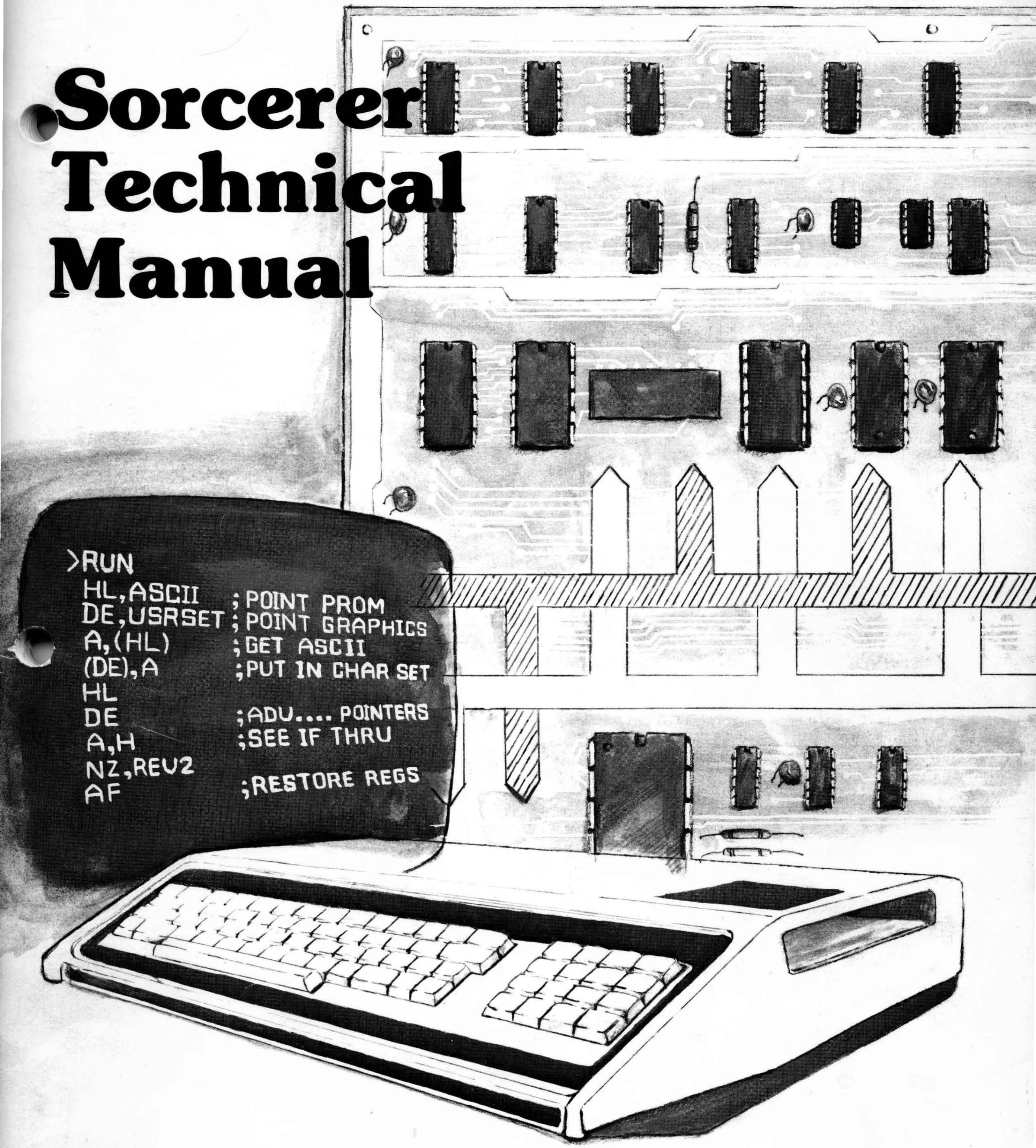
Sorcerer Technical Manual



```
>RUN
 HL,ASCII   ;POINT PROM
 DE,USRSET  ;POINT GRAPHICS
 A,(HL)     ;GET ASCII
 (DE),A     ;PUT IN CHAR SET
 HL
 DE         ;ADU.... POINTERS
 A,H        ;SEE IF THRU
 NZ,REV2
 AF         ;RESTORE REGS
```

By the Sorcerer of Exidy inc.

COPYRIGHT 1979 by EXIDY INCORPORATED
ALL RIGHTS RESERVED
390 Java Drive
Sunnyvale, California 94086

**FIRST EDITION
March 1979**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system (e.g., in memory, disk, or core) or be transmitted by any means, electronic, mechanical, photocopy, recording, or otherwise, without prior written permission from the publisher.

PRINTED IN U.S.A.

SORCERER TECHNICAL MANUAL
CONTENTS

ILLUSTRATIONS

TABLES

FOREWORD

This manual has two parts, covering Sorcerer software and hardware. The software material is useful to anyone who wants to master the Sorcerer's operating system. Readers should be familiar with hexadecimal notation and should recognize Z80 Assembly or machine instructions, but do not need much programming experience. Readers who are new to microcomputing should first read the Sorcerer's operating manual, A Guided Tour of Personal Computing, and then skim through the Sorcerer Development PAC^{TM} instruction manual, or any standard Z80 Assembly Language reference manual.

The hardware material is written for highly experienced service technicians, although some of this information is useful to all Sorcerer owners. We assume the reader is as qualified as our own test and service personnel. We strongly recommend that owners not attempt to repair or modify their own units.

All service should be done by an authorized Sorcerer dealer; unauthorized service will void our warranty.

PART ONE
SOFTWARE

MEMORY ADDRESSING AND ALLOCATION

Address, Input/Output Port, and Data Notation

The Sorcerer's Z80 CPU has sixteen address lines; every memory storage location has a sixteen digit binary number as its address. This means that the Sorcerer will recognize up to 65536 different memory addresses. (65535 is the largest number which can be written in no more than sixteen binary digits; but we start numbering the addresses at 0, instead of 1, so the total number is 65536).

It is more convenient to refer to these addresses in hexadecimal (base 16) notation than in binary (base 2) or decimal (base 10) notation. Any binary number with sixteen or fewer digits can be written as a hexadecimal number with four or fewer digits. Whenever we refer to a memory address we will always use a four digit hexadecimal number. To avoid confusion, we write the letter H at the end of the number to indicate ***hexadecimal*** notation. If the address is lower than 1000H, we add zeroes on the left to make four digits total. (Example: the second address in memory is 0001H.)

The Z80 has eight data lines — one for each bit of storage in a memory location. Every memory location contains a number from 0 to 255. (255 is the largest number which can be written in eight or fewer binary digits.) Again we use hexadecimal notation for the memory contents; we write each such number as two hexadecimal digits, followed by an H. If the number is less than 10H, we add a zero on the left to make two digits total. (Example: we write 12 decimal as OCH.)

NOTE

If the Z80 is instructed to read a memory address which is ***not*** connected to any RAM or ROM, it will usually assume the data is FFH.

Similarly, the first eight address lines can designate an input/output port (I/O port). Since these eight lines can be set to any number from 0 to 255, there are 256 I/O ports. We number them in hexadecimal, OOH to FFH.

Sorcerer Memory Map

Figure 1 shows the Sorcerer's memory allocations. Column A is the standard configuration (8K internal RAM and an 8K ROM PAC^{TM}); the diagram is not drawn to scale. Column B is the same as Column A, but redrawn to approximate scale.

Columns C and D show alternative configurations. The lower portions show the addresses of internal RAM in the 16K and 32K Sorcerers, while the upper portions show the addresses of the 4K and 16K ROM PACs.

The unassigned addresses between the top of internal RAM and the bottom of the ROM PAC can be given to the S-100 Expansion Unit. If the ROM PAC is unplugged, its address space is also available to the S-100 unit.

The Monitor stack, consisting of the Monitor RAM and the stack proper; is shown at the top of the internal RAM, the position it usually takes at power-on or reset (see Relocating the Monitor Stack, for details). But, note that if a block of expansion memory is assigned immediately above internal RAM, the stack will go to the top of that block.

Example:

In the 16K Sorcerer at power-on or reset, the Monitor stack occupies addresses 3F50H to 3FFFH. If 4K of memory (1000H addresses) is added to the Expansion Unit and assigned addresses 4000H through 4FFFH, the Monitor stack will occupy 4F50H through 4FFFH at power-on. However, if the new memory is assigned to 5000H through 5FFFH, the stack will remain at 3F50H to 3FFFH.

Screen RAM

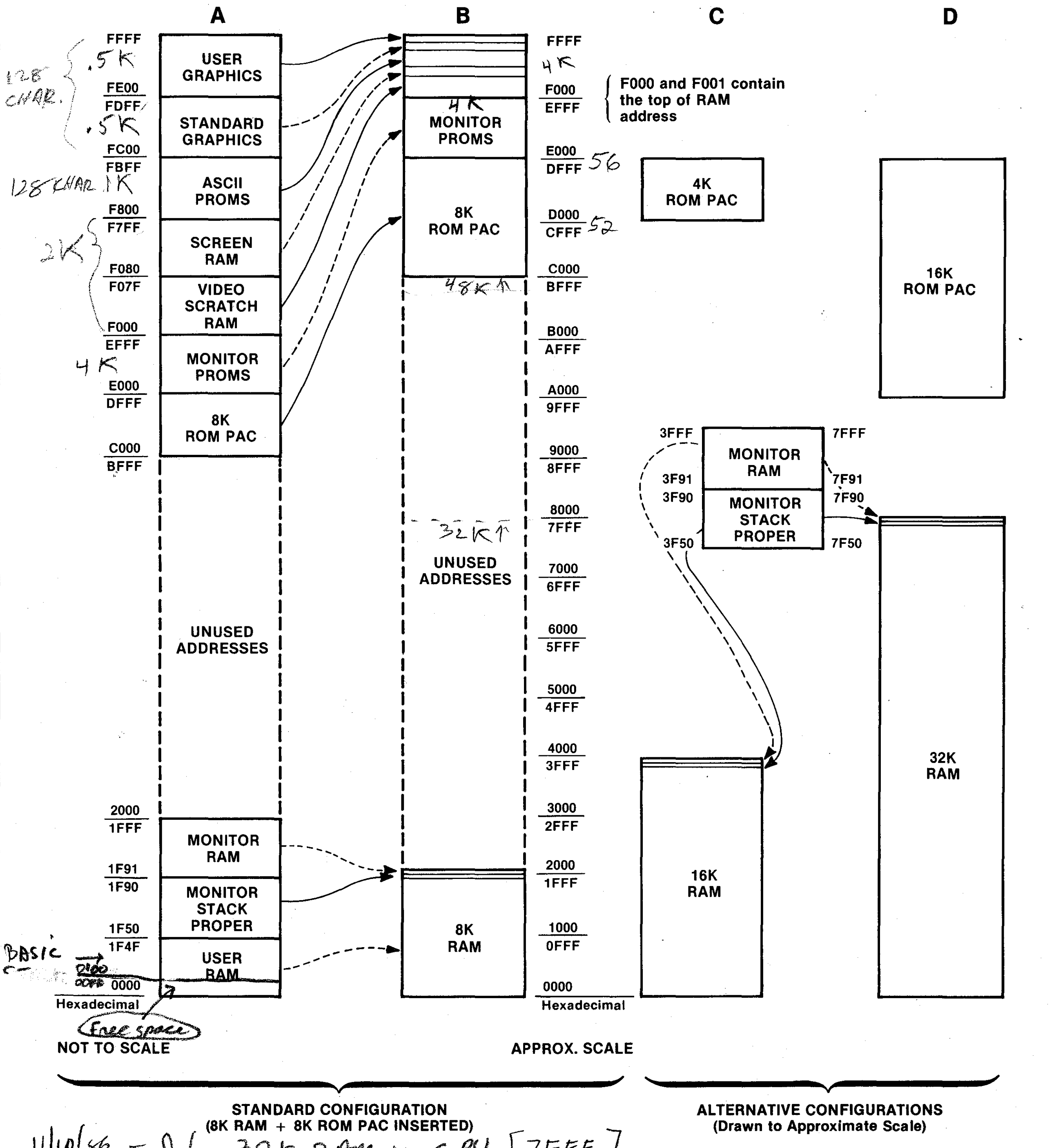
Every character (alphanumeric, graphic, or user-defined) is printed on the screen as an 8 x 8 array of dots. Each dot can be either ON (a white spot on the screen) or OFF (a dark spot). The exact shape of the character depends on which dots are turned on, and which are turned off.

Each row of the character is stored in the Sorcerer's character generator, in a separate memory address, with each dot of the row stored as one bit of that address. A 0 bit means OFF, and a 1 bit means ON.

To write a character onto the Sorcerer's screen, put its ASCII number into the screen RAM. The exact address you use will determine where the character will appear on the screen. The ASCII number directs the Sorcerer to eight successive addresses in the RAM or ROM character generator. These addresses store the eight lines of the character's dot matrix.

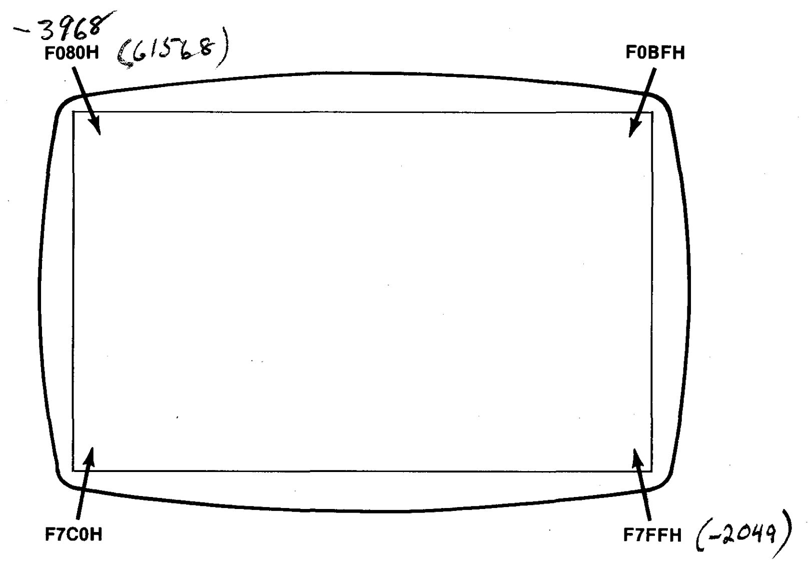
Each text line on screen is 64 characters long (40H), and there are 30 lines in all (1EH). The first address in screen RAM (F080H) is for the upper left corner of the screen. The next address (F081H) is for the second place in the first line, and so on to the end of the line (FOBFH). The pattern continues with the first place on the second line (FOCOH) and so forth. The last address (F7FFH) is for the lower right corner of the screen.

Figure 1. Sorcerer Memory Map



11/10/86 – I have 32K RAM in CPU [7FFF$_H$]
(+) 24K on 32K RAM CARD. [DFFF$_H$]
with switch to DC00, Top RAM = DBFF$_H$ (Applies to switch on S100 Box; switch
" " " FC00 " " = DFFF$_H$. on CPU has no effect).
I am using 48K CPM ['BFFF$_H$] at home; 46K at VAMC.
11/27/86 - Disabled top 4K on RAM card. Top RAM CFFF$_H$ [corrected failure to load at D C00]

Figure 2. Screen RAM Addresses for the Corners of the Screen



POWER-ON MONITOR

General Description

The Power-On Monitor (or Monitor, for short) is the Sorcerer's software operating system; it gives easy access to the Z80 microprocessor and control of peripheral devices. The Monitor occupies 4K of ROM, addresses E000H to EFFFH, and contains full software drivers for the keyboard and video, cassette tape units, and a Centronics printer. The Monitor's command set includes tape unit batch commands, which allow automatic processing. Monitor routines find the top of RAM and use the top 176 bytes (approximately) as the Monitor storage stack; this stack can be relocated anywhere in RAM.

The Power-On Monitor Command Set

The Monitor's commands are listed below by type.

Memory Test and Manipulation

DUMP

Format: DU XXXX (XXXX)
addr1 addr2

Remarks: Displays memory on the screen. If you give one address, you get the contents of that address; if you give two addresses, you get the contents of all addresses from the first to the second.

Examples:

DU 0100	Displays the contents of address 0100H.
DU FE00 FE07	Displays the contents of all addresses from FE00H to FE07H.

ENTER

Format: EN XXXX
addr

Remarks: Lets you enter hexadecimal numbers into the memory address. When you hit the carriage return, Sorcerer advances to the next address; when you hit carriage return alone, you just get the current address. A / (slash), followed by the carriage return, gets you out of ENTER.

Examples:

Entering data one address at a time:

EN FE00	The command to enter data, starting at FE00H.
FE00: FF	Sorcerer prints the addresses and colons.
FE01: 00	You must enter the numbers (following each with a RETURN).
FE02: FF	
FE03: 00	
FE04: /	The slash terminates the EN command.

Entering a block of data:

EN FE04 The command to enter data, starting at FE04H.

FE04: FF 00 FF 00 Sorcerer prints the address and colon; you must enter the numbers, separated by delimiters (in this case, spaces).

FE08: / Sorcerer reports the next available address; the slash terminates the EN command.

TEST

Format: TE XXXX XXXX (C)
addr1 addr2

Remarks: Tests each bit of RAM from address 1 to address 2 (8 bits per word). A blinking * appears while the test is running. No message if memory is good; otherwise prints BAD or OK for each bit. With C, tests continuously, giving a PASS COMPLETED message at each pass. Use this command only for testing RAM; all ROM addresses will be reported BAD, no matter what their condition.

NOTE

Bug in Monitor 1.1 [See SA 4(1):16 + 4(4):74]

NOTE

The bit test (command TE) will not give reliable results on the area of RAM occupied by the Monitor stack. If you wish to test this portion of RAM, first relocate the Monitor stack, as described under that heading. For the addresses of the stack, see Figure 1. p.19

You must also take special precautions when testing screen RAM and video scratch RAM (addresses F000H to F7FFH). For details see Performance Tests, Logic Board (under RAM bit test). p.30

MOVE

Format: MO XXXX XXXX (S) XXXX
addr1 addr2 block/addr3

Remarks: Copies the contents of a block of successive memory addresses into another block of addresses. Use this command two ways:

1. If you include the S, then the following hexadecimal number tells Sorcerer how many addresses to move. The contents of address 1 go into address 2, and the addresses following address 2 are copied after address 2.

2. If you omit the S, then Sorcerer moves the block of memory starting at address 1 and ending at address 2. The contents of address 1 go into address 3, and the following addresses are copies after address 3.

Examples:

MO FC00 FE00 S0010 This copies the 16 addresses (10H) starting at FC00H, into the 16 addresses starting at FE00H.

MO FC00 FC0F FE00 This copies the 16 addresses from FC00H to FC0FH into the 16 addresses starting at FE00H.

Program Storage and Loading

SAVE

Format: SA NAMEX XXXX XXXX (X)
name addr1 addr2 unit

Remarks: Saves memory on tape from the first address to the second address, on the tape unit specified (or on unit #1, if you don't specify). The header on the tape will contain the name (which must start with a letter), the beginning address, block size, file type, and GO address (see SE command).

Example:

SA TEST1 0100 0240 2 Saves the program in addresses 0100H to 0240H on tape unit #2, under the name TEST1.

FILES

use to verify SAVE

Format: FI (X)
unit

Remarks: Lists all cassette files and header information from the specified unit (or on unit #1, if you don't specify). CTRL C stops the listing and returns you to the Monitor. This command will list BASIC programs stored with CSAVE, but will **not** list arrays saved with CSAVE *

LOAD (G)

Format: LO(G) (NAMEX) (X) (XXXX)
name unit addr

Remarks: Loads the file named NAMEX into memory, from the tape unit specified (or unit #1, if you don't specify). Starts loading at the given address if specified; otherwise at the address in the file header on the tape. If you specify an address you must also specify the unit. If you don't give a name, then the first file will be loaded. If you include the G after LO, Sorcerer will run the program starting at the GO address in the file header, immediately after loading. This command will only load files saved in the Monitor. A file saved in Standard BASIC must be loaded with the command CLOAD.

Examples:

LO TEST1 Loads the program TEST1 from tape unit #1, starting at the address in the file header.

LO TEST1 2 Loads the program TEST1 from tape unit #2, starting at the address in the file header.

LO TEST1 1 0100 Loads the program TEST1 from tape unit #1 starting at the address 0100H.

LOG Loads the first program on tape unit #1 starting at the address in the file header, and runs the program.

LOG 2 0100	Loads and runs the first program on tape unit #2, starting at address 0100H; then jumps to the GO address in the file header.
LOG TEST1	Loads and runs the program TEST1 from tape unit #1, starting at the address on the file header.

Control Transfer

GO

Format: GO XXXX
addr

Remarks: Calls a program at the given address.

PP

Format: PP (X)

Remarks: Jumps to ROM PAC. If no parameter, then warm start, otherwise cold.

System Adjustments

PROMPT Removed in Vers 1.1

Format: PR=X

Remarks: Changes the Monitor > to whatever you specify. For example, PR= # changes the prompt to #.

[Default]

SET

Format: SE A=B

Remarks:

S=XX	[Ø]	Changes display delay to XX.
* T=X	[Ø]	Sets tape rate to 0=1200 baud (default); 1=300 baud. Cassette
F=XX		Sets file type in tape header to ASCII value of XX. (See note below.) If D8 is set, then file is non-auto-execute.
X=XXXX		Sets auto execution address in tape header.
O=V		Sets current output port to VIDEO
=P		PARLOT (parallel output)
=L	Parallel Port	CENTRONICS PRINTER DRIVER
=S		OUTAPE (tape output)
=XXXX		ADDRESS
I =K		Sets current input port to KEYBRD
=P		PARLIN (parallel input)
=S		INTAPE (tape input)
=XXXX		ADDRESS

	* T= Ø	1200	Baud cassette	
	1	300	"	"
Vers. 1.1	2	1200	Baud	RS232
	3	300	"	"

NOTE

There are 256 file types (00H through FFH). D8 is non-auto-execute. All others are arbitarily user assignable. However, BASIC will only load a file of type B – (i.e., most significant digit is B).

Examples:

SE S=FF	Sets the display delay to FFH.
SE F=D8	Sets file type to non-auto-execute.
SE O=L	Sets output to Centronics Printer.
SE O=1000	If you have loaded an output driver routine at address 1000H, this command sets the Sorcerer's output to that routine.

Batch Commands

CREAT

Format: CR

Remarks: Creates a batch tape on tape unit #1. Enter your batch commands a line at a time after the * prompts, and end each line with a carriage return. The tape will start, record your line, and stop. To re-enter the Monitor, hit the carriage return alone.

LIST

Format: LI

Remarks: Lists all the commands on your batch tape.

BATCH

Format: BA

Remarks: Tells Sorcerer to read the batch tape on tape unit #1 and execute all the commands on that tape. You can stop the routine by typing CTRL C, ESC or RUN/STOP; Sorcerer will stop automatically if it reads an invalid command, or if any error occurs.

