] when XXXX is a parameter required for that Select Function.

To view the default values in effect for a specific Select Function, enter SE$_v$HH[CR]. This will display the value in effect.

To view the entire data translation format menu: enter either the first two letters, FO, followed by a [CR] or SE$_v$B3[CR]. The entire menu and the format currently in effect will be displayed as in Figure 3-13.

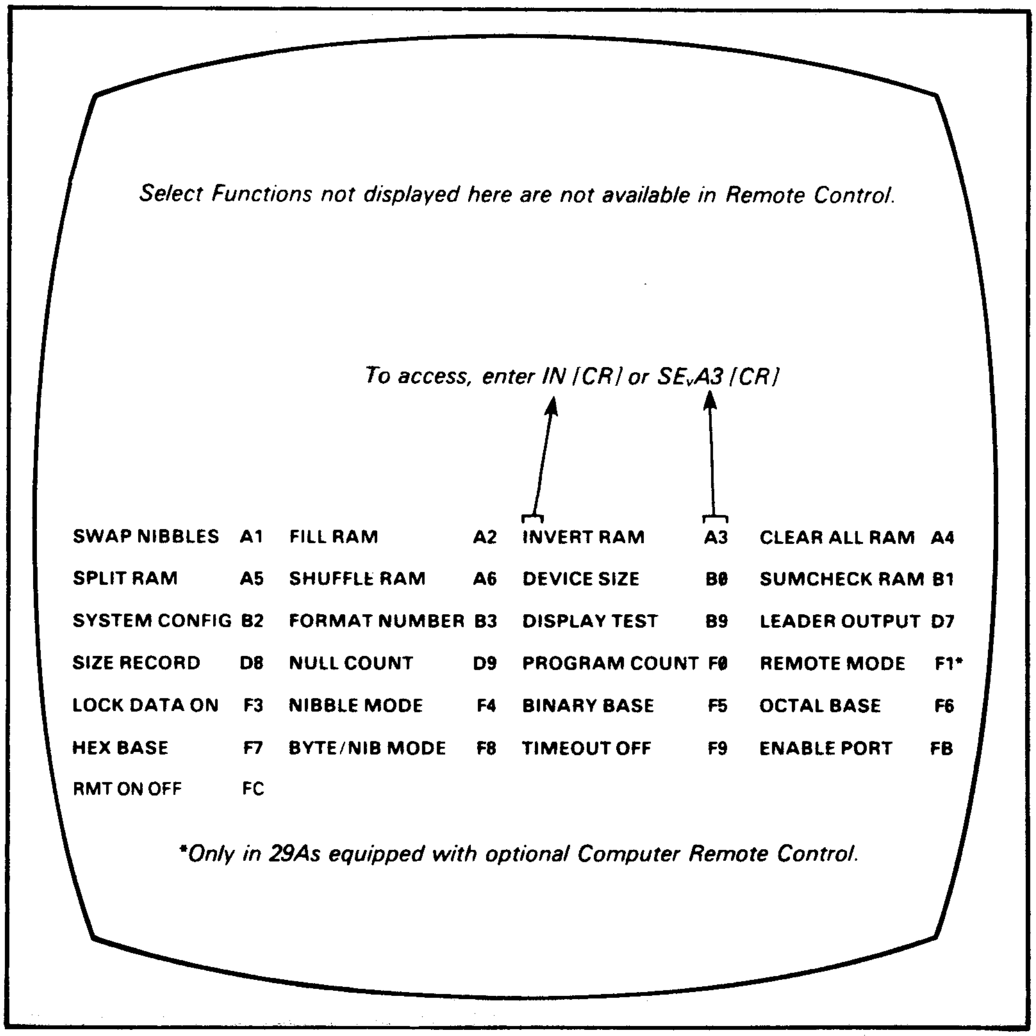


Figure 3-12. Select Function Menu in Remote Control

Default value or format currently in effect.

BINARY	10	DEC BINARY	11	BNPF STX	01	BNPF NO STX	05
BHLF STX	02	BHLF NO STX	06	B10F STX	03	B10F NO STX	07
5-BNPF STX	08	5-BPF NO STX	09	SPECTRUM STX	12	SPECT NO STX	13
OCTAL SP STX	30	OCTAL SP SOH	35	OCTAL PC STX	31	OCTAL PC SOH	36
OCTAL APOST	32	OCTAL SMS	37	HEX SPCE STX	50	HEX SPCE SOH	55
HEX PCNT STX	51	HEX PCNT SOH	56	HEX APOST	52	HEX SMS	57
HEX COMA STX	53	HEX COMA SOH	58	RCA COSMAC	70	FAIRCHILD	80
MOS TECH	81	MOTOROLA EX	82	INTEL MDS	83	SIGNETICS	85
TEKTRONIX	86	MOT EXORMAX	87	INTEL MCS	88	HP 64000 ABS	89
TI SDSMAC	00						

Figure 3-13. Data Translation Format Menu in Remote Control.

3.7 ERROR CODES

Table 3-7 gives descriptions and corrective actions for the 29A's error codes. Additional error codes may appear on the programmer display, depending on the presence of other equipment, such as a UniPak, MOS Pak, or other programming module.

Table 3-7. Error Codes

DISPLAY		DESCRIPTION	CORRECTIVE ACTION
SORC/DEST ERR	*15*	Illegal source/destination key sequence was entered.	Check key sequence and re-enter.
COMMAND ERR	*17*	Illegal key sequence while in standard Remote Control.	Check key sequence and re-enter.
NONBLANK	*20*	Device failed the blank test.	Press START and try to program the device. It will abort if the nonblank bits prevent programming.
ILLEGAL BIT	*21*	Not possible to program the device due to already programmed locations of incorrect polarity.	Erase the device if possible or discard it.
PROGRAM FAIL	*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.
VERIFY FAIL 1	*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.
VERIFY FAIL 2	*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.
NO PROG PAK	*25*	A device-related operation was attempted without any programming module installed.	Install the appropriate programming module.
PROG PAK RST	*26*	The programming electronics will not start operation due to a reset condition.	Usually an overcurrent caused by an incorrectly inserted or bad device.
RAM EXCEEDED	*27*	There is insufficient RAM to program the device; the total allotment of RAM resident is less than the word limit of the device.	Program smaller parts or buy enough extended RAM. If enough RAM is installed, it may be faulty.
FRAME ERR	*41*	The serial interface detected a start bit but the stop bit was incorrectly positioned.	Check the baud rate and stop bit switches.
OVERRUN ERR	*42*	The serial interface received characters when the programmer was unable to service them.	Check the baud rate and stop bit switches.
FRME + OVR ERR	*43*	Combination of FRAME ERR 41 and OVERRUN ERR 42.	Check the baud rate and stop bit switches.
I/O TIMEOUT	*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.
FAULTY ACIA	*47*	ACIA chip may have failed.	Contact your local Data I/O Service Center.
I/O OVERRUN	*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/O VFY FAIL	*52*	The data from the serial port did not match the data in RAM.	

Table 3-7. Error Codes

DISPLAY		DESCRIPTION	CORRECTIVE ACTION
NO RAM	*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.
RAM BIT ERR	*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.
RAM WRITE ERR	*63*	The programmer is unable to write the intended data in RAM.	Failure of the associated RAM chip; replace the failed chip.
RAM DATA ERR	*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.
IRQ ERR	*66*	The IRQ line to the processor was held low for no apparent reason.	Ignore. If the error persists, service the programmer.
	67	Programmer received a non-hex character in optional Computer Remote Control.	
DATA LOCKED	*68*	Data locked via Select Function F3.	Use the password to release data.
PARITY ERR	*81*	The incoming data has incorrect parity.	Check the parity switch and try again.
SUMCHK ERR	*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.
INVALID DATA	*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, 81-86)	Check the connection of all units in the system, data format and data source, and then try again.
INVALID FORM	*90*	Non-existent I/O format is selected in optional Computer Remote Control.	Enter a legal format code.
I/O FORM ERR	*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/O FORM ERR	*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.
BAD REC TYPE	*93*	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/O FORM ERR	*94*	The record type was in error. (Intel-Intellec 8/MDS format only.)	Check the connection of all units in the system, data format, and data source, and then try again.
	*96**	Illegal center point for RAM shuffle.	Check parameters and reenter.
BLOCK MOVE ERR	*97*	Block Move was attempted outside RAM boundaries.	Redefine parameters.
DEV EXCEEDED	*98*	Programming data exceeded the last device address.	

* Remote Control only; will not occur during front panel operation, hence no front panel display.

SECTION 4
CALIBRATION

4.1 INTRODUCTION

Calibration of the 29A consists of checking and adjusting the power supplies. This will normally be performed as part of the calibration of a programming 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 module's O & M manuals list the instructions for calibrating the modules, and this manual tells how to carry out these instructions on the 29A. To calibrate a programming module, consult its O & M manual and use this section as a supplement. Figure 4-1 graphically demonstrates the calibration procedure.

4.2 POWER SUPPLY CALIBRATION

Power operation of programming modules depends on the programmer's power supplies. 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 programmer's Controller Board and on the Calibration Extender. The power supplies can be measured in either place.

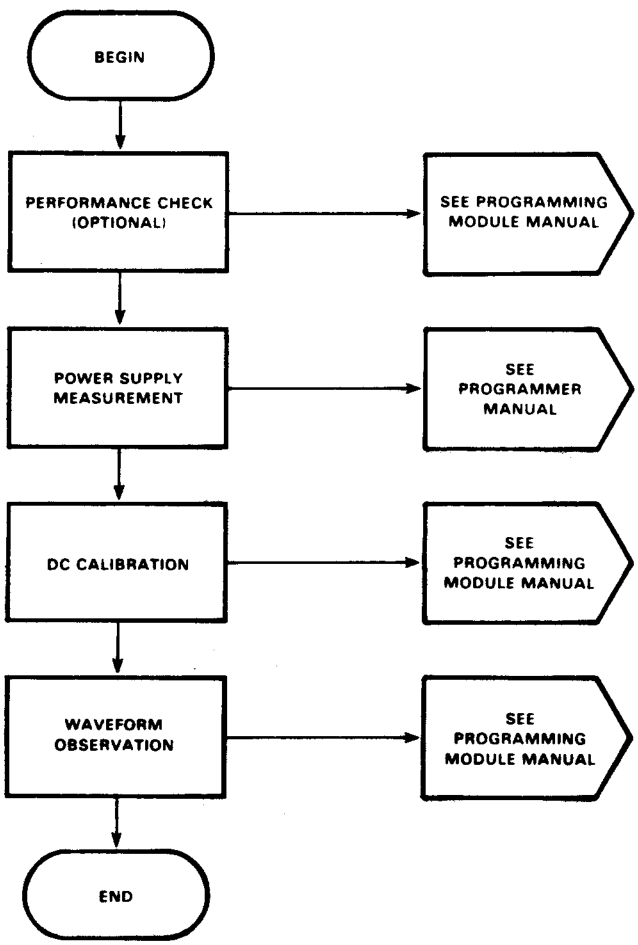


Figure 4-1. Calibration

Table 4-1. Calibration Voltage Ranges

POWER SUPPLY	VOLTAGE RANGE Min.	Nom.	Max.	TEST POINT	ADJUST
+5 V	5.05	5.10	5.15	TP1	R42
+24 V	23.50	24.00	24.50	**	R23
+48 V	49.50	49.70	49.80	**	R16
+ PROG V*	4.80	4.90	5.10	Below J7	*
-9 V	-9.50	-9.00	-8.50	**	R26
-5 V	-5.25	-5.00	-4.75	**	Non-adjustable
+12 V	11.40	12.00	12.60	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 ± 0.20 V.

** These test points are printed circuit feed-throughs to the left of J7.

4.2.1 REQUIRED EQUIPMENT

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

- Digital voltmeter (DVM), Fluke Model 8000A or equivalent
- Potentiometer adjustment tool (tweaker), or 1/8" flat blade screwdriver
- Jumper wire approximately 12 inches long
- IC removal tool or small screwdriver

4.2.2 SET-UP

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

1. Turn the power off. Remove the programming module and protective shield as described in Section 2.
2. Ground the DVM to TP2.
3. Ground TP3 to the programmer chassis with a jumper wire. Turn the power back on.
4. Measure the supplies at the test points shown in Figure 4-2. Results should be within the ranges listed in Table 4-1.

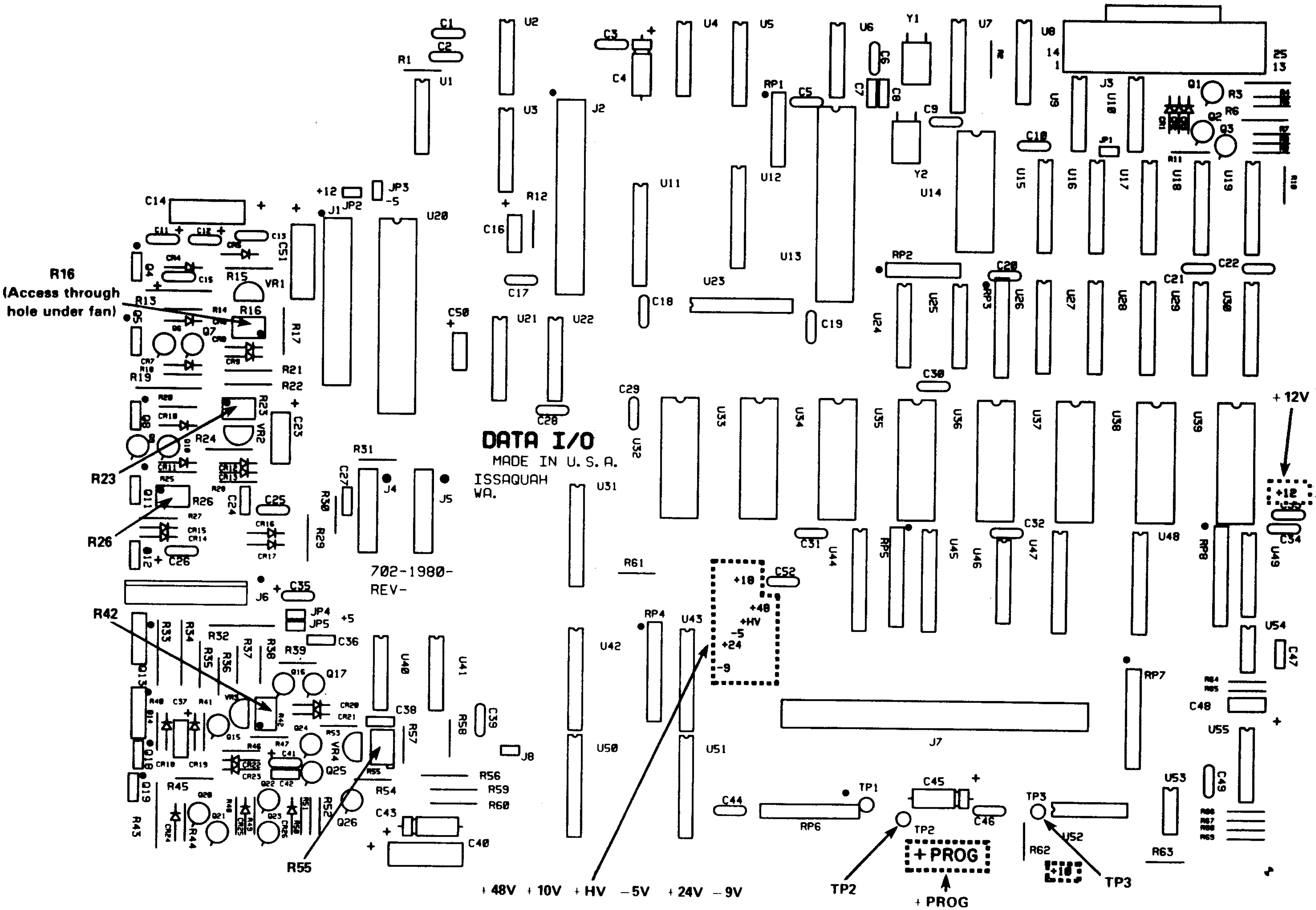


Figure 4-2. Power Supply Test and Adjustment Points

4.2.3 ADJUSTMENT PROCEDURES

If all voltages 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 power off and unplug the programmer. Remove any calibration equipment that has been installed.
2. Take off the top cover and remove the front panel of the programmer as shown in Figure 4-3. It is held in place with 4 screws, 1 in each corner and 2 more holding the center brace to the power supply assembly.
3. Disconnect the front panel cable.
4. Ground TP3.
5. Apply power.

CAUTION

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

6. Adjust the supplies as necessary, using the potentiometers cited in Table 4-1. Figure 4-2 shows their locations. If any supply can't be adjusted within the ranges of Table 4-1, refer to Section 6, Troubleshooting.

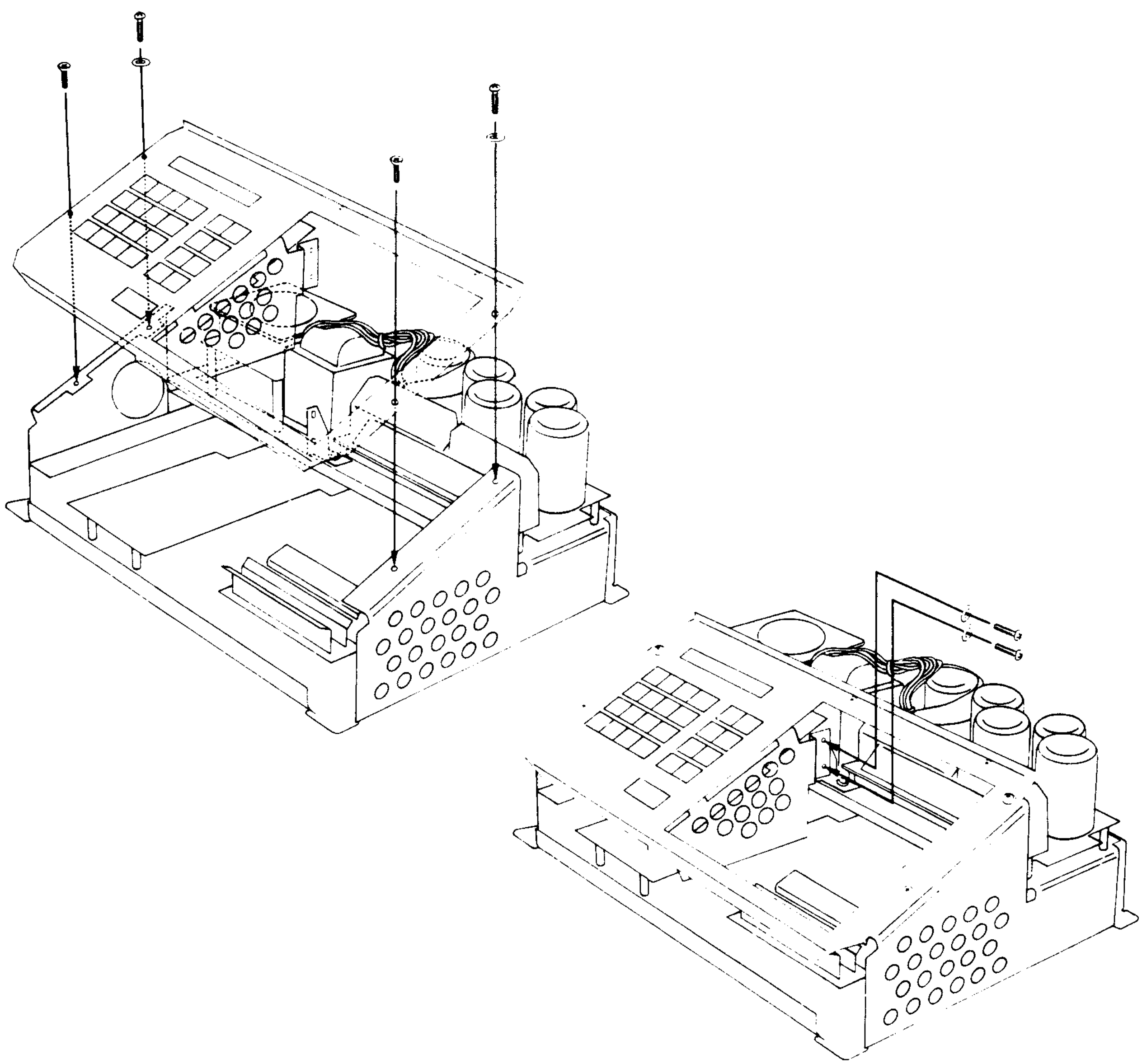


Figure 4-3. Front Panel Removal

4.3 THE PERFORMANCE CHECK

Some programming module manuals contain a "performance check" procedure. To do a performance check, follow the procedure in the programming module manual, executing the steps on the 29A as follows.

1. Set up the programmer and the programming module as shown in Figure 4-4, and turn the programmer power on.

CAUTION

Remove all devices from the programming module before putting the 29A in the Calibration Mode. DC voltages applied to the socket adapter during calibration will damage any device in a socket.

2. Enter the Calibration mode using Select Function C1.

   a. Press SELECT

   b. Enter C1

   c. Press START

3. The programmer is now ready for step 1 of the Measurement Chart. Follow the steps in order.

   - To increment one step, press START. To jump forward a number of steps, enter the step number desired and press START.
   - To decrement steps, press REVIEW.

   After making a measurement, increment to the next step.

4. If any voltages fall outside the specified range, perform a complete calibration.

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

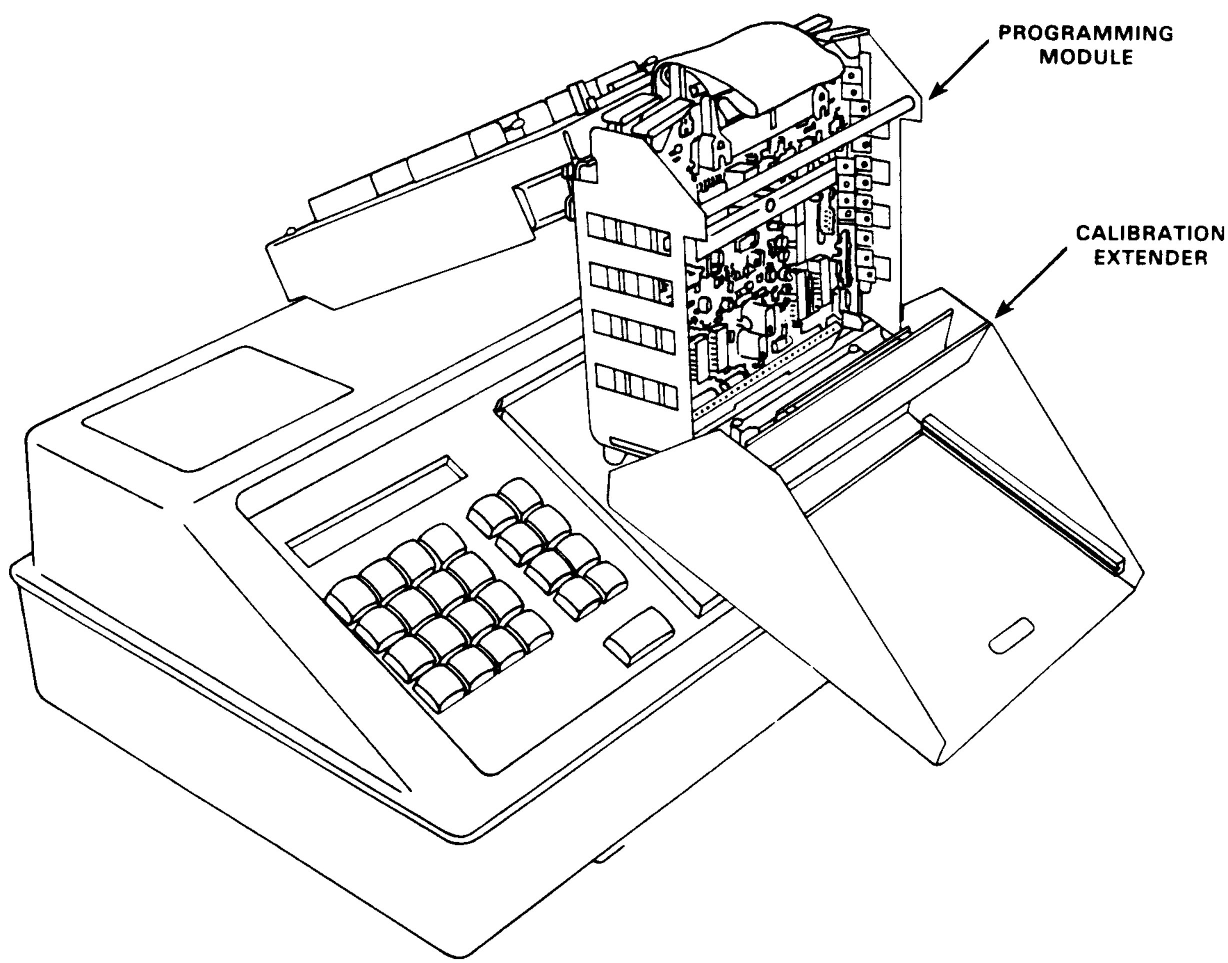


Figure 4-4. 29A with Programming Module Prepared for Calibration

4.4 CALIBRATION OF PROGRAMMING MODULES

Data I/O manufactures two types of programming modules for use with its programmers — expanded memory modules and standard modules. Calibration of standard programming modules requires the Data I/O Universal Calibrator and Calibration Extender. Expanded memory programming modules have additional ROM, providing software which eliminates the need for the Universal Calibrator. (The Calibration Extender is still necessary.)

This section contains two calibration procedures, one for expanded memory modules and one for standard modules. If the programming module manual cites the Universal Calibrator as required equipment, use the procedure for calibration of standard programming modules. If the Universal Calibrator is not required, use the procedure for calibration of expanded memory modules.

4.4.1 CALIBRATION OF EXPANDED MEMORY PROGRAMMING MODULES

The procedure for calibration of expanded memory programming modules is as follows:

1. Measure and adjust the 29A's power supplies as necessary.

2. Enter the calibration mode via Select Funtion C1.

   a. Press SELECT

   b. Enter C1

   c. Press START

3. Refer to the instruction manual for the programming module under calibration. Section 4, Calibration, will contain required equipment, calibration instructions, and Measurement Charts. Step numbers on the programmer display correspond to those on the Measurement Charts.

   - 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 order and make adjustments as necessary.

4. For waveform observation, follow the directions in the programming module manual. Use Select Function A2 to fill RAM with the correct programming data. Consult the appropriate programming module manual for the proper RAM data.

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

NOTE

Always exit the calibration mode before attempting to program devices.

4.4.2 CALIBRATION OF STANDARD PROGRAMMING MODULES

Calibration of standard programming modules is as follows:

1. Measure and adjust the programmer power supplies as necessary.

2. Put the programmer in 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.

3. Refer to the instruction manual for the programming module. Section 4 contains required equipment, calibration instructions, and the Calibration charts.

   - 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.

   The programmer displays and Calibration Chart instructions are given in Table 4-2.

   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 O & M manual.

4. To exit calibration, 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) 0000DFF R00 L F — 0000: Address; DFF: DO Data (Part Data); R00: DI Data (RAM Data); L: Start Line, Low/High (L or H); F: Forward/Reverse Line (F or R)	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.

SECTION 5
MAINTENANCE

5.1 INTRODUCTION

The primary maintenance requirement is periodic cleaning of the fan filter and regular inspection of the machine's interior. To inspect the interior, remove the cover as described in Section 5.2.

CAUTION

Avoid operating the programmer with the cover removed. The cover serves to direct airflow for cooling as well as protect the unit against dust and damage.

5.2 COVER REMOVAL

First remove any programming module. Turn the power off and disconnect the power cord. Turn the programmer upside down and remove the 4 screws at the corners of the base. Then turn the programmer right side up and remove the cover.

5.3 CLEANING

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

CAUTION

Do not use a caustic or abrasive agents; these will damage the Model 29A.

Clean the fan filter (located on top of the unit) every 3 months with normal usage, and up to twice a month with heavy usage. Remove the filter and clean it in running water to rinse out accumulated dust. Dry it thoroughly before reinstalling. Press it back into its recess, first one side and then the other.

5.4 INSPECTION

Periodic inspection can be a hedge against malfunction. A good time to schedule inspection is before calibration. Check cable connections, card seating, mounting of socketed components, etc., for obvious mechanical problems.