OVER

Format: OV

Remarks: This is the batch command which terminates the batch mode and returns you to Monitor. This is similar to the END statement in Standard BASIC; it should be the last command on your batch tape.

Entry Points (Jump Table)

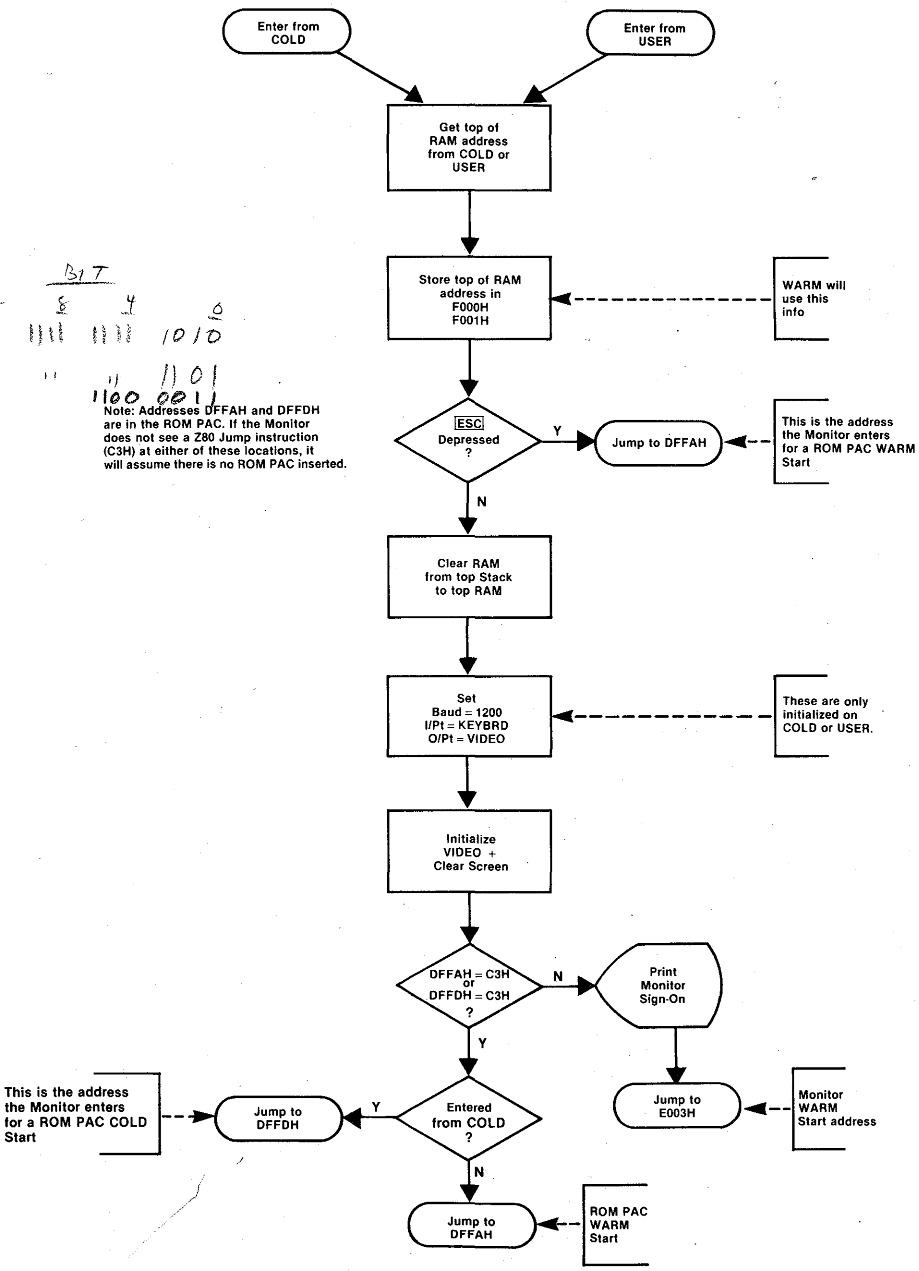
The Power-On Monitor has three initial entry points, nine I/O entry points, and two tape unit entry points.

Initial Entry

Address	Name	Remarks
E000H	COLD	Sorcerer enters at this address when first turned on, or when the RESET keys are depressed. The Monitor searches for the top of contiguous RAM, starting at address 0000; it then jumps to the initialization routine (see Figure 3).
E003H	WARM	This address is used to re-enter the Monitor. Sorcerer establishes the Monitor stack from the top RAM address contained in F000H and F001H. It then prompts the user and looks for a Monitor command. WARM does not change any of the system parameters (baud rate, file type, prompt character, etc.).
E006H	USER	Entry at this address allows relocation of the Monitor stack after power-on. (See Relocating the Monitor Stack.) Sorcerer reads the HL register pair to find the last byte of available RAM; it then jumps to the initialization routine (see Figure 3).

Figure 3. Power-On Monitor Initialization Routine

BIT
12 8 4 0
DFFA 1101 1111 1111 1010
DFFD " " " 1101
1100 0011
C3


Enter from COLD
Enter from USER
Get top of RAM address from COLD or USER
Store top of RAM address in F000H F001H
WARM will use this info
Note: Addresses DFFAH and DFFDH are in the ROM PAC. If the Monitor does not see a Z80 Jump instruction (C3H) at either of these locations, it will assume there is no ROM PAC inserted.
ESC Depressed ?
Y
Jump to DFFAH
This is the address the Monitor enters for a ROM PAC WARM Start
N
Clear RAM from top Stack to top RAM
Set Baud = 1200 I/Pt = KEYBRD O/Pt = VIDEO
These are only initialized on COLD or USER.
Initialize VIDEO + Clear Screen
DFFAH = C3H or DFFDH = C3H ?
N
Print Monitor Sign-On
Y
This is the address the Monitor enters for a ROM PAC COLD Start
Jump to DFFDH
Y
Entered from COLD ?
Jump to E003H
Monitor WARM Start address
N
Jump to DFFAH
ROM PAC WARM Start

To set Z flag means it is in Z state [actually contains a 1]   POSITIVE LOGIC 1 = +3-5V

To reset Z flag returns it to NZ state [Actually contains a Φ]   Positive logic Φ = GRND

I/O Entry Points

The Monitor has two I/O vector entry points: an input and an output. The Monitor SE command allows you to route these vectors to any peripheral device. When Sorcerer is first powered on, these vectors are initialized to KEYBRD and VIDEO (see below). The other seven I/O entry points are for specific I/O devices.

Address	Name	Remarks
E009H	RECEVE	This is the I/O vector input point. You do not have to load any registers before entering. The A and F registers will change but all others remain the same. When you enter at RECEVE, the Monitor searches the assigned input device for a character. If it doesn't find one, it sets the Z flag and returns. If it does find a character, it loads it (in ACSII) into the A register, resets the Z flag, and then returns.
E00CH	SEND	This is the I/O vector output point. Before calling SEND, load the ASCII number of a character into the A register. This entry sends the content of the A register to the output device; no registers are changed. There is a built-in delay in output, which is determined by the Monitor command SE S = XX.
E00FH	INTAPE or RS232 I/O FC	This is the cassette tape input point. Only the A and F registers are disturbed. The Monitor will return from this routine on only two conditions: • If CTRL C, ESC, or RUN/STOP has been depressed, the Monitor returns with the Z flag set. • If a character has been read from the tape, the Monitor returns with the Z flag reset, and the character loaded in the A register.
E012H	OUTAPE } I/O FC or RS232 }	This is the cassette tape output point. Load a character (ASCII) into the A register before calling OUTAPE. The content of the A register is sent to the current cassette unit; no registers are changed.
E015H	QUIKCK	This entry does a quick check of CTRL C, ESC, and RUN/STOP; the A register is changed. The Monitor returns with the Z flag set if CTRL C, ESC, and RUN/STOP have ***not*** been pressed. If the Monitor returns with the Z flag reset, the A register will contain: • CTRL C (03H), if CTRL C was pressed. ΦΦΦΦ ΦΦ11 • ESC (1BH), if ESC or RUN/STOP was pressed. 0001 1Φ11
E018H	KEYBRD	This is the keyboard entry point. Entry here destroys the content of the A register. If no character is present, the Monitor will return with the Z flag set. If a character is present, it will be loaded into the A register, and the Monitor will return with the Z flag reset.
E01BH	VIDEO	This is the video output entry point. The content of the A register is printed on the video screen at the cursor position; no registers are altered. The cursor responds to control characters, as shown in Table 1. Shifts cursor re position to RT.

Table 1. Video Screen Cursor Control

Character	ASCII CODE (Hexadecimal)	Effect
Line Feed ^J	0A (10)	Moves the cursor down one line or, if it is already at the bottom of the screen, scrolls the screen up one line.
Form Feed ^L	0C (12)	Clears the screen and puts the cursor in the upper left corner (home position).*
RETURN	0D (13)	Positions the cursor at the leftmost position of the current line.
CTRL A	01 (1)	Moves the cursor one space to the left (without erasing) unless it is already at the left margin — if so, no effect.
CTRL H	08 (8)	Backspaces the cursor one position, erasing the character in the new cursor position.
CTRL Q	11 (17)	Puts the cursor in home position.
CTRL S	13 (19)	Moves the cursor one space to the right, unless it is already at the right margin — if so no effect.
CTRL W	17 (23)	Moves the cursor up one line, unless it is already at the top of the screen — if so, no effect.
CTRL Z	1A (26)	Same as Line Feed.
All other control characters below ASCII number 20H		No effect. 02, 03, 04, 05, 06, 07, 09, 0B, 0E, 0F

*The Form Feed also resets the standard graphics to their normal values (the symbols shown on the key caps).

10, 12, 14, 15, 16, 18, 19, 1B, 1C, 1D, 1E, 1F

E01EH	PARLIN	This is the parallel input entry point. The input character is loaded into the A register from I/O port FFH, with standard handshaking. Refer to Figure 16 for a timing chart of the handshake signals; refer to Table 6 for their Pin numbers.
E021H	PARLOT	This is the parallel output entry point. The content of the A register is sent to the output port (FFH).

Examples:

Here is a typical input routine:

```
CD 09 E0          INPUT: CALL RECEVE
CA XX XX          JP Z,INPUT
```

Notice that RECEVE is called over and over, until the Z flag shows something has actually been received. Note: Z flag means no character found.

Here is a typical output routine:

```
3E 48             LD A, 'H'
CD 0C E0          CALL SEND
```

This prints an H on screen at the cursor position.

Tape Unit Entry Points

Address	Name	Remarks
E024H	CMOTON	Entry at this point turns on the motor of one or the other tape unit, depending on the contents of the B register: if B is set to 1, unit #1, if B is set to anything else, then unit #2. Sorcerer will wait approximately three seconds before returning from this entry, to allow motor start-up. Entry at this point destroys the contents of both the A and B registers.
E027H	CMOTOF	Entry at this point turns off the motors of both tape units. On entry, there is approximately a one second delay; then the units are turned off and Sorcerer returns. The contents of both the A and B registers are destroyed.

Interfacing Programs

Passing control between user programs and the Monitor

The Monitor GO command can jump to any address in the memory and initiate a Z80 call at that address. On entry to a program, the stack pointer will be valid (pointing to a RAM location) and the first two addresses of the stack will contain a Monitor return address.

Therefore, a Z80 RET command (C9H) in your program will return you to the Monitor, provided your program does not destroy or alter the stack. If you do alter the stack, you should re-enter the Monitor at WARM (E003H); this will destroy your program stack, but re-establish the Monitor stack.

The only other way to enter a user program from the Monitor is with the Monitor LOG command, which loads a program from a tape unit and then jumps to the start address. Once again, a Z80 RET command (C9H) in the program will return you to the Monitor, if the stack is intact.

> SPECIAL CASE
>
> The Monitor PP command passes control to whichever ROM PAC is inserted. Refer to the initialization routines, Figure 3.

If your program alters the top of RAM, (where the Monitor RAM and stack proper are located), you should relocate the Monitor stack, as explained under that heading.

Input and Output, General

In general, you should route all I/O through the vector I/O points, RECEVE and SEND. This lets you assign peripheral devices to your program with the Monitor SE command. However, if you want to use a specific device, use the entry point for that device.

Examples:

Here is a typical input routine, using the I/O vector input:

```
CD 09 E0 INPUT:   CALL RECEVE
28 FB             JR Z, INPUT
```

The A register now contains a character.

Here is a typical output routine, using the I/O vector output:

```
3E 48             LD A, 'H'
CD 0C ~~D~~E0          CALL SEND
```

This sends the character H to the output device.

Alternatively:

```
78                LD A, B
CD 1B E0          CALL VIDEO
```

This outputs the contents of the B register to the screen.

Cassette I/O Routines (for use with the Serial Data Cable, Exidy Part No. DP4005)

In general, write your routines this way:

1. Turn on the specified unit with a CMOTON call.
2. Read or write data with an INTAPE or OUTAPE call.
3. Turn off the motor with a CMOTOF call.

When you write data on a cassette, you should prefix it with a header. We recommend a standard header of 100 zeroes followed by a one.

? Tyro 3E_H → LDA + LDB ?
SEE P.22

~~State indicated~~
6 LOGIC 1 = Hi = NZ

Bit 0 = RIGHT
LEAST
SIGN. BIT

Example:
This routine writes the contents of addresses 0000 to 00FFH on tape unit #1:

```
3E          WRITE:  LD B, 1
CD 24 E0            CALL CMOTON  Turns on the motor of
                                 unit #1.
3E 32 30            LD B, 32H ; = 50D
3E 00       NULL:   LD A, 0
CD 12 E0            CALL OUTAPE
10 F9               DJNZ NULL    Writes 50 zeroes (32H).
3C                  INC A
CD 12 E0            CALL OUTAPE  Writes a one.
21 00 00            LD HL, 0
7E          LOOP:   LD A, (HL)
CD 12 E0            CALL OUTAPE
23                  INC HL
7C                  LD A, H
FE 01               CP 01H       ; Ø1 = ETX
20 F6               JR NZ, LOOP  Writes the memory
                                 from 0000 to 00FFH.
CD 27 E0            CALL CMOTOF  Turns off the motor.
```

I/O Port Assignments

Tables 2, 3 and 4 give the Monitor's I/O port assignments for serial, parallel, and casette I/O.

Table 2. Serial Data Port Programming Assignment

I/O Port #	Bit #	Input	Output
FC	0 1 2 3 4 5 6 7	Input data bits NZ = TRUE	output data bits
FD	0 NZ	transmit buffer empty	bits per character (NB1)
	1 NZ	data available	bits per character (NB2)
	2	over-run	number of stop bits
	3	framing error	parity select
	4	parity error	no parity

Z → Trans Buffer full
Z = No data avail

Table 3. Parallel Data Port Programming Assignment

I/O Port #	Bit #	Input	Output
FF	0 1 2 3 4 5 6	input data bits	output data bits
	7	NZ BUSY (CNTRNU)	~~DATA~~ Strobe for CNTRNU
FE	6	?	OUTPUT data acknowledged
	7	INPUT data available	

BOTH INPUT ?
IØB6
IØB7

IN + OUT are two totally separate physical locations

Table 4. Cassette Recorder Programming Format

I/O Port #	Bit #	Input	Output
FC	0 to 7	input data bits	output data bits
FD	0	transmit buffer empty	
	1	data available	
FE	0-3	– KEYBOARD	– KEYBOARD
	4	– KEYBOARD	motor control #1
	5	– Video Trace	motor control #2
	6		baud rate (0 = 300 baud, 1 = 1200 baud)
	7		interface (0 = cassette, 1 = RS232)

FE – 3Ø = Carrier Scope

Notice - Port/Bit used for one thing only.

Writing Your Own Operating System

You may wish to replace the Sorcerer's Power-On Monitor with your own EPROM-resident operating system. If you have a later model Sorcerer, you must first have the logic board re-jumpered to accept EPROMs (see Hardware Modifications, EPROM — ROM Conversion).

At power-on and reset, the Z80 CPU trys to execute code at address 0000. Since the Monitor starts at address E000H, there is a circuit (the ROM decode, see Theory of Operation, Logic Board) which maps address E000H onto 0000 at power-on and reset. When you write your own operating system, you must put a Z80 jump instruction (C3H) at E000H; otherwise, the Sorcerer will never enter your system. Addresses E001H and E002H must contain the low order and high order bytes respectively of your system's actual starting addresss, which must be between E003H and E7FFH. *

Doesn't physically change ØØØØ H on reset

THE MONITOR BATCH SYSTEM

Introduction

The Monitor batch system lets the Sorcerer automatically run a sequence of programs, and search and update a sequence of files. The four batch commands are explained above (see Power-On Monitor Command Set).

A ***batch tape*** is a sequence of user created Monitor commands, recorded on a tape cassette. Use the Monitor command CR to create this tape, and the command LI to verify its contents. Use the command BA to automatically execute the commands on the batch tape, and use the command OV as the last command on the tape, to return control to the Monitor. Note that the Monitor does not have any commands which would allow you to edit a batch tape. If you make a mistake while creating your tape, you will have to go back and start over.

Example:

Here is a sample batch routine. The following commands are on a batch tape ready to be played on tape unit #1. Tape unit #1 also contains the two programs PROG1 and PROG3. Tape unit #2 contains a blank tape, for recording the file FILE1.

```
LOG PROG1 1
PROG1 is here
LO PROG3 1 2000
PROG3 is here
SE O=L
GO 2000
SE FA=AA
SA FILE1 0000 1000 2
OV
```

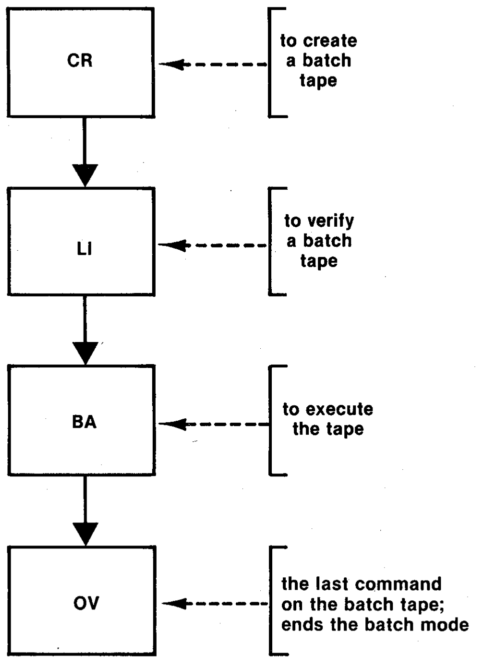
This batch routine loads the program PROG1 from tape unit #1 and executes it. Next, it loads the program PROG3 from tape unit #1 starting at address 2000H. It then routes the output to a Centronics printer and executes PROG3. Then it sets the tape file type to AA and saves the memory from addresses 0000 to 1000H on tape unit #2, under the name FILE1. Finally, it returns control to the Monitor.

Any error will end the batch mode and return you to the Monitor. You can also return to the Monitor by hitting CTRL C, ESC, or RUN/STOP.

Tape Unit Control

Only tape unit #1 can be used for creating or executing a batch tape; however, you can put a sequence of programs on tape unit #2 and run them with batch tape LOG commands. To use both tape units, you need a Serial Data Cable (Exidy Part No. DP4005); this cable also provides automatic control of the tape unit motors.

Figure 4. Batch Command Sequence



We strongly recommend you use the Serial Data Cable — without automatic motor control, your batch tape may run past a command before the preceding command has been executed. If you do not use the data cable, you can avoid this problem in two ways:

First you can put plenty of blank tape between commands on the batch tape. When you enter each command (under the CR command), let the tape run awhile before hitting RETURN.

Second, you can start and stop the tape unit manually (although this defeats the purpose of the batch system). The Sorcerer prints each command on the screen after reading it from the batch tape; this is the time to stop the recorder. After executing the command, Sorcerer prints a prompt; you can then turn on the recorder.

Batch Tape Format

The Monitor puts your commands onto a batch tape in this format:

One hundred 0s and a 1	a command in ASCII	one byte CRC (Cyclic Redundancy Check)

As each character is written on (or read from) the batch tape, Sorcerer performs a CRC routine to update the CRC byte. The final value of this byte is then written on the tape (under the CR command) or compared with the value already on the tape (under the BA command). The following routine is essentially the same as the Monitor's CRC routine. We list it here to show how the Monitor's CRC routine works. You may wish to use this routine in your own tape I/O software.

1. Choose an address XXYY in which to store the CRC byte (the Monitor uses an address in Monitor RAM). Here, XX and YY are the high order and low order bytes of the address, respectively.
2. Clear the CRC address.
3. On entry into the routine, the A register must contain a character.

CRC Routine

```
C5          PUSH  BC
F5          PUSH  AF
3A YY XX    LD    A,(CRC)
47          LD    B,A
F1          POP   AF
F5          PUSH  AF
90          SUB   B
47          LD    B,A
A8          XOR   B
2F          CPL
90          SUB   B
32 YY XX    LD    (CRC),A
F1          POP   AF
C1          POP   BC
```

RELOCATING THE MONITOR STACK

Introduction

The stack is a term which loosely denotes three separate areas of memory in the Sorcerer:

- The Monitor RAM
- The Monitor stack proper
- The stack pointer

To ***relocate the stack*** means to simultaneously move the Monitor RAM and stack proper, and reset the stack pointer.

The Monitor runs in ROM, but needs temporary scratchpad storage in RAM; this storage is the Monitor RAM, which is used to hold intermediate values during Monitor routines. The Monitor stack proper is used for temporary storage of Z80 registers (using PUSH and POP instructions), and to store return locations for Z80 CALL instructions. The stack pointer is a two byte register in the CPU which holds the lowest address currently used for stack storage.

At power-on, Sorcerer searches RAM for the top RAM address. The 112 addresses (70H) from the top downward are used as the Monitor RAM; the next 64 addresses approximately (40H) are the Monitor stack proper.

Example:

In the 8K Sorcerer at power-on, the top of RAM is 1FFFH and the stack starts at 1F90H. This means that addresses 1F91H to 1FFFH are Monitor RAM storage, and 1F90H to approximately 1F50H are the Monitor stack proper.

Reasons to Relocate the Stack

If the stack is disturbed, the system may crash. This can happen in three ways:

1. A tape file can overwrite the stack (the file header may address the file into the stack area).
2. A user program may overwrite the stack.
3. A user program may disturb the stack if it calls the Monitor I/O entry points.

You can recover from the crash by hitting the RESET keys, but you will still lose the contents of all RAM. To prevent a crash in any of these cases, you must relocate the stack to an area of RAM which won't be used by your program or tape file. This area must contain at least 176 bytes (B0H) — 112 bytes for Monitor RAM storage, and 64 bytes for the Monitor stack proper.