Particular care is required if heat-damaged components are found. It is important to find and correct the cause of overheating in order to prevent recurrence of the damage.

SECTION 6
TROUBLESHOOTING

6.1 INTRODUCTION

The following troubleshooting information is an aid to understanding malfunctions and locating hardware failures. Section 6.2 discusses the procedures for establishing the type of trouble in the unit and the steps necessary for further servicing. Section 6.3 directs the service technician to the portion of the circuitry implicated when the machine displays an error. 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 failure.

Additional useful information will be found in the Circuit Descriptions, Section 7, and the Schematics in Appendix E.

6.2 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 6-1 shows the overall troubleshooting procedure.

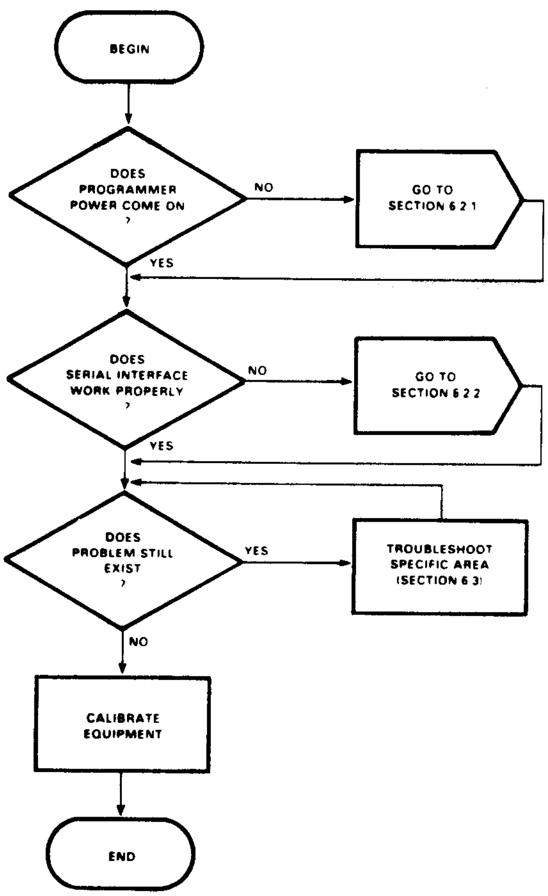


Figure 6-1. Troubleshooting

6.2.1 PROGRAMMER DOES NOT OPERATE OR OPERATES ERRATICALLY

1. Check that the AC power cord is firmly plugged in and the power switch is on.

2. Check the AC power selection against the line voltage. Refer to Figure 6-2. The voltage at which the programmer will operate is shown on the voltage selector card. For proper operation, the line voltage must be within +5% or −10% of the voltage shown on the card.

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 the module (or a clip lead).

4. Remove any cables attached to the serial interface.

5. Check the power supplies according for the calibration procedure in paragraph 4.2. If this reveals a problem, refer to paragraph 6.3.1.

6. Check the installation of all hardware. Check the orientation and connections of all cables as well as the seating of the PC boards. Check all jumpers also.

7. Check the Controller as described in paragraph 6.3.2.

8. If steps 1 through 7 do not reveal the problem, contact your Data I/O Service Center.

6.2.2 SERIAL I/O FAILURES

1. Make sure the programmer and other equipment to which it's attached are set for the same parity, baud rate, and number of stop bits.

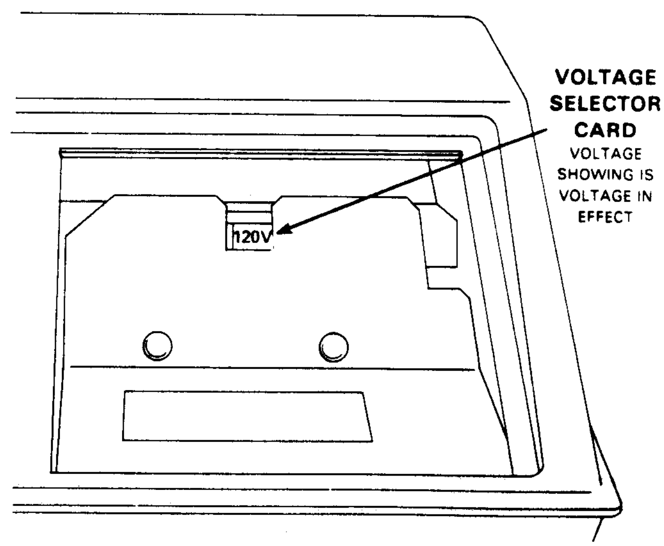


Figure 6-2. Voltage Selector Card

2. Check the operation of the peripheral equipment according to the manufacturer's procedures.

3. Troubleshoot the serial interface circuitry (paragraph 6.3.3).

6.3 TROUBLESHOOTING SPECIFIC AREAS

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

6.3.1 POWER SUPPLIES

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.

If one of the power supplies is at the wrong potential and cannot be adjusted, refer to Table 6-1 to check the circuitry that may be causing the problem. Also check other associated components.

To replace any components associated with the +5 V, −5 V, or +12 V supplies, first remove jumpers JP1 through JP5 until the supplies fall into range. Then replace the jumpers and calibrate the programmer.

6.3.2 CONTROLLER

1. Visually 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 $\overline{V\bullet\emptyset2}$ 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.

6.3.3 SERIAL INTERFACE CIRCUITRY

To check the serial interface circuitry, proceed as follows:

1. With an oscilloscope, observe U14, pin 3 on the Controller Card. The frequency of the signal at pin 3 should be 16 times the selected baud rate. (Baud rate = 1/time divided by 16).

EXAMPLE: At 110 baud, the observed pulse period should be 0.57 ms.

$$\text{Baud Rate} = (1/t) \div 16 = \frac{1 \div (5.7 \times 10^{4}s)}{16} = (1754) \div 16 = 109.6 \text{ baud (110 baud)}$$

A failure in step 1 indicates a problem with U7 and associated circuitry.

2. a. Initiate a Copy from the data RAM to the serial port.

   b. Check the voltages at U14, pins 24 and 23. Both should read 0 V to 0.5 V.

   c. Press START.

   d. Use the oscilloscope to observe pin 2 of the serial interface. The 29A should be transmitting data at the selected baud rate.

A failure in step 2 (a through d) indicates a failure in the ACIA (U14) or the drivers (U9 or U10).

Table 6-1. Power Supply Voltages

SUPPLY	PIN ON UNIVERSAL CALIBRATOR OR CALIBRATION EXTENDER	VOLTAGES Min.	Nom.	Max.	ASSOCIATED CIRCUITRY 702-1981	702-1980
+5†	+5	5.05	5.10	5.15	BR1, F2	CR21, Q13, Q14
+24*	+24	23.50	24.00	24.50	CR1, CR2, F1, F3	CR5, Q8
+48*	+48	49.50	49.70	49.80	CR3, CR4, F4, F5	CR16, Q5
−9	−9	−9.50	−9.00	−8.50	CR5, CR6, CR7, CR8, F6, F7	CR13, Q11
−5†	JP3 on 702-1980	−5.25	−5.00	−4.75	CR5, CR6, CR7, CR8, F6, F7	CR14, Q12
+PROG V*	+PROG	4.80	5.00	5.10	CR1, CR2, F1, F3	CR24, Q19
+12†	JP2 on 702-1980	11.40	12.00	12.60	CR5, CR6, CR7, CR8, F6, F7	CR4, Q4

* 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.

SECTION 7
CIRCUIT DESCRIPTION

7.1 INTRODUCTION

This section describes the 29A Universal Programmer's circuitry. Included are general architecture, address map, components and assembly cabling.

7.2 ARCHITECTURE

The 29A is a microprocessor-based system using bus architecture. It is designed around a 6802 microprocessor located on the Controller Board. Figure 7-1 is the system block diagram.

The circuitry of the programmer's main components is described in sections 7.4.1 through 7.4.4. Block diagrams are presented, and schematics for each are available in Appendix D.

7.2.1 THE BUS

The Bus consists of a 16-bit address bus, 8-bit data bus, power supply lines, and several control lines. These are detailed in Table 7-1. 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 7-2, and the timing of a read cycle is shown in Figure 7-3.

The buffered bus is available at the programming module interface (J7) and the option port (J2).

Table 7-1. Bus (at J2)

PIN	FUNCTION	PIN	FUNCTION
1	$\overline{\text{V•Ø2}}$	21	R/$\overline{\text{W}}$
2	$\overline{\text{HALT}}$	22	$\overline{\text{R}}$
3	$\overline{\text{NMI}}$	23	$\overline{\text{IRQ}}$
4	$\overline{\text{TOR}}$	24	$\overline{\text{EN}}$
5	DØ	25	D1
6	D2	26	D3
7	D4	27	D5
8	D6	28	D7
9	A1	29	AØ
10	A3	30	A2
11	A5	31	A4
12	A7	32	A6
13	not used	33	A15
14	+5	34	A14
15	+5	35	A13
16	+5	36	A12
17	GND	37	A11
18	GND	38	A10
19	GND	39	A9
20	GND	40	A8

7.2.2 ADDRESS MAP

The address map of Table 7-2 shows the location in hexadecimal of each decoded function of the programmer.

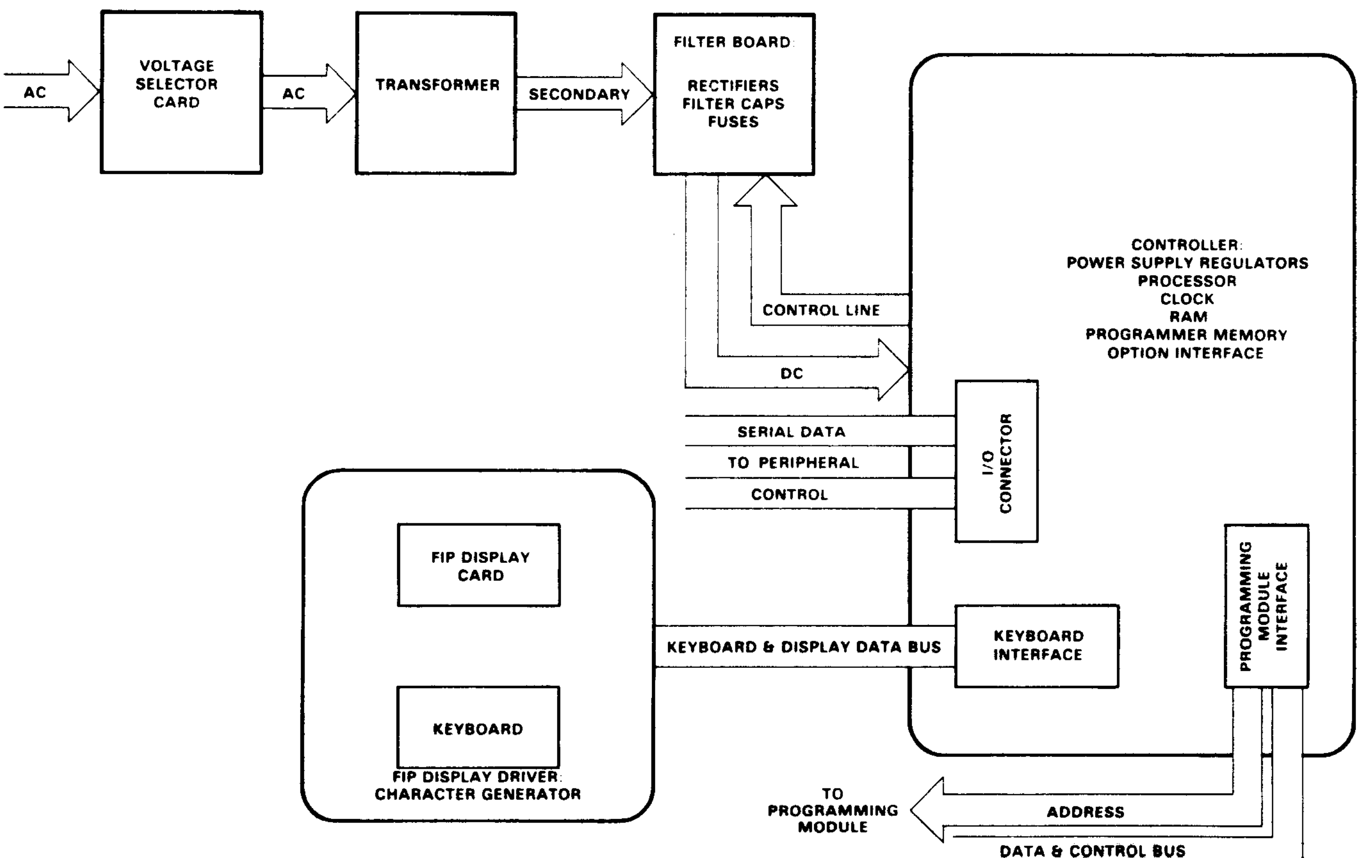


Figure 7-1. System Block Diagram

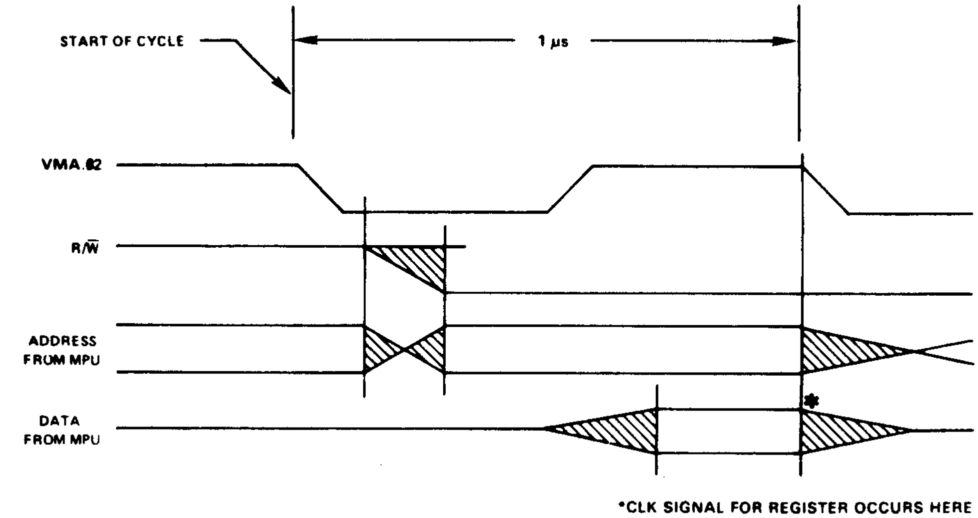


Figure 7-2. Write Timing

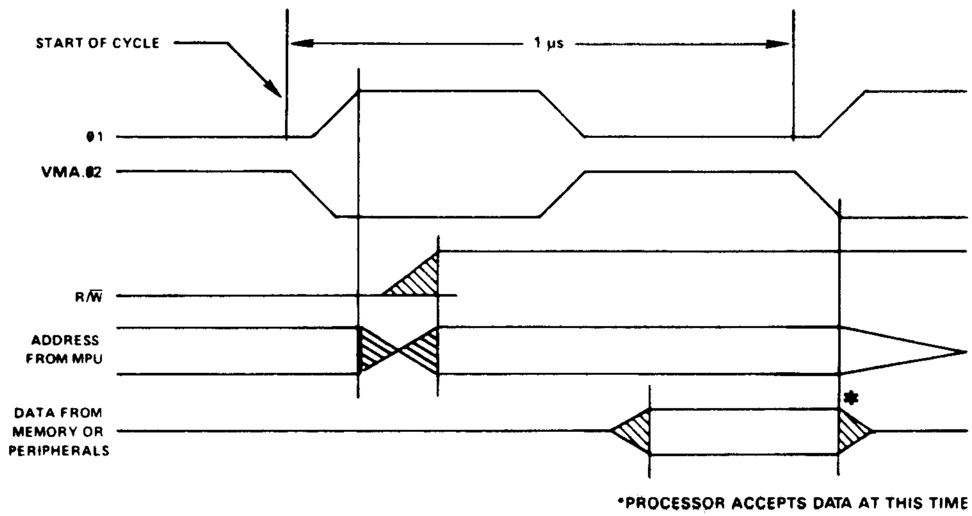


Figure 7-3. Read Timing

Table 7-2. Address Map

ADDRESS RANGE	FUNCTION	ADDRESS RANGE	FUNCTION
0000-03FF	Internal scratch RAM (Controller Board, U15 and U26)	E200—E2FF	I/O area
0400—1FFF	Unassigned	E200	Address register, high order (U51, Controller Board)
2000—3FFF	Data RAM 8K (Controller Board, U16-U19, U27-U30)	E201	Address register, low order (U42, Controller Board)
4000—5FFF	Data RAM 8K maximum (Extended Memory Board, U1-U12, U15-U18, U20-U27)	E202	Data gate/data register (U44, Controller Board)
6000—9FFF	Expanded-memory programming modules	E203	Control register/status gates (U45 and U46, Controller Board)
A0000—DFFF	Program Memory 16K, (U32-U39, Controller Board)	E204 and E205	KBD/display (U20, Controller Board)
E000—E0FF	Interface control register (U31, Controller Board)	E206 and E207	Serial I/O (Controller Board, U14)
E100—E1FF	Switch gates (U52, Controller Board)	F800—FFFF	Restart vector, 2K

7.3 COMPONENT LAYOUT

Figure 7-4 shows the cabling between assemblies, along with associated connector cables and part numbers.

The component layout of the 29A is shown in Figure 7-5.

7.4 MAIN COMPONENTS

The following paragraphs describe the circuitry of each board in the programmer. References are made to the individual block diagrams. The schematic for each board is located in the Appendix D.

7.4.1 POWER SUPPLIES

Figure 7-6 is a block diagram of the power supply. Each will be discussed separately.

The **AC power switch** has a built-in overload circuit breaker reset by turning the equipment off, pausing, and turning it on again.

The **power transformer** has multiple primary windings for various input voltages. The secondary develops appropriate voltages for the rectifiers and filters.

Four **rectifier and capacitor filter networks** on the Filter Card provide the DC voltages for the various regulators. See Figure 7-7. PNP transistor Q1 is connected to the center tap of the high-voltage winding of the transformer. When Q1 is turned off by the high voltage shutdown control, no current can flow to the +40 and +HV unregulated voltage outputs.

The voltage regulator blocks are shown in Figure 7-8. Each block is discussed below.

The **5-volt regulator** consists of a TL430 shunt regulator (VR3) driving an MJE 240 (Q18) which, in turn, drives 2 TIP 35A pass transistors (Q13 and Q14). Feedback