- 7ØH
+ 4Ø H
BØ H

You must also relocate the stack before using the Monitor RAM test (command TE) on the area of RAM occupied by the stack.

How to Relocate the Stack

First, choose a suitable address XXYY for the top of the Monitor RAM; here, XX and YY are the high order and low order bytes of the address, respectively. Second, use the Monitor EN command to put these Z80 instructions into the addresses 0000 to 0005H:

```
21 YY XX C3 06 E0
```

This Z80 program loads the address XXYY into the HL register pair and then jumps to the Monitor USER entry point. Finally, give the command GO 0000.

Example:

To move the stack so that the top of RAM is at 0750H:

```
You type:          EN 0000
Sorcerer replies:  0000:_
You type:                 21 50 07 C3 06 E0 /
Then type:         GO 0000
```

Sorcerer moves the stack and prints this message:

```
EXIDY STANDARD MONITOR
VERSION 1.0
COPYRIGHT (C) 1978 BY EXIDY INC.
THE TOP OF RAM IS 0750 HEX.
STACK BEGINS FROM 06E1 HEX.
```

page 1 (contains Ø)

FURTHER TOPICS

RS232 Input/Output

NB

The RS232 is enabled by setting bit 7 of I/O port FEH. However, this is the same port the keyboard uses, so when Sorcerer scans the keyboard, bit 7 is reset and RS232 is disabled. Therefore, the Power-On Monitor cannot communicate directly with an RS232 device; it needs some driving software. There are two RS232 driving programs available.

First is the Sorcerer Dumb Terminal program. We supply it on cassette to our dealers; customers who want this program may obtain duplicates from their dealers. The program loads at address 0100H and takes about 1.2K of memory. Put it into the Sorcerer with the Monitor LOG command.

When the program comes on, it asks for a choice of baud rate (0 = 300 baud, 1 = 1200 baud) and of duplex (0 = full duplex, 1 = half duplex). After both choices are made, the Sorcerer becomes a dumb terminal and remains so until you exit the program with RUN/STOP . Note that CTRL C and ESC will not terminate this program.

At full duplex, characters typed on the Sorcerer's keyboard will appear on the RS232 device's screen or printer, but not on the Sorcerer's screen. At half duplex, characters typed on the Sorcerer's keyboard will appear on both screens. In either case, characters sent by the RS232 device will appear on the Sorcerer's screen. The choice of full or half duplex depends on whether or not the RS232 device sends the Sorcerer an echo of each character it receives.

While acting as a dumb terminal, the Sorcerer can print graphics (including user-defined graphics) and cursor controls, as directed by the RS232 device. However, it can't function independently as a computer.

The second program is the RS232 Printer Driver. This program lets the Sorcerer use an RS232 output while still functioning as a computer. The program is short and relocatable, but it will slow your software; we list it below.

See SUN 1:11,1979

RS232 PRINTER DRIVER

```
      Obj Code            Label                               Comments

                          RS232:     EQU    $
      F5                             PUSH   AF              ;SAVE A
      F5                             PUSH   AF              ;SAVE TWICE
      CD 1B E0                       CALL   VIDEO           ;PRINT ON SCREEN
      FD 7E 3D                       LD     A,(IY + 3DH)    ;GET BAUD RATE      sets Bd 7 to 1
      F6 80         sets high Byte=1 OR     80H             ;ENABLE RS232 -
      D3 FE                          OUT    0FEH,A  code to port  }Sends data in A to FE (output Port)
      F1                             POP    AF              ;RESTORE CHAR
      CD 12 E0                       CALL   OUTAPE          ;SEND TO RS232  - out on I/O FCH
      DB FD               NOTFIN:    IN     A,0FDH          ;WAIT TILL UART DONE -
      CB 47                          BIT    0,A  —           Transmit Buffer empty → NZ
      28 FA                          JR  Z  ~~N~~Z,NOTFIN
      F1                             POP    AF              ;RESTORE ORIGINAL CHAR
      C9                             RET
```

Also must change Parity to Mark

Use the Monitor EN command to put the RS232 Printer Driver into any free area of RAM. You can then call the RS232 output through the Monitor's I/O vector output (SEND); assign the RS232 device to SEND with the Monitor command SE O = (starting address).

Example:

Put the RS232 Printer Driver into addresses 1000H to 1017H. Then give the command SE O = 1000. This will route all Monitor SEND calls to the RS232 device.

Alternatively, your Assembly language programs can call the RS232 device directly; just include an EQU statement assigning the label RS232 to the starting address of the RS232 Printer Driver.

Example:

```
RS232:  EQU 1000H
        LD B, 0FFH
LOOP:   LD A, B
        CALL RS232
        DJNZ LOOP
```

Whether you call the RS232 device directly, or through SEND, use the Monitor SE T = command to set the baud rate.

Converting Processor Technology Software

See Article will read CUTS on SOL

There is a large amount of readily available software written for Processor Technology Corporation (PTC) equipment. You may wish to use some of these programs on your Sorcerer.

Processor Technology Assembler and machine code programs will run on the Sorcerer if you modify their I/O routines. There is no simple way to convert the tape I/O routines, since the PTC tape formats are incompatible with the Sorcerer's. However, the other I/O routines are easy to convert. Here are the PTC I/O entry points:

Address	Name	Remarks
C01CH	AOUT	This is an output to an absolute device. The B register is sent to the pseudo-port specified by the A register. The pseudo-ports are: 0 = Video 1 = Serial 2 = Parallel 3 = User-defined
C019H	SOUT	This is the PTC I/O vector output, corresponding to the Monitor's SEND (address E00CH). The B register goes to the current pseudo-port (specified by a PTC command corresponding to the Monitor's SE O =).
C022H	AINP	This is the PTC absolute input. The pseudo-port specified by the A register is input into the A register. The Z flag is set or reset on the same conditions as the Monitor's input entry points.
C01FH	SINP	This is the PTC I/O vector input, corresponding to the Monitor's RECEVE (address E009H). The current pseudo-port is input into the A register, and the Z flag is set or reset.

Notice that the PTC input routines put the incoming data into the same register as the Monitor's input routines, and set or reset the Z flag under the same conditions. Therefore, all you need do to convert a Processor Technology input routine is replace all calls to SINP or AINP with calls to RECEVE, KEYBRD, or PARLIN. You should replace SINP calls with RECEVE calls; however, you should give some thought to the AINP calls. Check the A register to see which pseudo-port was intended for the AINP; then replace with a RECEVE, KEYBRD, or PARLIN call, as appropriate.

The PTC output routines take their data from the B register, while the Monitor's routines use the A register. Therefore, to convert a PTC output routine you must replace each call to SOUT or AOUT with the instruction LD A,B followed by a call to SEND, VIDEO, or PARLOT.

Example:

Here is a routine written for Processor Technology equipment:

```
CD 19 C0                 CALL SOUT
CD 1F C0        INPUT:   CALL SINP
28 FB                    JR Z, INPUT
```

Here is the routine converted to run on the Sorcerer:

```
78                       LD, A, B
CD 0C E0                 CALL SEND
CD 09 E0        INPUT:   CALL RECEVE
28 FB                    JR Z, INPUT
```

The Standard BASIC USR(X) Function

The function USR(X) allows a BASIC program to call a machine language subroutine. The format for the call is:

V = USR(0)

Here, 0 is used as a dummy argument, and V stands for any dummy variable. Note that the call to USR will change the value of V.

When BASIC is loaded, C3H is entered at address 0103H. Therefore addresses 0103H through 0105H are a Z80 jump command; 0104H and 0105H contain the low order and high order bytes respectively of the jump address. When USR is called, Sorcerer jumps to 0103H, reads the address in 0104H and 0105H and jumps to that address.

Before calling USR, use the POKE command to load 0104H and 0105H with your subroutine's starting address.

Example:

100 POKE 260, 16	puts 10H into 0104H
110 POKE 261, 0	puts 0 into 0105H
200 A = USR(0)	calls the machine language subroutine at address 0010H

C [illegible] 260

[illegible] 10

BAUD RATE see @ (IY + 3D_H)

~~see p.32 Rate is @ 0033H~~

NOTE

You must use decimal notation for memory addresses and their contents, when using the PEEK and POKE commands. You must use hexadecimal notation when using the Power-On Monitor or when writing in Z80 machine language.

Standard BASIC has its own stack, separate from the Monitor stack; the USR subroutine uses the BASIC stack. So long as your USR routine leaves the stack unaltered, a Z80 RET command (C9H) will return control to BASIC. Addresses 0000 through 00FFH are not used by BASIC or the Monitor; you can safely use these addresses as scratchpad RAM for your USR subroutine, or to store the subroutine.

NOTE

The simplest way to pass parameters between a BASIC program and a USR subroutine is by means of PEEK and POKE commands. Use a POKE to put a decimal number into a decimal address; your Z80 subroutine can then pick the number (in hexadecimal) out of that address (also hexadecimal). Use PEEK to read a number into your BASIC program, after it has been deposited by your USR subroutine.

See p. 17 Uses address ØØØØ for monitor at power on + reset.

Find where Basic stack is located.

ØFEHPORT

KEYBOARD		VIDEO
IN	OUT	IN
Ø	Ø	5 - Vertical Retrace
1	1	
2	2	
3	3	
4		

PART TWO
HARDWARE

1/10/81 - Suspicion of 11D
Screen Clear ROM
for interference pattern Tapping → pattern.
Finally seemed to stop.

Video

2B Scr. Control
6D+7D Data Buff
6A+6B Counter [illegible]
8D Shift Reg
5D
3A,4A,5A H+V Sync

HARDWARE DESIGNATIONS

We refer to an IC device by its location on the board. Thus 1A is the device in column 1, row A of the board. Context will make clear which board is intended.

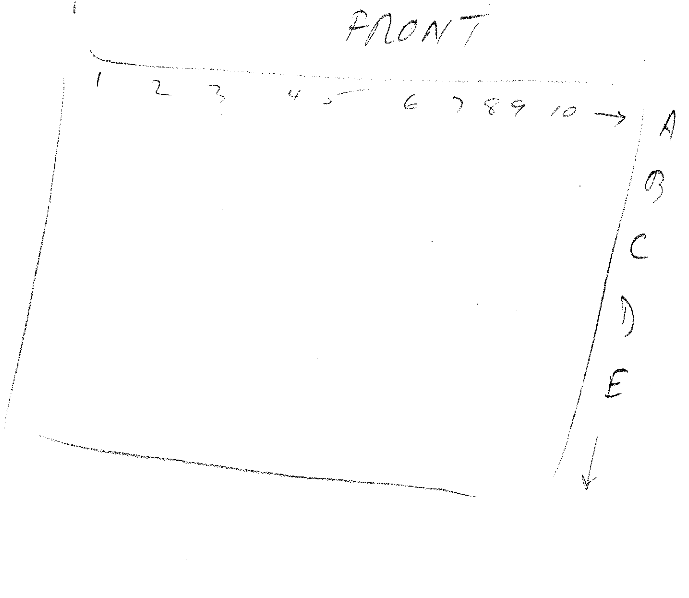
We refer to a pin of an IC device (and sometimes the signal at that pin) by a hyphenated number following the location. Thus 1A-5 is pin 5 of device 1A.

If an IC chip contains more than one device, we refer to each by one of its pins. Thus 1A-5 also designates one of the devices on chip 1A — the one containing pin 5. Context will make clear whether a designation such as 1A-5 refers to a pin or to a device.

NOTE

All service should be done by an authorized Sorcerer dealer; unauthorized service will void our warranty.

Logic Board - large lower board
Tape interface board - smaller top one - Schematic I



HARDWARE MODIFICATIONS

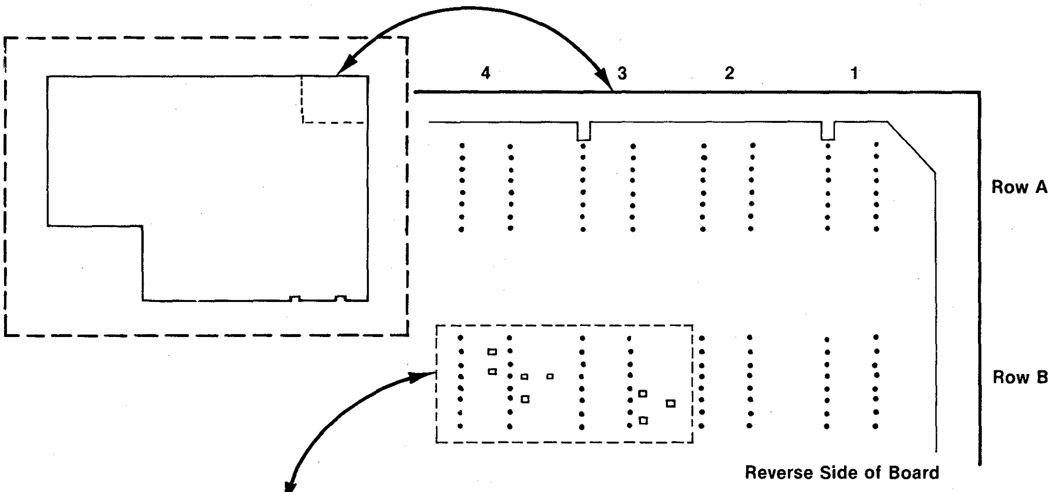
110 V 60 Hz to 220 V 50 Hz Conversion

For use in the USA, the Sorcerer is wired for 110 V 60 Hz power. To convert to 220 V 50 Hz, both the power supply and the vertical sync generator must be modified.

1. For 100 V, the power supply transformer primary windings are connected in parallel; to convert to 220 V, disconnect them and rewire in series. Do this by cutting the black-yellow and black-red wires at the line filter, and soldering them together.

2. The vertical sync generator has three jumper sets at board locations 3B and 4B, on the ***reverse side*** of the logic board (see Figure 5). To convert from 60 Hz to 50 Hz, cut the printed traces and rewire, as shown in Figure 6.

Figure 5. Vertical Sync Jumpers, Location



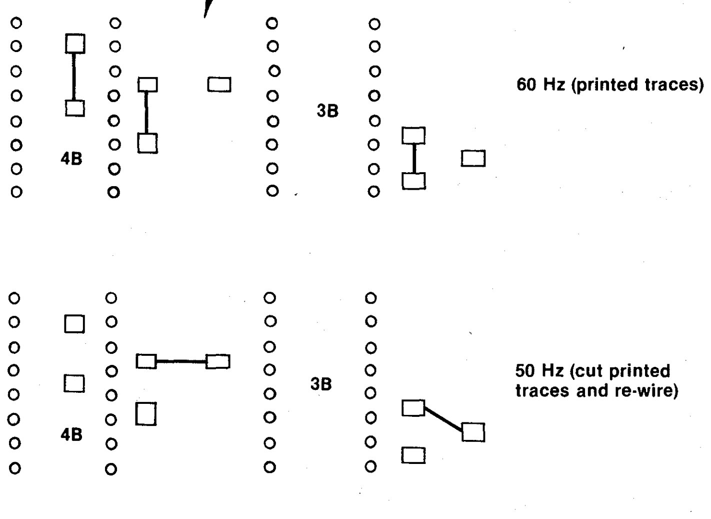


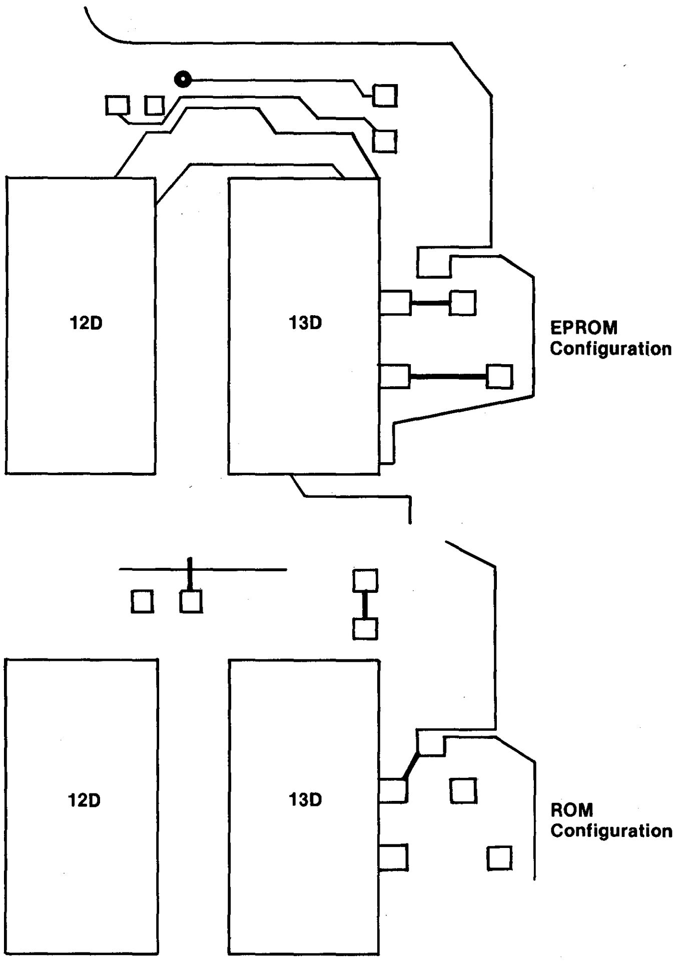
Figure 6. Vertical Sync Jumpers, Configuration

EPROM — ROM Conversion

The Sorcerer logic board accepts either EPROMs or ROMS in sockets 11D, 12D, and 13D (the Power-On Monitor and Character Generator). However, the board must be modified when switching from one type of device to the other. There are three jumper locations at 13D which select between EPROMs and ROMs (see Figure 7).

In early model Sorcerers the Power-On Monitor is EPROM-resident, and printed jumpers configure the board for EPROMs. In the later model Sorcerers, the Monitor is ROM-resident, and the board is configured for ROMs. If the owner of a later model Sorcerer wishes to put his own operating system on EPROMs, he must have the logic board re-jumpered; cut the printed traces and rewire as shown in Figure 7.

Figure 7. EPROM-ROM Jumpers



Internal Memory Expansion

The Sorcerer computer normally has 8K of internal RAM; this is expandable to 16K or 32K. The RAM sockets are located in two rows in the upper right corner of the logic PC board (see Figure 8).

There are two options for internal memory expansion:

- Install RAMs in one row of RAM sockets, or in both rows.
- Use either 4K RAMs or 16K RAMs.

This gives four possibilities:

No. Rows	RAM Size	Total RAM (Bytes)	Remarks
1	4K	4K	This configuration is never used.
2	4K	8K	The normal configuration.
1	16K	16K	1 Memory Expansion Kit (row A).
2	16K	32K	2 Memory Expansion Kits (rows A and B).

The Sorcerer Memory Expansion Kit contains eight 16K RAMs — enough to increase the Sorcerer's memory from 8K to 16K or from 16K to 32K. To increase memory from 8K to 32K use two Expansion Kits.

There are two jumper locations at 11A on the logic PC board (see Figures 8 and 9). The top location determines whether one or two rows of RAM sockets are to be used:

Top jumper absent = 1 row of RAMs
Top jumper present = 2 rows of RAMs

The top jumper is printed on the logic board; to remove it, cut carefully with a sharp knife. You can reconnect it later by soldering a wire between the jumper terminals.

The lower jumper location determines whether 4K or 16K RAMs are to be used:

Lower jumper absent = 4K RAMs
Lower jumper present = 16K RAMs

> NOTE
>
> If only one row is used, the RAMs ***must*** be installed in Row A. ***Do not*** insert a RAM upside down; this will destroy the RAM, and may damage the rest of the logic board. Pin 1 is at the upper left of each socket.

Figure 8. RAM Sockets and RAM Jumpers, Locations

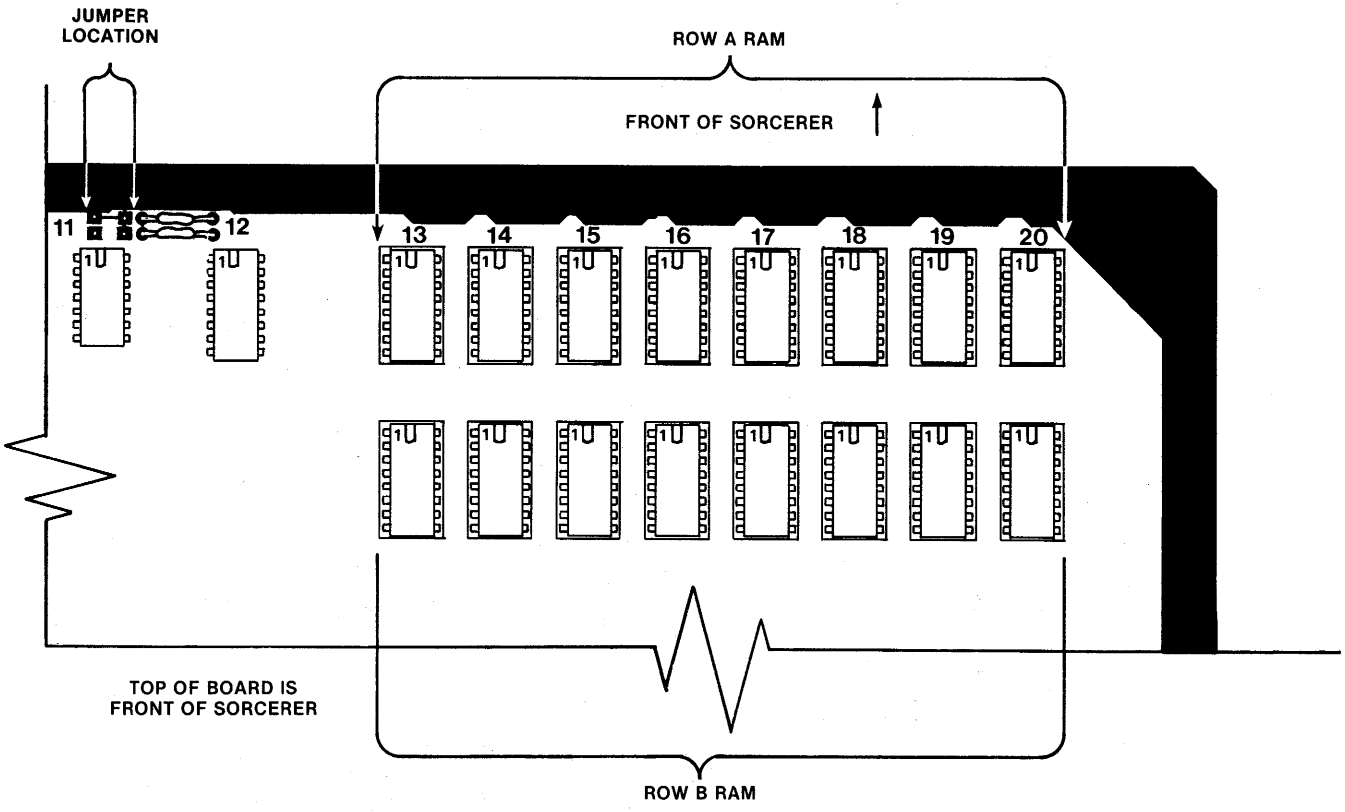
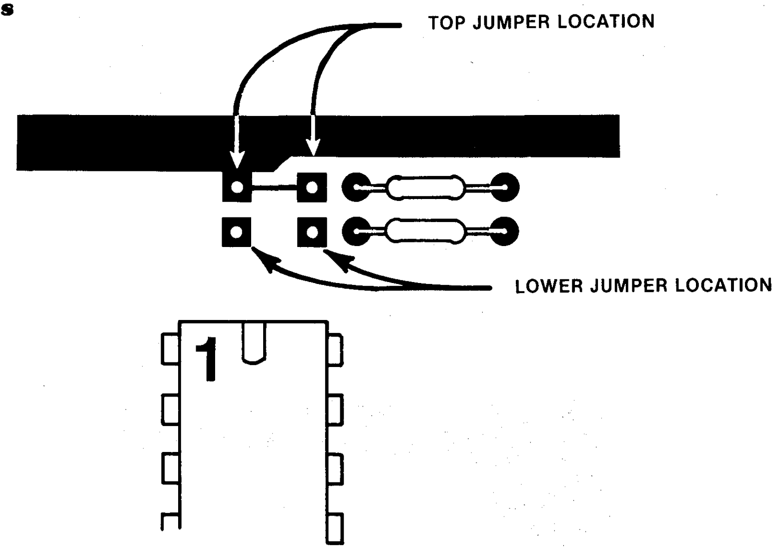


Figure 9. RAM Jumpers



PERFORMANCE TESTS

The following tests will determine whether the Sorcerer is functioning properly, and will help locate malfunctions. The numbered headings describe the tests; where necessary, lettered subheadings give step-by-step instructions for the tests.

Logic Board

1. Parallel I/O — Connect the output to the input and check whether the unit reads and writes properly. See Table 6 for the parallel interface connector pinouts.

CHAR RAM
FCØØ-
FFFF

2. RAM bit test — Use the Monitor TE command to test the screen RAM, character generator RAM, and main RAM (see the memory map, Figure 1, for the addresses). You must test the screen RAM in two steps:

   - First, put the cursor into the top half of the screen, and run the bit test on addresses F400H to F7FFH.
   - Second, put the cursor into the bottom half of the screen, and run the test on addresses F002H and F3FFH.

   Do not use the bit test on addresses F000H and F001H.

   Before testing the main RAM, relocate the Monitor stack to another section of RAM. See p. 19

1st TE ØØØØ - 7EFF
2nd MO STACK!
3rd TE 7F00 - 7FFF

3. 16K RAM timing test — The 16K and 32K Sorcerers use 16K rather than 4K RAMs. To check the timing on the 16K RAMs use a dual trace triggering scope (Tektronix 465 or equivalent) and two high frequency, high impedance, low capacitance probes (Tektronix P6108 or equivalent).

   a. Use the Monitor EN command to enter the following Z80 routine into addresses FE00H to FE06H:

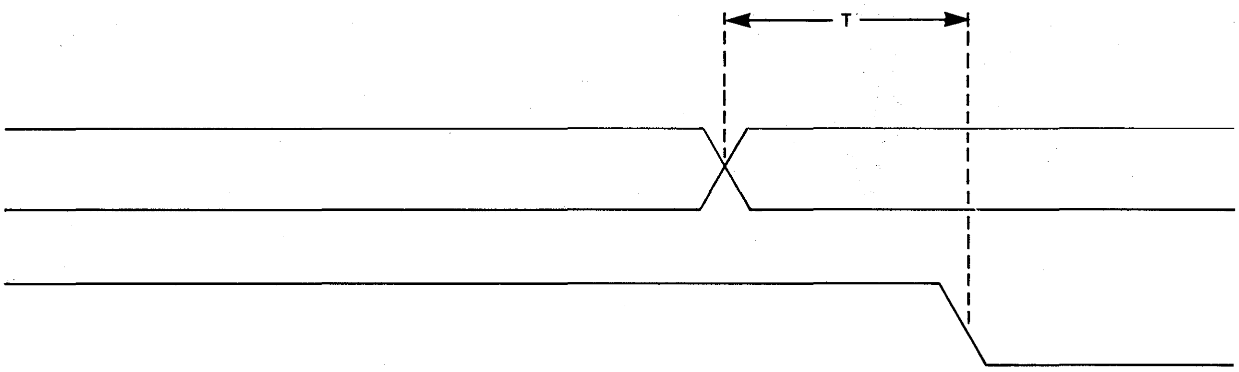
      21 00 21 7E C3 00 FE

      Then give the command GO FE00.

   b. Place the first probe at 20B-13 and the second at 20B-15; lock the scope to the falling edge of the signal at 20B-13.

   c. You should see the signals shown in Figure 10. The delay T should be at least 40 nsec.

Figure 10. 16K RAM Timing Signals



Tape Interface Board

These tests require a Serial Data Cable (Exidy Part No. DP4005) and an oscilloscope with a 50 mV per cm setting.

1. Check both recorder remote-control outputs.
   a. Plug the REM 1 plug of the Serial Data Cable into the recorder remote-control jack. The recorder is now under computer control and should not be able to run. If the recorder is one you haven't tried before and it runs continuously, there may be a polarity problem. Reversing the leads on the REM plug will solve the polarity problem. Wrong polarity will not harm the computer or the recorder.
   b. Use the Monitor FI 1 command to turn the recorder on. Hit RESET to turn it off. Repeat this, to be sure that the transistor Q2 didn't burn out when you turned the recorder off.
   c. Repeat step b, using the REM 2 plug and the FI 2 command.
2. Measure the frequency at P11-9 for 300 baud (210 μs) and 1200 baud (52 μs).
   a. Hit RESET. This resets the baud rate to 1200; however, the Sorcerer will not actually send the 1200 baud carrier until you give a Monitor tape command (such as LO, SA, or FI). Until then, Sorcerer sends the 300 baud carrier signal.
   b. Check for 210 μs cycle time at pin 9 of connector P11 on the cassette boards. (Pin 1 is on the left and pin 10 is on the right.)
   c. Give the Monitor command FI (this activates the 1200 baud carrier).
   d. Check for 53 μs cycle time at P11-9.
3. At 300 baud, check ***all*** of the MIC and AUX outputs for proper voltage levels and freedom from noise due to trace shorts.
   a. The correct voltages are 40 ±5 MV p-p for MIC and 210 ±20 mV p-p for AUX. These values are lower than the nominal 50 mV and 250 mV, since the tape interface atenuates the 300 baud carrier more than the 1200 baud carrier.
   b. If there is excessive noise on either of the AUX outputs, write a 1200 baud file onto tape using the noisy output, and then read the file back to insure there were no errors. The length of the file should be 4K (E000H to EFFFH). Be sure that the AUX output is plugged into the AUX input on the recorder and not into the MIC input.
   c. Reset the Sorcerer to insure the 300 baud carrier.
   d. Use a ground clip on the scope probe.
   e. Make sure no MIC or AUX cables are plugged into a recorder (otherwise the signal will be loaded down).
   f. Check MIC 1 both from the phono jack and from the 25-pin serial interface connector.
   g. Check MIC 2, AUX 1, and AUX 2 from the 25-pin connector.
4. Write and read a 300 baud, 128 byte file (E000H to E080H) using MIC 2 and EAR 2 from the 25-pin serial interface connector.
   a. Plug the EAR 2 plug of the Serial Data Cable into the EAR jack on the recorder.
   b. Put the Sorcerer into the Power-On Monitor and type SE T = 1.
   c. Put a scratch tape in the recorder and make sure it's rewound ***completely,*** otherwise you will get errors.
   d. Put the recorder in record.
   e. Type SAVE TEST E000 E080, but don't hit RETURN (this command will record pseudo-random data taken from addresses E000H through E080H).
   f. Wait until the tape leader is past and then hit RETURN.
   g. When the recording is finished, rewind the tape and load it back in with the Monitor command LO.
   h. If you get a prompt character and the word "ERROR" didn't appear anywhere, the read was good.
5. When the UART is not sending data, the CASSWRIT signal goes high. The tape interface manchester encoder translates this as a steady stream of logic 1s — this is the normal cassette carrier signal. Connect EAR 2 and AUX 2 and check for this carrier at 1200 and at 300 baud. (Use AUX 2 rather than MIC 2 because the 50 mV MIC 2 signal is too weak.)
6. Test the tape interface's ability to write and read data generated by a diagnostic program.
   a. Enter the following Z80 program into addresses 0000 to 0048H:

Saved on Work tape and Master 1 as TESTP

```
Addr   Obj Code      Label      Mnemonic              Comments

0000   18 27         WRITE:     JR     WR             ;START-POINT FOR WRITING
                     ;READING ROUTINE
0002   CD 33 00      READ:      CALL   BAUD           ;START-POINT FOR READING
0005   CD 0F E0                 CALL   INTAPE
0008   FE E2                    CP     0E2H
000A   20 0C                    JR     NZ,ERROR
                     ;PRINT FLASHING ASTERISKS
000C   3E 2A                    LD     A,'*'
000E   CD 1B E0                 CALL   VIDEO
0011   3E 08                    LD     A,08H          ;BACKSPACE
0013   CD 1B E0                 CALL   VIDEO
0016   18 EA                    JR     READ
                     ;PRINT "E"
```

(continued)

```
Addr   Obj Code       Label      Mnemonic                      Comments

0018   3E 45          ERROR:     LD     A,'E'
001A   CD 1B E0                  CALL   VIDEO
001D   CD 33 00       LOOP:      CALL   BAUD
0020   CD 0F E0                  CALL   INTAPE
0023   FE E2                     CP     0E2H
0025   20 FE                     JR     NZ,LOOP
0027   18 D9                     JR     READ
                      ;WRITING ROUTINE
0029   CD 33 00       WR:        CALL   BAUD
002C   3E E2                     LD     A,0E2H
002E   CD 12 E0                  CALL   OUTAPE
0031   18 F6                     JR     WR
                      ;BAUD RATE AND EXIT-CHECK SUBROUTINE
0033   FD E5          BAUD:      PUSH   IY
0035   CD A2 E1                  CALL   GETIY                  ;A MONITOR SUBROUTINE
0038   FD 7E 3D                  LD     A,(IY + TAPES)
003B   FD 77 45                  LA     (IY + CMTRFG),A
003E   D3 FE                     OUT    0FEH,A
0040   FD E1                     POP    IY
0042   CD 15 E0                  CALL   QUIKCK
0045   C2 03 E0                  JP     NZ,WARM
0048   C9                        RET
```

b. Use the Monitor command SE to set the baud rate to 300 or 1200.

c. Turn on the recorder and give the command <u>GO 0000.</u> This causes the program to write data.

d. Let the program write for at least 60 seconds. Then stop the recorder, and halt the program with CTRL C, ESC, or RUN/STOP.

e. Rewind the tape and start the recorder playing. Give the command GO 0002; this causes the program to read the recorded data.

f. Sorcerer will print:

- A flashing asterisk (*) if it reads the data correctly.
- An E, for each error.
- Nothing, if it sees no data.

7. Use the diagnostic program in Step 6 above, to write and read data, while you adjust the recorder's tone and volume controls. You should be able to read and write correctly over at least half of each control's tuning range (it doesn't matter which half).

CASSETTE INPUT/OUTPUT TROUBLE-SHOOTING HINTS

1. Make sure the tape is well past the leader before starting to record.

2. Make sure the two cassette cables are firmly plugged into the jacks on the recorder and the Sorcerer.

   a. First, plug the Sorcerer MIC output (at the RCA Jack) to the recorder's MIC input.

   b. Then try connecting the Sorcerer's MIC and AUX outputs to the recorder's MIC and AUX inputs, in all possible combinations. MIC 2 and the two AUX outputs are on the serial interface connector (see Table 5).

   c. In all cases, connect the recorder's EAR output to the Sorcerer's EAR input.

3. If the recorder is running on batteries, try using a line cord instead. If the recorder is running on a line cord, try using batteries (the power supply may filter poorly).

4. Try a different tape cassette, preferably low noise, high output, 15 to 20 minutes to a side. Longer tapes usually are satisfactory, but have more internal drag, and therefore don't run at uniform speed. Also, slight imperfections not normally a problem with audio recording can cause errors when recording data; it's best to use certified tape.

5. Use this procedure to check the Sorcerer's ability to sync onto incoming tape data:

   a. Give the Monitor command FI; this outputs a 1200 baud carrier signal. (The Monitor command SE T = 0 will set the tape information rate to 1200 baud, but no carrier signal will be sent until another tape command is given.)

   b. Record the 1200 baud carrier for 30 to 60 seconds.

   c. Locate the tape interface LED at the back of the Sorcerer. This LED is visible through the back grill behind the EAR jack, and should be on except when data is being played from the recorder.

d. Play back the recorded 1200 baud carrier; the LED should go out (or flicker very faintly). If this does not happen, try different combinations of tone and volume setting. If the LED still stays on, try plugging the MIC cable into the recorder's AUX input.

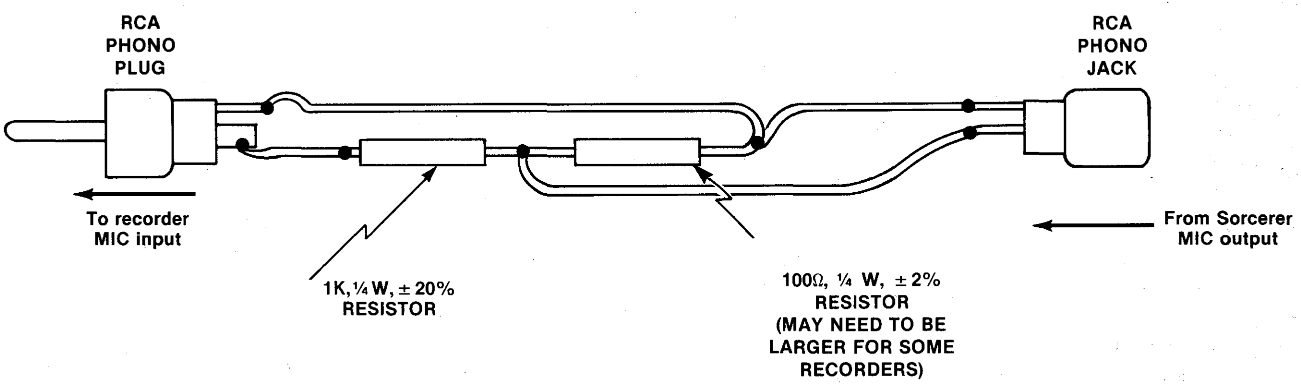
e. As a last resort, locate the potentiometer VR1 on the tape interface board (location 1D). There are two ways to adjust VR1:

- Play the recorded 1200 baud carrier while moving VR1 back and forth over its entire range. The LED should stay out for at least half the range. Determine which subrange of settings turn the LED off, and set VR1 to the middle of that subrange.
- Disconnect the recorder, and jumper one of Sorcerer's AUX outputs to one of its EAR inputs. Now adjust VR1 against the LED at both 300 and 1200 baud. Note that this method is independant of your recorder's idiosyncracies.

6. Marginal performance may be due to out of tolerance parts in the input filtration train, especially the second order linear phase high-pass filter.

7. Some recorders have very sensitive MIC inputs which are overloaded by the Sorcerer's 50 mV MIC output. You can usually get good results by connecting the recorder's AUX input to one of Sorcerer's AUX outputs; however, the recoder's AUX may not work with Sorcerer's MIC. If you wish to use. the Sorcerer MIC output, you must use an attenuator plug, which reduces the 50 mV output to a level the recorder's MIC will accept. Figure 11 shows how to make an attenuator plug.

Figure 11. Attenuator Plug



SORCERER INTERFACE CONNECTORS — SPECIFICATIONS

The audio and video connectors are female RCA jacks, and mate with standard male RCA jacks. The serial and parallel interface connectors are standard female 25-pin D connectors and mate with standard male 25-pin plugs; the pinouts are given in Tables 5 and 6. The S-100 interface connector is a 50-pin male connector (dual 25 on .1 centers) printed onto the edge of the logic board; it mates with a female 50-pin edge connector. The pinouts for the S-100 interface are given in Table 7.

Table 5. Serial Interface Pinouts (Right Hand 25 Pin Connector)

Pin #	Signal	Pin #	Signal
1	Shield 1	13	Motor #2 +
2	RS232 OUT	14	Shield 2
3	RS232 IN	15	MIC 1
4	Ground	16	MIC 2
5	AUX 1	17	Ground
6		18	AUX 2
7	Ground	19	Ground
8		20	EAR 1
9	+12 Volts	21	EAR 2
10	Unused	22	Unused
11	RS232 IN	23	RS232 OUT
12	Motor #1 +	24	Motor #1—
		25	Motor #2—

Table 6. Parallel Interface Pinouts (Left Hand 25 Pin Connector)

PORT

IN, FE-6 (Monitor)

IN, FE-7 (monitor)

Pin #	Signal	Pin #	Signal
1	Ground	13	Input Bit 6
2	Output data accepted	14	Unused
3	Output data available	15	+5 volts
* 4	Output bit 7	16	Output bit 0
5	Output bit 6	17	Output bit 1
6	Output bit 5	18	Output bit 2
7	Output bit 4	19	Output bit 3
8	Ground	20	+5 volts
9	Input data available	21	Input data accepted
10	Input bit 0	22	Input bit 1
11	Input bit 2	23	Input bit 3
12	Input bit 4	24	Input bit 5
		*25	Input bit 7

*BIT 7 used as STROBE/BUSY IN CENTRONICS PRT. DRVR. See p. 34.

Table 7. Sorcerer 50-Pin Edge Connector Pinout Table

Pin #	Signal	Pin #	Signal
1	$\overline{\text{PRESET}}$ (out of Sorcerer)	26	Address bit 11
2	$\overline{\text{INT}}$	27	Address bit 13
3	$\overline{\text{WAIT}}$	28	Address bit 14
4	Data Bus Enable (into Sorcerer)	29	Address bit 0
5	$\overline{\text{BUSRQ}}$	30	Address bit 12
6	$\overline{\text{NMI}}$	31	Address bit 2
7	BUSACK	32	Address bit 1
8	Data Bus Direction (into Sorcerer)	33	Address bit 4
9	$\overline{\text{RAM}}$ $\overline{\text{DR}}$ or $\overline{\text{ROM}}$ $\overline{\text{ENABLE}}$	34	Address bit 3
10	ϕ1	35	Address bit 6
11	ROM PRE	36	Address bit 5
12	Reset Acknowledge	37	Data bit 0
13	ϕ2 (Clock out)	38	Address bit 7
14	UP8K	39	Data bit 2
15	$\overline{\text{MREQ}}$	40	Data bit 1
16	$\overline{\text{M1}}$ ← (MI?)	41	Data bit 4
17	$\overline{\text{RD}}$	42	Data bit 3
18	$\overline{\text{IORQ}}$	43	Data bit 6
19	$\overline{\text{RFSH}}$	44	Data bit 5
20	$\overline{\text{WR}}$	45	RESET (into Sorcerer)
21	Address bit 8	46	Data bit 7
22	$\overline{\text{HALT}}$	47	Unused
23	Address bit 10	48	I/O
24	Address bit 9	49	Ground
25	Address bit 15	50	Ground

Table 8. Parallel Data Cable Pinouts

Sorcerer End Pin #	Printer End Pin #	Signal
1	19 to 30	Ground
2	10	Acknowledge from Printer
3		Unused
* 4	1	Data strobe from Sorcerer
5	8	Data bit 6
6	7	Data bit 5
7	6	Data bit 4
8	19 to 30	Ground
9 to 15		Unused
16	2	Data bit 0
17	3	Data bit 1
18	4	Data bit 2
19	5	Data bit 3
20 to 24		Unused
* 25	11	Busy
	9	Unused - Ground to get logic as in Epson
	12 ~~13~~ to 18	Unused
	31 to 36	Unused

Note: Pins #15+20 have +5V on some systems. (SA 3(4):67)

* Bit 7 used for handshake. - Centronics PRT/DVR. See below

No Data Bit 7! [TRENDCOM 200]

DATA CABLES

Parallel Data Cable (Centronics Printer)

This cable (Exidy part number DP4003) connects the Sorcerer parallel interface to the input of a Centronics or Centronics-compatible printer. Table 8 gives the pinouts for the cable. * The data strobe signal from the Sorcerer is bit 7 of port FFH. The busy signal from the printer is input to bit 7 of port FFH. The acknowledge signal from the printer is used to reset the data available bit in the Sorcerer handshake latch (8F-8). The data strobe and acknowledge signals are both negative going; the busy signal is positive going. See Figure 16 for a timing chart of these signals.

NOTE

Do not plug the parallel cable into the Sorcerer's serial interface connector, as this will damage both the printer and the Sorcerer.