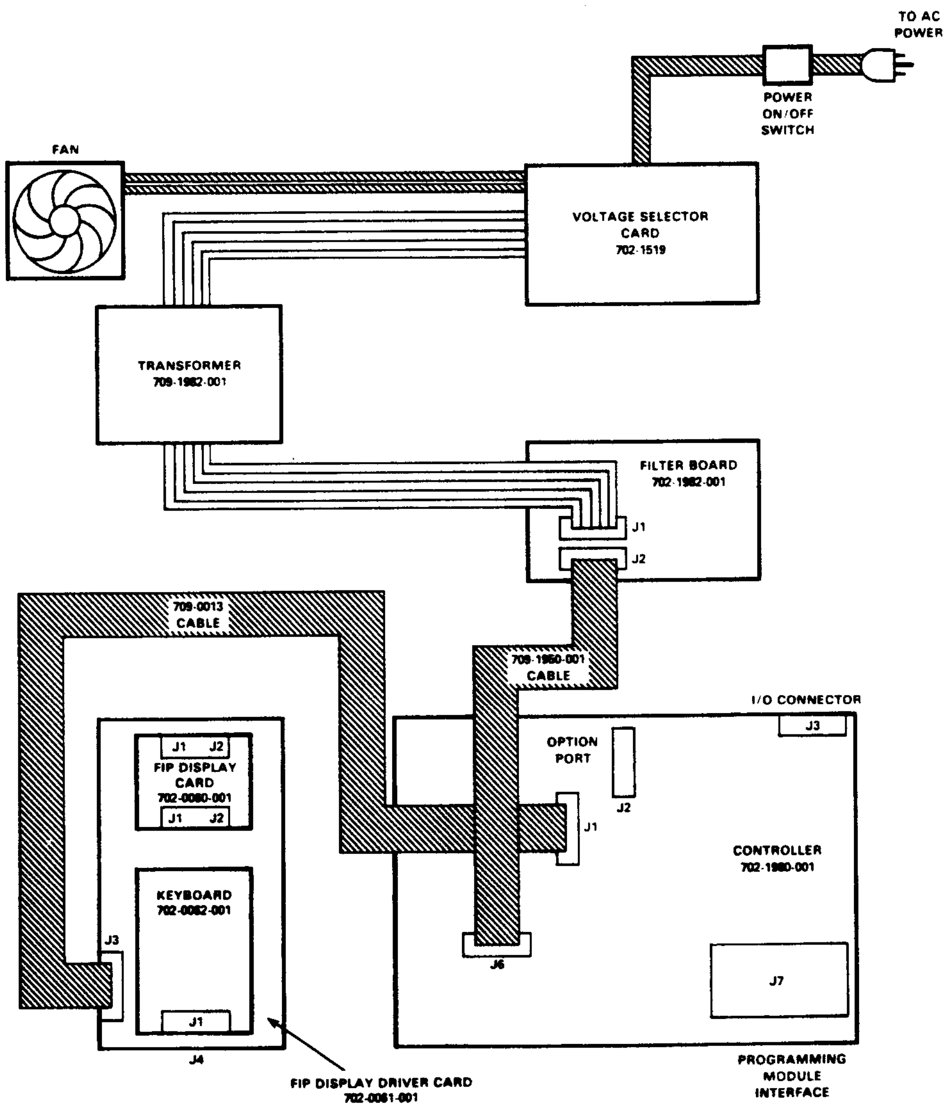


Figure 7-4. Interconnection Diagram

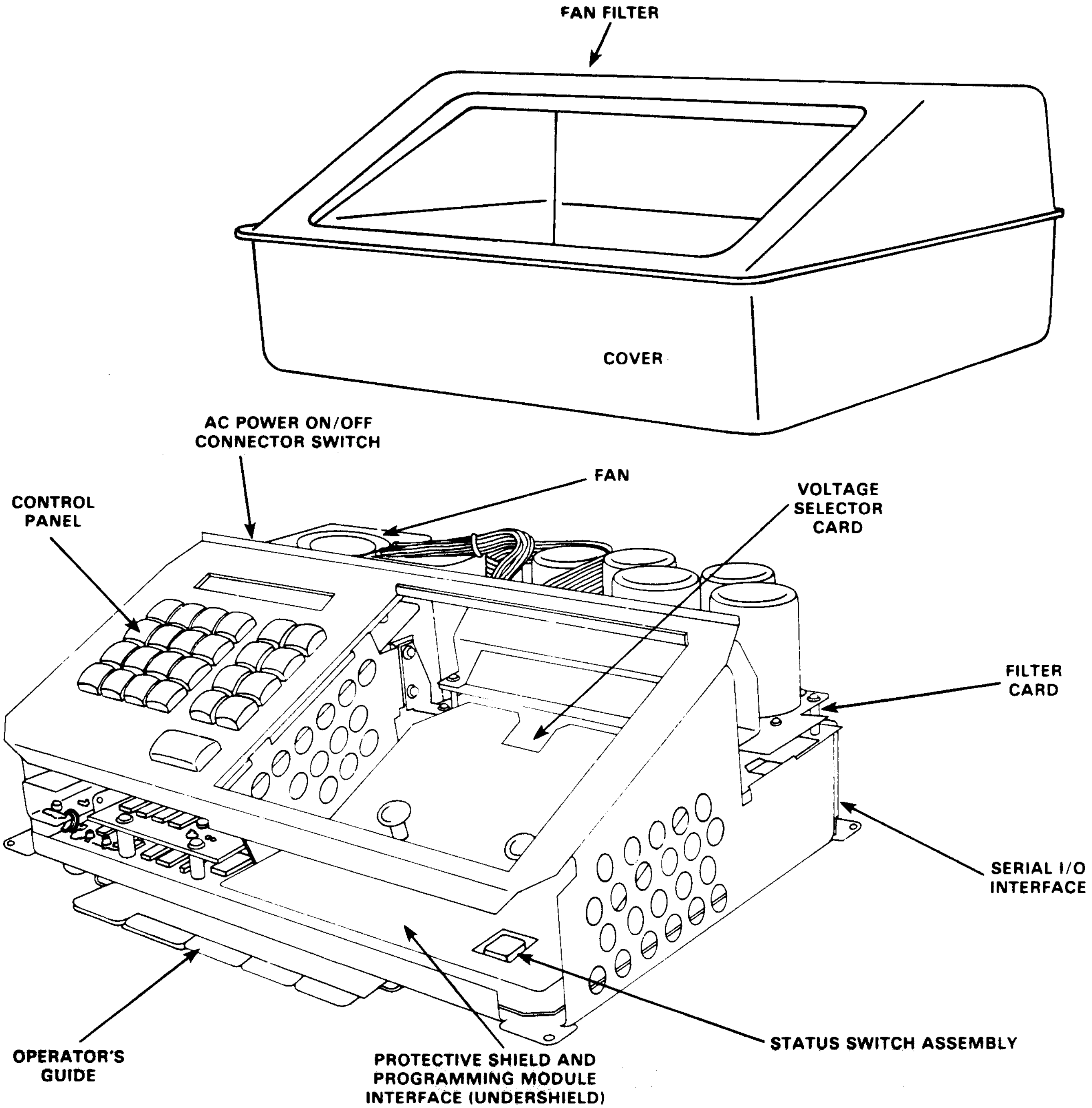


Figure 7-5. Component Layout

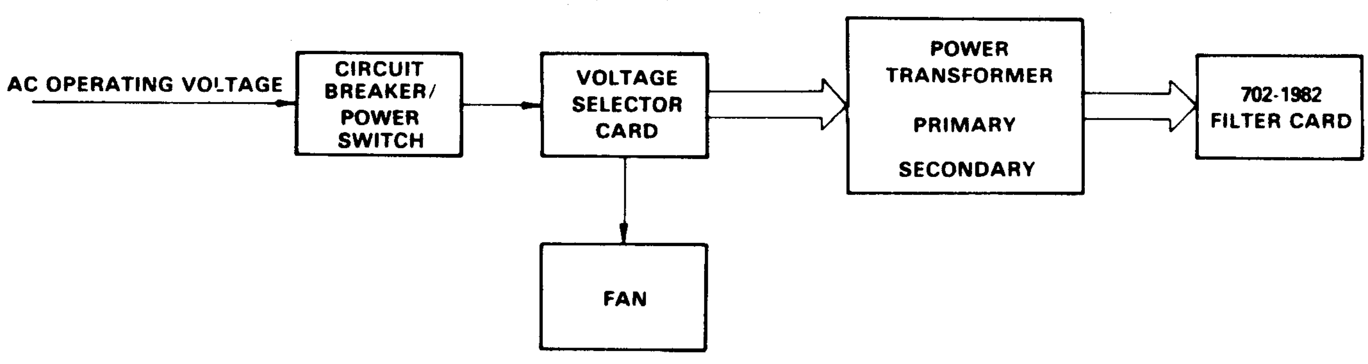


Figure 7-6. Block Diagram, Power Supply

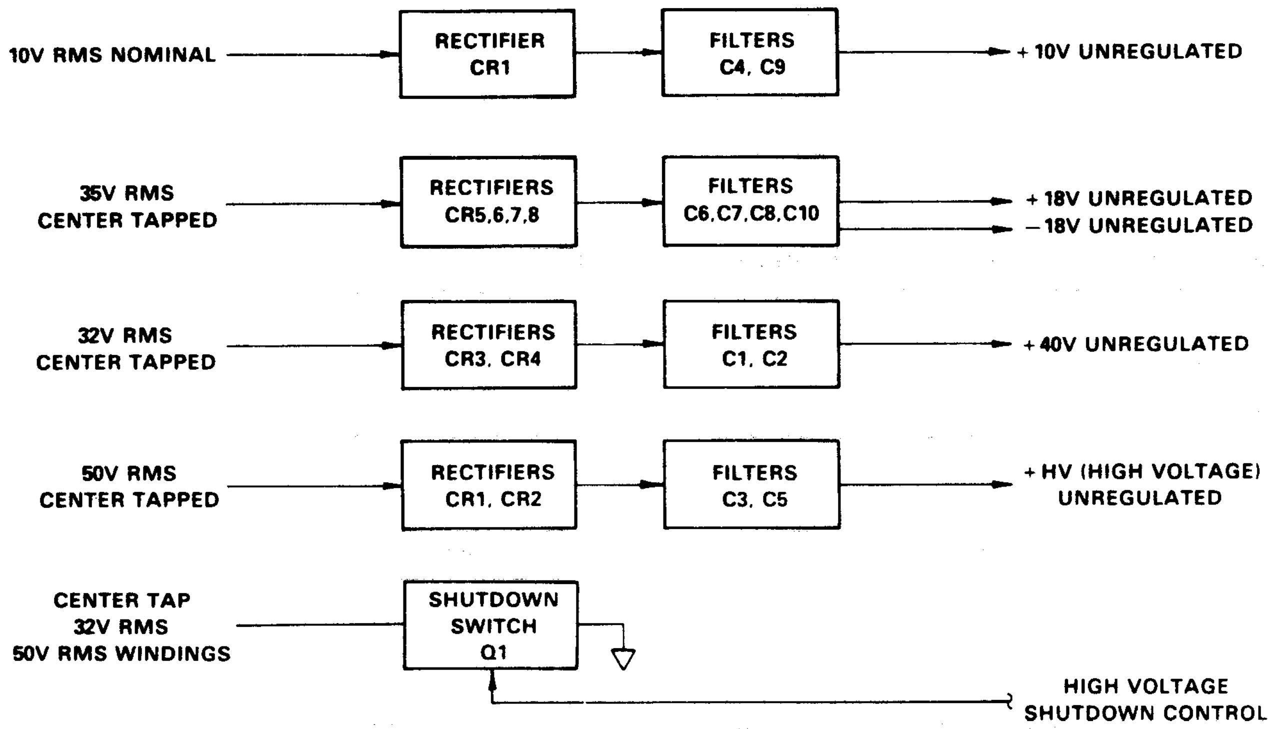


Figure 7-7. Block Diagram, Filter Card

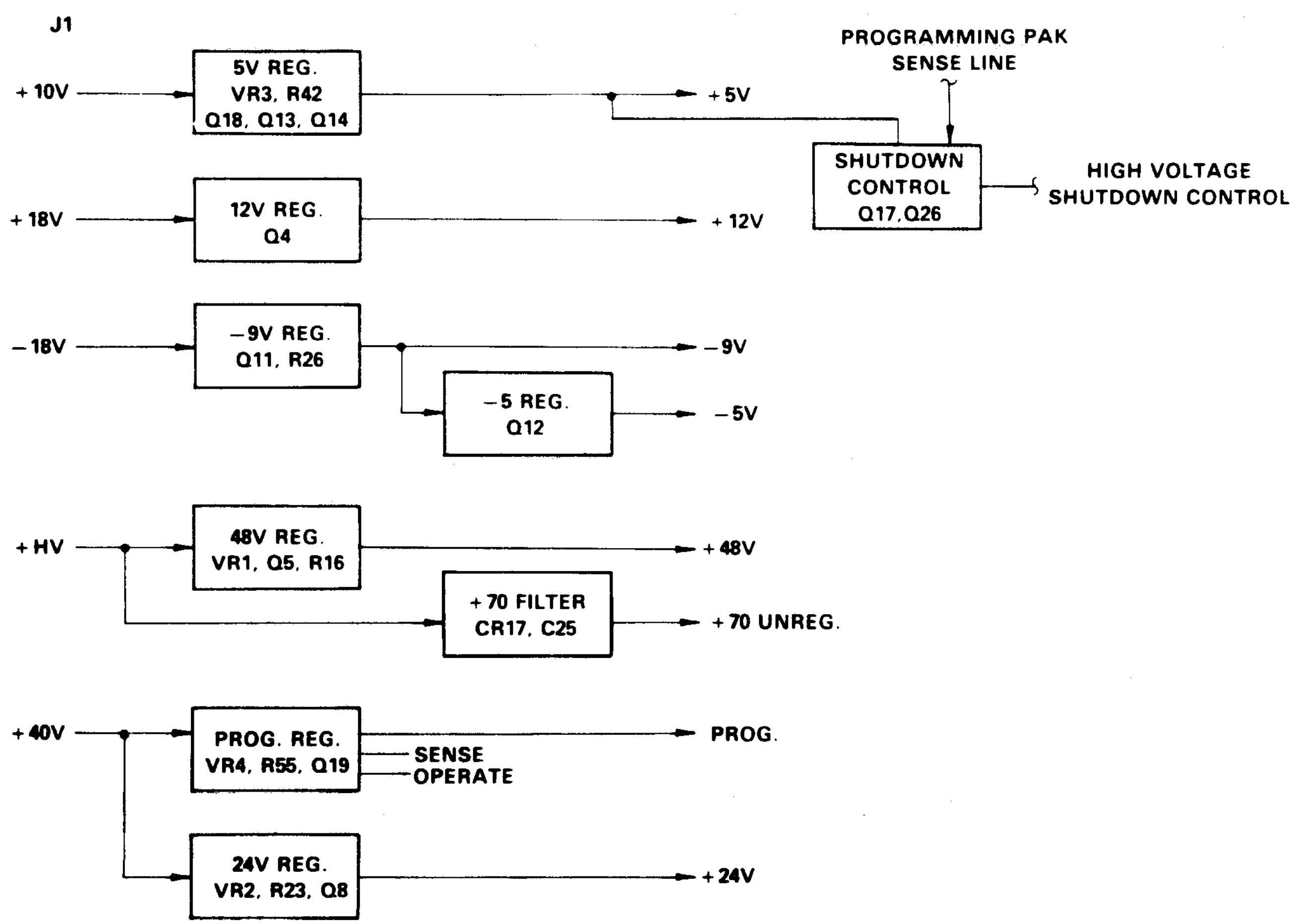


Figure 7-8. Block Diagram, Voltage Regulator

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 reference, 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 overvoltage 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.

The **programmable supply** 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 5 V reference for the plus input of the differential pair (Q24). Two feedback nodes are connected to the Sense and Operate lines. By connecting various resistors to these lines, the output voltage can be "programmed" to any level between 5 and 40 V. Foldback current limiting is provided in the same manner as in the 5 V supply. The **+12 V, −9 V, and −5 V** supplies use standard monolithic regulators.

The shutdown control signals the card to turn off the +HV and +40 V unregulated supplies when a programming module is removed. This in turn shuts down the +48, +24, +PROG and +70 V 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 overvoltage condition on the supply lines. An overvoltage condition could occur from failure of power supply components.

7.4.2 CONTROLLER

The Controller is shown in block diagram form in Figure 7-9. Each block will be discussed in this section.

The **processor** drives the bus (paragraph 7.2.1) through the address buffers and data buffer. Control signals are also developed by the processor and sent through buffers to the bus.

The **decode PROMs** monitor the address bus and R/$\overline{W}$ to select the various gates, registers and other devices connected to the data bus. Refer to the memory map of Table 7-2. $\overline{V\bullet\emptyset2}$ is connected to the chip select of each decode PROM to provide the correct timing for writing to registers or reading gates or memory.

The **programming module interface** is provided by the address registers, status gates, data register, data gates, and control register. These gates and registers interface the 29A to Data I/O programming modules.

When a programming module is removed, the processor is held reset by a high on 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

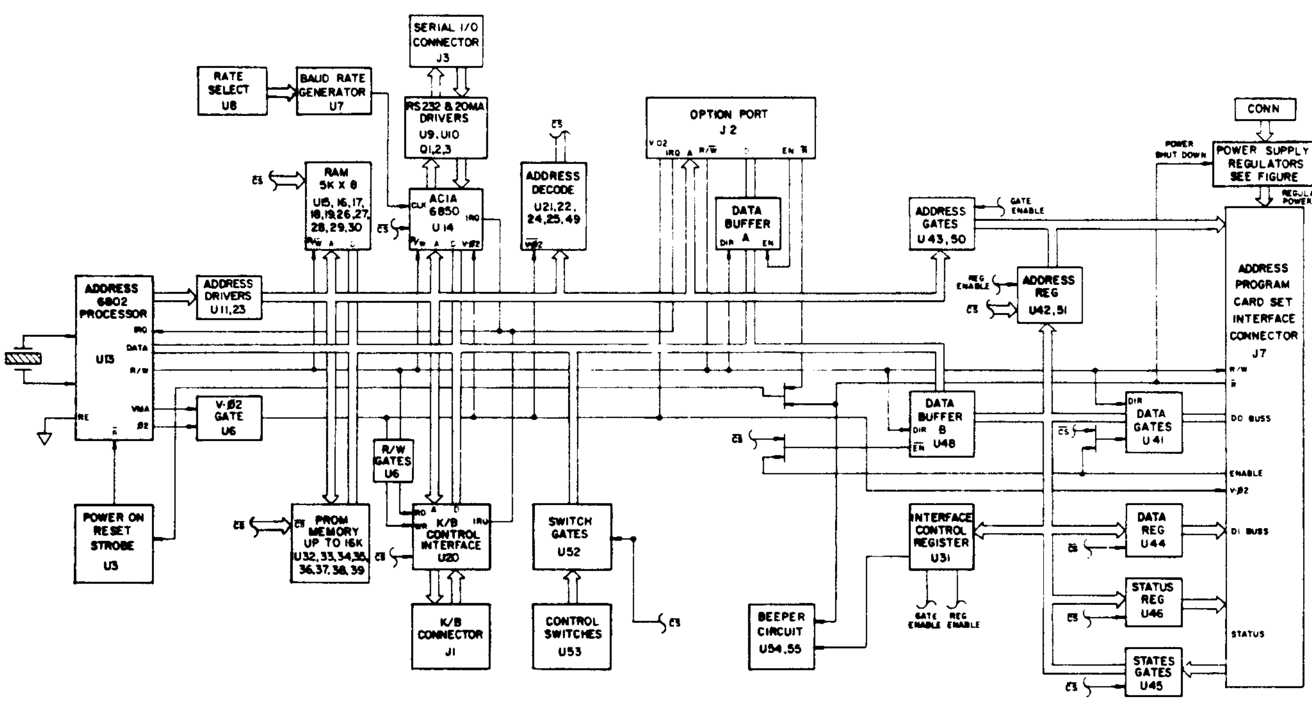


Figure 7-9. Block Diagram, Controller

interface can be gained with software control of the interface control register. The programming module interface can be set up so that the processor 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 7-2) and communicates with the processor using the Interrupt Request Line ($\overline{IRQ}$).

The **keyboard/display interface** is provided by an 8279 (U20). This device contains a small RAM and a first-in-first-out (FIFO) register along with scanning control circuitry. The 8279 is configured for N-key rollover. The 8279 occupies 2 address locations (Table 7-2) and uses $\overline{IRQ}$ to interface with the processor.

The **on-board program memory** occupies up to 16K bytes of PROM (U32-39) decoded in 2K segments.

Temporary data storage on the Controller consists of 4K bytes of RAM (U16-19, U27-30) decoded in 1K segments, and 1K bytes of scratch RAM. An additional 4K bytes are available on the Expanded Memory Board. See paragraph 7.4.4.

The **option port** (J2) provides access to the buffered processor bus for service and the addition of optional RAM. The buffers on this port are enabled at the appropriate address by an external enable line.

7.4.3 CONTROL PANEL

The control panel consists of a hex keyboard, 4 mode keys, 3 source/destination keys, 2 control keys, and a 16-character, 14-segment display. See Figure 7-10.

Display operation starts with 4 scan lines, RS0-RS3, from the keyboard interface chip 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 (6 bits) associated with each scan count passes to a character generator PROM (U11) whose 4-bit output is demultiplexed into 16 lines by a decoder (U12) and 4 quad flip-flops (U7 through U10). These outputs are used by the display anode drivers (U5 and U6).

Keyboard operation starts with 3 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 5 return lines.

7.4.4 EXPANDED RAM

The standard 29A includes 4K bytes of RAM on the Controller Board, but is capable of addressing 16K of RAM. RAM in excess of 4K bytes resides on the optional Extended Memory Board. This board has the capacity for 24 1K x 4 RAMs for a total memory capacity of 12K x 8 bits of memory. Decoding is accomplished on the card by a 74LS151 driving two 74LS138 devices. The 74LS138 outputs are enabled by $\overline{V \bullet \phi 2}$ to their active-low state.

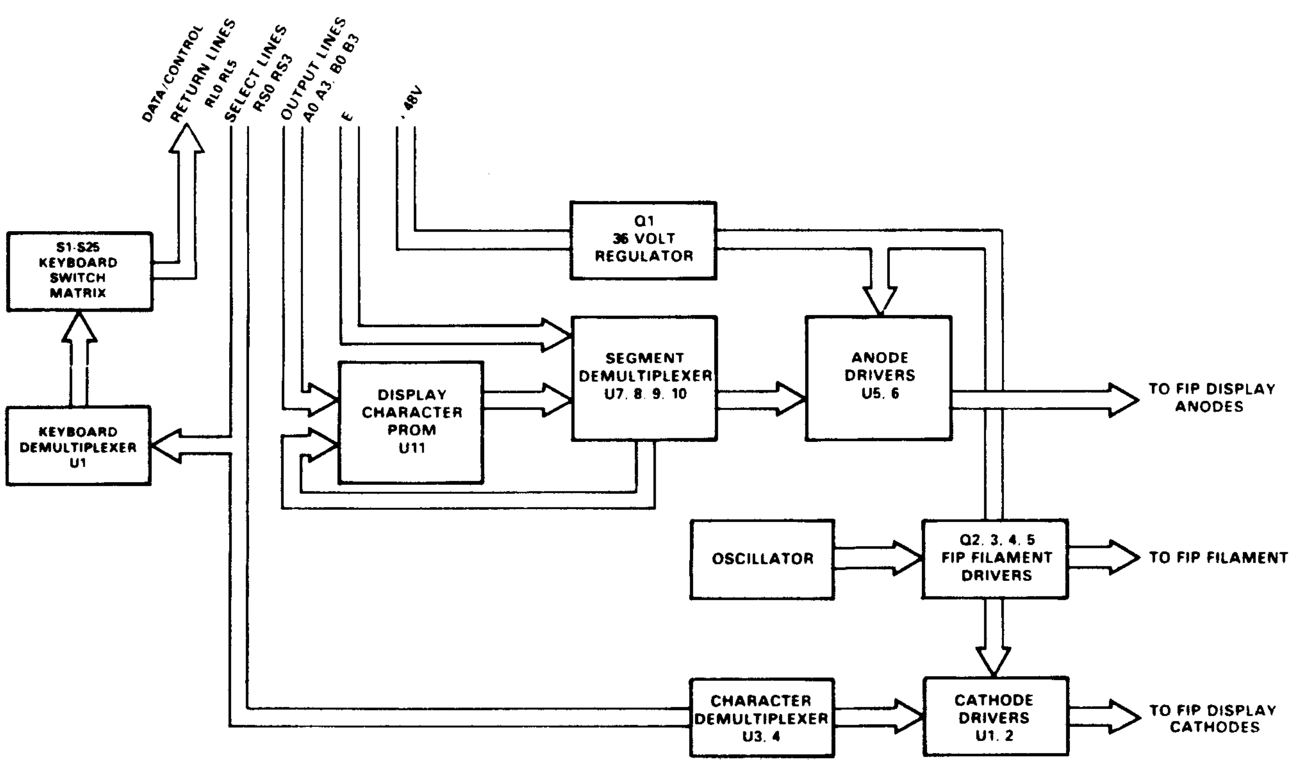


Figure 7-10. Block Diagram, Keyboard and FIP Display Driver

APPENDIX A
DATA TRANSLATION FORMATS

A.1 INTRODUCTION

This appendix defines the data translation formats available for the 29A. The 29A 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 29A'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 > PORvZZZZZZZZ
  (ZZZZZZZZ: Address Offset)
- Copy RAM to Port
  FORMAT: Motorola Exormax (Format #87)
  DISPLAY: RAM > PORvZZZZZZ
  (ZZZZZZ: Address Offset)
- Copy Port to RAM
  FORMAT: HP 64000 Absolute (Format #89)
  DISPLAY: POvZZZZZZZZ/YYYY
  (ZZZZZZZZ: Address Offset; YYYY: Block Size)
- Copy Port to RAM
  FORMAT: Motorola Exormax (Format #87)
  DISPLAY: POv--ZZZZZZ/YYYY
  (ZZZZZZ: Address Offset; YYYY: Block Size)
- Pressing REVIEW to review I/O parameters.
  FORMAT: HP 64000 Absolute* and Motorola Exormax
  DISPLAY: XXXX/YYYY > ZZZZZZ
  (XXXX: RAM Begin; YYYY: Block Size; ZZZZZZ: Address Offset)

A.2 DATA VERIFICATION

For data verification the 29A 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.

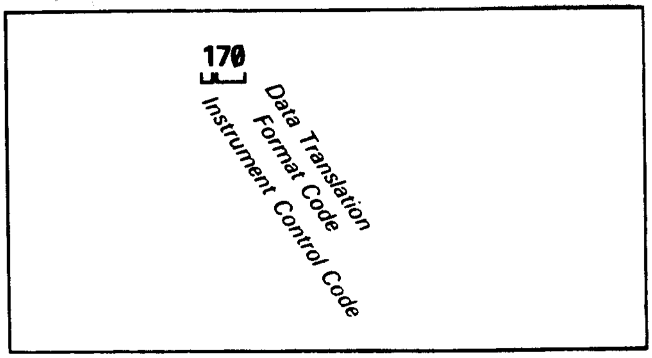


Figure A-1. Formatting the Instrument Control Code and Data Translation Format Code

A.3 CODES

Each format is assigned a 2-digit data translation format code which the operator enters into the programmer to transfer data in that format. 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.

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

A.4 TRANSLATION FORMATS

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

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	Send "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.

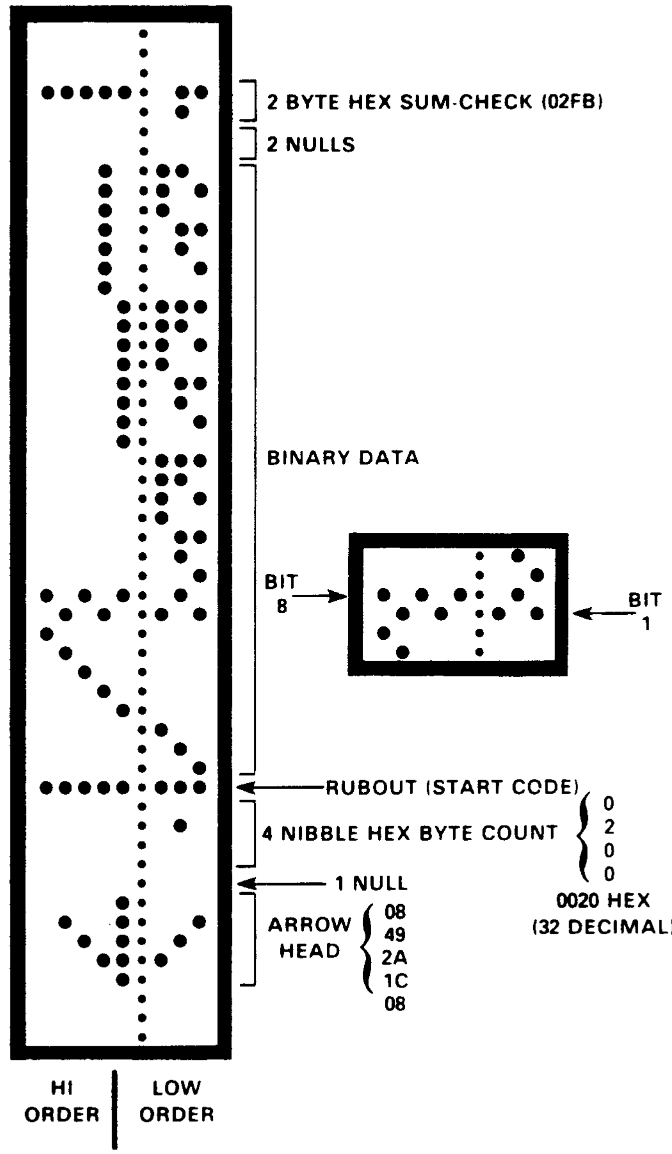


Figure A-2. Input or Output Binary Tape

A.4.1 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, a nonprintable rubout in even parity, follows the byte count. The end of data is signalled by 2 nulls and a 2-byte sum-check of the data field. Refer to Figure A-2.

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 the COPY, VERIFY, SELECT, or EDIT keys.

A.4.2 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.4.3 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, and 1's and 0's, respectively. See Figure A-3. The ASCII Binary formats do not have addresses.

Figure A-3 shows four data byte 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. 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 in "no parity"), and the end code is a nonprintable ETX, control C (or a hex 03 in "no parity").

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-B1ØF, 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 the COPY, VERIFY, SELECT or EDIT key.

Data is output in 4-byte lines with a space between bytes. The transmission is preceded and followed by 50 null characters.

A.4.4 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.4.5 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.

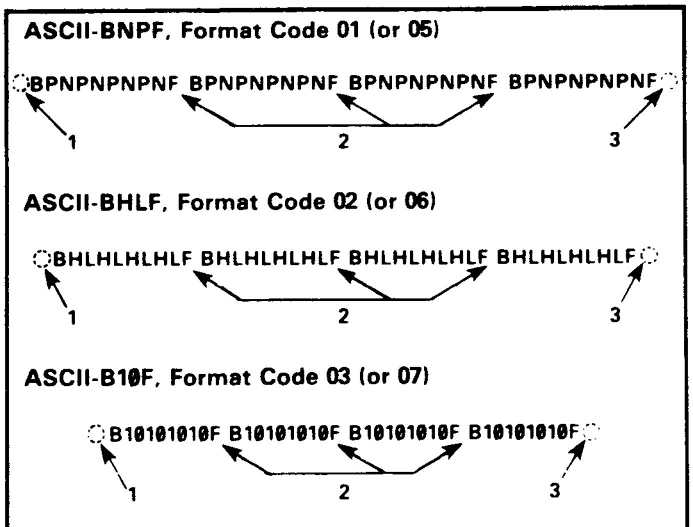
Figure A-4 shows 2 data bytes coded in the Spectrum format. Bytes are sandwiched between the space and carriage-return characters and are normally separated by line feeds. The start code is a nonprintable STX, control B (or hex 02 in "no parity"), and the end code is a nonprintable ETX, control C (or hex 03 in "no parity").

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.

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 the COPY, VERIFY, SELECT, or EDIT keys.

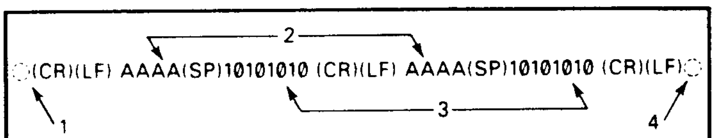
Data output to a printer will have one address and one



NOTES

1. **Start code is a nonprintable STX – CTRL B (start code is optional).**
2. **Characters such as spaces, carriage returns and line feeds may appear between bytes.**
3. **End code is a nonprintable ETX – CTRL C.**
4. **Data can also be expressed in 4-bit words**

Figure A-3. ASCII Binary Formats



NOTES

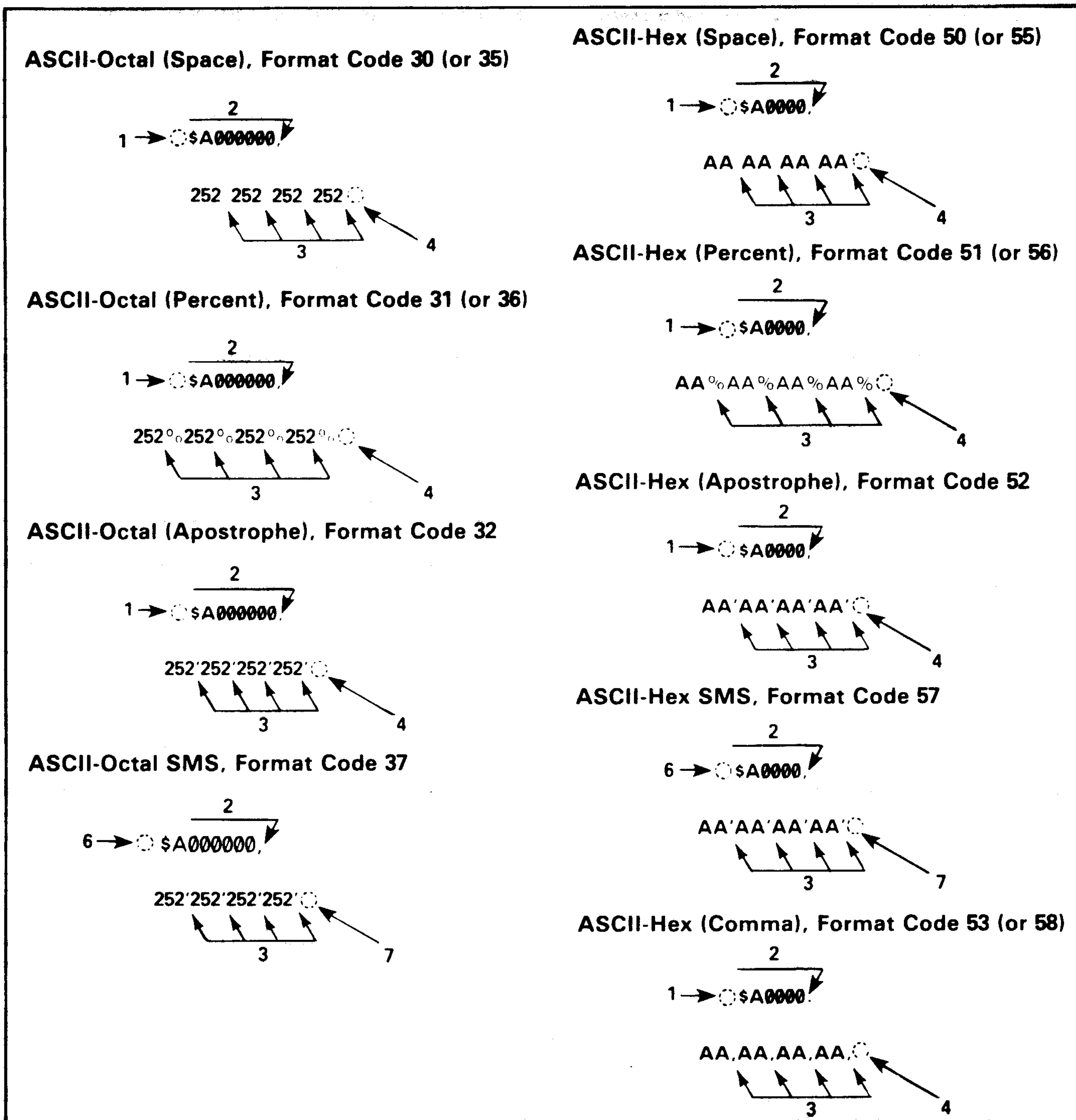
1. **Start code is a nonprintable STX (start code is optional).**
2. **Address code is 4 hex digits.**
3. **4 or 8 data bits appear between space and carriage return.**
4. **End code is a nonprintable ETX.**

Figure A-4. Spectrum Format

byte of data on each line. The programmer first sends an STX (optionally), then the data, and finally an ETX. The transmission is preceded and followed by 50 null characters.

A.4.6 ASCII OCTAL & 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 illustrate 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



NOTES:

1. Start code is nonprintable STX – CTRL B (optionally SOH – CTRL A).
2. Optional address code may precede any data byte. Up to six address digits in octal formats, four in hex.
3. Execute code.
4. End code is a nonprintable ETX – CTRL C.
5. Data can also be expressed in 4-bit form.
6. Start code is nonprintable SOM – CTRL R.
7. End code is a nonprintable EOM – CTRL T.

Figure A-5. ASCII-Octal and Hex Formats

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.

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.

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.

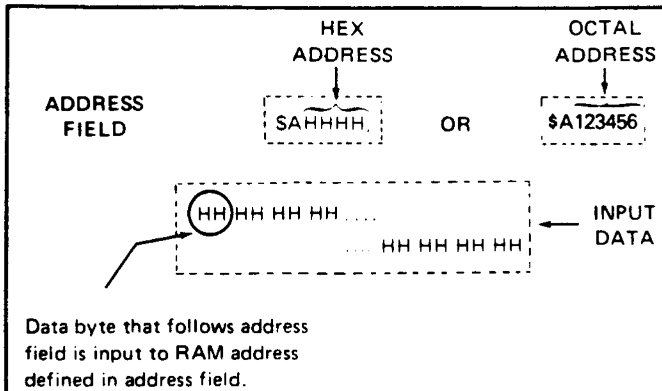
NOTE

At least sixteen characters must follow an end code to avoid a timeout error.

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.

Output is begun by invoking the Output command. The programmer divides the output data into 8-line blocks.

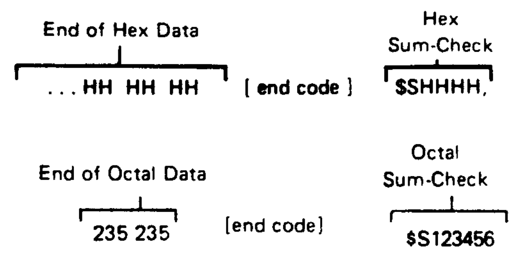
Data transmission is begun with the start code, a



NOTES:

- H = hex character. 1,2,3,4,5, and 6 = octal characters.
- Address has 2-4 or 3-6 octal characters. The programmer assumes leading zeros.
- Address must be preceded by "$A" and followed by ",".
- Line feed and carriage return optional.
- The most significant octal digit may be 0 or 1 when expressing 16 bits as 6 octal characters.

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



NOTES:

- Sum-check field consists of 2-4 hex or 3-6 octal digits sandwiched between "$S" and ",".
- Sum-check field immediately follows end code.
- Sum-check field optional in the input mode and always included in the output mode.
- The most significant octal digit of the sum-check may be 0 or 1 when expressing 16 bits as 6 octal characters.

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

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. The transmission is preceded and followed by 50 null characters.

A.4.7 RCS COSMAC FORMAT, CODE 70

Data in this format begins with a start record consisting of the start character (!M or ?M), an address field, and a space. See Figure A-8.

The start character ?M is sent to the programmer only by a development system. This happens when the operator enters the interrogation ?M at a terminal (linked in parallel with the programmer to the development system), followed by the address in the development system memory where data transmission is to begin, followed by a number of bytes to be transferred, then by a carriage return. The development system responds by sending ?M to the programmer, followed by the starting address, and a data stream which conforms to the data input format described below. Transmission stops when the specified number of bytes have been transmitted.

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.

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

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. The transmission is preceded and followed by 50 null characters.

A.4.8 MICROPROCESSOR FORMATS

Data in these formats is organized into records characterized by expressed addresses and error-check codes. Each format has record start characters and sum-checks. Records are independent; that is, the programmer can accept addresses in nonsequential order. (The Fairchild Fairbug format differs from the other microprocessor formats in address setting. See the Fairchild Fairbug format description).

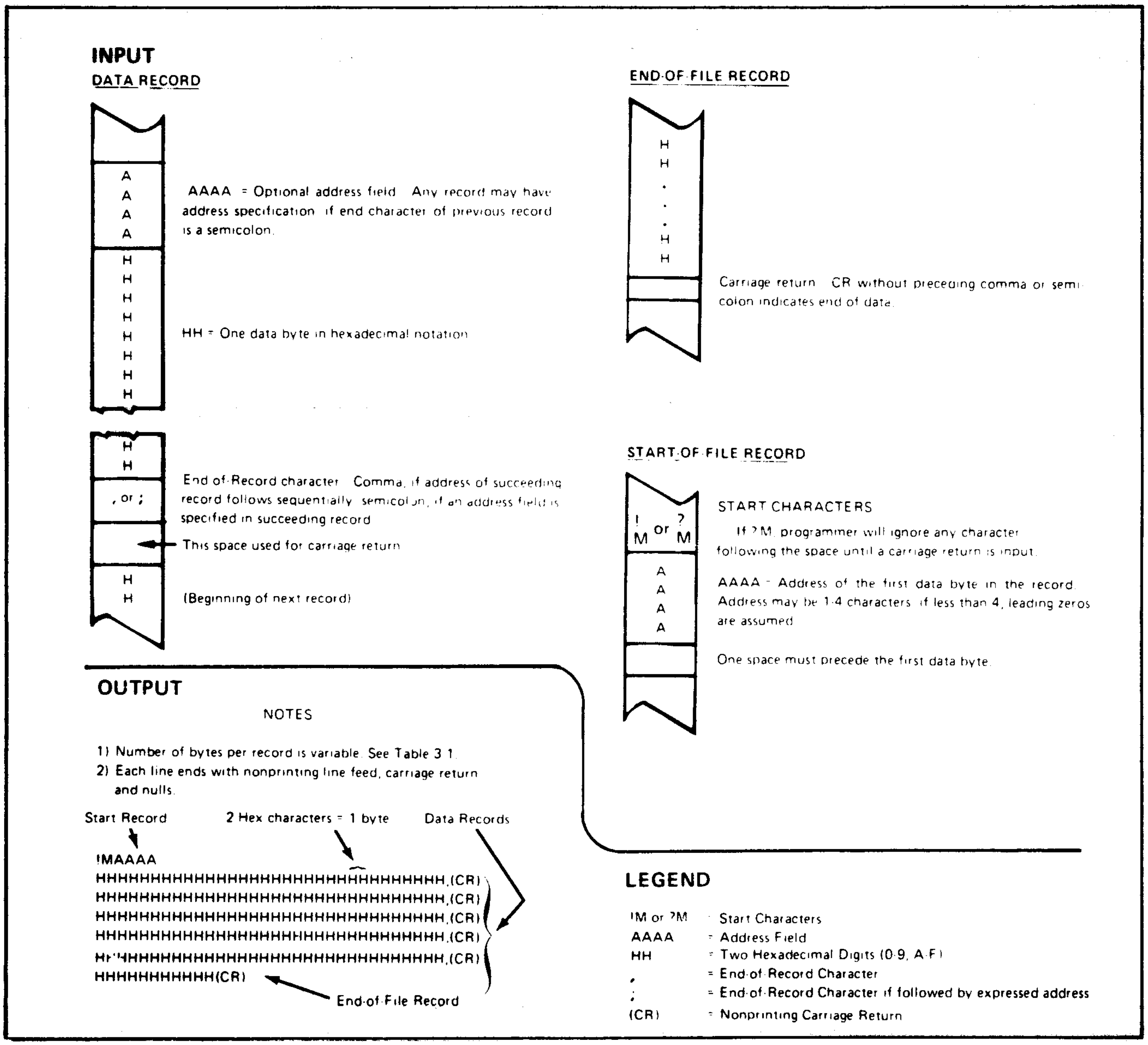


Figure A-8. Specifications for RCA Cosmac Data Files

• 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 1-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.

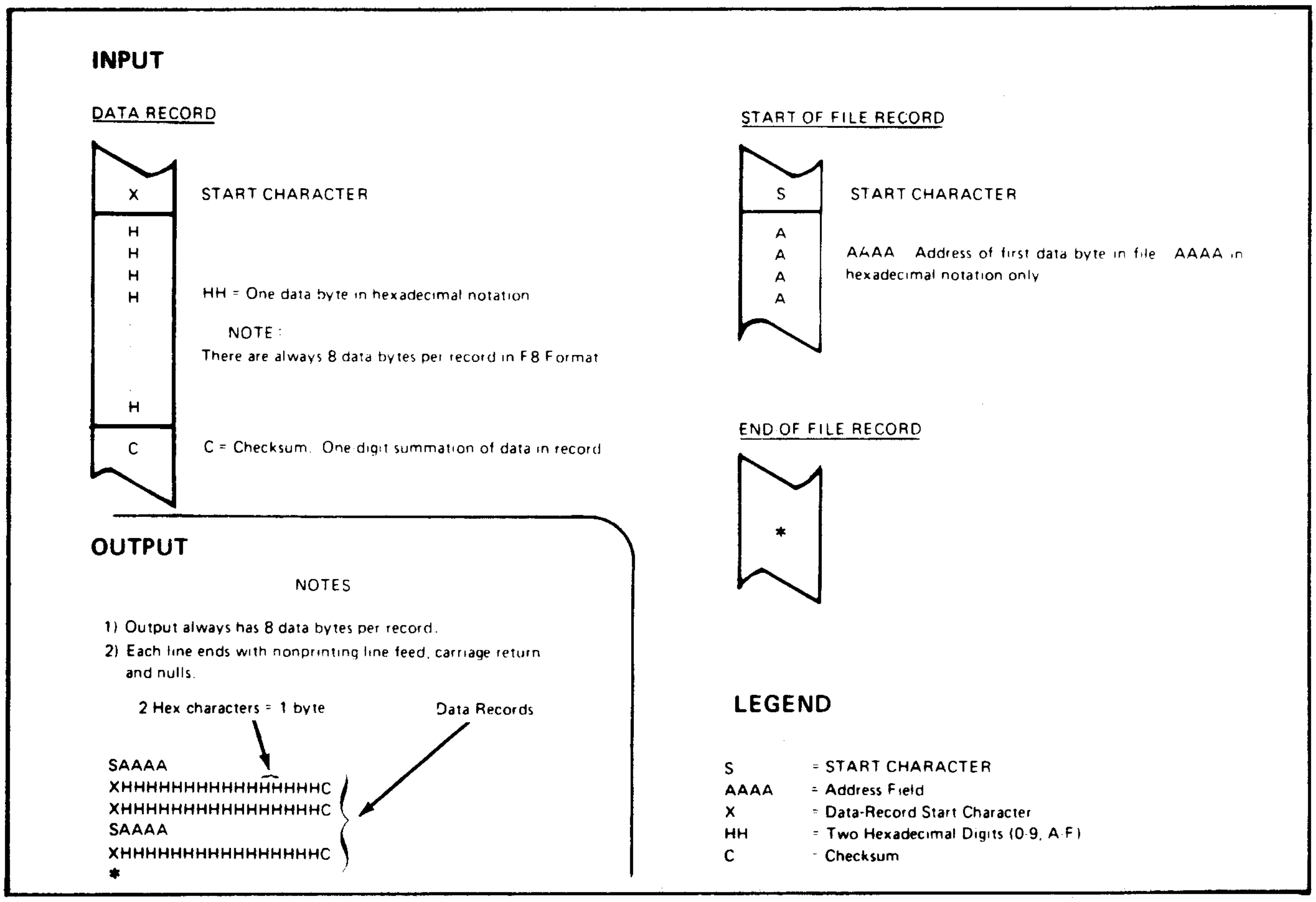


Figure A-9. Specifications for Fairchild Fairbug Data Files

- **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.

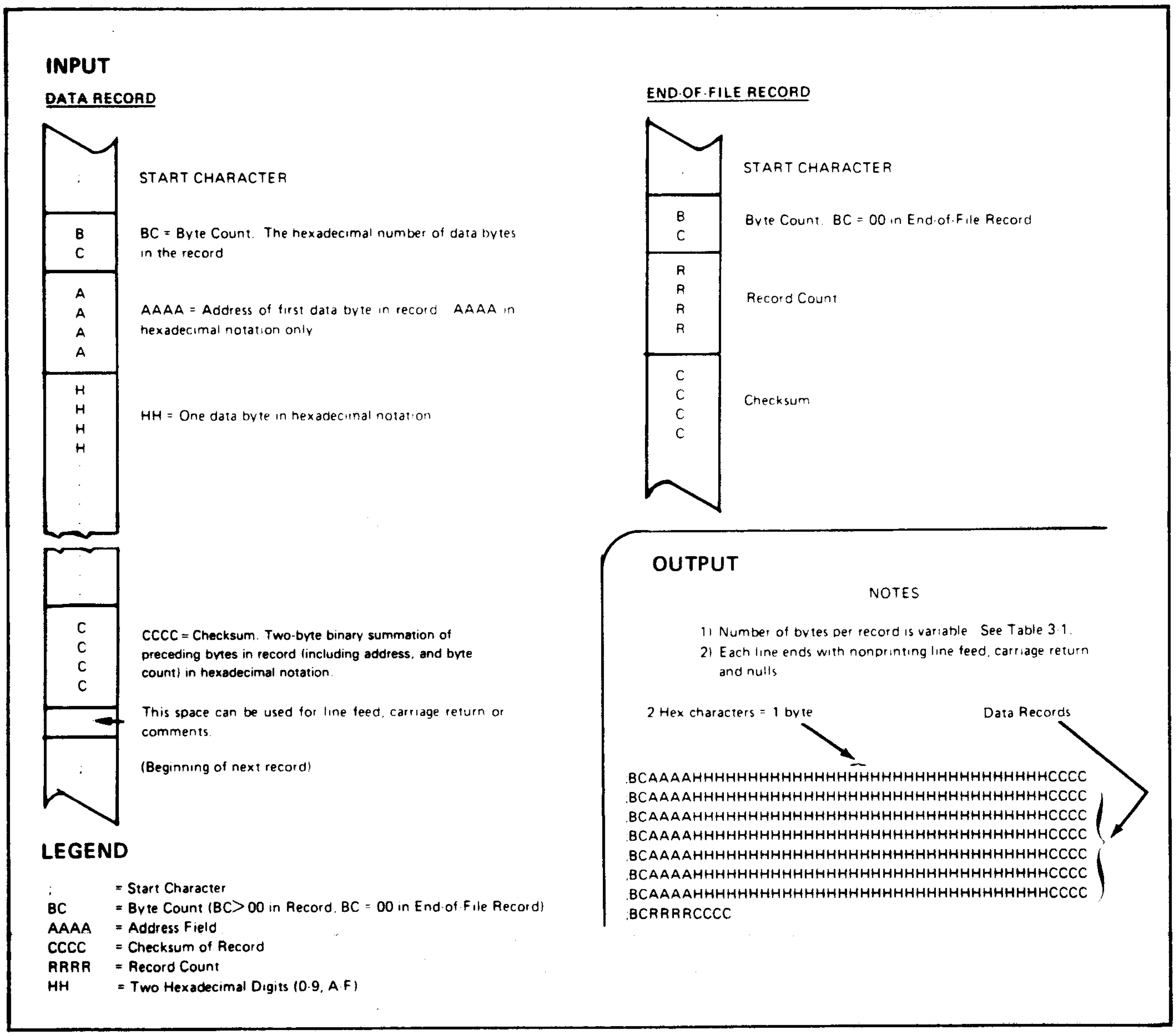


Figure A-10. Specifications for MOS Technology Data Files

- **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 3 less than the byte count. The suffix is a 2-character checksum.

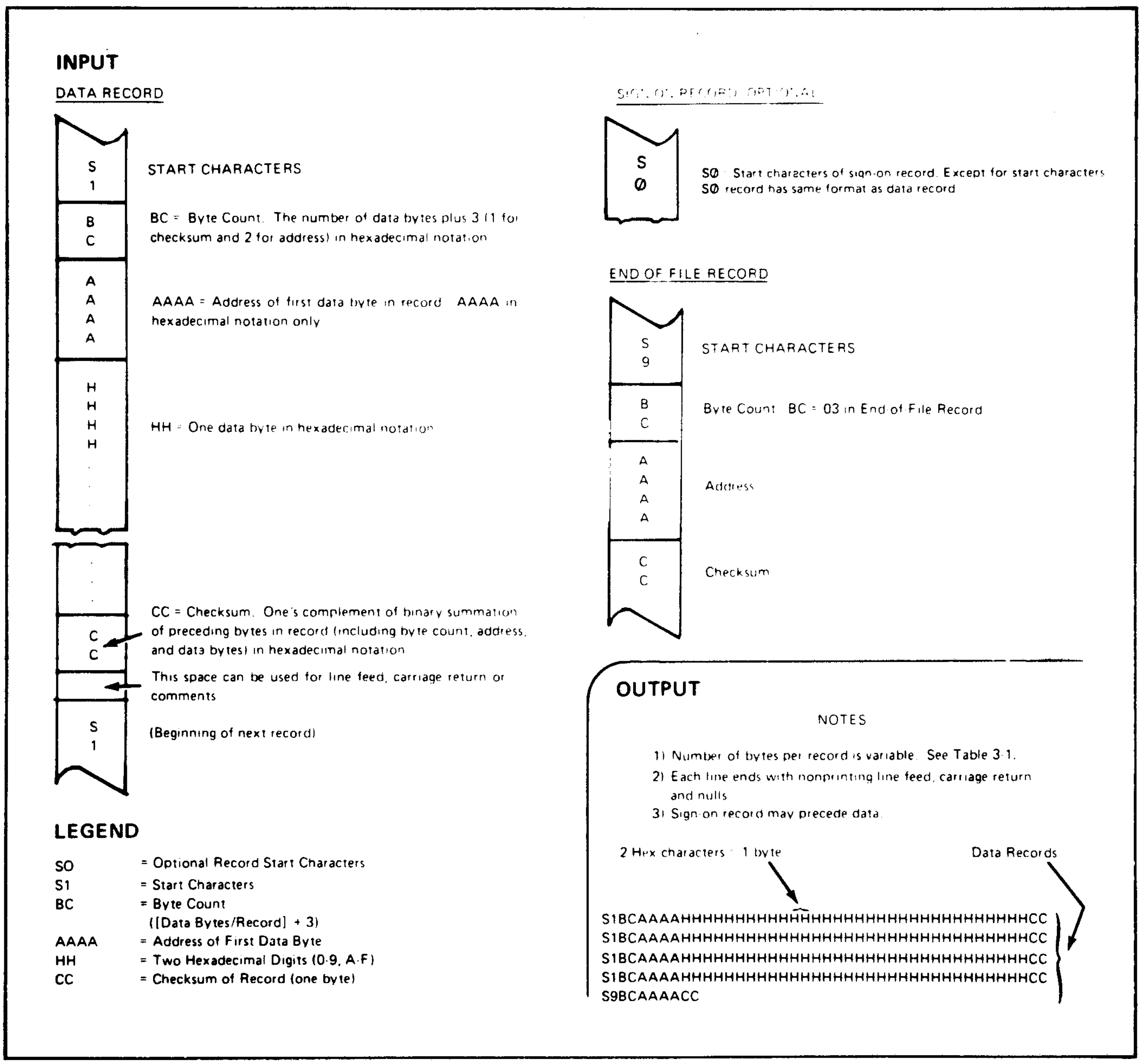


Figure A-11. Specifications for Motorola Data Files

• 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.

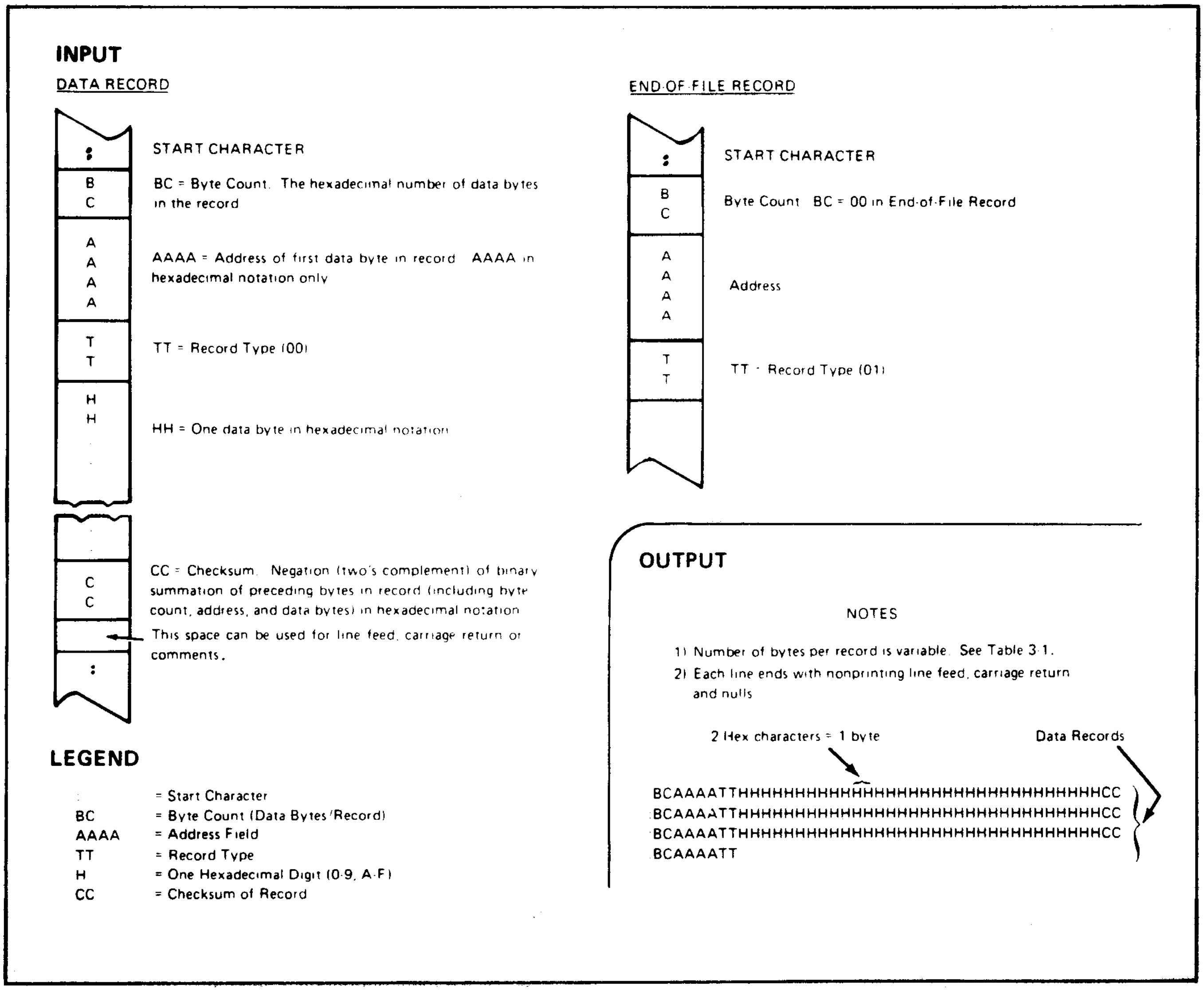


Figure A-12. Specifications for Intel Intellec 8/MDS Data Files

- **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.

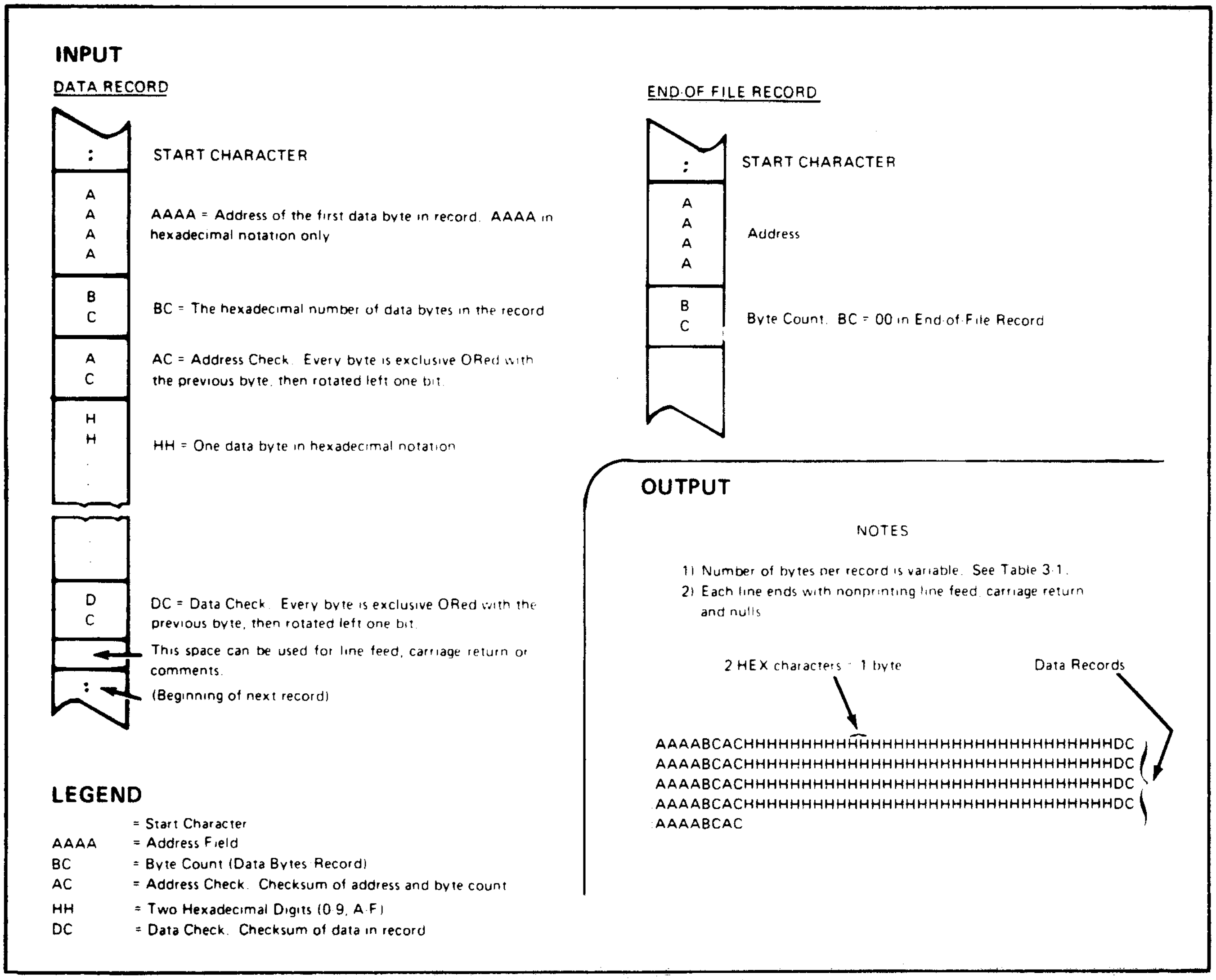


Figure A-13. Specifications for Signetics Absolute Object Data Files

• **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 has 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. The transmission is preceded and followed by 50 null characters.

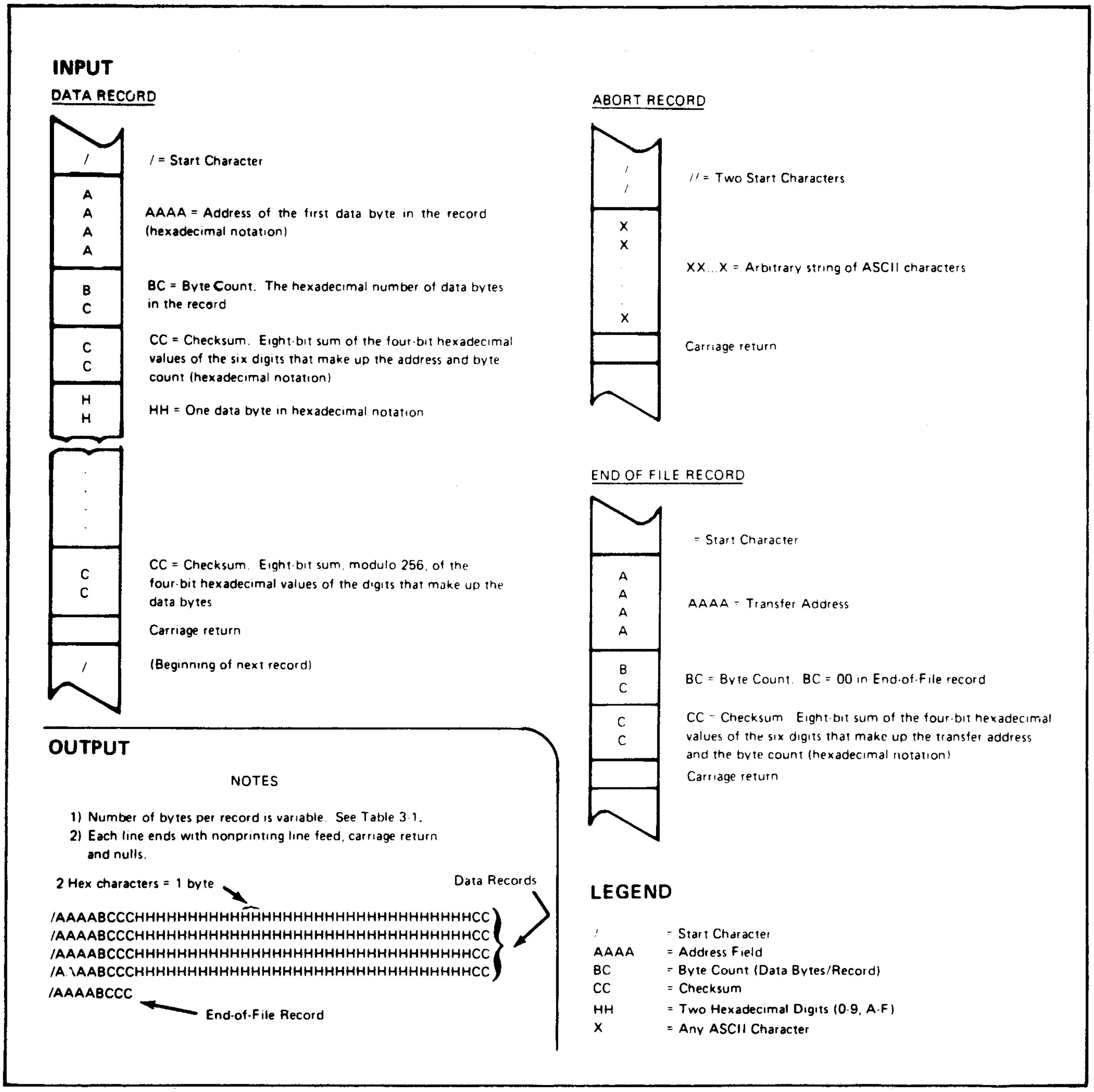


Figure A-14. Specifications for Tektronix Hexadecimal Data Files

• 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 less than the byte count. The suffix is a 2-character checksum.