Serial Data Cable (Tape Unit)

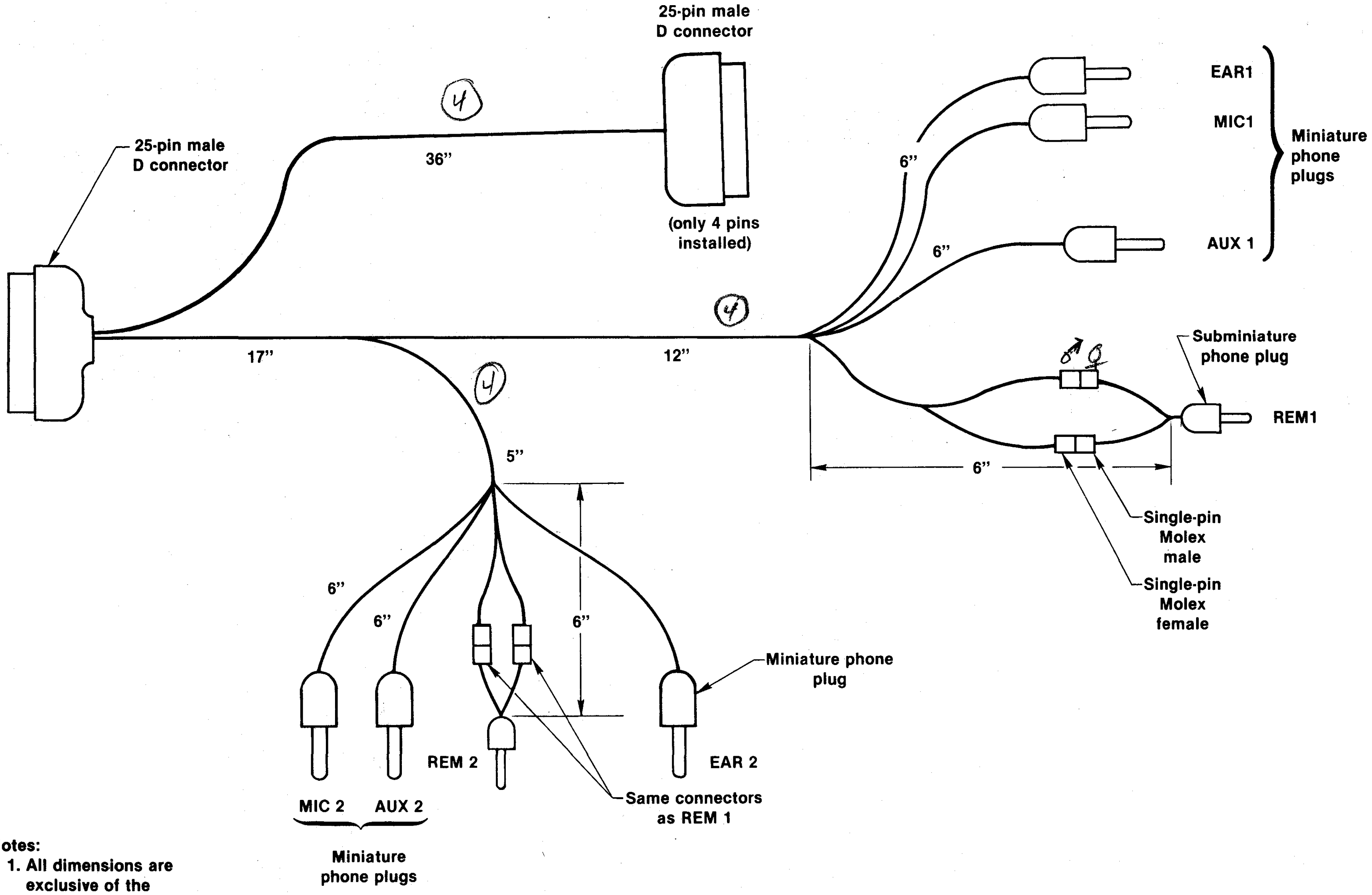
This cable (Exidy part number DP4005) connects the Sorcerer's serial interface to one or two cassette recorders or to an RS232 device; when connected to a recorder, it allows the Sorcerer to control the recorder's motor. Single-pin Molex connectors in the motor control lines allow you to change the polarity of the motor control signal, to suit your recorder.

Figure 12 shows the layout of this cable, and Figure 13 is the schematic.

NOTE

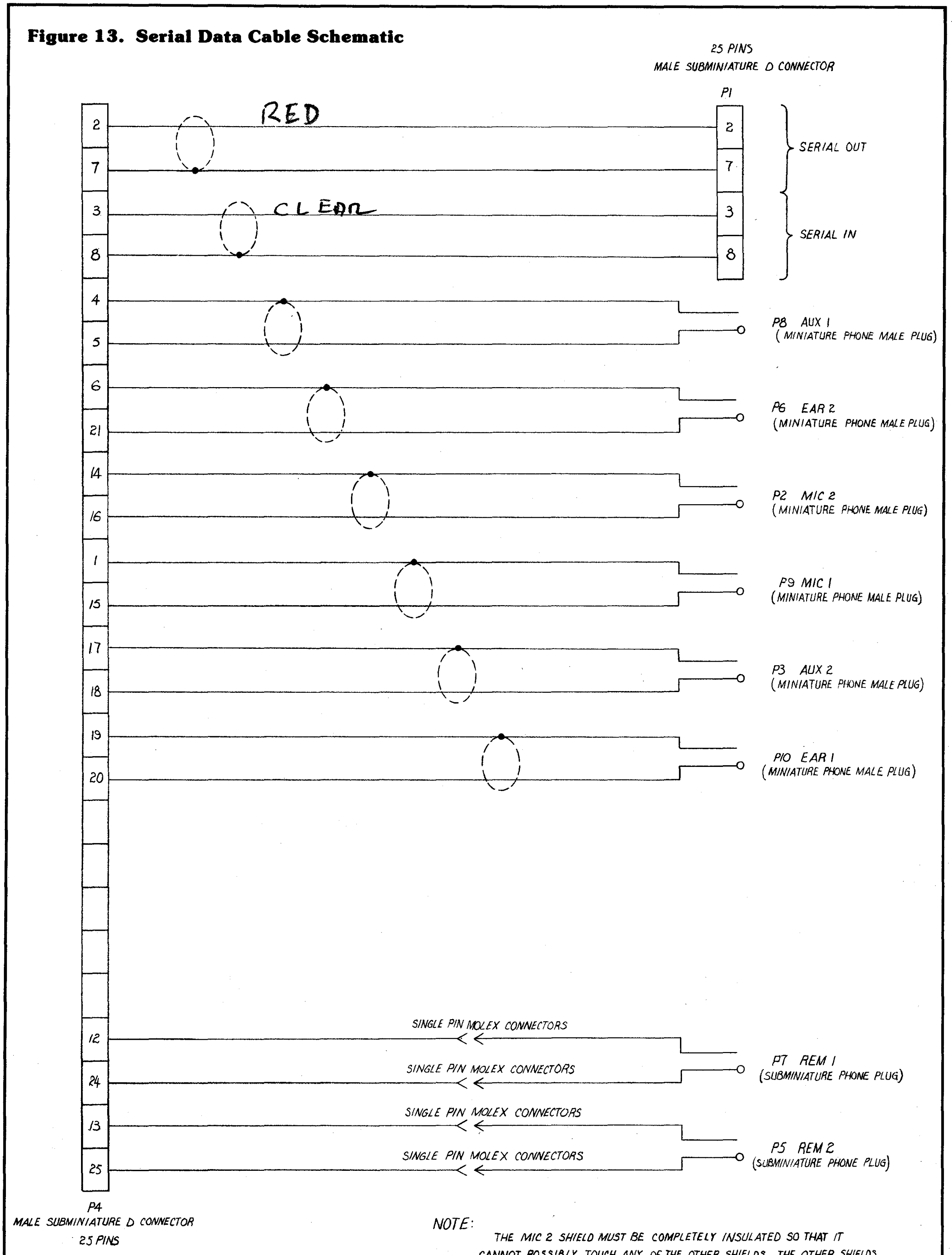
Do not plug the serial cable into the Sorcerer's parallel interface connector, as this will damage the Sorcerer.

Figure 12. Serial Data Cable Assembly Diagram



Notes:
1. All dimensions are exclusive of the connector body
2. Label all connectors

Figure 13. Serial Data Cable Schematic


25 PINS
MALE SUBMINIATURE D CONNECTOR
P1
RED
CLEAR
2
7
3
8
SERIAL OUT
SERIAL IN
4
5
P8 AUX 1
(MINIATURE PHONE MALE PLUG)
6
21
P6 EAR 2
(MINIATURE PHONE MALE PLUG)
14
16
P2 MIC 2
(MINIATURE PHONE MALE PLUG)
1
15
P9 MIC 1
(MINIATURE PHONE MALE PLUG)
17
18
P3 AUX 2
(MINIATURE PHONE MALE PLUG)
19
20
P10 EAR 1
(MINIATURE PHONE MALE PLUG)
12
SINGLE PIN MOLEX CONNECTORS
24
SINGLE PIN MOLEX CONNECTORS
P7 REM 1
(SUBMINIATURE PHONE PLUG)
13
SINGLE PIN MOLEX CONNECTORS
25
SINGLE PIN MOLEX CONNECTORS
P5 REM 2
(SUBMINIATURE PHONE PLUG)
P4
MALE SUBMINIATURE D CONNECTOR
25 PINS

NOTE: THE MIC 2 SHIELD MUST BE COMPLETELY INSULATED SO THAT IT CANNOT POSSIBLY TOUCH ANY OF THE OTHER SHIELDS. THE OTHER SHIELDS DO NOT NEED INSULATION.

Theory of Operation Sections

Figure 14. Sorcerer Main Logic Block Diagram

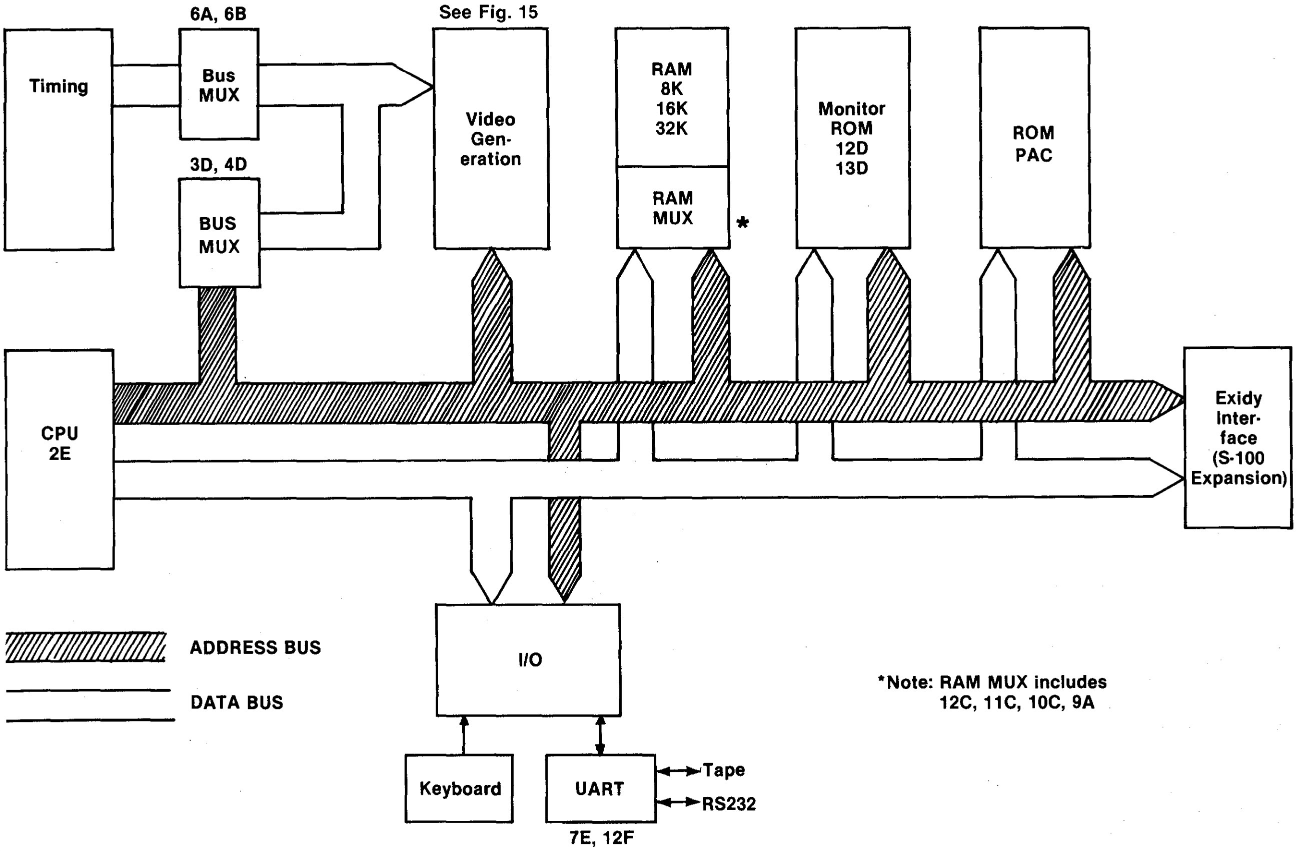
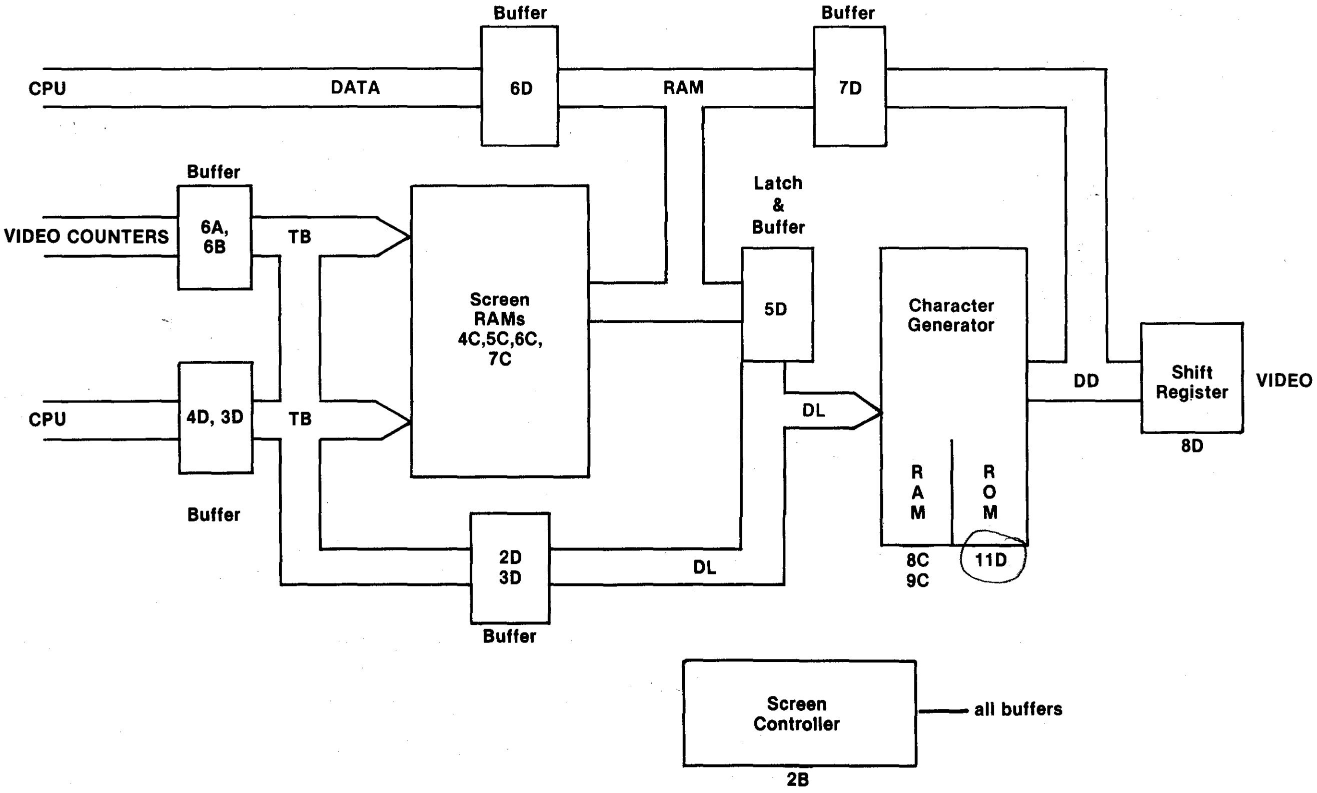


Figure 15. Video Generator Block Diagram



THEORY OF OPERATION, LOGIC BOARD

Video Clock Generator

The 12.638 MHz crystal, 1A-12 and 1A-10 form an oscillator with 1A-6 as a buffer. The flip-flop 1B divides this clock signal in half, providing CLK6 (6 MHz approximately) and its inversion 5CLK6.

Horizontal Sync Generator

3A, 4A and 5A-5 provide the horizontal sync and blanking generation. 3A and 4A form a binary up-counter; the starting count is determined by the state of 5A-6. E1 through E256 are the horizontal scan element counts. 7B-9 shuts off the video during horizontal blanking; ~~7B-5~~ shuts off the video when the CPU accesses the screen. 6A is part of the video counter buffer.

7B-6 [SA 3:172]

Vertical Sync Generator

1B-9 (SA 3:172)

3B, 4B and ~~5A-9~~ work similarly to 3A, 4A and 5A-5; the start of count depends on the state of 5A-7 and the jumpering to 3B and 4B. L1, L2 and L4 are the line counts of the 8 x 8 character matrices; L8 through L256 are the line counts for the text lines on screen. 7A and 1B-9 latch the vertical sync. 6B-18 is part of the video count buffer. The jumpers select 60 Hz or 50 Hz input (see Hardware Modifications, 110 V 60 Hz to 220 V 50 Hz).

Bit 5, Port ØFEH can be tested to time vert. sync. pulse.

CPU Clock Generator

5B-5 and 5B-9 divide the 12 MHz clock by 3, producing approximately 4 MHz. 9A-5 divides by 2 to give approximately 2 MHz.

Video Interface

See Fig 15

The pre-programmed 6331 PROM 2B is the heart of the screen controller. 2B controls all the buffers on the page, as follows:

- When the CPU is ***not*** addressing the screen RAMs, 6D and 7D go high-impedance.
- When the CPU is writing on the screen, the counter buffer (6A and 6B) and the data bus buffer (7D) go high-impedance.
- When reading the character generator ROM, or reading and writing character generator RAM, an address comes from the CPU through the TB and DL buses; both data bus buffers are on.

CPU Circuitry

The reset circuitry is based on a pair of one-shots, 13E-5 and 13E-9. The CPU $\overline{\text{BUSAK}}$ signal is buffered by 9E and fed to all the buffers on the page. At $\overline{\text{BUSAK}}$, they all go high-impedance; this allows a Direct Memory Access (DMA) into the Sorcerer.

The CPU is a Zilog Z80; refer to the Zilog Z80 CPU Technical Manual, Zilog part number 03-0029-01.

ROM Circuitry

At restart, the Z80 begins executing code at address 0000; however, the Power-On Monitor resides near the top of the memory map. The ROM decode shifts the CPU into the Monitor at reset. The reset signal pulls 11E-9 low; this disables the RAMs and unconditionally enables the ROMs. The first instruction in the Monitor is Jump E062H. For any address from E000H to FFFFH, 9B gives the UP8K signal; address E062H sends 8A-15 low, which sets the latching flip-flop 11E, and enables the bottom ROM.

NOTE

The RAM is always enabled, except during the first three instruction fetches after a reset.

I/O and UART Circuitry

5E and 6E handle I/O requests. 5E-8 gives the $\overline{\text{I/O}}$ request signal, and enables both halves of 6E. The $\overline{\text{RD}}$ and $\overline{\text{WR}}$ signals are the other enables for the first and second halves of 6E, respectively. The following I/O port designations come into 6E and A0 and A1:

see I/O Schem. #6-F

Signal	Port	A1	A0
UART data (serial interface)	FCH	0	0
UART status	FDH	0	1
Sorcerer housekeeping input	FEH	1	0
Sorcerer user output (parallel interface)	FFH	1	1

The Sorcerer input port and the parallel input port are 3-state buffered by 2C and 7H, respectively. The enable signals for the buffers come from 6E. The Sorcerer output port and user output port are 8-bit latched, by 1C and 8H respectively. Figure 16 shows the parallel output port timing signals.

The UART is a General Instruments AY-3-1015; refer to the manufacturer's technical publications. 12F buffers the UART output.

UART uses Pos. Logic, +Ve
+5V = LOGIC 1
ØV = LOGIC Ø

Interfaces

Cassette/UART Interface: This circuit communicates with the tape interface board.

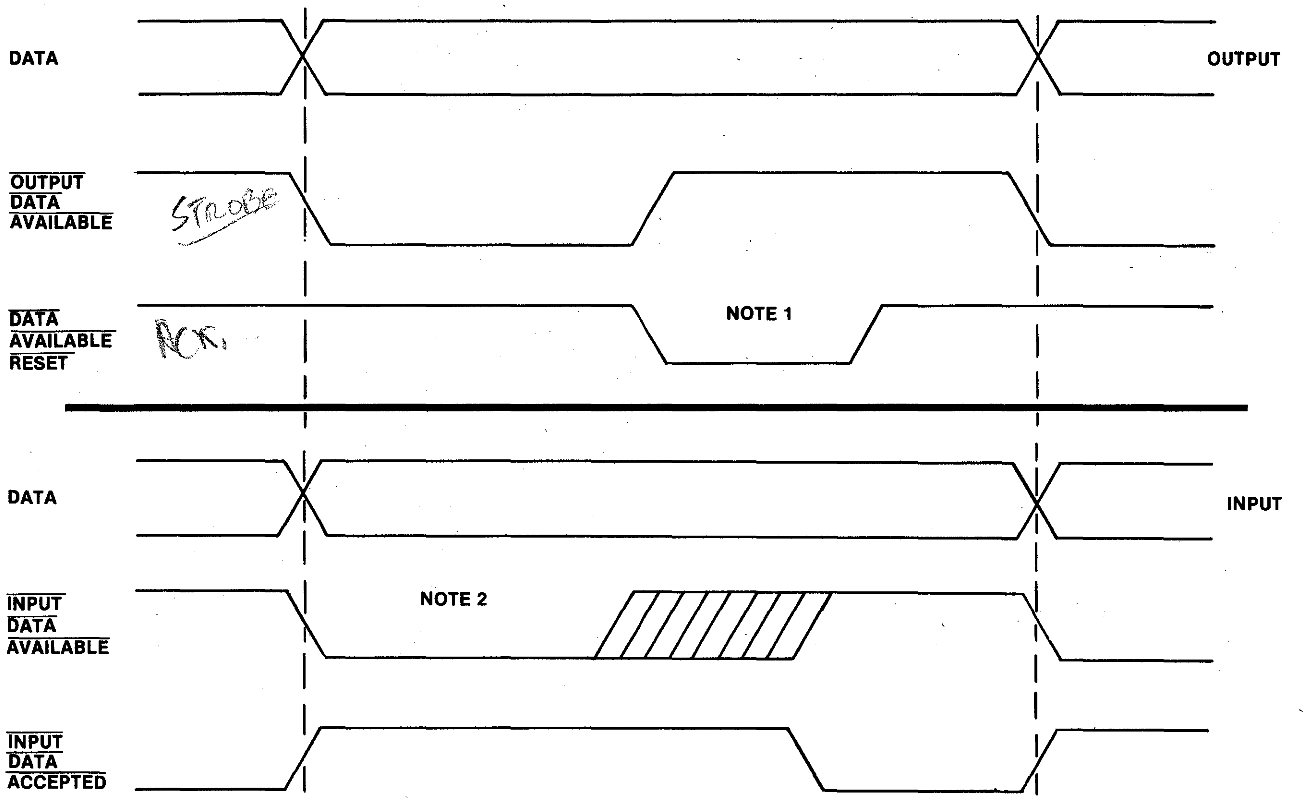
Cassette Motor Drivers: 9F and Q2 form a Darlington pair; the *reversed* diode CR2 is surge protection for Q2. 9H, Q3, and CR3 are exactly similar.

Power Supply: The transformer has two primary windings in parallel; to convert the power supply to 220 V input, disconnect the windings and reconnect in series (see Hardware Modifications, 110 V 60 Hz to 220 V 50 Hz).

Exidy Bus Drivers: The CPU Control, Address, and Data signals are bi-directionally buffered by 2H, 3H, 4H, 5F, and 6H. The bi-directional buffering allows DMA.