•INTEL MCS-86 HEXADECIMAL OBJECT, CODE 88

The Intel 16-Bit Hexadecimal Object file record format is basically the same as the Intel Intellec 8/MDS (Code 83). It starts with nine characters (four fields) that define the start of record, byte count, load address, and record type. It ends with a 2-character checksum. Figure A-12 (Intel Intellec 8/MDS) illustrates this format.

There are four record types:

Ø = data record
Ø1 = end record (signals end of file)
Ø2 = extended address record (added to the offset to determine the absolute destination address)
Ø3 = start record (ignored)

Record type Ø2, 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. Because the data bytes are sent in nonsequential fashion, the address offset must be entered into the programmer every time the data transfer is initiated. It is the same as a data record with only four data digits. It's address field is always ØØØØ.

NOTE

Aways specify the address offset when using this format, even when the offset is zero.

• HEWLETT-PACKARD 64000 ABSOLUTE FORMAT, CODE 89

Hewlett-Packard Absolute is a binary format with control and data-checking characters. See Figure A-16.

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-17.

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:

Ø - 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 ØØØØ. 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.

INPUT

DATA RECORD

S1 or S2	START CHARACTERS. S1 if following address field has 4 characters; S2 if address field has 6 characters.
BC	BC = Byte Count. The number of data bytes plus 3 (1 for checksum and 2 for address) in hexadecimal notation.
AAAAAA	AAAA or AAAAAA = Hexadecimal address of first data byte in record.
HHHH . . .	HH = One data byte in hexadecimal notation.
CC	CC = Checksum. One's complement of binary summation of preceding bytes in record (including byte count, address, and data bytes) in hexadecimal notation.
	This space can be used for line feed, carriage return or comments. The programmer will ignore any characters inserted between the last checksum and the start characters of the next record.
S1 or S2	(Beginning of next record)

SIGN-ON RECORD (OPTIONAL)

S0	S0 = Start characters of sign-on record. Except for start characters S0 record has same format as data record.

END-OF-FILE RECORD

S8 or S9	START CHARACTERS. Must be S9 if previous data record began with S1. May be S8 or S9 if previous record began with S2.
BC	Byte Count. BC = 03 in End-of-File Record.
AAAA	Address. Always 0000 in End-of-File Record.
CC	Checksum

LEGEND

SO	=	Optional Record Start Characters
S1	=	Start Characters
BC	=	Byte Count ((Data Bytes/Record) + 3)
AAAA / AAAAAA	=	Address of First Data Byte
HH	=	Two Hexadecimal Digits (0-9, A-F)
CC	=	Checksum of Record (one byte)

OUTPUT

NOTES

1) Number of bytes per record is variable.
2) Each line ends with nonprinting line feed, carriage return and nulls.
3) Sign-on record may precede data.

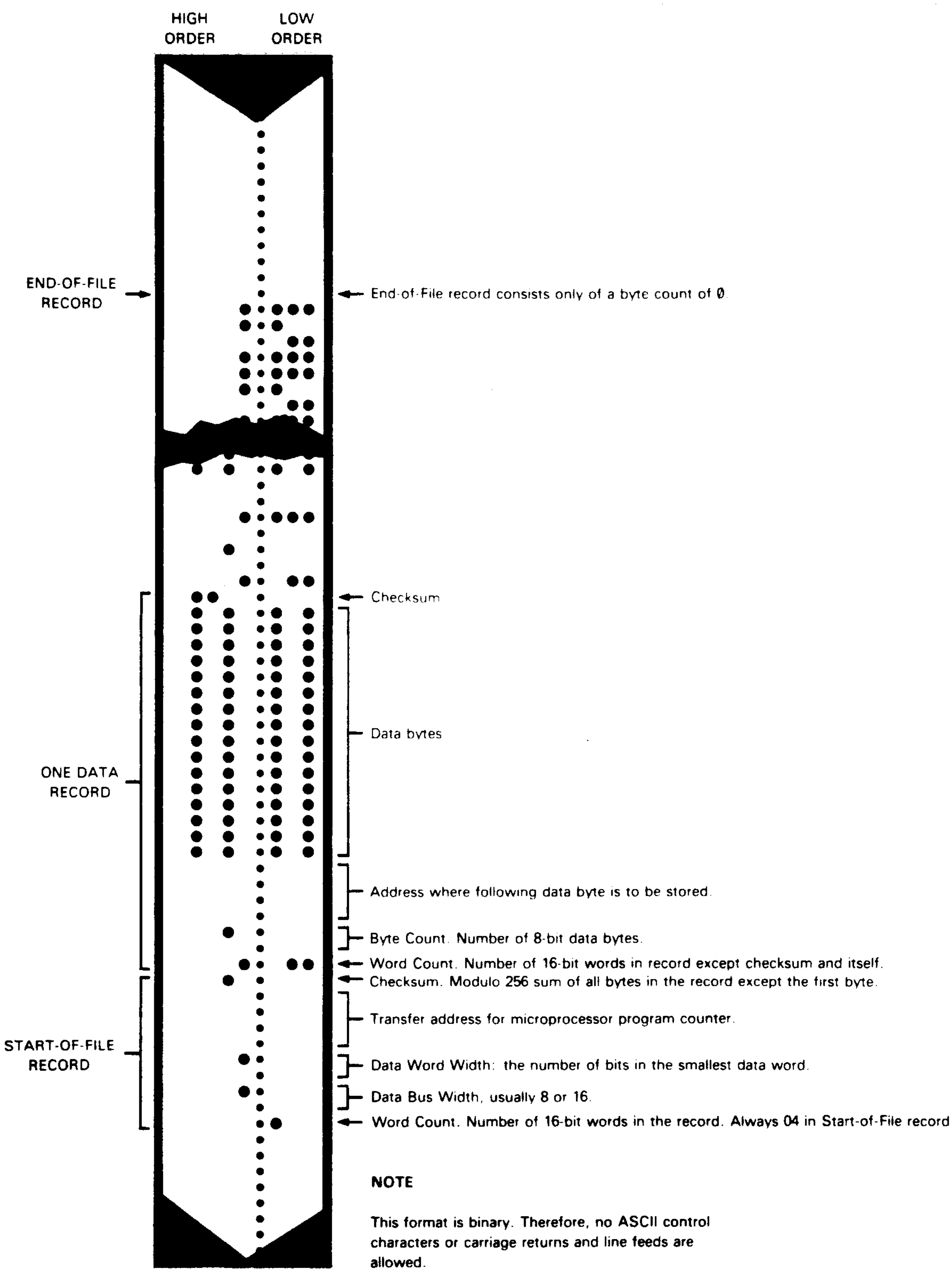
2 Hex characters = 1 byte

Data Records

```
S1BCAAAAHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
S1BCAAAAHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
S1BCAAAAHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
S1BCAAAAHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
S2BCAAAAAAHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
S2BCAAAAAAHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHCC
S9BCAAAACC
```

End-of-File Record

Figure A-15. Specifications for Motorola Exormax Data Files



NOTE

This format is binary. Therefore, no ASCII control characters or carriage returns and line feeds are allowed.

Figure A-16. Specifications for Hewlett Packard Absolute Format Data Files

INPUT

START-OF-FILE RECORD

Ø	Tag Character. Always followed by a file header.
X X X X Y Y Y Y Y Y Y Y	File Header. XXXX = hexadecimal byte count of 16-bit data bytes in the record. YYYYYYYY = name of file. May be any ASCII character in hexadecimal notation.
9	Optional Tag Character. Always followed by a load address. Tag characters 9 and B may come in any order.
A A A A	Load Address: 4-character hexadecimal address of first data byte in the record.
B	Tag Character. A "B" must precede every 4 data characters.
H H H H	HH = 2-character hexadecimal data byte.
B	
H H H H	
7 or 8	Tag Character. Must be followed by a 4-character checksum field. If 8, programmer will ignore the checksum.
C C C C	Hexadecimal Checksum. Two's complement of the sum of the 8-bit ASCII values of the characters in the record from the first tag character through the checksum tag (7 or 8).
F	Tag Character. Denotes end of the data record. All characters following F before line feed and carriage return will be ignored.

DATA RECORD

9	Optional Tag Character. Always followed by a load address. Tag characters 9 and B may come in any order.
A A A A	AAAA = Load address: 4-character hexadecimal address of first data byte in the record.
B	Tag Character. A "B" must precede every 16-bit data byte.
H H H H	HH - 2-character hexadecimal data byte.
B	
H H H H	
B	
H H H H	
7 or 8	Tag Character. Must be followed by a 4-character checksum field. If 8, checksum will be ignored by programmer.
C C C C	Checksum in hexadecimal notation. Two's complement of the sum of the 8-bit ASCII values of the characters in the record from the first tag character through the checksum tag (7 or 8).
F	Tag character. Denotes end of data record. All characters following F before line feed and carriage return will be ignored.

END-OF-FILE RECORD

OUTPUT

NOTES

1)Number of bytes per record is variable.
2)Each line ends with nonprinting line feed, carriage return and nulls.

Start-of-File Record — Data Records

```
ØXXXXYYYYYYYY9AAAABHHHHBHHHH...7CCCCF
BHHHHBHHHHBHHHHBHHHHBHHHH....7CCCCF
BHHHHBHHHHBHHHHBHHHHBHHHH....7CCCCF
BHHHHBHHHHBHHHHBHHHHBHHHH....7CCCCF
```

End-of-File Record — 2 Hex Characters = 1 byte

LEGEND

Ø,7,8,9,B,F = Tag Characters
AAAA = Address Field
XXXX = Byte Count
YYYYYYYY = File Name
HH = Two Hexadecimal Digits (0-9, A-F)
CCCC = Checksum

Figure A-17. Specifications for Texas Instruments SDSMAC Data Files

APPENDIX B
OPTIONAL COMPUTER REMOTE CONTROL

NOTE

This Optional Computer Remote Control package is the same as that used with Data I/O System 19s and System 17s. Computer software written for the System 19s or 17s is directly compatible with 29As equipped with this option. Only 29As equipped with Optional Computer Remote Control can accomplish the operations in this section. The following programmers have Optional Computer Remote Control:

990-0029-009
990-0029-010
990-0029-011
990-0029-013
990-0029-014
990-0029-015

B.1 INTRODUCTION

The Optional Computer Remote Control is designed to allow complete control of the 29A 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 for ease of incorporation into a computer program. For use with a terminal, the standard Remote Control described in Section 3 of this manual is more applicable.

B.2 INSTALLATION

The 29A with Optional Computer Remote Control must be connected to the computer according to RS232C or 20 mA current loop specifications. The function of each serial port connector pin is described in Section 2 of this manual. Refer to Table 2-3 to determine the necessary connector pins for serial data transfers. The programmer's baud rate, parity, and stop bit settings are also described in Section 2.

B.3 OVERVIEW

Figure B-1 illustrates the basic components of the 29A under Optional Computer Remote Control. Remote control commands are written into a computer's operating software, allowing it to control the 29A in much the same way as it would control any other peripheral such as a disc drive or printer.

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

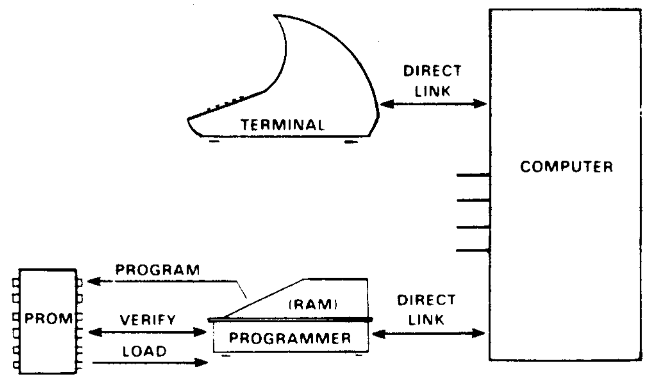


Figure B-1. Optional Computer Remote Control Components

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

B.4 RESPONSE CHARACTERS

The programmer sends a response character to the computer after every command. Table B-1 summarizes these.

Whenever an error occurs, the 29A will send an F to

Table B-1. 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 follows it with a carriage return.
F	Fail	Informs the computer that the programmer has failed to execute the last command entered. The programmer follows it with a carriage return.
?	Question	Informs the computer that the programmer does not understand a command or the command was invalid. The programmer follows it with a carriage return.

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 error codes (described in Section 3 of this manual) that have occurred. The F command codes all errors into a 32-bit error status word, as shown in Figure B-2.

When a command is invalid or not understood by the programmer, it will send a ?. When this occurs, examine the last command entered to check its validity.

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		
29		
28		
27		
26	4	Serial-overrun error (42)
25	2	Serial-framing error (41, 43)
24	1	Buffer overflow, i.e., V 15 characters (48)
		PROGRAMMING ERRORS
23	8	Any device-related error
22	4	Start line not set high (26)
21	2	L2 + L3 V 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		
13		
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., V word limit (27, 28, 51, 56, 57)
8	1	Data not hexadecimal (84, 86, 91) Insufficient data received, ASCII-Hex and Octal (54)
		RAM ERRORS
7	8	RAM-hardware error (64, 66 or any RAM error)
6		
5	2	L2 + L3 V 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 end not on 1K boundary (62)

EXAMPLE:

What errors are indicated in this error status word: 80C80081?

8 — the word contains error information
0 — no receive errors
C — (= 8 + 4): 8 = Device-related error
4 = Start line not set high (error 26)
8 — device is not blank (error 20)
0 — no input errors
0 — no input errors
8 — RAM error (error 62, and possible 64 and 66)
1 — RAM end is not on 1K boundary (error 62)

NOTES

1. **The numbers in parentheses are 29A error codes, defined in Section 3.**
2. **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.**
3. **After being read, the error-status word resets to zeros.**

Figure B-2. Error-Status Word

B.5 ENTERING AND EXITING REMOTE CONTROL

To enter Optional Computer Remote Control use Select Function F1. It is detailed in Table 3-4 of this manual. While in Optional Computer Remote Control the programmer display will show REMOTE MODE and the rotating action symbol. On entering remote control, the programmer will retain all RAM data.

To exit Optional Computer Remote Control via the 29A 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.

B.6 COMMAND SUMMARY

Table B-2 is a summary of Optional Computer Remote Control commands. Figure B-3 is a flowchart of the command protocol. Section B.7 gives further detailed descriptions of the command groups and individual commands.

Table B-2. Command Summary

COMMAND	NAME	DESCRIPTION	RESPONSE	NOTES
CONTROL COMMANDS				
RETURN		Execute a command.		1,2
ESC		Aborts a command.	> CRLF	1,2
BREAK		Aborts a binary transfer.	> CRLF	1,2
UTILITY COMMANDS				
G	Software-configuration	This command sends a 4-digit hex number (XXXX) representing the software revision level in the programmer.	XXXX > CRLF	1,2
HHHH<	Set Begin RAM Address	Defines first RAM address to be used for data transfers. Also functions as the RAM Source Address in RAM-RAM Block Move. The default value is Ø. Setting this value clears any previously entered Block Size.	> CRLF	1,2
HHHH ;	Set Block Size	Sets number of bytes to be transferred. The default value is the device size, for device-related operations; RAM limit less the Begin RAM Address for I/O operations; no default for RAM-RAM Block Move.	> CRLF	1,2
HHHH :	Set Begin Device Address	Sets the first device address to be used in data transfers. Also functions as the RAM-destination address in RAM-RAM Block Move. The default value is Ø.	> CRLF	1,2
HH]	Select External Function	This command accesses Select Codes (HH) carried in extended software on some programming modules.	> CRLF	1,2
S	Sum-check	Causes programmer to calculate the sum-check of RAM data up to word limit of the installed programming electronics and output it to the computer.	XXXX > CRLF	1,2
F	Error-Status Inquiry	Programmer return a 32-bit word that codes errors accumulated. Error-status word resets to zeros after interrogation. (Error-status word is shown in Figure B-2.)	XXXXXXXX > CRLF (See Figure B-2)	1,2
X	Error-Code Inquiry	Programmer outputs the error codes stored in scratch-RAM and then clears them from memory. Refer to the error list in Section 3.	XXXX > CRLF	1,2
H	No Operation	This is a null command and always returns a prompt character (>.	>CRLF	1,2
Z	Escape Remote Control	Return control to the programmer.	None	

Table B-2. Command Summary (cont.)

COMMAND	NAME	DESCRIPTION	RESPONSE	NOTES
DEVICE COMMANDS				
T	Illegal-Bit Test	Test for illegal bit in device.	> CRLF	1,2
B	Blank Check	Check that no bits are programmed in device.	> CRLF	1,2
[	Family and Pinout Inquiry	Programmer sends a 4-digit number (FFPP) where FF is the Family Code and PP is the Pinout Code in effect.	FFPP> CRLF (Expanded Memory Programming Modules only) FCRLF (Standard Programming Modules)	1,2
FFPP @	Select Family and Pinout	A 2-digit Family Code (FF) and a 2-digit Pinout Code (PP) set up the programming module for programming a particular device.	> CRLF (Expanded Memory Programming Modules only)	1,2
R	Respond	Programmer indicates status determined by programming module and socket adapter and outputs AAA/B/C or AAAA/B/C, where AAA or AAAA = device word limit, B = byte size and C = VOL/VOH status (1 = VOL; 0 = VOH).	AAA/B/C > CRLF	1,2
L	Load	Load device data into RAM.	> CRLF	1,2
P	Program	Program RAM data into device.	> CRLF	1,2
V	Verify	Verify device against RAM.	> CRLF	1,2
I/O COMMANDS				
D	Select Odd Parity	Sets odd parity for input and output data. The default value is the programmer's parity-switch setting.	> CRLF	1,2
E	Select Even Parity	Sets even parity for input and output data. The default value is the programmer's parity-switch setting.	> CRLF	1,2
N	Select No Parity	Sets no parity for input and output data. The default value is the programmer's parity-switch setting.	> CRLF	1,2
J	Set 1 Stop Bit	Sets 1 stop bit for input and output data. The default value is the programmer's stop-bit switch.	> CRLF	1,2
K	Set 2 Stop Bits	Sets 2 stop bits for input and output data. The default value is the programmer's stop-bit switch.	> CRLF	1,2
FC A	Select Translation Format	Two Characters (FC) before A define the data translation format for I/O data transfer. The default value is MOS Technology Format, #81.	> CRLF	1,2
HH M	Select Record Size	Two hex characters before M define output record size. The default value is 16 bytes per record (8 bytes per record in Fairchild Fairbug).	> CRLF	1,2

Table B-2. Command Summary (cont.)

COMMAND	NAME	DESCRIPTION	RESPONSE	NOTES
HH U	Set Nulls	Two hex characters before U set the number of nulls output after carriage returns and enables line feeds. The default value is no nulls and no line feeds.	> CRLF	1,2
HHHH W	Set Address Offset	Four hex characters before W define the offset added on output and subtracted on input. The default value is Ø (output) or first incoming address (input).	> CRLF	1,2
=	Disable Timeout	Disables the 25-second I/O timeout. Restored only at power on.	> CRLF	1,2
I	Input	Input data from computer to RAM.	> CRLF	1,2,3
O	Output	Output data from RAM to computer.	> CRLF	1,2,3
C	Compare	Compare RAM data with data in computer.	> CRLF	1,2,3
Y	Parity-Error Inquiry	Responds with the hex number of parity errors since last Y command, since power on, or since last parity command (D, E, or N).	XXXX > CRLF	1,2
EDITING COMMANDS				
Q	Swap Nibbles	Exchanges high- and low-order halves of every word in RAM.	> CRLF	1,2
\	RAM-RAM Block Move	Initiates data transfer from one RAM location to another. The Begin RAM Address, block size, and Begin Device Address must be set first.	> CRLF	1,2
HHHH ?	Split RAM Data	For 16-bit microprocessor data. Splits even- and odd-numbered bytes into two blocks separated by a center point, HHHH, which must be a power of 2 between Ø and RAM midpoint. The default value is the RAM midpoint.	> CRLF	1,2
HHHH >	Shuffle RAM Data	For 16-bit microprocessor data. Merges block above center point HHHH with block below. Center point must be a power of 2 between Ø and RAM midpoint. The default value is the RAM midpoint.	> CRLF	1,2
^	Clear All RAM	Clears all of the 29A's data RAM to zeros.	> CRLF	1,2

NOTES

1. *LF = Line Feed, CR = carriage return*
2. *Line Feeds are present only if the null command (U) has been sent.*
3. *Response occurs at end of data transmission with proper termination.*

B.7 COMMAND GROUPS

This section gives detailed descriptions and usage of the command groups and individual commands used in Optional Computer Remote Control.

B.7.1 CONTROL COMMANDS

These commands are used to execute or suspend a command.

RETURN. 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**. These commands cause the programmer to unconditionally halt (abort) any operation in progress, output a V, and await further instructions from the computer.

B.7.2 UTILITY COMMANDS

These commands set or check various operating parameters related to operations.

SOFTWARE CONFIGURATION NUMBER — **G RETURN**

On command, the programmer sends the 4-digit hex number representing the particular configuration or revision level of software resident in the 29A.

SET BEGIN RAM ADDRESS — **HHHH < RETURN**

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. Setting the Begin RAM Address clears any previously entered Block Size. The default value is Ø.

SET BLOCK SIZE — **HHHH ; RETURN**

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 for **Block Moves.**

SET BEGIN DEVICE ADDRESS — **HHHH : RETURN**

This command, preceded by a 4-digit 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 Ø.

SELECT EXTERNAL FUNCTION — **HH] RETURN**

Accesses Select Functions or Select Codes carried in the extended software of some programming modules. Consult the programming module manuals for more information.

SUM-CHECK — **S RETURN**

This command instructs the programmer to calculate the 4-digit hex sum-check of RAM from Ø to RAM word limit, device word limit, or the limit defined by the ; command, whichever is smaller. Sum-check is defined in the Glossary in Appendix C.

ERROR-STATUS INQUIRY — **F RETURN**

On this command, the programmer returns a 32-bit word,displayed as 8 hex characters, that codes errors accumulated. The error-status word resets to all zeros after interrogation. See Figure B-2.

ERROR CODE INQUIRY — **X RETURN**

The programmer responds to this command with hex error codes previously stored. After execution, the error codes are cleared from memory. Section 3 of this manual lists and describes all the error codes.

NO OPERATION — **H RETURN**

This is a null command and always returns a prompt (V).

ESCAPE REMOTE CONTROL — **Z RETURN**

This command returns control to the 29A keyboard. All RAM data and operating parameters except the address offset are retained.

B.7.3 DEVICE COMMANDS

This group of commands executes the operations used in device programming. Figure B-3 illustrates their respective protocols.

NOTE

Illegal-Bit Test, Blank Check, Load, Program, and Verify are performed from the Begin RAM Address to the device word limit, RAM word limit or Block Size, whichever is smaller. In the case of devices larger than RAM, these commands are performed starting at the Begin Device Address defined by the : command.

ILLEGAL-BIT TEST — **T RETURN**

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 RETURN**

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 — **[RETURN**

The 29A 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 @ RETURN**

Selects a 2-digit Family Code (FF) and a 2-digit Pinout Code (PP). This command is only valid when using programming modules requiring Family and Pinout Codes.

RESPOND — **R RETURN**

The programmer checks the programming module and outputs:

AAAA / B / C

- AAAA: device word limit
- B: byte size
- C: VOL (1) or VOH (Ø)

LOAD — **L RETURN**

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 RETURN**

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 RETURN**

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.

B.7.4 I/O COMMANDS

This group of commands sets up the 29A to transmit or receive data through the serial port. This includes inputting or outputting data, selecting a data translation format, setting parity, address controls, and other considerations incidental to I/O data transfers.

SELECT ODD PARITY — **D RETURN**

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 RETURN**

This command 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 RETURN**

This command 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 RETURN**

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 RETURN**

On receiving this command, the programmer sets two stop bits for serial data transfers. The default value is the programmer's stop-bit switch.

SET TRANSLATION FORMAT — **FC A RETURN**

This command selects the input or output data translation format expressed by the format code (FC) in the command. Table B-3 lists the format codes. The default value is MOS Technology Format, #81. All the data translation formats available are detailed in Appendix A of this manual.

Table B-3. 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 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 without the SOH (CTRL A) start code, these alternate data translation format codes are used.

SELECT RECORD SIZE — HH M RETURN

The 2 hex characters (HH) before M 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 — ^

Clears all of the 29A data RAM to zeros.

SET NULLS — HH U RETURN

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 — = RETURN

This command disables the 25-second I/O timeout. The timeout can be restored only be turning off the programmer and then turning it n again.

INPUT DATA — I RETURN

This command instructs the programmer to accept formatted data from the computer.

OUTPUT DATA — O RETURN

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 (Control S), and will resume on receipt of the X-ON character, DC-1 (Control Q).

COMPARE DATA — C RETURN

This command instructs the programmer to compare data in RAM with data in the computer.

PARITY-ERROR INQUIRY — Y RETURN

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).

B.7.5 EDITING COMMANDS

This group of commands is used for manipulating data in the 29A data RAM.

SWAP NIBBLES — Q RETURN

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 — \ RETURN

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 ? RETURN

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 Ø 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.

Each block of data can then be programmed into an 8-bit device, and the 2 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.

The center-point default value is the RAM midpoint.

SHUFFLE RAM DATA — HHHH V RETURN

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 at address Ø, to create the two blocks, which are then merged for serial transfer. The center point must be a power of 2 between Ø and RAM midpoint. The center-point default value is the RAM midpoint.

ADDRESS OFFSET — HHHH W RETURN

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.

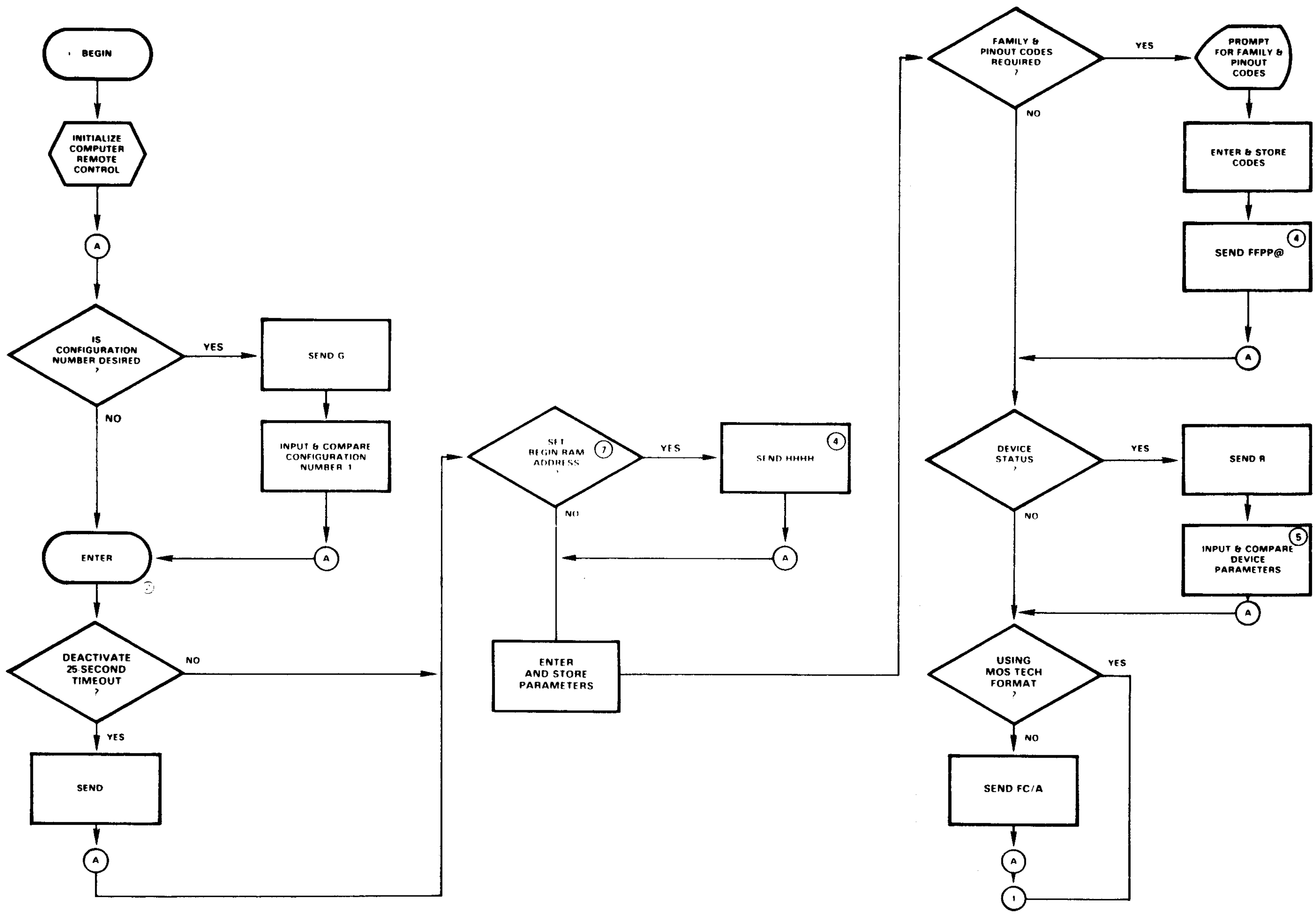


Figure B-3. Optional Computer Remote Control (Page 1 of 5)

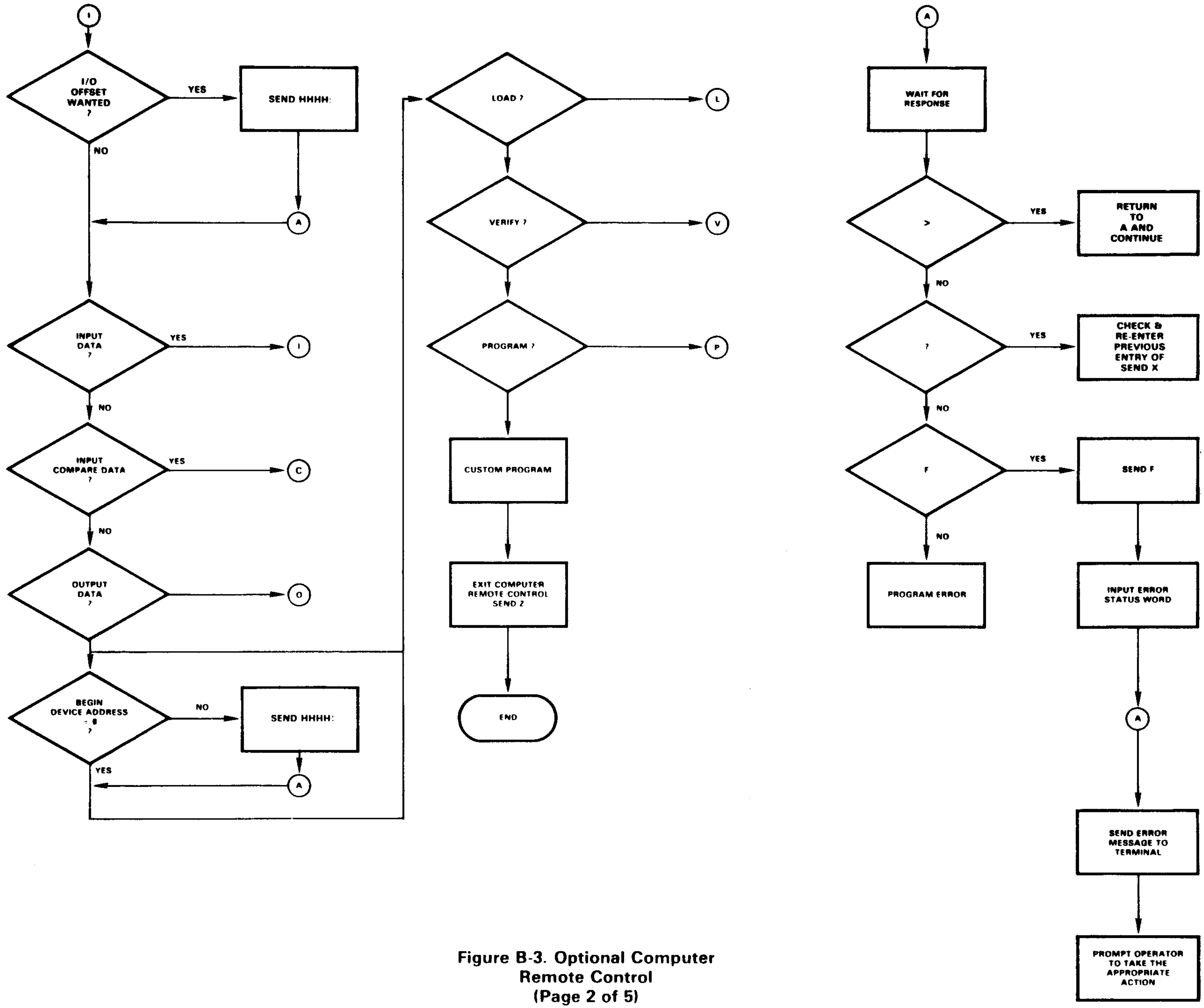


Figure B-3. Optional Computer
Remote Control
(Page 2 of 5)

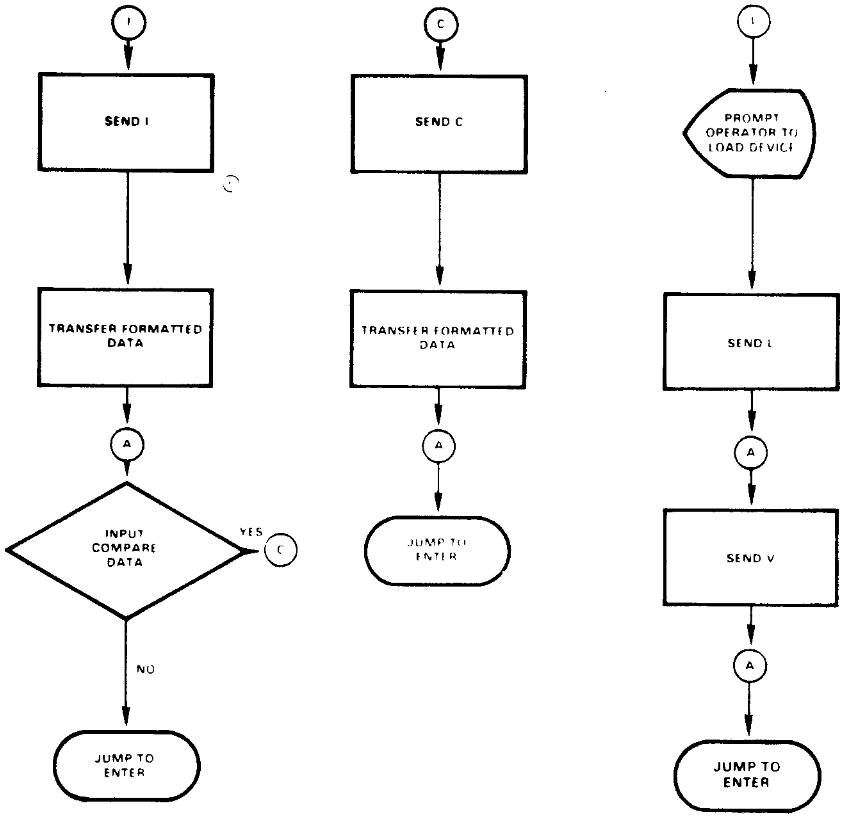


Figure B-3. Optional Computer Remote Control (Page 3 of 5)

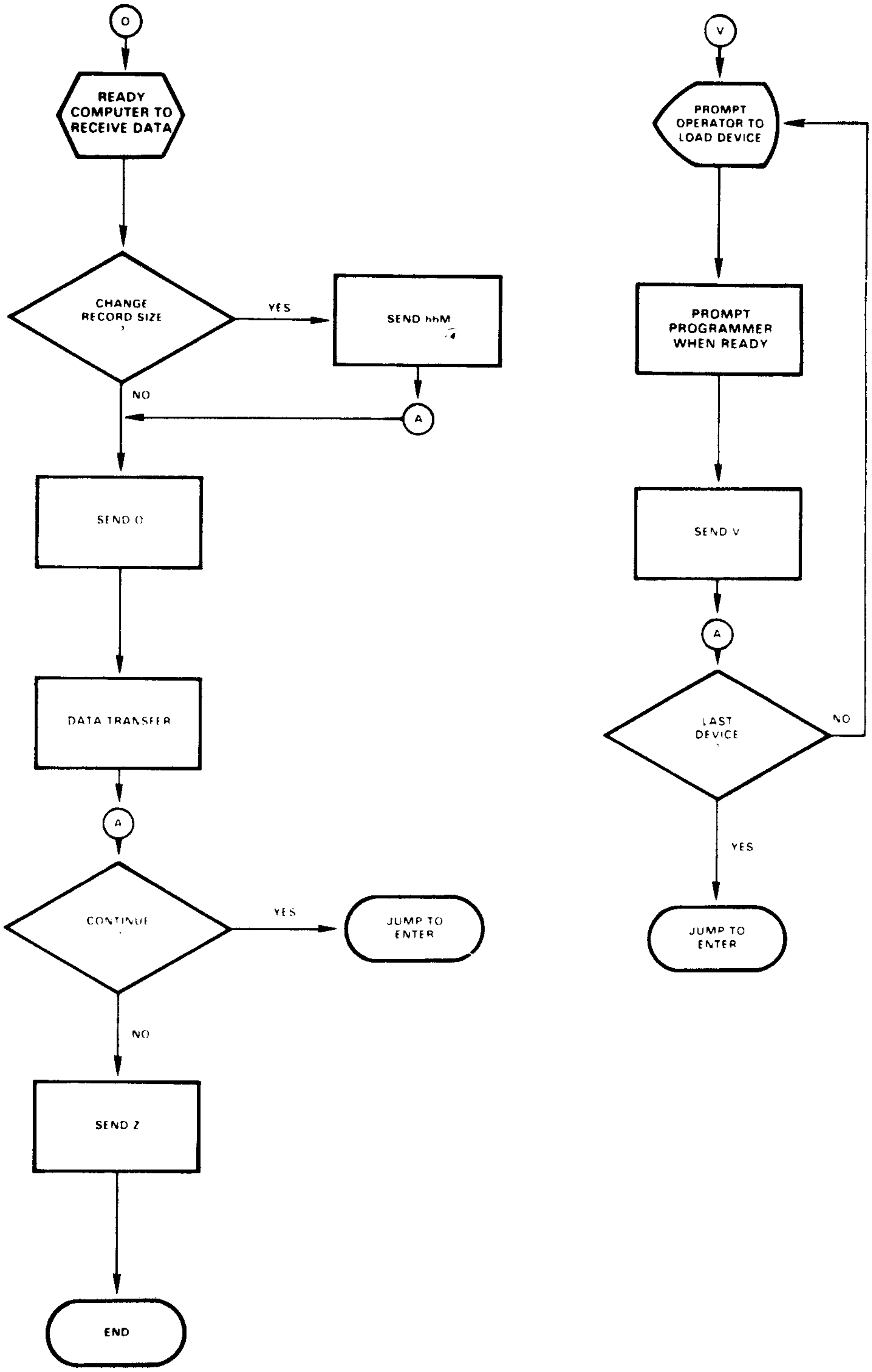


Figure B-3. Optional Computer Remote Control (Page 4 of 5)

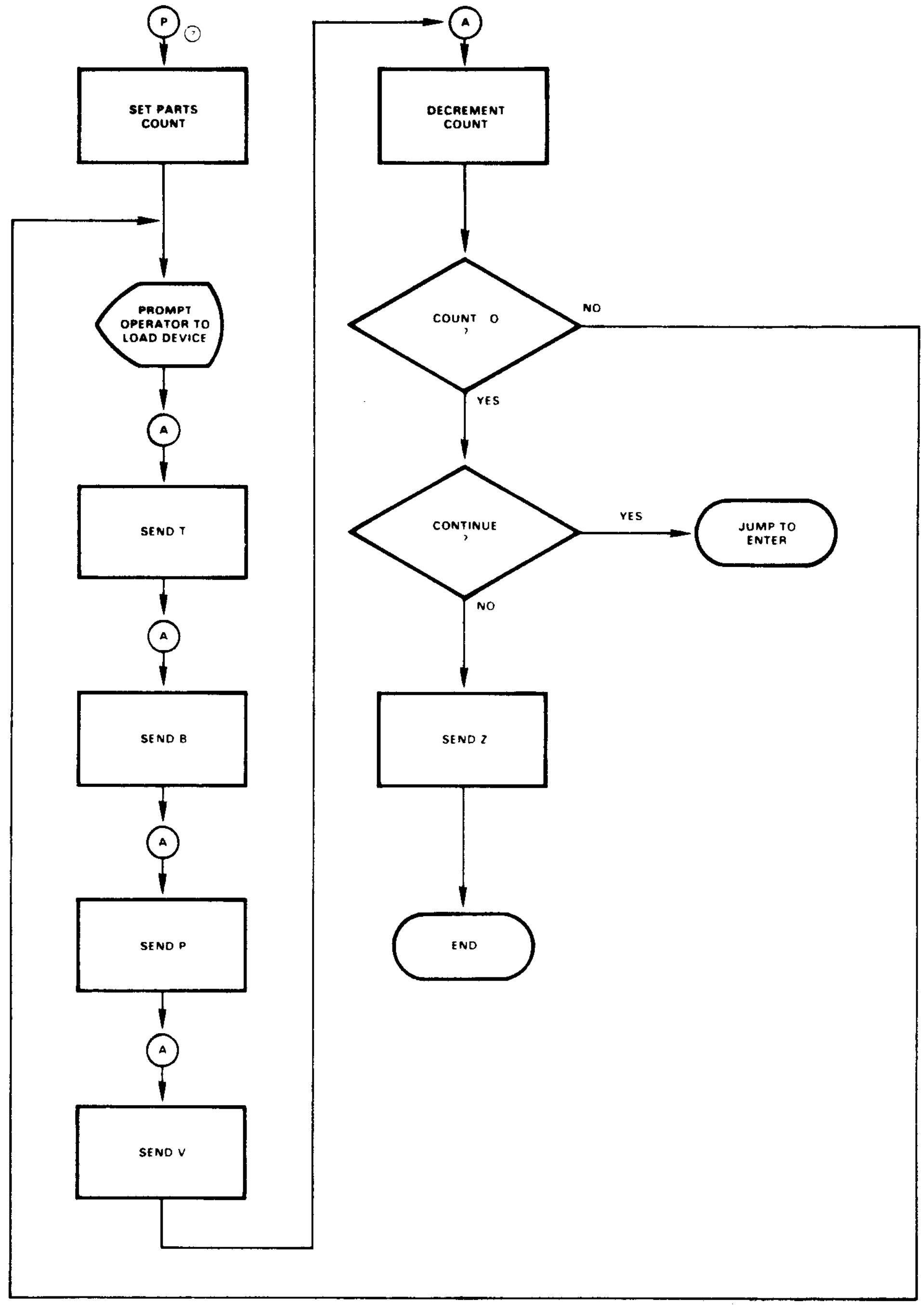


Figure B-3. Optional Computer Remote Control (Page 4 of 4)

NOTES

1. *The configuration number command allows the program to query the programmer as to which version of remote control it is using.*
2. *The following commands can be invoked at any time in the sequence:*

   S Sum-check
   X Error Code Inquiry
   H No Operation

   Each must be followed by a carriage return.
3. *The default value is zero nulls. By entering a number of nulls, a line feed will accompany the carriage return.*
4. *The letters hhhh, hh, FFPP, and FC are used to denote variables that must be entered at this time.*
5. *Device parameters are output in the following format:*

   XXXX / Y / Z
   word limit
   word size (4 or 8)
   VOH or VOL (0 = VOH; 1 = VOL)
6. *A 500 ms delay is advised in some cases.*
7. *To clear Begin RAM send < without a prefix.*

APPENDIX C
REFERENCE MATERIAL

Table C-1. Glossary

address field. Optional set of control characters in a data translation format. It defines the address of the next data byte.

address offset. A 4-digit hex value subtracted from addresses on input and then added to addresses output from the programmer. The result is added to the begin device address or begin RAM address as applicable.

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. Also the form for data output by the unit.

default value. The value the unit uses for a parameter unless the operator specifies another value.

device. Any PROM, EPROM, MOS PROM, or programmable logic array.

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.

FIP. Flourescent Indicator Panel.

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 memory used for performing calculations.

select function. A 2-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.

sum-check. A summation of bits calculated according to the rules of simple addition and usually expressed as a 4-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.

HEX DATA	BINARY DATA
84	10000100
C1	11000001
62	01100010
24	00100100
01CB	0000 0001 1100 1011
↑ Sum-check in hexadecimal notation	↑ Sixteen-bit binary sum-check

Figure C-1. Sample Sum-check Calculation

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).

Table C-2. Abbreviations

The following is a list of abbreviations commonly used in Data I/O Instruction manuals.

A_4 - address line 4
ADDR - address
BC - bin count, number of bins
BR - bridge rectifier
C - capacitor
$\overline{CE}$ - Chip Enable
Clk - clock
Clk. Inh. - clock inhibit
Cntl. - control
Cont. - control
CR - diode
CTS - Clear To Send
D_5 - data line 5
DAC - Digital to Analog Converter
DC - division count, number of partitions
DCD - Data Carrier Detect
DCU - Data I/O Data Control Unit
DI_2 - data input 2
DIR - Directory
DO_2 - data output 2
DS - display
DSR - Data Set Ready
DTR - Data Terminal Ready
DVM - digital voltmeter
Emul - emulate
ERR - error
ESC - escape
F - fuse
FC - Translation Format Code
FFPP - Family Code (FF) and Pinout Code (PP)
FIP - Flourescent Indicator Panel
FPGA - Field Programmable Gate Array
FPLA - Field Programmable Logic Array
FPLS - Field Programmable Logic Sequencer
FPRP - Field Programmable ROM Patch
FRME - frame
Fwd. - Forward
Gnd. - Ground
HLT. - Halt
HV - high voltage
ID_4 - identification line 4
IFL - Integrated Fused Logic
I/O - Input/Output
$\overline{IRQ}$ - Interrupt Request
J - jack or connector
JP - jumper
K - relay
LIM - limit
LSB - Least Significant Bit
LSD - Least Significant Digit
MSB - Most Significant Bit
MSD - Most Significant Digit
NMI - Non-Maskable Interrupt
NO CONT SECT - No Contiguous Sector
Oper. - operate
PA_{15} - programmer address line 15
PAK - programming module
PCS - Program Card Set
PD_6 - programmer data line 6
PN - Part Number
PR/$\overline{OE}$ - Preset/Output Enable
Prog - Program
Prog. Pulse - Program Pulse
PROM - Programmable Read Only Memory
Q - transistor
R - resistor
RAM - Random Access Memory
Read Inh. - Read Inhibit
Rec. - receive
Rev. - reverse
RP - resistor pack
RST - Reset
R/$\overline{W}$ - Read/Write
RX - Receive Data
S - switch
TOR - Turn On Reset
TX - Transmit Data
U - integrated circuit device
$\overline{V \bullet \emptyset 2}$ - the "AND"-ing of the Valid Memory Address line and the phase 2 line
Ver. A - Verify pass A
Ver. B - Verify pass B
VFY - verify
VMA - Valid Memory Address
VR - Voltage Regulator
V_{REF} - Voltage Reference
W/L - Word Limit
Write Inh. - Write Inhibit

Table C-3. 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

Table C-4. ASCII & IEEE Code Chart

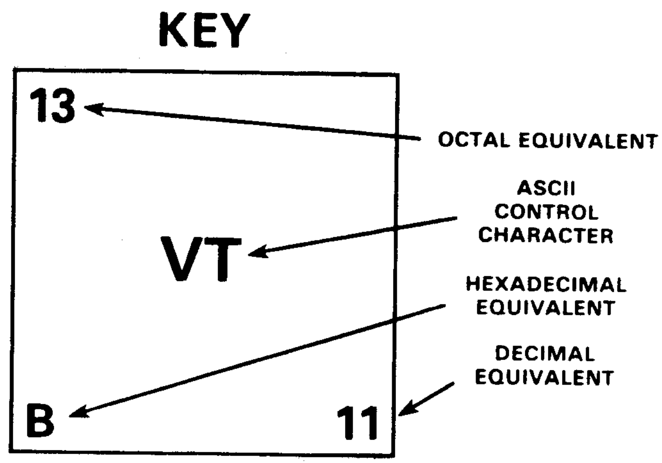


Table C-4. (cont.)

Bits 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 0 0	20 DLE 10 16	40 SP 20 32	60 0 30 48	100 @ 40 64	120 P 50 80	140 ` 60 96	160 p 70 112
0 0 0 1	1 SOH 1 1	21 DC1 11 17	41 ! 21 33	61 1 31 49	101 A 41 65	121 Q 51 81	141 a 61 97	161 q 71 113
0 0 1 0	2 STX 2 2	22 DC2 12 18	42 " 22 34	62 2 32 50	102 B 42 66	122 R 52 82	142 b 62 98	162 r 72 114
0 0 1 1	3 ETX 3 3	23 DC3 13 19	43 # 23 35	63 3 33 51	103 C 43 67	123 S 53 83	143 c 63 99	163 s 73 115
0 1 0 0	4 EOT 4 4	24 DC4 14 20	44 $ 24 36	64 4 34 52	104 D 44 68	124 T 54 84	144 d 64 100	164 t 74 116
0 1 0 1	5 ENQ 5 5	25 NAK 15 21	45 % 25 37	65 5 35 53	105 E 45 69	125 U 55 85	145 e 65 101	165 u 75 117
0 1 1 0	6 ACK 6 6	26 SYN 16 22	46 & 26 38	66 6 36 54	106 F 46 70	126 V 56 86	146 f 66 102	166 v 76 118
0 1 1 1	7 BEL 7 7	27 ETB 17 23	47 ' 27 39	67 7 37 55	107 G 47 71	127 W 57 87	147 g 67 103	167 w 77 119
1 0 0 0	10 BS 8 8	30 CAN 18 24	50 (28 40	70 8 38 56	110 H 48 72	130 X 58 88	150 h 68 104	170 x 78 120
1 0 0 1	11 HT 9 9	31 EM 19 25	51) 29 41	71 9 39 57	111 I 49 73	131 Y 59 89	151 i 69 105	171 y 79 121
1 0 1 0	12 LF A 10	32 SUB 1A 26	52 * 2A 42	72 : 3A 58	112 J 4A 74	132 Z 5A 90	152 j 6A 106	172 z 7A 122
1 0 1 1	13 VT B 11	33 ESC 1B 27	53 + 2B 43	73 ; 3B 59	113 K 4B 75	133 [5B 91	153 k 6B 107	173 { 7B 123
1 1 0 0	14 FF C 12	34 FS 1C 28	54 , 2C 44	74 < 3C 60	114 L 4C 76	134 \ 5C 92	154 l 6C 108	174 ¦ 7C 124
1 1 0 1	15 CR D 13	35 GS 1D 29	55 - 2D 45	75 = 3D 61	115 M 4D 77	135] 5D 93	155 m 6D 109	175 } 7D 125
1 1 1 0	16 SO E 14	36 RS 1E 30	56 . 2E 46	76 > 3E 62	116 N 4E 78	136 ^ 5E 94	156 n 6E 110	176 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 / 2F 47	77 ? 3F 63	117 O 4F 79	137 _ 5F 95	157 o 6F 111	177 Rubout 7F 127
	Addressed Commands	Universal Commands	Listen Addresses		Talk Addresses		Secondary Addresses or Commands	

Table C-5. 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 separater
GS	group separater
HT	horizontal tabulation
LF	line feed
NAK	negative acknowledge
NUL	null
RS	record separater
SI	shift in
SO	shift out
SOH	start of heading
STX	start of text
SUB	substitute
SYN	synchronous idle
US	unit separater
VT	vertical tab

APPENDIX D
SELECT FUNCTION FC

D.1 INTRODUCTION

Select Function FC, Remote On Off, allows you to control your remote control capability remotely. It applies to both standard and Optional Computer Remote Control.

After selecting FC, enter hexadecimal values for the ASCII codes you want to use as on/off codes. The defaults at power-up are **00 00**, which define no on/off codes. (Therefore, a null can't be used as an on/off code.)

When entry of the code is complete, the remote mode will be in an "off" state, waiting for the "on" code. All characters input will be discarded until the "on" code is received. All subsequent characters are processed in the normal fashion until an "off" code is received, restoring the remote mode back to the "off" state.

Setting the on/off codes back to zeros disables this feature and sets the remote mode to the "on" state.

Select Function FC has no effect on input done via the data translation formats. Characters in the translator data stream matching the on/off codes have no effect on the state of the port.

D.2 APPLICATIONS

With this capability, the 29A can be "daisy chained" to a terminal and the terminal connected to the computer via one port only. Previous applications required a separate port for the programmer. Figure D-1 illustrates this.

Another possible application is multiplexing more than one programmer to the same port as shown in Figure D-2. By assigning different "on" codes for each programmer, they can be selected on an individual basis. The only problem is that units that are off will look for enable codes while doing translator input to units that are on. This can be solved by careful selection of the enable codes for all translators without binary format.