S-100 Control: When anything is happening on the logic board or the tape interface board, 5H receives an input. This disconnects the S-100 Expansion Unit.

Figure 16. Parallel Port Timing Chart

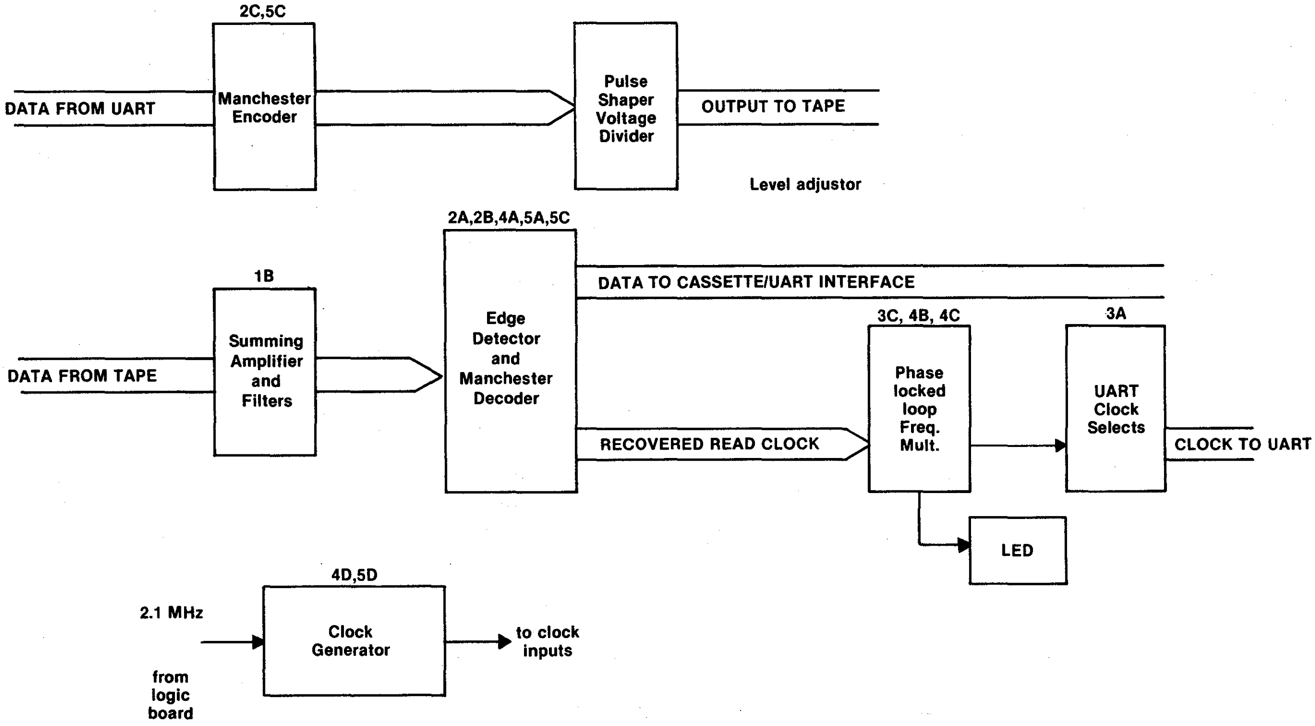


Notes:

1. Data Available Reset Pulse = 300ns < DAR < 10μs

2. Input Data Available minimum 300ns, maximum = any length as long as it is not still low when input data accepted goes low

Figure 17. Tape Interface Block Diagram



THEORY OF OPERATION, TAPE INTERFACE

General

The tape interface board translates between the UART data format (non return to zero) and the tape cassette format (frequency shift). The frequency-shift format uses a high frequency for logic 1, and a low frequency for logic 0 (see Figure 18). At 1200 baud, a logic 1 is 1 cycle of 1200 Hz and a logic 0 is ½ cycle of 600 Hz; at 300 baud, a logic 1 is 8 cycles of 2400 Hz and a logic 0 is 4 cycles of 1200 Hz. In both cases, the time required to transmit a logic 1 is the same as the time to transmit a logic 0.

The interface also adjusts the output signal levels to approximately 250 mV p-p for the tape recorder AUX input and approximately 50 mV p-p for the tape MIC input. A jumper at board location 1C allows a 4 V p-p signal instead, for digital recorders (jumper points A,C).

Manchester Encoder

Flip-flop 5C-5 synchronizes the input data with the 1200 Hz clock, triggering on the positive edge of the clock pulse. The signal is inverted in passing through 5C-5, but is otherwise unchanged. 2C-5 and 2C-9 frequency encode the data, giving a high frequency for logic 1 and a low frequency for logic 0.

Level Adjustor/Pulse Shaper

C9 rounds the corners of the square waves (audio recorders don't like square waves); C10, 11, 12, 13 are DC isolators. The output voltage jumper (location 1C) is part of this circuit.

Clock Selector

This circuit selects a clock rate for the manchester encoder, dependent on the selected baud rate.

Frequency Divider (x 1/55)

2D and 3D form a six stage binary up-counter, which counts from 9 to 64 and then sends a clock pulse from 3D-13 and a carry from 3D-15 to 5B-9; the carry starts the next count cycle.

Clock Pulse Generator

This circuit is simply a five stage binary up-counter.

Summing Amplifer/Low Pass Filter

This circuit allows input from two tape units, by preventing the non-operating unit from disturbing the signal of the operating unit. The output is 62% of the input signal, partially filtered. C1 is a DC isolater; C2 and the parallel 130 ohm resistor (R6) form a 3700 Hz low-pass filter. The 3.3 Megohm resistor (R5) biases 1B-7 to the center of its range (+4.5 V to −5 V) so that the positive and negative portions of the wave are equally clipped.

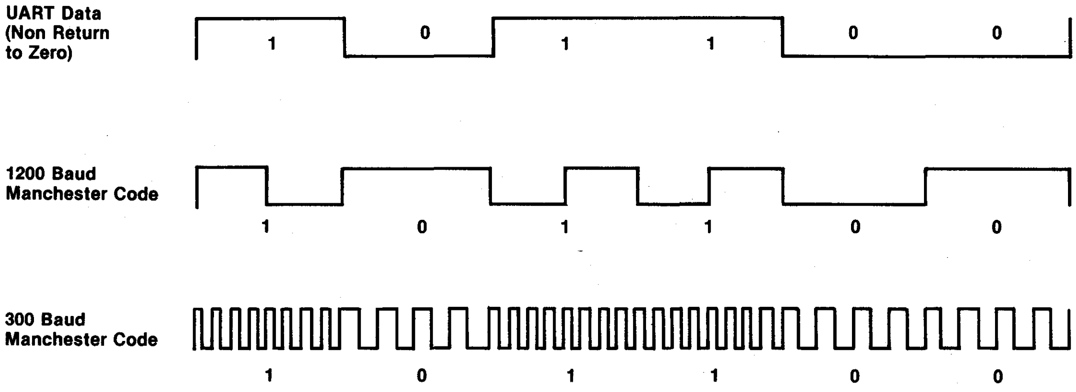
Second-Order Linear-Phase High Pass Filter

This circuit removes flutter. C3, C4, R9, and R10 determine the high-pass cutoff at 300 Hz.

3400 Hz Low-Pass Filter

This filter reduces high-frequency noise and prevents op-amp oscillation. C6 is a DC isolator.

Figure 18. UART and Cassette Data Formats



Comparator/Low-Pass Filter

This filter converts the analog input signal to digital pulses. C7 and R13 reduce frequencies above 480 Hz. Any input signal above 0 V is converted to approximately +5 V at output; any input signal below 0 V is converted to approximately 0 V at output.

Bi-Directional Edge Detector

This circuit provides a positive-going pulse for each transition at the input. The Schmidt-trigger inverter 2A-2 reduces signal noise and squares the pulse edges. 2A-4, 2A-6, R16, and C8 form a delay line, with a delay on the order of a few microseconds. 2B-3 is an exclusive-OR gate which provides a high-going signal during the time the input signal has ***not*** passed through the delay line.

Manchester Decoder

2B-6 is used as an inverter, to force CASSREAD high when RS232 is chosen. 5A-5 is set by the pulses from the bi-directional edge detector and produces a recovered clock signal. 4A is a binary up-counter which functions as one-shot to reset 5A-5. 4A counts from 4 to 15 before firing — this determines a critical period during which 5A-9 is either set or reset, and thus determines a maximum pulse width. Pulses narrower than this max width are considered 1s; wider pulses are 0s. 5C-9 synchronizes the 5A-9 signal with 5A-5's recovered clock signal.

Frequency Multiplier (x 8 or x 16)

3C is a phase-locked loop and 4B is a frequency-divider. 3C has a voltage controlled oscillator, which is adjusted so that the signal out of 4B equals the recovered clock (that is, the signal out of 3C-4 is adjusted so that the signal into 3C-14 equals the signal in at 3C-3). 3C locks onto the recovered clock and provides a clock signal for the UART, at 16 times the data rate. 4C selects the working frequencies, depending on the chosen baud rate. C18 and its 150K resistor form a low-pass filtered feedback loop for the voltage controlled oscillator; C16, VR1, (location 1C) and the 68K and 100K (pin 1) resistors set the center frequency and frequency range.

Sync Indicator

2B-9 acts as a buffer. 3C-1 is high if the phase-locked loop is in sync. C17 filters out small pulses.

UART Clock Selector

This circuit selects working clock signals based on choice of RS232 or Cassette, and baud rate.

1/29/81 Stan Knight (Exidy)
Required change in older Sorcerer I to make them compatible c̄ S100 box. An
see Tech Note 10
External interrupt was coming in & not being handled.

Jumper 11F pin #18 to 5E pin #12 ?, 11

Cut trace connecting 5E pins #11 + #12.

NOTE: Did not work. Locked up sorcerer completely. My guess is that this was needed on the very earliest Sorcerer's (pre S100) and that this was corrected by time got to mine. Clearly the problem has to be in S100 not sorcerer since my S100 doesn't work c̄ TLJ's while hers does.

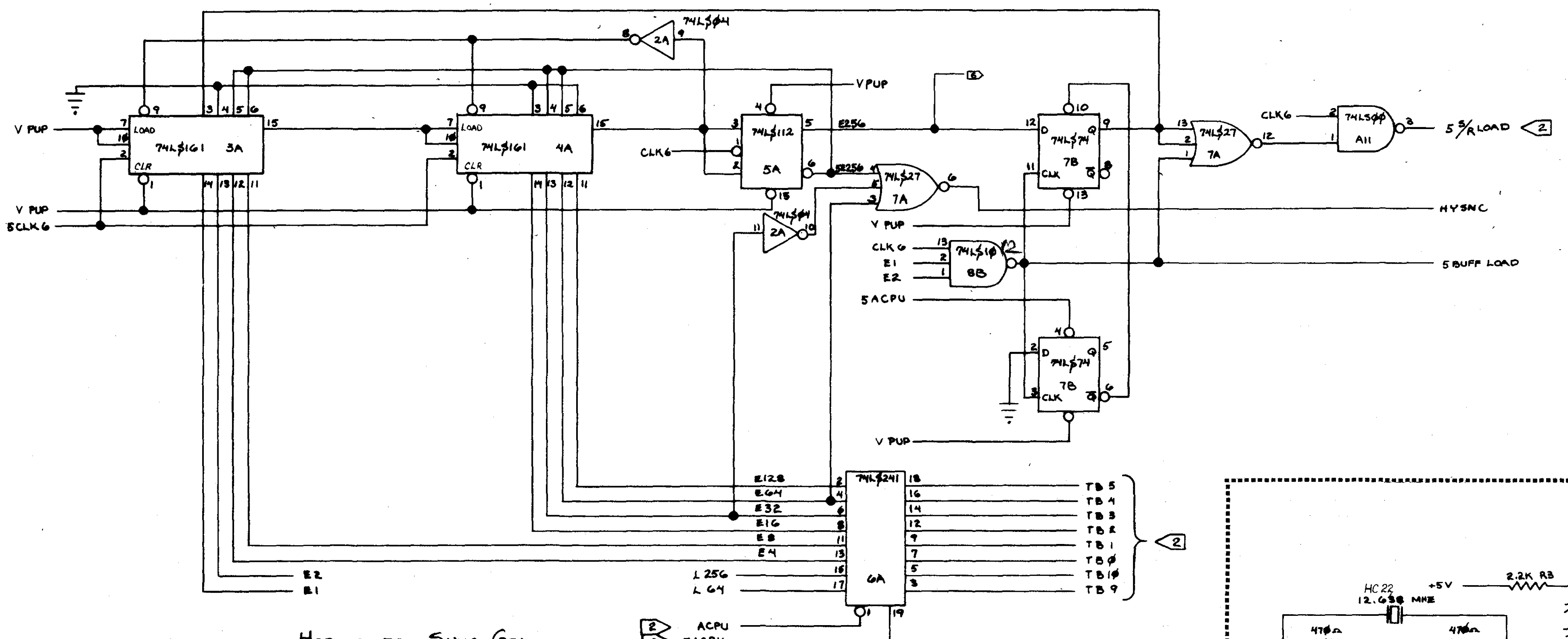
4/25/82: See Tech note #10. Or No wrong?

2A
3A
4A
5A
6A
7B
7A
A11 (?)
8B

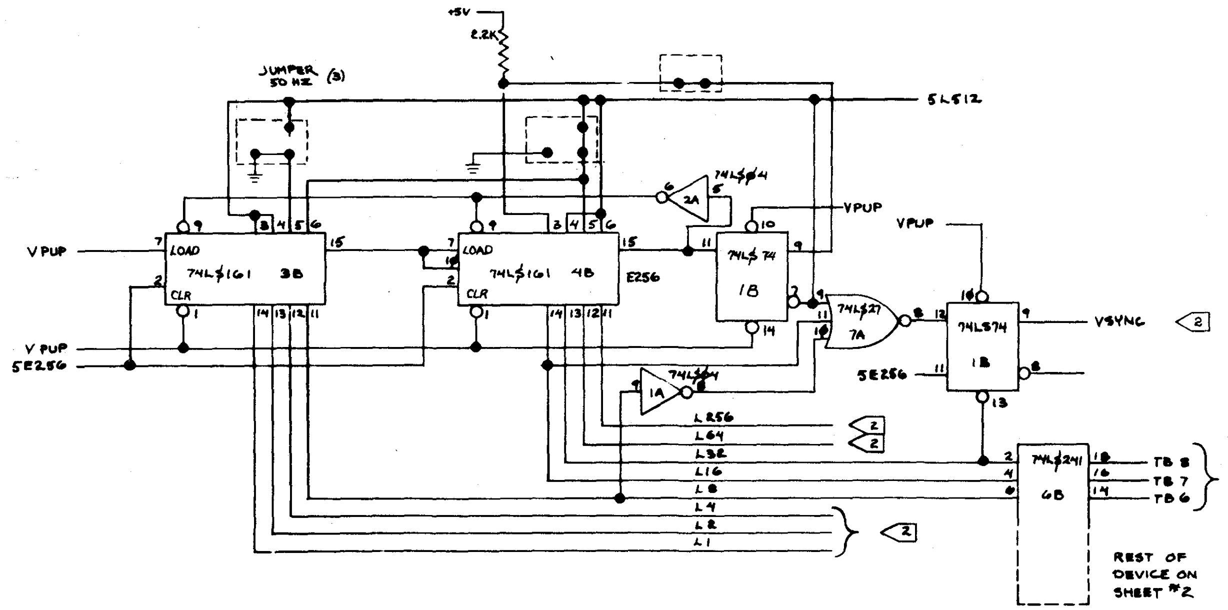
1A
1B
5B

1A
1B
2A
3B
4B
7A
6B

ARROWS
2
3
7

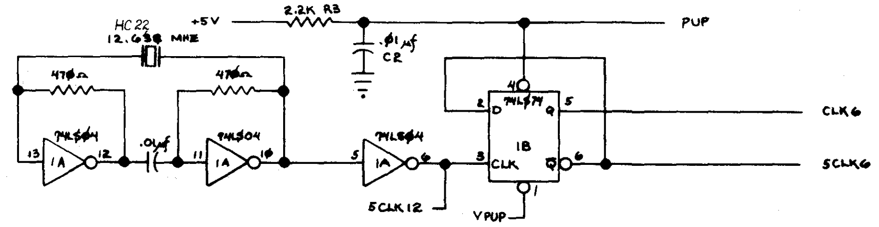


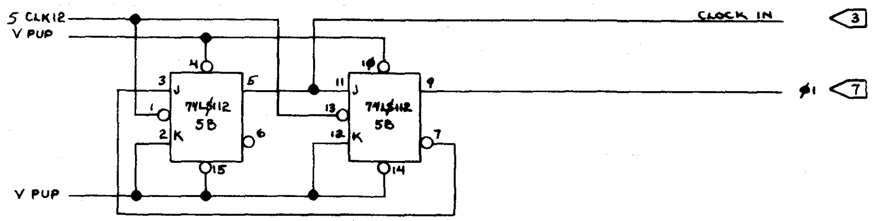
Horizontal Sync Gen



Vertical Sync Generator

Video Clock Generator





CPU CLOCK GENERATOR
PART I

Exidy inc.

Exidy Data Products Division
969 W. Maude Avenue, Sunnyvale, CA 94086

SCALE: N/A
APPROVED BY:
DRAWN BY: RHM
DATE: 8-3-78
REVISED
TITLE: MASTER TIMING FOR EXIDY STANDARD VIDEO
PROJECT NO. SC30-3135-1A
DRAWING NUMBER 1-A

4D
3D
2D
6B½

6D
7D

4C
5C
6C
7C

5D

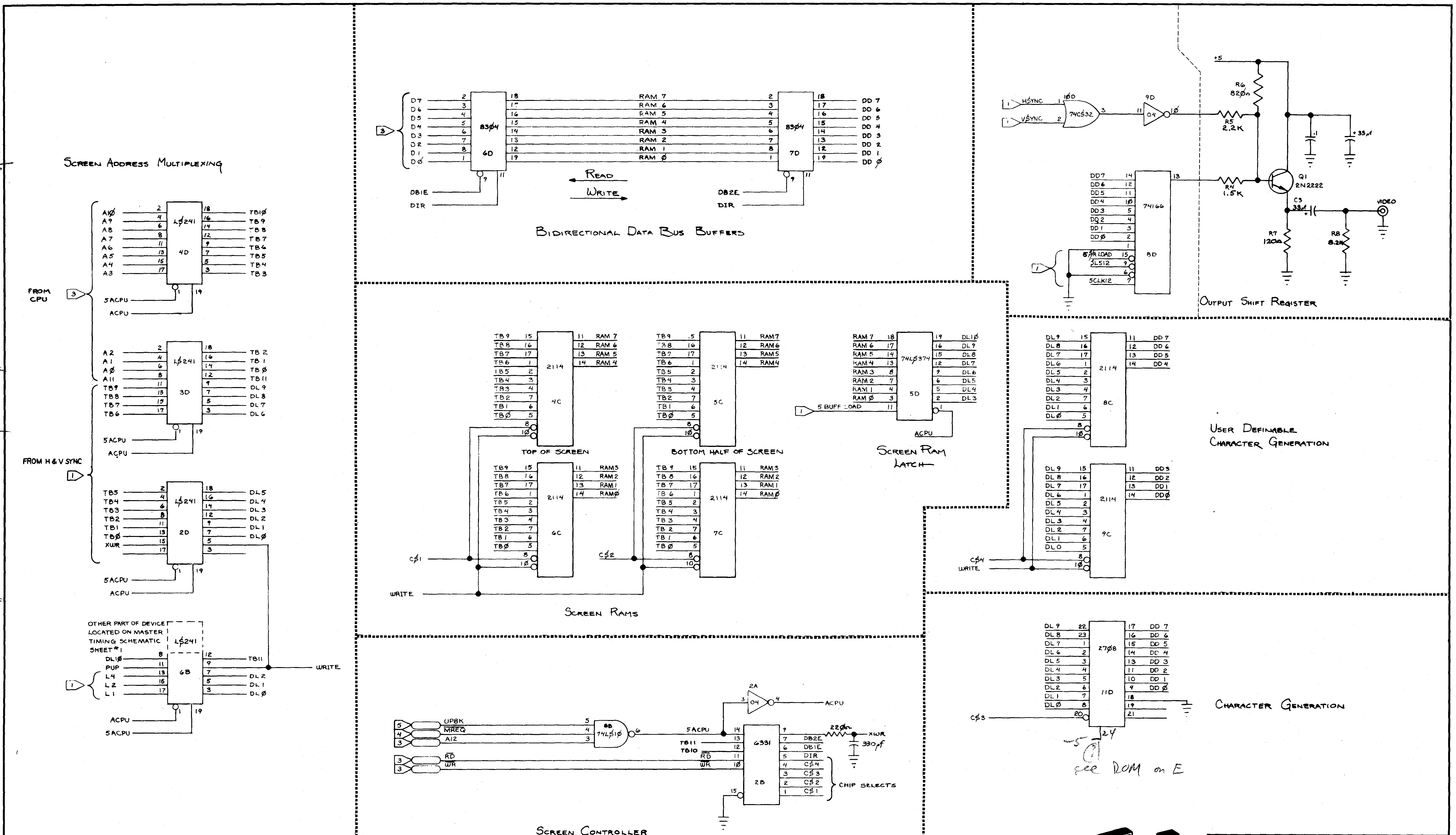
8B
04?
2B

10D
9D
8D

8C
9C

11D

Arrows
3
1
4
5



Exidy inc.