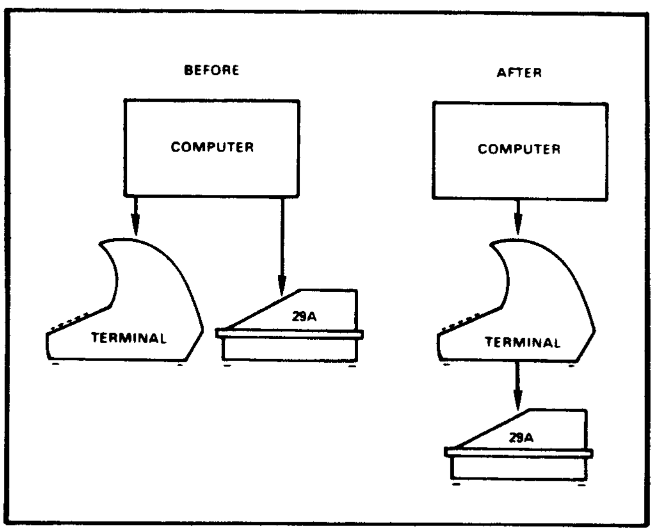


Figure D-1. Chaining Off

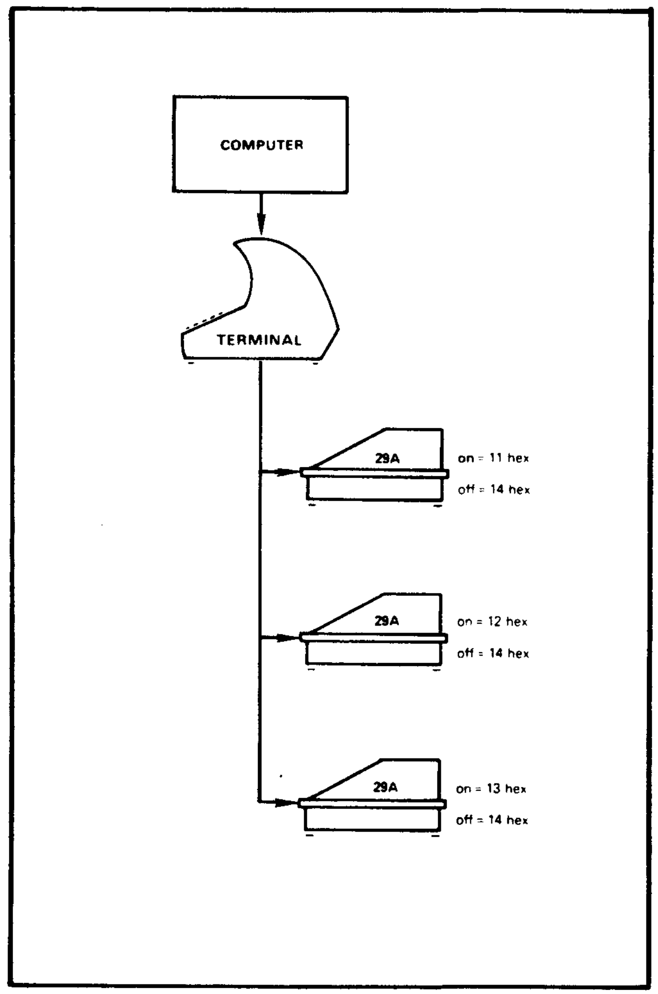


Figure D-2. Multiplexing Programmers

29A REMOTE CONTROL

OPERATION	DIRECT ENTRY	INTERACTIVE ENTRY ENTRY	 TERMINAL DISPLAY
Input from Port	COvPOvXXXXvYYYYvTOvRAvZZZZ+[CR]	CO[CR] PO[CR] XXXX,YYYYvTOvRA[CR] ZZZZ[CR]	COPY DATA FROM> POR ADDR, SIZE> COPY POR>RAM ADDR>
Load from Device	COvDEvXXXXvYYYYvTOvRAvZZZZ+[CR]	CO[CR] DE[CR] XXXX,YYYYvTOvRA[CR] ZZZZ[CR] *	COPY DATA FROM> DEV ADDR, SIZE> COPY DEV>RAM ADDR>
Program Device	COvRAvXXXXvYYYYvTOvDEvZZZZ+[CR]	CO[CR] RA[CR] XXXX,YYYYvTOvDE[CR] ZZZZ[CR] *	COPY DATA FROM> RAM ADDR, SIZE> COPY RAM>DEV ADDR>
Output to Port	COvRAvXXXXvYYYYvTOvPOvZZZZ+[CR]	CO[CR] RA[CR] XXXX,YYYYvTOvPO[CR] ZZZZ[CR]	COPY DATA FROM> RAM ADDR, SIZE> COPY RAM>POR ADDR>
Block Move	COvRAvXXXXvYYYYvTOvRAvZZZZ+[CR]	CO[CR] RA[CR] XXXX,YYYYvTOvRA[CR] ZZZZ[CR]	COPY DATA FROM> RAM ADDR,SIZE> COPY RAM>RAM ADDR>
Verify Device	VEvRAXXXXvYYYYvTOvDEvZZZZ+[CR]	VE[CR] RA[CR] XXXX,YYYYvTOvDE[CR] ZZZZ[CR]	VERIFY DATA FROM> RAM ADDR,SIZE> VE RAM>DEV ADDR>
Input Verify	VEvRAvXXXXvYYYYvTOvPOvZZZZ+[CR]	VE[CR] RA[CR] XXXX,YYYYvTOvPO[CR] ZZZZ[CR]	VERIFY DATA FROM> RAM ADDR,SIZE> VE RAM>POR ADDR>

Edit:

• To view last address edited	ED [CR]
• To view a specific address	EDvAAAA[CR]
• To enter data at a specific address	EDvAAAAvDD[CR]
• To increment addresses	[CR]
• To decrement addresses	/[CR]

XXXX = *begin source address*
YYYY = *Block Size*
ZZZZ = *begin destination address*
AAAA = *Edit address*
DD = *data*
FFPP = *Family and Pinout Codes*

v = *space bar*
* = *If Family and Pinout Codes are required, the terminal will prompt for them at this point.*
+ = *If Family and Pinout Codes are required, enter ",FFPP" after ZZZZ.*

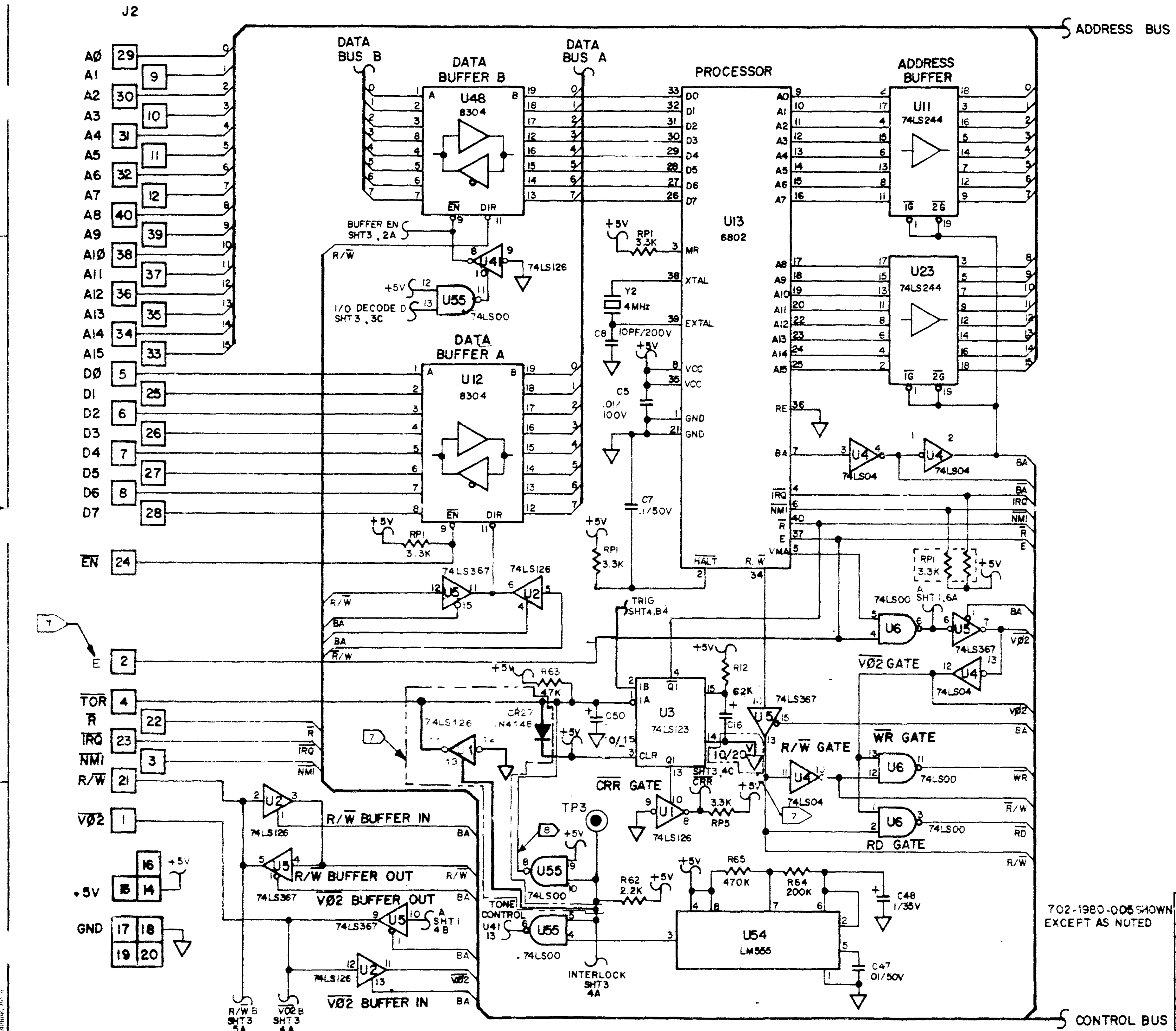
Detach and keep near your terminal.

Remote Control Guide

APPENDIX E
SCHEMATICS

30-702-1980	Controller Board
30-702-1982	Filter Board
30-702-1983	Expanded Memory Board (Optional)
30-702-0061	FIP Display Driver Card
30-702-1648	Keyboard

REVISIONS					
LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
A	RELEASE	J.K.			5-81
B	ECN 4371	BV			1-82
C	ECN 4769	CL			
D	ECN 4793				
E	ECN 4800	CL			
F	ECN 4829	RF			5-26-3



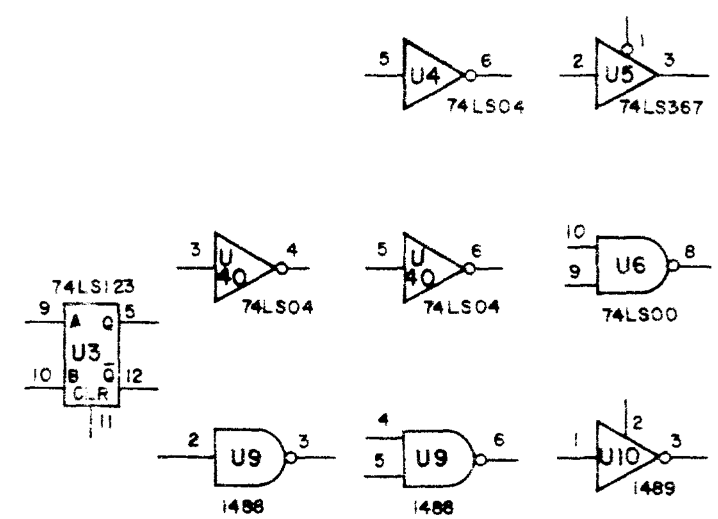
702-1980-005 SHOWN EXCEPT AS NOTED

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:

   C52, CR27, J8, JP5, Q26, R69, RP8, TP3, U55, VR4, Y2

4.
5. UNUSED GATES:



6. POWER SUPPLY CHART ON SHEET 4
7. -004 ONLY.
8. -004 ONLY NC.

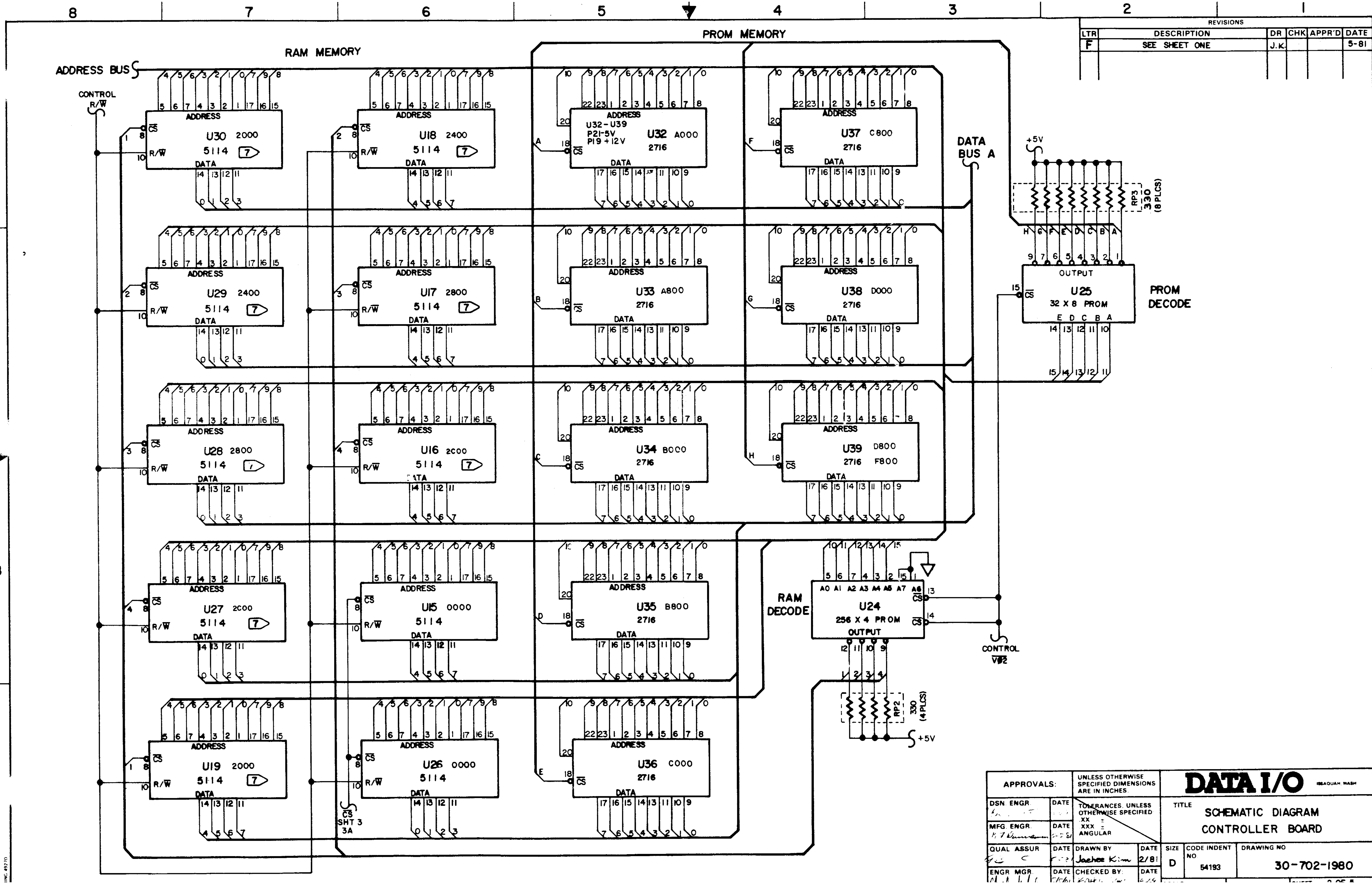
APPROVALS	DATE	UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
DSN ENGR		TOLERANCES UNLESS OTHERWISE SPECIFIED .XX ± .XXX ± ANGULAR
MFG ENGR		
QUAL ASSUR		DRAWN BY Jaehee Kim 2/81
ENGR MGR		CHECKED BY:

DATA I/O

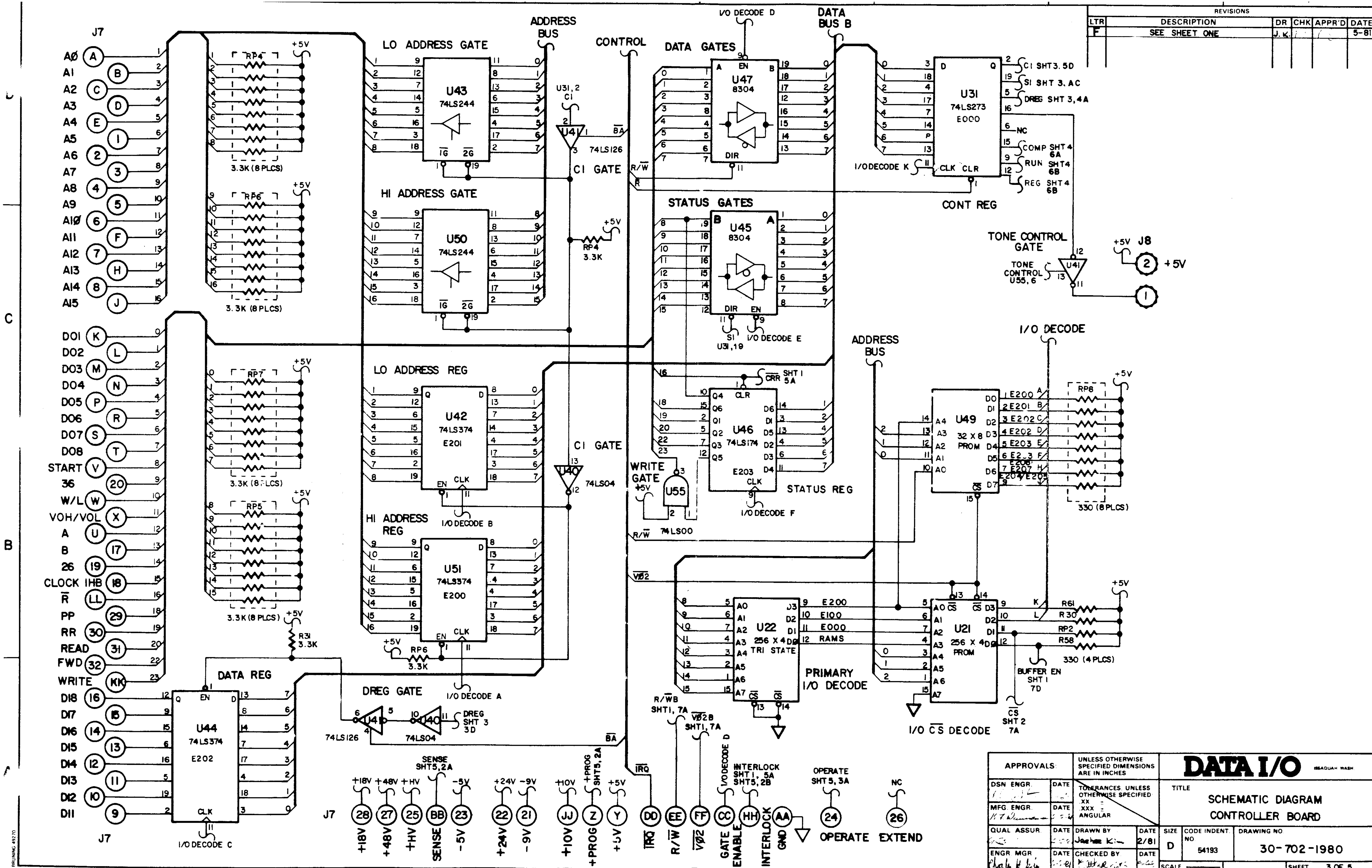
TITLE: SCHEMATIC DIAGRAM CONTROLLER BOARD

SIZE	CODE INDENT NO	DRAWING NO
D	54193	30-702-1980

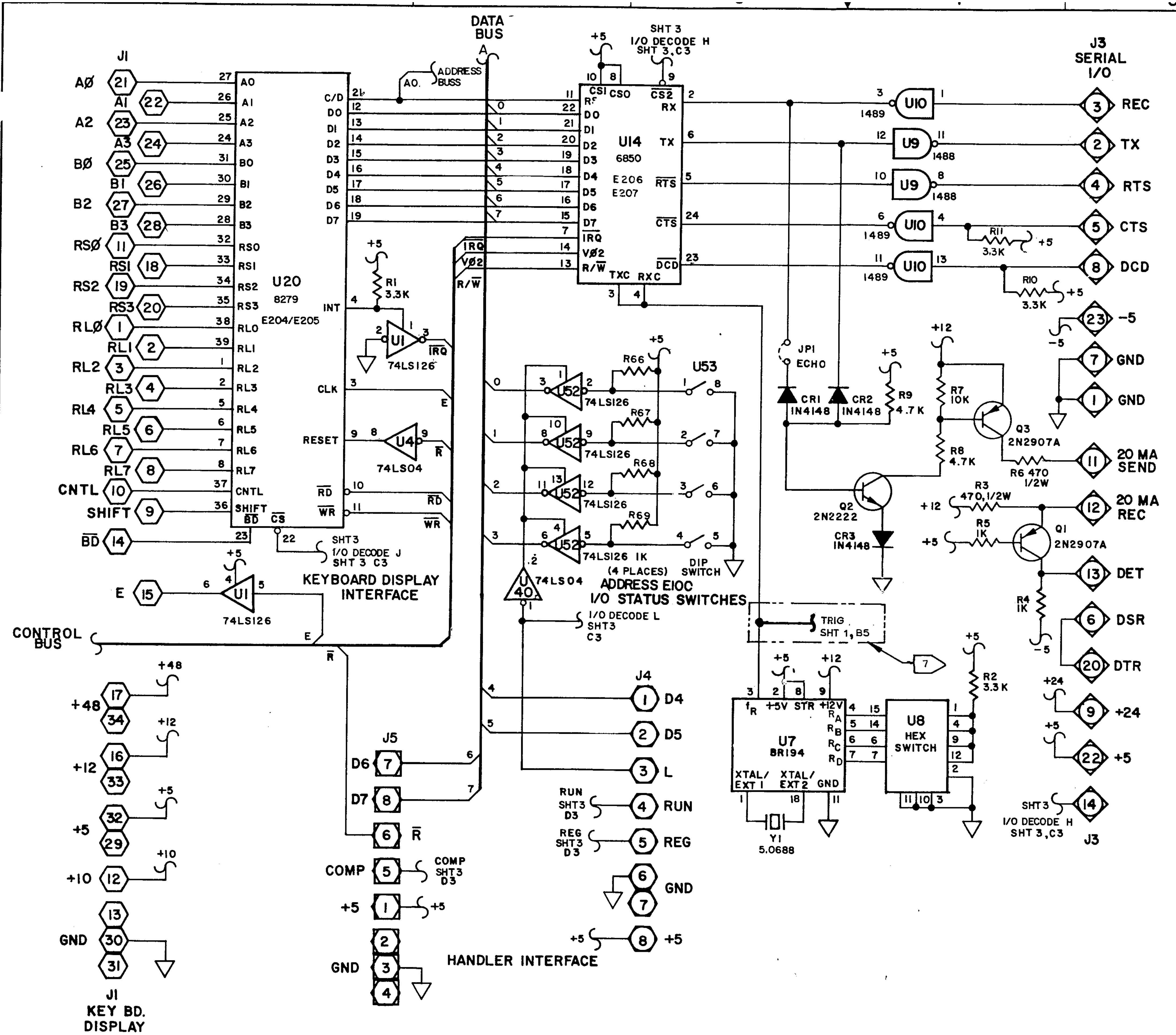
SCALE — SHEET 1 OF 5



PROM MEMORY
RAM MEMORY
ADDRESS BUS
CONTROL R/W
U30 2000 5114
U18 2400 5114
U29 2400 5114
U17 2800 5114
U28 2800 5114
U16 2C00 5114
U27 2C00 5114
U15 0000 5114
U19 2000 5114
U26 0000 5114
U32-U39 P21-5V P19 +12V
U32 A000 2716
U33 A800 2716
U34 B000 2716
U35 B800 2716
U36 C000 2716
U37 C800 2716
U38 D000 2716
U39 D800 F800 2716
DATA BUS A
+5V
RP3 330 (8 PLCS)
U25 32 X 8 PROM
PROM DECODE
RAM DECODE
U24 256 X 4 PROM
CONTROL VΦ2
RP2 330 (4 PLCS)
CS SHT 3 3A
REVISIONS
LTR DESCRIPTION DR CHK APPR'D DATE
F SEE SHEET ONE J.K. 5-81
APPROVALS
DSN. ENGR.
MFG. ENGR.
QUAL ASSUR
ENGR MGR
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES, UNLESS OTHERWISE SPECIFIED XX = XXX = ANGULAR
DRAWN BY Jaehee Kim 2/81
CHECKED BY
DATA I/O ISSAQUAH, WASH
TITLE SCHEMATIC DIAGRAM CONTROLLER BOARD
SIZE D
CODE INDENT NO 54193
DRAWING NO 30-702-1980


REVISIONS
LTR DESCRIPTION DR CHK APPR'D DATE
F SEE SHEET ONE J.K. 5-81
J7
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
DO1 DO2 DO3 DO4 DO5 DO6 DO7 DO8
START 36 W/L VOH/VOL A B 26 CLOCK IHB R̄ PP RR READ FWD WRITE
DI8 DI7 DI6 DI5 DI4 DI3 DI2 DI1
RP4 3.3K (8 PLCS)
RP6 3.3K (8 PLCS)
RP7 3.3K (8 PLCS)
RP5 3.3K (8 PLCS)
R31 3.3K
ADDRESS BUS
CONTROL
LO ADDRESS GATE
U43 74LS244
HI ADDRESS GATE
U50 74LS244
LO ADDRESS REG
U42 74LS374 E201
HI ADDRESS REG
U51 74LS374 E200
I/O DECODE A
I/O DECODE B
DATA REG
U44 74LS374 E202
I/O DECODE C
DREG GATE
U41 74LS126
U40 74LS04
DREG SHT 3 3D
U31, 2 C1
C1 GATE
74LS126
RP4 3.3K
74LS04
I/O DECODE D
DATA GATES
U47 8304
DATA BUS B
STATUS GATES
U45 8304
S1 U31, 19
I/O DECODE E
CRR SHT 1 5A
U46 74LS174 E203
STATUS REG
I/O DECODE F
WRITE GATE
U55 74LS00
R/W̄
VØ2
U31 74LS273 E000
C1 SHT3.5D
S1 SHT 3.4C
DREG SHT 3,4A
NC
COMP SHT 4 6A
RUN SHT4 6B
REG SHT 4 6B
I/O DECODE K
CONT REG
TONE CONTROL GATE
TONE CONTROL U55, 6
U41
J8 +5V
I/O DECODE
U49 32 X 8 PROM
RP8 330 (8 PLCS)
E200 E201 E202 E203 E204 E205 E206 E207
U22 256 X 4 TRI STATE
E200 E100 E000 RAMS
PRIMARY I/O DECODE
U21 256 X 4 PROM
R61 R30 RP2 R58 330 (4 PLCS)
BUFFER EN SHT 1 7D
CS SHT 2 7A
I/O CS DECODE
R/W̄B SHT1, 7A
VØ2B SHT1, 7A
+18V +48V +HV SENSE -5V +24V -9V +10V +PROG +5V
SENSE SHT5, 2A
+PROG SHT5, 2A
I/O DECODE D
INTERLOCK SHT1, 5A SHT5, 2B
OPERATE SHT 5, 3A
NC
IRQ R/W̄ VØ2 GATE ENABLE INTERLOCK GND OPERATE EXTEND
APPROVALS
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES UNLESS OTHERWISE SPECIFIED .XX .XXX ANGULAR
DSN ENGR. MFG. ENGR. QUAL ASSUR. ENGR MGR
DRAWN BY Jaehee Kim 2/81
CHECKED BY
DATA I/O ISSAQUAH WASH
TITLE SCHEMATIC DIAGRAM CONTROLLER BOARD
SIZE D CODE INDENT. NO 54193
DRAWING NO 30-702-1980
SCALE
SHEET 3 OF 5

REVISIONS					
LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
F	SEE SHEET ONE	J.K			5-81



POWER SUPPLY CHART

CHIP REF	GND	+5V	+12V	-5V
U1-U4	7	14		
U5	8	16		
U6	7	14		
U9	7		14	1
U10	7	14		
U11, U12	10	20		
U14	1	8,10,12		
U15-U19	9	18		
U20	20	40		
U21, U22	8	16		
U23	10	20		
U24, U25	8	16		
U26-U30	9	18		
U31	10	20		
U32-U39	12	24	19	21
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 ISSAQUAH, WASH.

DSN. ENGR. — DATE
MFG. ENGR. — DATE 5-5-81
QUAL. ASSUR. — DATE
ENGR. MGR. — DATE

TOLERANCES, UNLESS OTHERWISE SPECIFIED: .XX ± .XXX ± ANGULAR

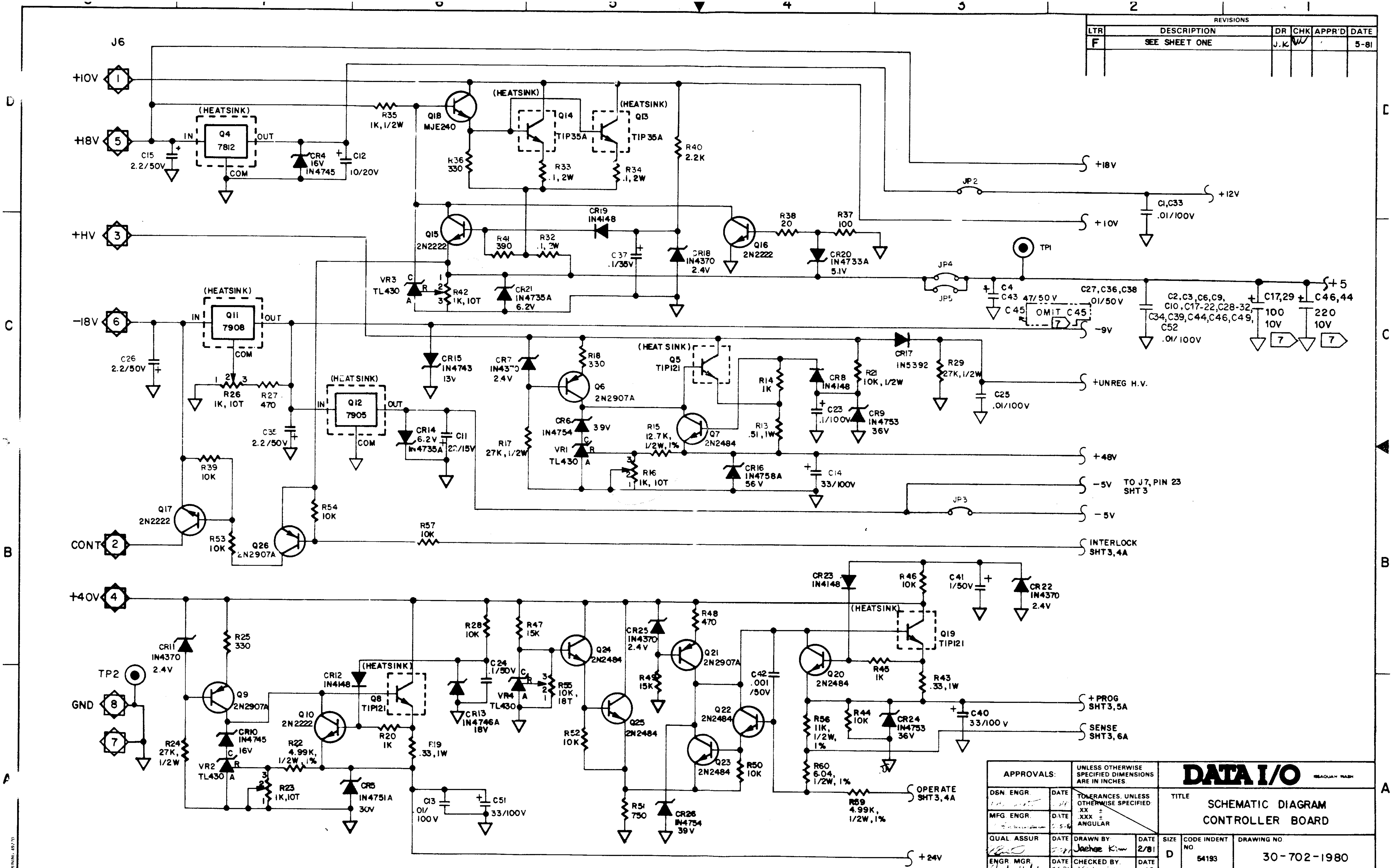
DRAWN BY: Joohee Kim — DATE 2/81
CHECKED BY: — DATE 5-481

TITLE: SCHEMATIC CONTROLLER BD.

SIZE D | CODE IDENT. NO. 54193 | DRAWING NO. 30-702-1980

SCALE NONE | SHEET 4 OF 5

BRUNING 49270

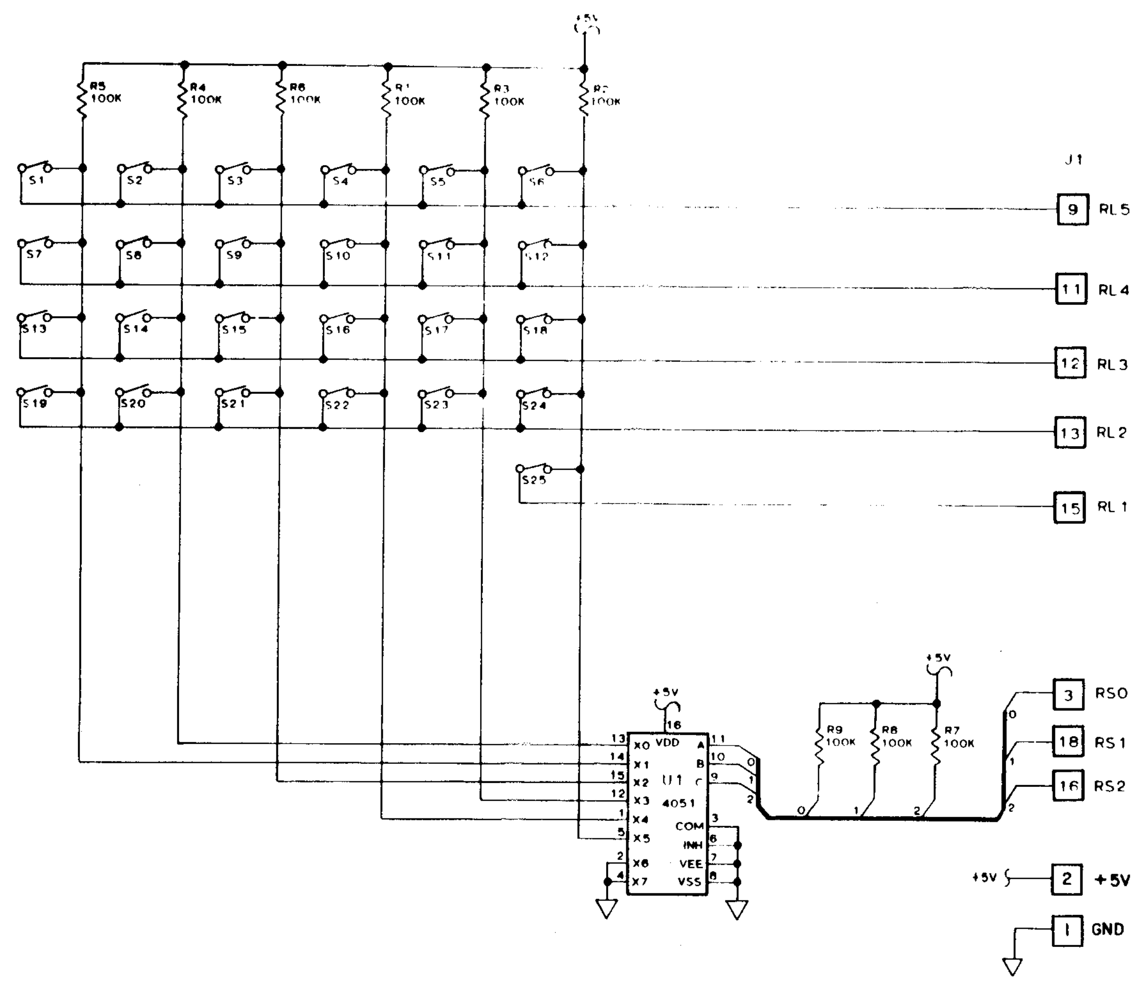


REVISIONS

LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
F	SEE SHEET ONE	J.K			5-81

APPROVALS:	UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES	DATA I/O ISSAQUAH WASH	
DSN ENGR / DATE	TOLERANCES, UNLESS OTHERWISE SPECIFIED .XX ± .XXX ± ANGULAR	TITLE SCHEMATIC DIAGRAM CONTROLLER BOARD	
MFG ENGR / DATE			
QUAL ASSUR / DATE	DRAWN BY Jaehee Kim DATE 2/81	SIZE D CODE IDENT NO 54193	DRAWING NO 30-702-1980
ENGR MGR / DATE	CHECKED BY / DATE	SCALE	SHEET 5 OF 5

LTR	DESCRIPTION	DR	CHK	APPR'D	DATE
A	RELEASE	NW			7-8-81



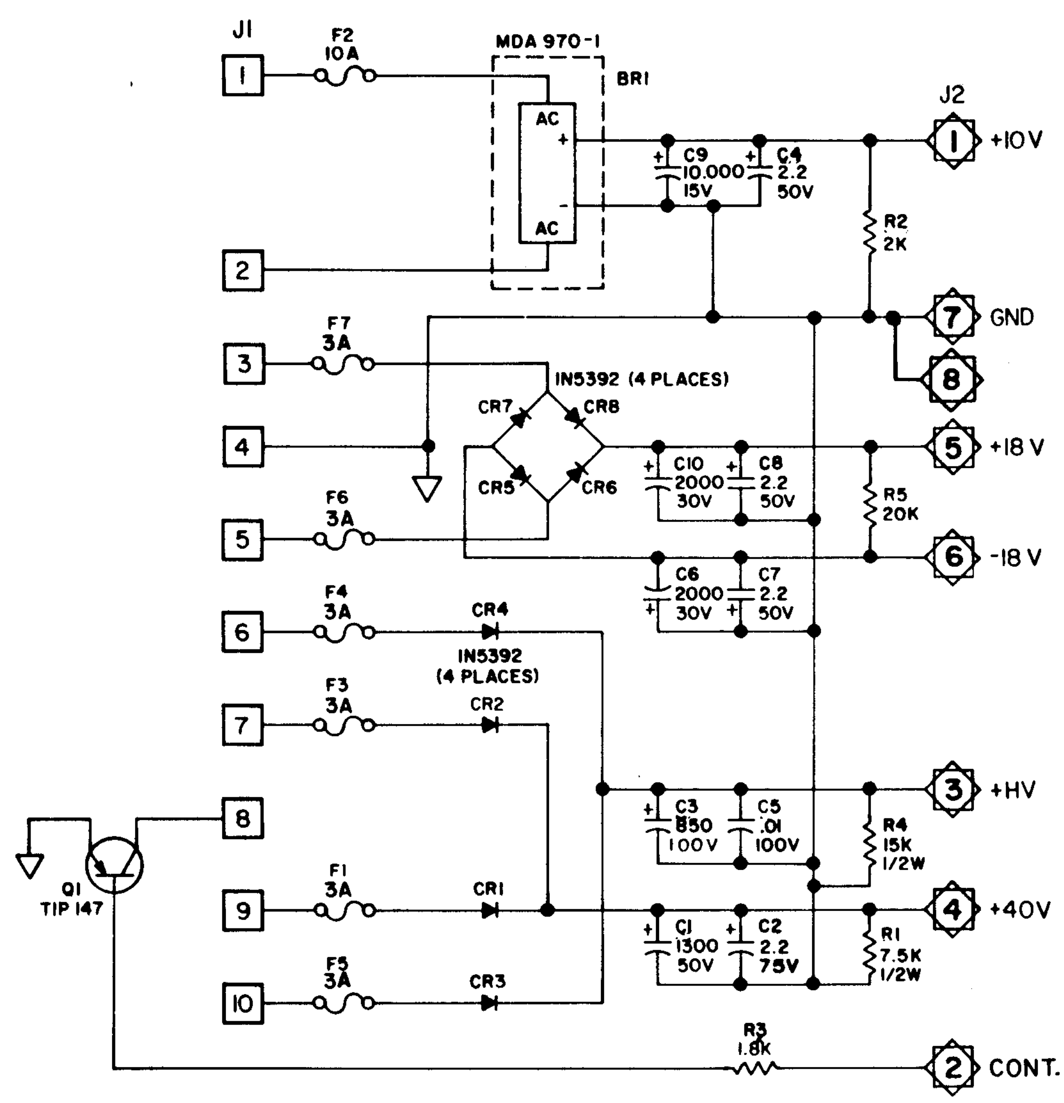
NOTES UNLESS OTHERWISE SPECIFIED
1 ALL RESISTORS ARE 1/4W AND IN OHMS, 5%
2 LAST REFERENCE DESIGNATOR USED: R9, J1, S25, U1
3 702-1648-001 SHOWN

DATA I/O ISSAQUAH WASH

SCHEMATIC DIAGRAM
MODEL 29A
KEYBOARD

SIZE D — CODE IDENT NO 54193 — DRAWING NO 30-702-1648 — SHEET 1 OF 1

ZONE	LTR	DESCRIPTION	CK	PE	APP	DATE
	A	RELEASE				1-80
	B	ECN 4218				5-81
	C	ECN 4299	KAT			8-81



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 CR8
BR1 C10
F7 R5
4. -001 SHOWN

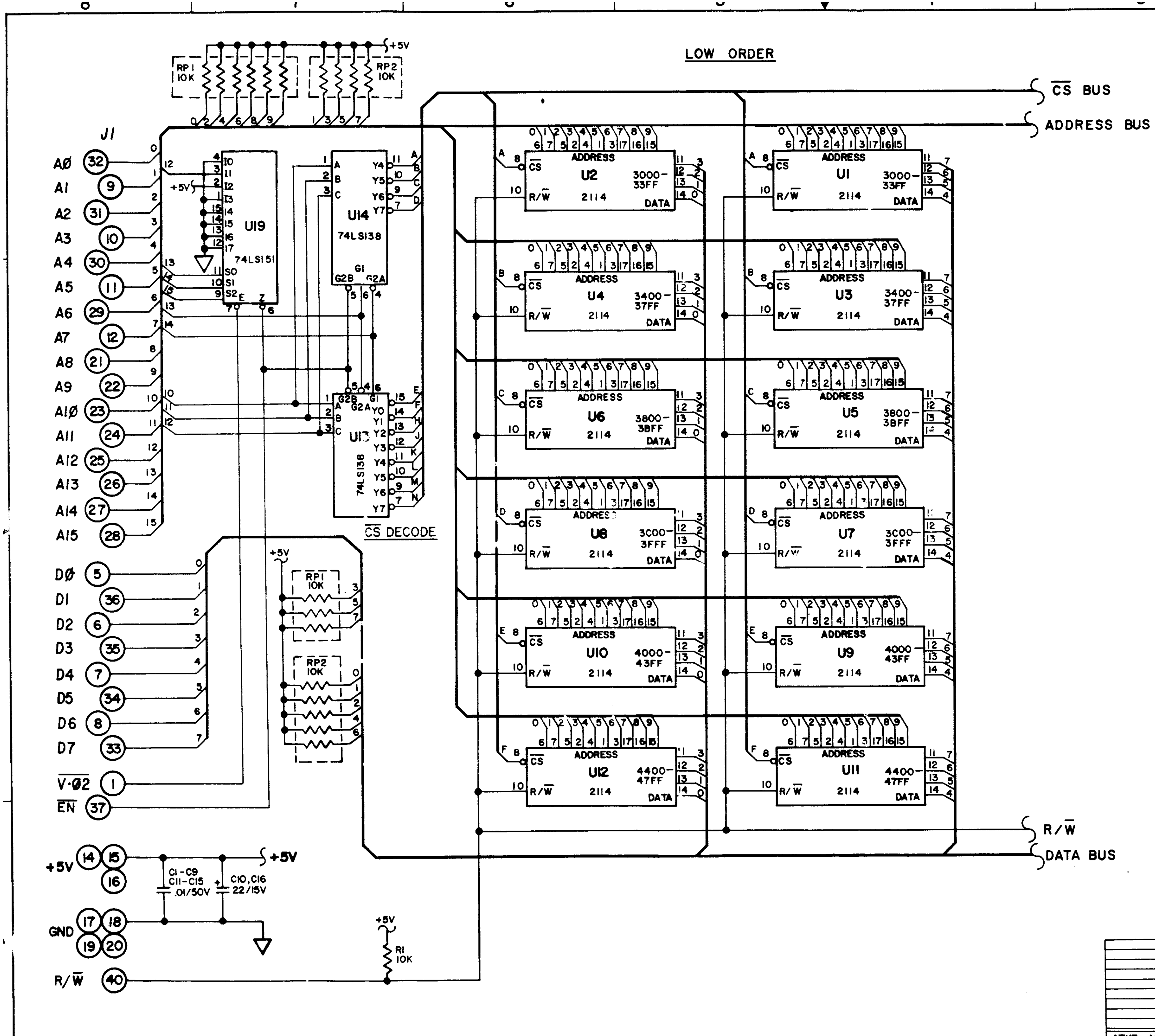
THIS DOCUMENT CONTAINS INFORMATION CONSIDERED PROPRIETARY, AND SHALL NOT BE REPRODUCED WHOLLY OR IN PART, NOR DISCLOSED TO OTHERS WITHOUT THE SPECIFIC WRITTEN PERMISSION OF DATA I/O CORP.

DATA I/O ISSAQUAH WASH

TITLE SCHEMATIC DIAGRAM
FILTER BOARD

SIZE C — DRAWING NO. 30-702-1982 — SCALE NONE — SHEET 1 OF 1

DATE 1-28-80



REVISIONS						
ZONE	LTR	DESCRIPTION	CK.	PE.	APP	DATE
	A	RELEASE				3-81

NOTES:

UNLESS OTHERWISE NOTED

1. ALL CAP. VALUES ARE IN MICROFARADS.
2. U13, U14, U19 PIN 8 = GND, PIN 16 = +5v.
3. U1-U12 & U15-U18, U20-U27 PIN 9 = GND, PIN 18 = +5v.
4. -001 SHOWN.

DATA I/O

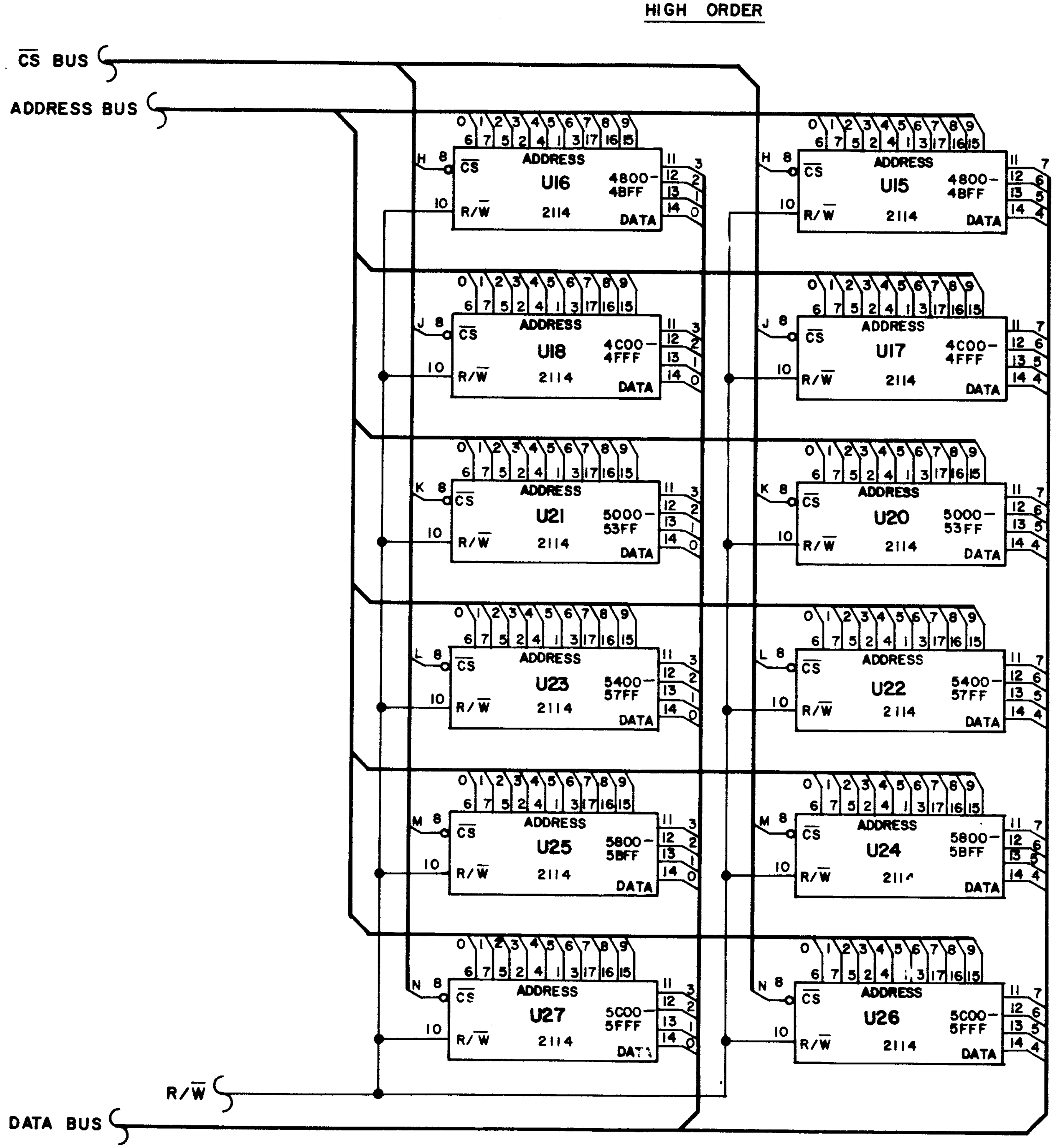
TITLE SCHEMATIC DIAGRAM EXTENDED MEMORY

APPROVALS:	DATE	TOLERANCES
PROJ. ENG.		DECIMALS .X ± .XX ± .XXX ± ANGULAR
MFG. ENG.		DO NOT SCALE DRAWING
Q.C.		DRAWN BY JOBK, DATE 2/81
ENG. MGR.		APPROVED BY, DATE 4/81

SIZE D | CODE IDENT NO | DRAWING NO 30-702-1983

SCALE NA | SHEET 1 OF 2

NEXT ASSY.



ZONE	LTR	DESCRIPTION	CK.	PE.	APP	DATE
	A	SEE SHT. 1				3-81

APPROVALS: DATE — UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES

TOLERANCES: DECIMALS .X ± .XX ± .XXX ± ANGULAR — DO NOT SCALE DRAWING

PROJ. ENG. / MFG. ENG. / Q.C. / ENG. MGR. / DRAWN BY: / APPROVED BY:

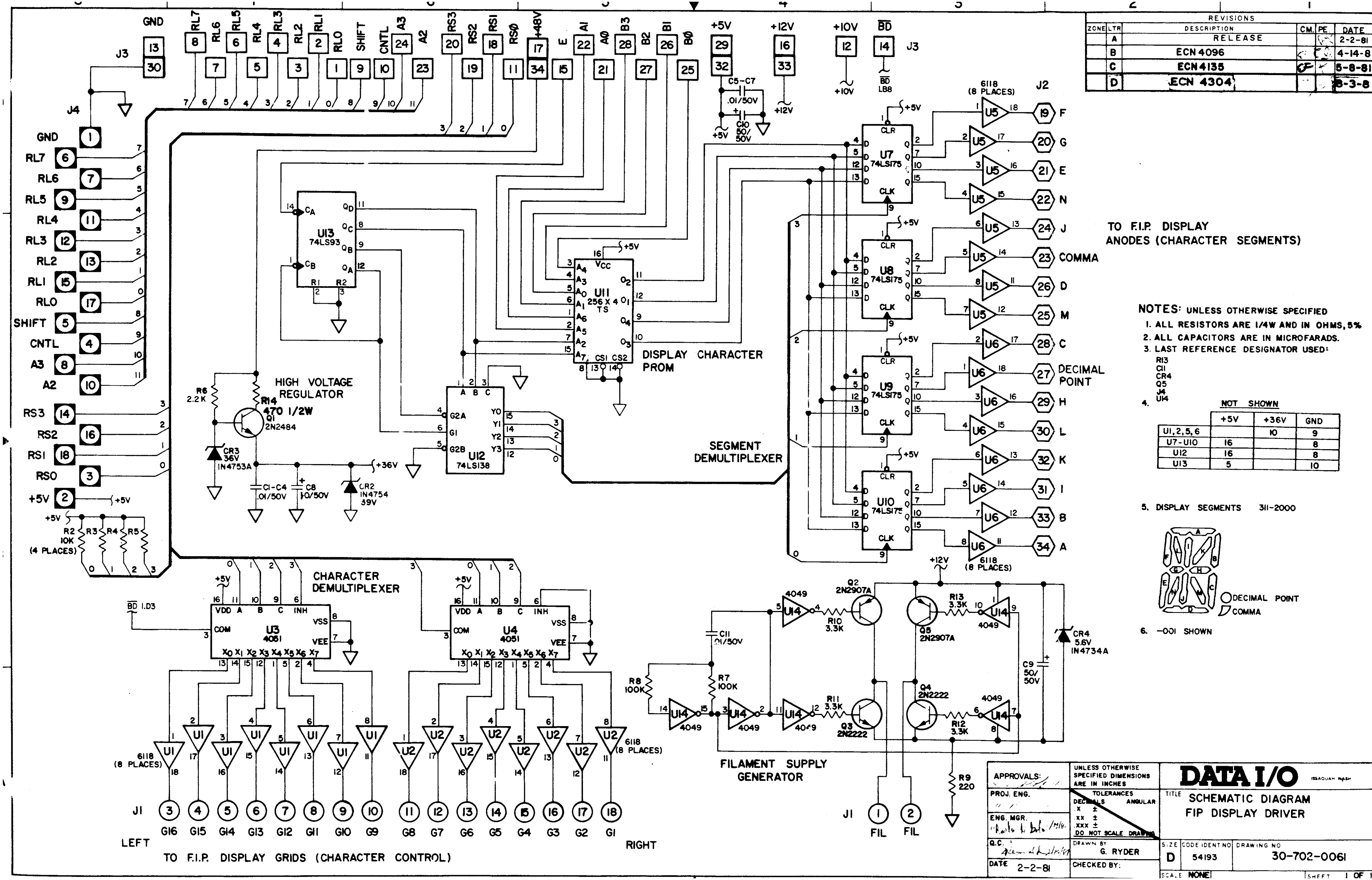
NEXT ASSY.

DATA I/O ISSAQUAH WASH

TITLE: SCHEMATIC DIAGRAM EXTENDED MEMORY

SIZE D — CODE IDENT NO. — DRAWING NO 30-702-1983

SCALE — SHEET 2 OF 2



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
J3
J4
J2
J1
GND
RL7 RL6 RL5 RL4 RL3 RL2 RL1 RL0 SHIFT CNTL A3 A2 RS3 RS2 RS1 RSØ +48V E A1 AØ B3 B2 B1 BØ +5V +12V +10V BD
C5-C7 .01/50V
C10 50/50V
HIGH VOLTAGE REGULATOR
R6 2.2 K
R14 470 1/2W
Q1 2N2484
CR3 36V 1N4753A
C1-C4 .01/50V
C8 10/50V
CR2 1N4754 39V
+36V
R2 10K (4 PLACES)
R3 R4 R5
U13 74LS93
U12 74LS138
U11 256 X 4 TS
DISPLAY CHARACTER PROM
SEGMENT DEMULTIPLEXER
U7 74LS175
U8 74LS175
U9 74LS175
U10 74LS175
6118 (8 PLACES)
F G E N J COMMA D M C DECIMAL POINT H L K I B A
TO F.I.P. DISPLAY ANODES (CHARACTER SEGMENTS)
CHARACTER DEMULTIPLEXER
BD 1.D3
U3 4051
U4 4051
FILAMENT SUPPLY GENERATOR
U14 4049
Q2 2N2907A
Q5 2N2907A
Q3 2N2222
Q4 2N2222
R10 3.3K
R11 3.3K
R12 3.3K
R13 3.3K
R7 100K
R8 100K
R9 220
C11 .01/50V
C9 50/50V
CR4 5.6V 1N4734A
FIL FIL
G16 G15 G14 G13 G12 G11 G10 G9 G8 G7 G6 G5 G4 G3 G2 G1
LEFT
RIGHT
TO F.I.P. DISPLAY GRIDS (CHARACTER CONTROL)
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:
R13
C11
CR4
Q5
J4
U14
4. NOT SHOWN
| +5V | +36V | GND
U1, 2, 5, 6 | | 10 | 9
U7-U10 | 16 | | 8
U12 | 16 | | 8
U13 | 5 | | 10
5. DISPLAY SEGMENTS 311-2000
DECIMAL POINT
COMMA
6. -001 SHOWN
APPROVALS
PROJ. ENG.
ENG. MGR.
Q.C.
DATE 2-2-81
UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES
TOLERANCES
DECIMALS ANGULAR
.X ±
.XX ±
.XXX ±
DO NOT SCALE DRAWING
DRAWN BY G. RYDER
CHECKED BY:
DATA I/O
ISSAQUAH WASH
TITLE SCHEMATIC DIAGRAM FIP DISPLAY DRIVER
SIZE D
CODE IDENT NO 54193
DRAWING NO 30-702-0061
SCALE NONE
SHEET 1 OF 1