Exidy Data Products Division
969 W. Maude Avenue, Sunnyvale, CA 94086

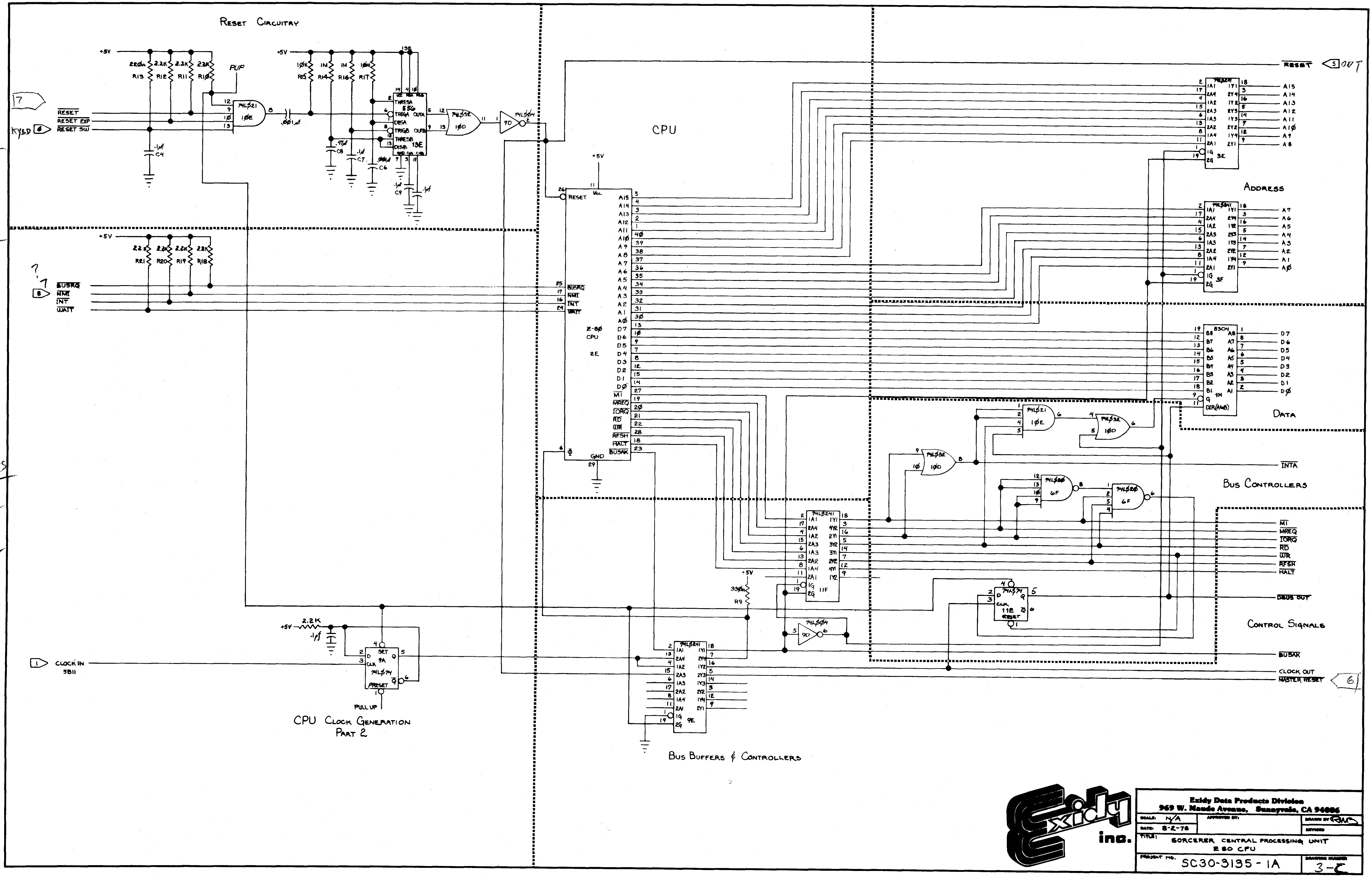
SCALE: N/A
APPROVED BY:
DRAWN BY: [illegible]
DATE: 8-2-78
REVISED
TITLE: SORCERER VIDEO INTERFACE
30 LINES 64 CHARACTERS ADDRESS F000 > FFFF
PROJECT NO. SC30-3135-1A
DRAWING NUMBER 2-B

10E
10D
9D
9A

2E
11F
9D
9E

3E
3F
1H
10D
10E
6F
11E

Arrows
6
8
1
5



Exidy inc.

Exidy Data Products Division
969 W. Maude Avenue, Sunnyvale, CA 94086

SCALE: N/A
DATE: 8-2-78
APPROVED BY:
DRAWN BY RW
REVISED
TITLE: SORCERER CENTRAL PROCESSING UNIT
Z 80 CPU
PROJECT NO. SC30-3135 - 1A
DRAWING NUMBER 3-C

12A
11A×3
10B×2
12B
11B

11C
12C
12E
9A
9B
10C
10A

B-20A
B-20B

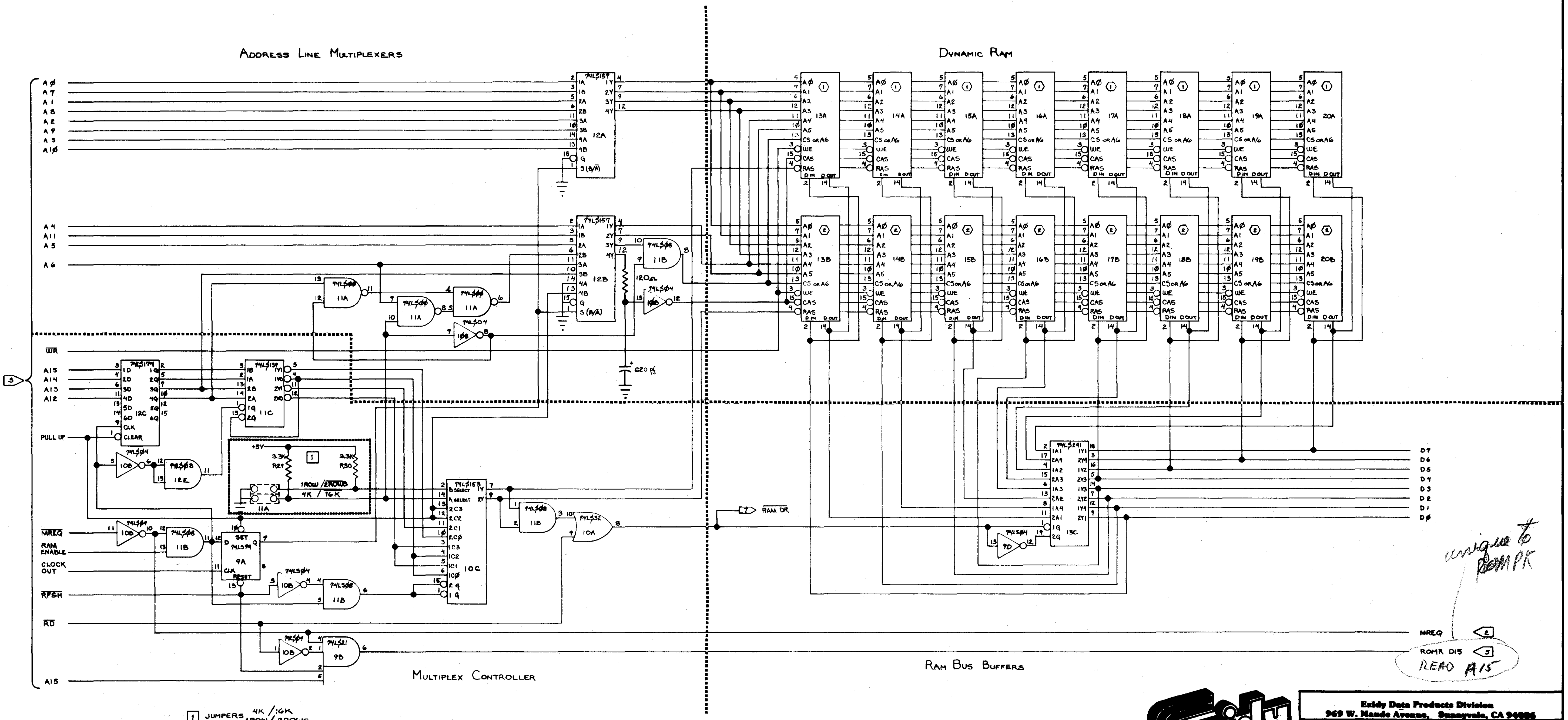
9D
13C

ARROWS
3
7
2
5

NOTES :

(1) 2104 OR EQUIVALENT IN 4K VERSION, 2116 OR EQUIVALENT IN 16K VERSION.

(2) EMPTY IN 4K AND 16K VERSION, 2104 OR EQUIVALENT IN 8K VERSION, 2116 OR EQIVALENT IN 32K VERSION.



unique to ROMPK

READ A15

Exidy Data Products Division
969 W. Maude Avenue, Sunnyvale, CA 94086

SCALE: N/A	APPROVED BY:	DRAWN BY RMB
DATE: 8-2-78		REVISED
TITLE: SOCERER DYNAMIC RAM SECTION		
PROJECT NO. SC30-3135 - 1A		DRAWING NUMBER 4-D

Exidy inc.

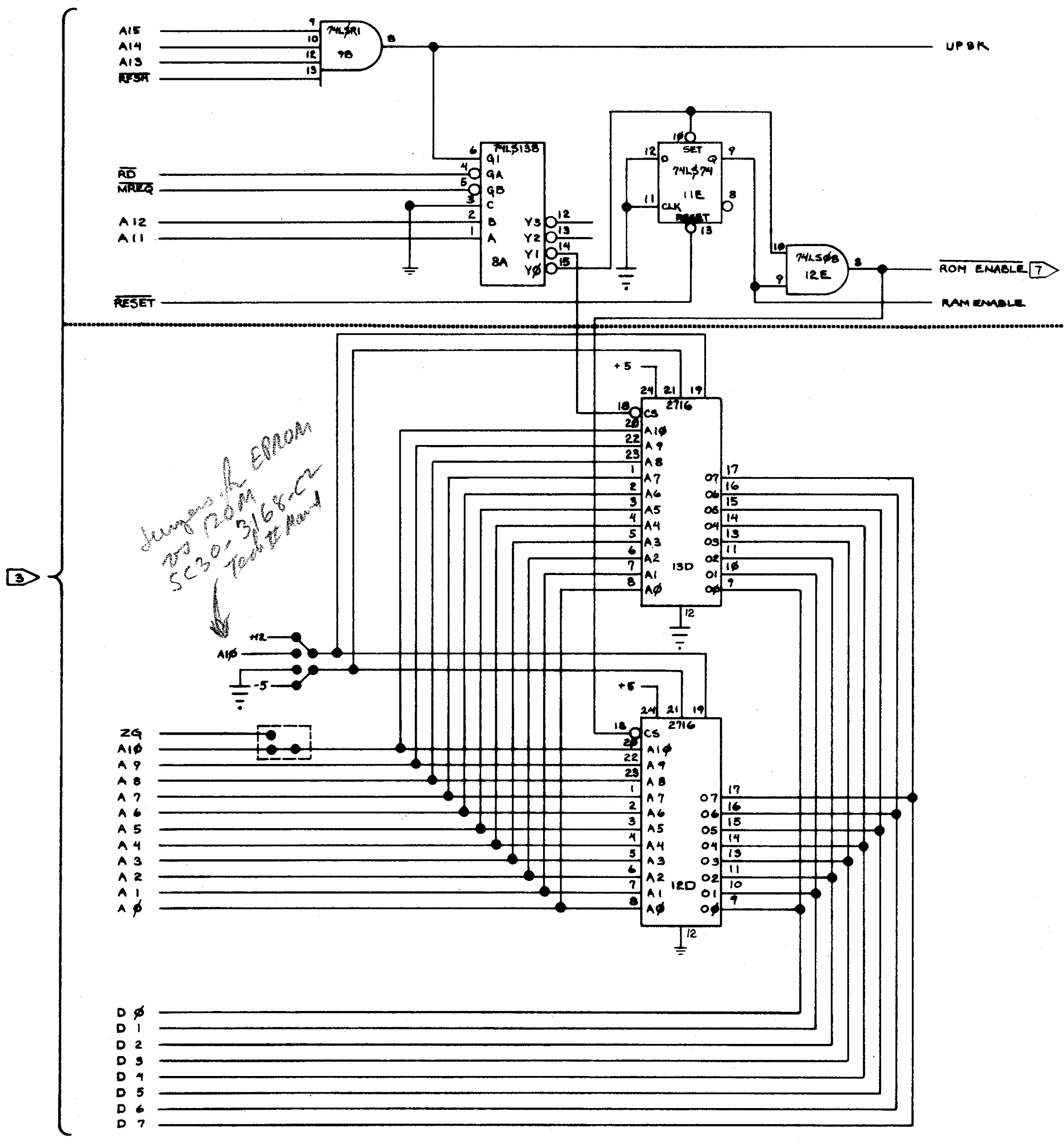
9B
8A
11E
12E

12D
13D

ARROWS

3
4
7

ROM Decode



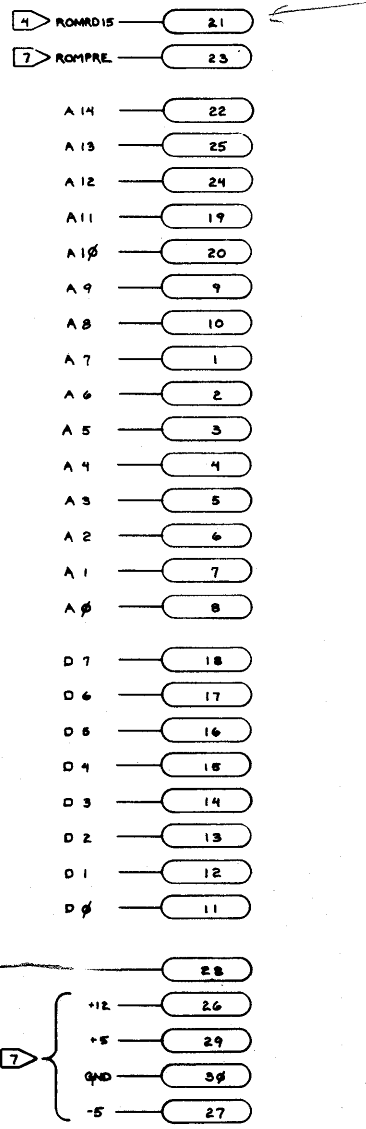
Jumpers for EPROM vs ROM SC30-3168-C2 Tech II Manual

ROM

Tech II Manual shows.

13D→2E – EYMO I 2

12D→1E – " 1



ROM PACK CONNECTOR
30 PIN CONNECTOR

ROM READ A15 (?)
Unique to ROMPAK

Tech II
SC30-3168-C2
E18 E19
WR
Removable
Jumper
? for EPROMS
vs Roms

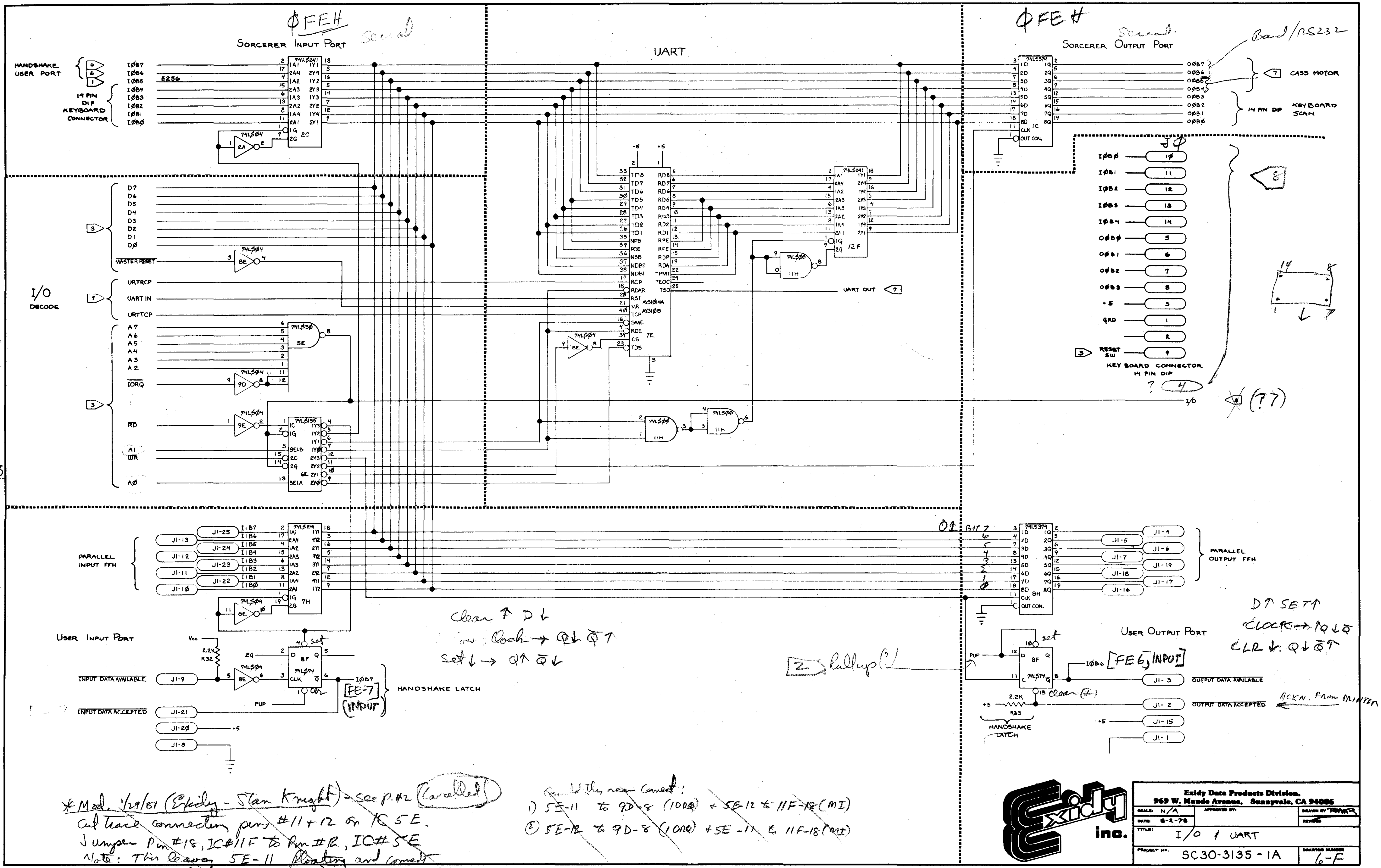
Exidy inc.

Exidy Data Products Division
969 W. Maude Avenue, Sunnyvale, CA 94086

SCALE: N/A	APPROVED BY:	DRAWN BY RMB
DATE: 8-2-78		REVISED
TITLE: ROM SECTION		
PROJECT NO. SC30-3135 - 1A		DRAWING NUMBER 5-E

2A
2C
8E
5E
9D
9E
6E
7H
8E
8F
7E
11H
12F
1C
8H
8F

ARROWS
6
1
3
7
8



*Mod. 1/29/81 (Exidy - Stan Knight) - see p. #2 (Cancelled)
~~Cut trace connecting pins #11 + 12 on IC 5E.~~
~~Jumper Pin #18, IC#11F to Pin #12, IC#5E~~
~~Note: This leaves 5E-11 floating and connects~~
~~5E-12 to both IORQ thru 9D and to M1~~

Not changed in Sorcerer II

~~Could they mean connect:~~
~~1) 5E-11 to 9D-8 (IORQ) + 5E-12 to 11F-18 (M1)~~
~~② 5E-12 to 9D-8 (IORQ) + 5E-11 to 11F-18 (M1)~~

Exidy inc.

Exidy Data Products Division,
969 W. Maude Avenue, Sunnyvale, CA 94086

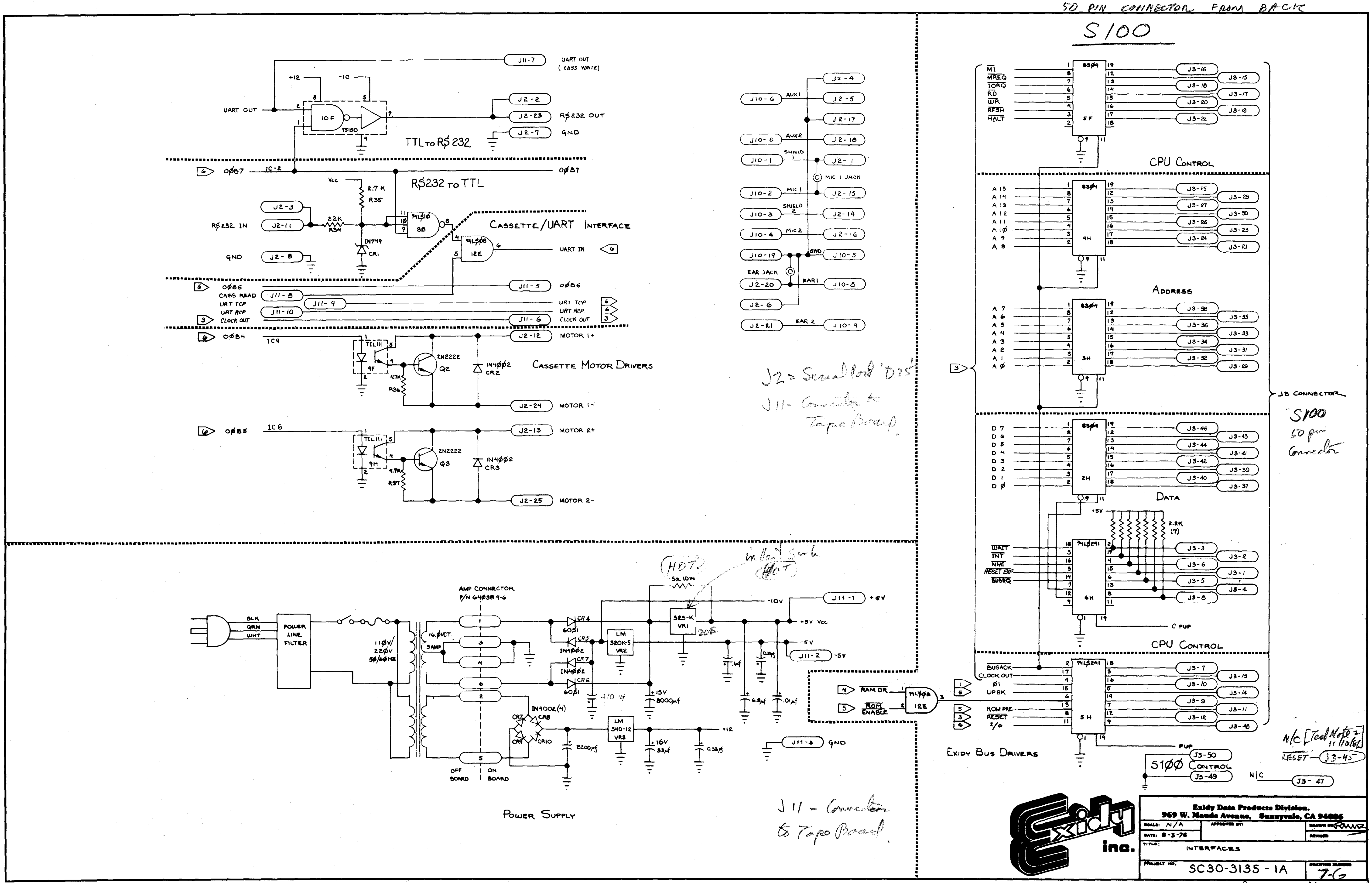
SCALE: N/A	APPROVED BY:	DRAWN BY
DATE: 8-2-78		REVISED
TITLE: I/O & UART		
PROJECT NO. SC30-3135 - 1A		DRAWING NUMBER 6-F

7 5 3 1
8 6 4 2
50 PIN CONNECTOR FROM BACK

10F
8B
12E
9F
9H

5F
4H
3H
2H
6H
5H

ARROWS
6
3
4
5
1



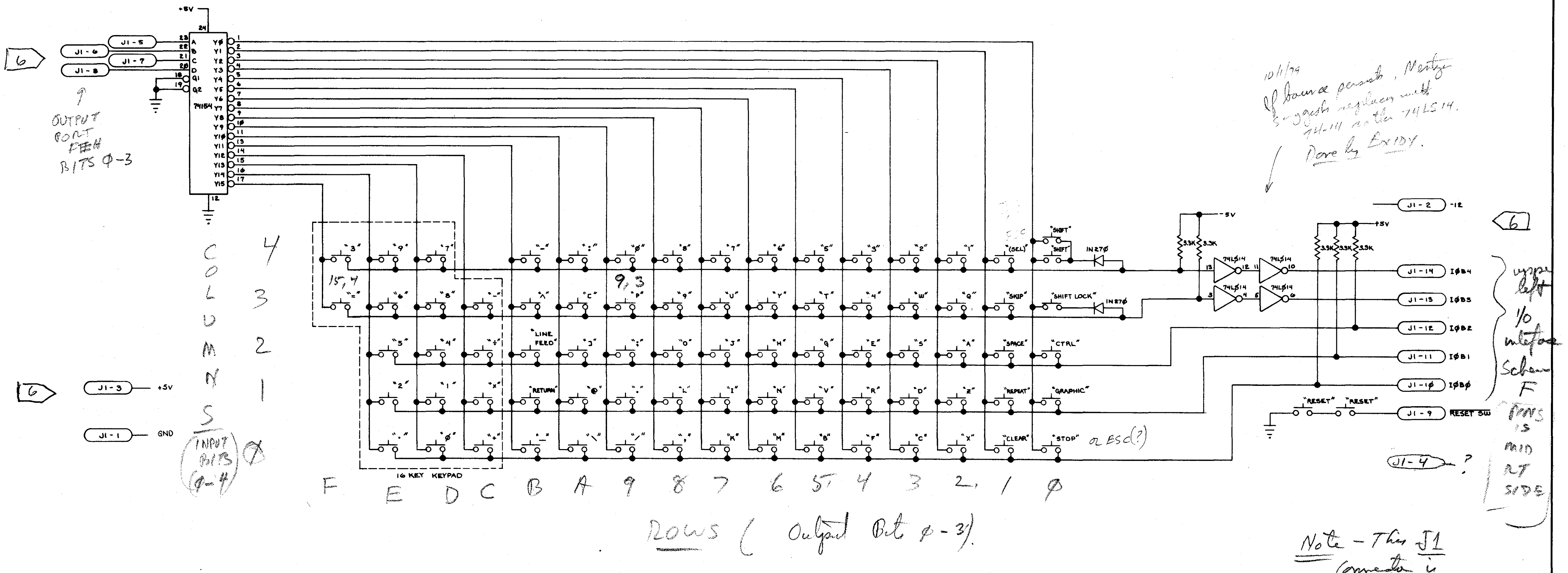
Exidy inc.

Exidy Data Products Division,
969 W. Maude Avenue, Sunnyvale, CA 94086

SCALE: N/A	APPROVED BY:	DRAWN BY RMR
DATE: 8-3-78		REVISED
TITLE: INTERFACES		
PROJECT NO. SC30-3135 - 1A		DRAWING NUMBER 7-G

See also SA 4(6): 135, Sept 82

2/13/83 7408 IC & socket installed by Mentzer.



10/1/79 If bounce persists, Mentzer suggests replacing with 74-14 in the 74LS14. Done by Exidy.

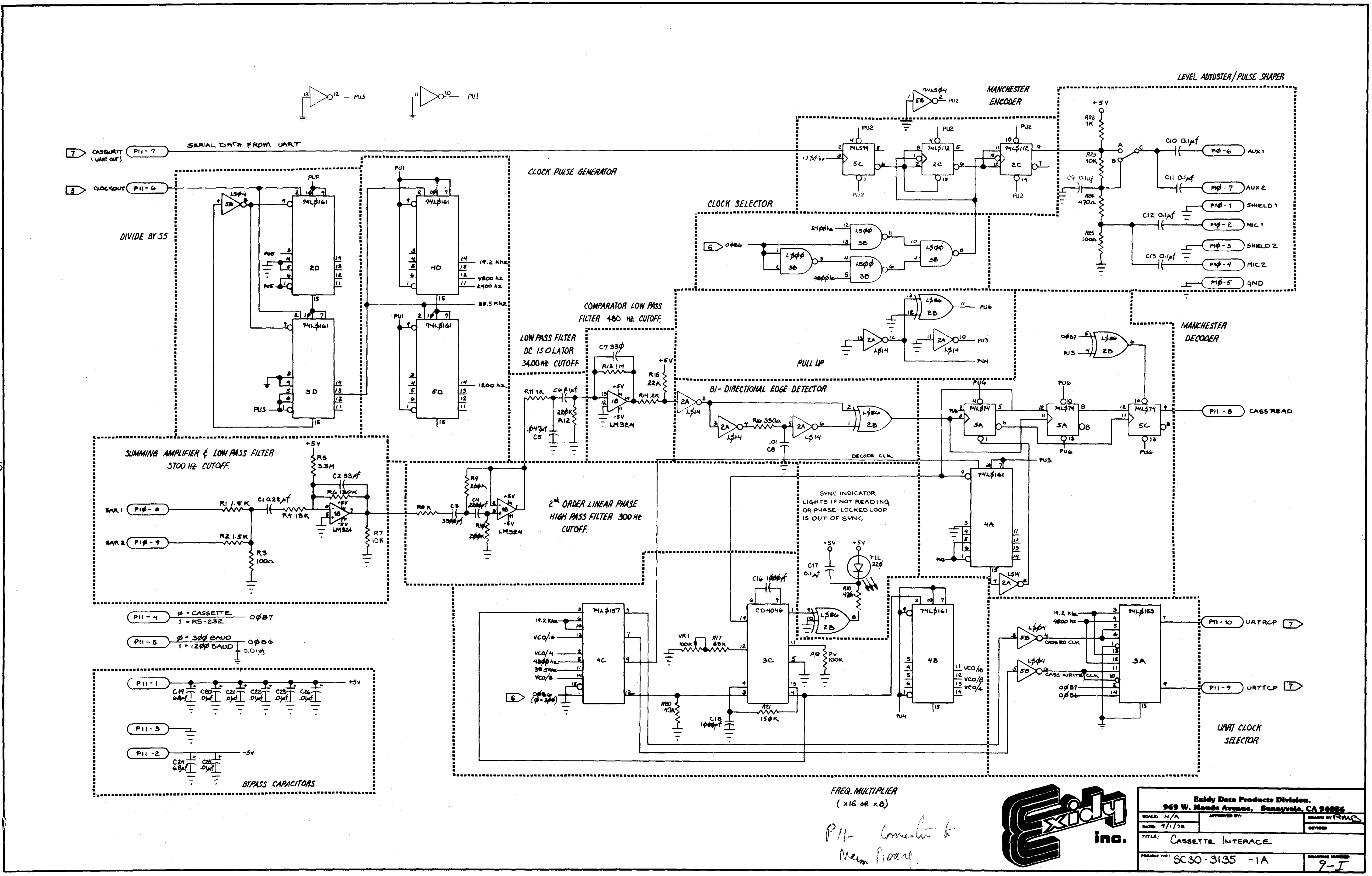
Note - This J1 connector is labelled J0 on #6

Exidy inc.

Exidy Data Products Division 969 W. Maude Avenue, Sunnyvale, CA 94086		
SCALE: N/A	APPROVED BY:	DRAWN BY:
DATE: 8-3-78		REVISED
TITLE: KEYBOARD SCHEMATIC		
PROJECT NO: SC30-3135-1A		DRAWING NUMBER 8-H

TAPE ~~In.Face~~ BRD
5B
2D
3D
4D
5D
1B
4C
5C
2C
3B
2A
2B
5A
4A
3C
4B
3A

ARROWS
6
7
3

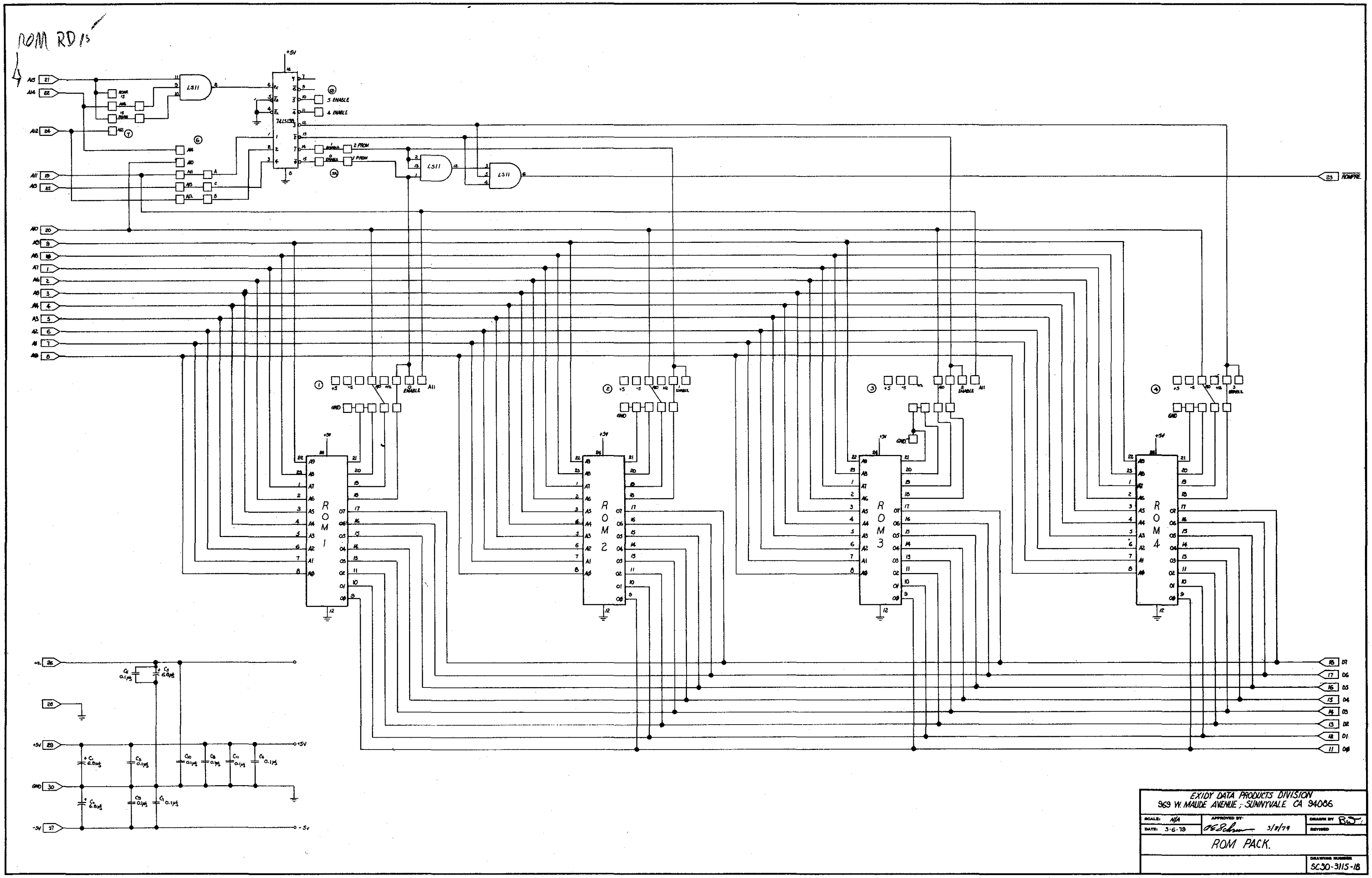


P11- Connector to Main Board

Exidy inc.

Exidy Data Products Division,
969 W. Maude Avenue, Sunnyvale, CA 94086

SCALE: N/A	APPROVED BY:	DRAWN BY RMB
DATE: 7/1/78		REVISED
TITLE: CASSETTE INTERACE		
PROJECT NO: SC30-3135 -1A		DRAWING NUMBER 9-I



EXIDY DATA PRODUCTS DIVISION
969 W. MAUDE AVENUE ; SUNNYVALE CA 94086

SCALE: N/A
DATE: 3-6-79
APPROVED BY: 3/8/79
DRAWN BY BWT.
REVISED

ROM PACK.

DRAWING NUMBER
SC30-3115-18

INDEX TO SCHEMATICS

A - Timing
B - Video Interface
C - CPU
D - Dynamic RAM
E - ROM
F - I/O + UART
G - Interfaces; Cassette, UART, power supply, S100
H - Keyboard
I - Cassette Interface Board
J - Rom Pak

Parts Lists

INDEX TO DEVICES

	A	B	C	D	E	F	G	H
1	A	A	F					C
2	AF	B	F	B	C			G
3	A	A		B	C	C		G
4	A	A	B	B				G
5	A	A	B	B	F	G		G
6	A	AB	B	B	F	C		G
7	A	A	B	B	F			F
8	E	ABG	B	B	F	F		F
9	CD	DE	B	BCDF	CF	G		G
10	D	D	D	BC	C	G		
11	D (?A)	D	D	B	CE	C		F
12	D	D	D	DE	EG	F		
13	D	D	D	E				
14-20	D	D						

LOGIC ARROWS	
1	BC FG
2	AD
3	ABDEFGI
4	BEG
5	BCDG
6	CFGI
7	ADEFI
8	CF

Complete Assemblies

Part	Qty/ Sorcerer	Exidy Part #
Logic Board Assembly	1	SC77-3135
Tape Interface Board Assembly	1	SC77-3125
Keyboard Assembly	1	SC77-3145
Transformer Assembly	1	SC63-4026A
ROM PAC PCB Assembly (does not include case or PROMs)	1	SC77-3115

Logic Boards ICs Unprogrammed

Part	Qty/ Board	Locations	Exidy Part #
74LS00	2	11A, 11H	SC48-2300
74LS04	5	1A, 2A, 8E, 9D, 10B	SC48-2302
74LS08	2	11B, 12E	SC48-2312
74LS10	1	8B	SC48-2306
74LS20	1	6F	SC48-2303
74LS21	2	9B, 10E	SC48-2316
74LS27	1	7A	SC48-2304
74LS30	1	5E	SC48-2324
74LS32	2	10A, 10D	SC48-2315
74LS74	5	1B, 7B, 8F, 9A, 11E	SC48-2305
74LS112	2	5A, 5B	SC48-2071
74LS138	1	8A	SC48-2307
74LS139	1	11C	SC48-2321
74LS153	1	10C	SC48-2322
74LS155	1	6E	SC48-2325
74LS157	2	12A, 12B	SC48-2323
74LS161	4	3A, 3B, 4A, 4B	SC48-2308
74LS166	1	8D	SC48-2309
74LS174	1	12C	SC48-2333
74LS241	13	2C, 2D, 3D, 3E, 3F 4D, 5H, 6A, 6B, 6H, 7H, 9E, 11F	SC48-2328
74S241	2	12F, 13C	
74LS374	3	1C, 5D, 8H	SC48-2314
CPU - Z80	1	2E	SC48-0Z80
2014 (4K RAM)	16	RAM sockets	SC48-2318
2114	6	4C, 5C, 6C, 7C, 8C, 9C	SC48-2334
75150	1	10F	SC48-2335
8304	7	1H, 2H, 3H, 4H, 5F, 6D, 7D	SC48-2327
UART AY-3-1015	1	7E	SC48-2319
TIL 111	2	9F, 9H	SC48-3040
NE 556	1	13E	SC48-2212

Logic Board Pre-Programmed PROMs

Part	Qty/ Board	Locations	Exidy Part #
2716 PROM EXMON 1	1	12D	SC48-2350
2716 PROM EXMON 2	1	13D	SC48-3000
2708 PROM Character Generator	1	11D	SC48-3006
6331 PROM (BRUCE) Screen Control	1	2B	SC48-6331

Logic Board Discrete Semiconductors

Part	Qty/ Board	Locations	Exidy Part #
IN4002	7	9H, 14E, 15E, 16D, 16E, 17D, 17E	SC46-3025
IN749	1	9B	SC46-3051
2N2222	3	9F, 9H, 9H	SC47-3039
LM 323 K	1	20E	SC48-2336
LM 340T-12	1	15D	SC48-2338
LM 320T-5	1	15D	SC48-2337
60S1	2	16E, 17E	SC46-3016
12.638 Crystal	1	1A	SC45-3038

Logic Board Resistors

Part	Qty/Board	Exidy Part #
*5 ohm 10W	1	SC55-5005
120 ohm ¼W	2	SC59-5139
220 ohm ¼W	2	SC59-5004
330 ohm ¼W	1	SC59-5136
470 ohm ¼W	2	SC59-5135
820 ohm ¼W	1	SC59-5144
1.5K ¼W	1	SC59-5116
2.2K ¼W	24	SC59-5110
2.7K ¼W	1	SC59-5105
4.7K ¼W	2	SC59-5095
8.2K ¼W	1	SC59-5085
10K ¼W	2	SC59-5080
1 Meg ¼W	2	SC59-5025

* Not present in early model boards

Logic Board Capacitors

Part	Qty/Board	Exidy Part #
330 pF 5% ceramic disc	1	SC23-4065
.001 µF 16V ceramic disc	2	SC23-4060
.01 µF 16V ceramic disc	1	SC23-4050
.1 µF ceramic disc	45	SC23-4035
.33 µF 16V ceramic disc	24	SC23-4031
.1 µF dip tant.	1	SC21-4023
.47 µF dip tant.	1	SC21-4025
33 µF 16V dip tant.	2	SC21-4010
33 µF 35V dip tant.	6	SC21-4010
*100 µF 25V electrolytic axial leads	1	SC20-4006
470 µF 25V electrolytic axial leads	1	SC20-4004
1500 µF 25V axial leads	1	SC25-1000
8000 µF 15V radial leads	1	SC25-1001

* deleted from new boards

Tape Interface Board, ICs and LED

Part	Qty/ Board	Locations	Exidy Part #
74LS00	1	3B	SC48-2300
74LS04	1	5B	SC48-2302
74LS14	1	2A	SC48-2340
74LS74	2	5A, 5C	SC48-2305
74LS86	1	2B	SC48-2341
74LS112	1	2C	SC48-2075
74LS153	1	3A	SC48-2322
74LS157	1	4C	SC48-2323
74LS161	6	2D, 3D, 4A 4B, 4D, 5D	SC48-2308
LM 324	1	1B	SC48-2342
CD 4046 BC	1	3C	SC48-2343
TIL 220 (LED)	1	1E	SC46-3040

Logic Board Mechanical Parts

Part	Qty/Board	Exidy Part #
Heat Sink Wakefield #690-3	1	
Heat Sink Thermalloy #6072	1	SC68-3041
6-pin AMP header #640388-6 MTS	1	SC61-8064
6-pin AMP receptacle #640426-6 MTS	1	SC61-8065
25-pin D Soldertail female connector	2	SC61-8063
RCA Jack — Female Phono Pin Type	3	SC61-8066
30-pin Female Edge Connector #5253-310-1 (15 Position) AMP	1	SC61-8053
10-pin Male Molex Connector	2	SC61-8041
14-pin Soldertail (gold) DIP Socket	1	SC61-8060
16-Pin Soldertail (gold) DIP Socket	16	SC61-8062
24-Pin Soldertail (gold) DIP Socket	3	SC61-8045
C934002 40-Pin DIP Socket	1	SC61-8035
Screw 6-32x3/8" (Machine)	3	
#6 Star Washer	3	
6-32 Hex Nut	3	
Double sided adhesive foam tape (3M) 1/2" square	1	
Blank PC Board	1	SC77-3130

Tape Interface Board Resistors

Part	Qty/Board	Exidy Part #
100 ohm ¼W 5%	2	SC59-5140
330 ohm ¼W 5%	1	SC59-5136
470 ohm ¼W 5%	2	SC59-5135
1 K ¼W 5%	3	SC59-5125
1.5 K ¼W 5%	2	SC59-5116
4.7 K ¼W 5%	1	SC59-5095
10 K ¼W 5%	2	SC59-5080
13 K ¼W 5%	1	SC59-5078
68 K ¼W 5%	1	SC59-5050
100 K ¼W 5%	1	SC59-5045
130 K ¼W 5%	1	SC59-5061
150 K ¼W 5%	1	SC59-5040
200 K ¼W 5%	2	SC59-5035
220 K ¼W 5%	1	SC59-5034
1 Meg ¼W 5%	1	SC59-5025
3.3 Meg ¼W %	1	SC59-5018
100 K Pot (CTS)	1	SC54-5021

Tape Interface Board
Capacitors

Part	Qty/Board	Locations	Exidy Part #
* 2200 pF COG	1	1B (C4)	SC25-1003
* 3300 pF COG	1	1B (C3)	SC25-1004
* 1000 pF COG	2	3C, 4C (C16, C18)	SC25-1005
* .047 µF X7R	1	1B (C5)	SC25-1006
* 330 pF X7R	2	1B, 1B (C2, C7)	SC25-1007
0.2 µF disc	1		SC23-4030
0.1 µF disc	7		SC23-4035
0.01 µF disc	7		SC23-4050
6.8 µF tant	2		SC21-4015
*Must be silvered mica or temperature-stable ceramic.			

Tape Interface Board
Mechanical Parts

Part	Qty/Board	Exidy Part #
Molex 09-52-310 10-Pin Female	2	SC61-8042
Blank Board	1	SC77-3120

Keyboard

Part	Qty/Board	Exidy Part #
Keyboard PCB	1	SC77-3140
Key switch pads — 63	1	SC72-3050
Key top set — 63	1	SC72-3050
IN270 germanium diode	2	SC46-3015
74LS14	1	SC48-2340
74L154	1	SC48-2320
3.3 K ¼ watt res.	5	SC59-5100
14-pin DIP socket	1	SC61-8060
Key pad & tops — 4x4	1	SC73-3051

ROM PAC
Excluding PROMs

Part	Qty/PAC	Exidy Part #
ROM PAC PCB	1	SC77-3110
ROM PAC case	1	SC91-4000
Label overlay	1	SC89-2003
74LS11	1	SC48-2332
74LS138	1	SC48-2307
0.1 µF cer. disc	8	SC23-4035
6.8 µF DIP tant.	3	SC21-4015
24-pin DIP socket	4	SC61-8045

Standard BASIC ROM PAC
Pre-Programmed PROMs

Part	Qty/PAC	Exidy Part #
2716 PROM EXBS1-1	1	SC48-2346
2716 PROM EXBS1-2	1	SC48-2347
2716 PROM EXBS1-3	1	SC48-2348
2716 PROM EXBS1-4	1	SC48-2349

Transformer Assembly Parts

Part	Qty/Ass'y	Exidy Part #
Transformer	1	SC63-4026
A.C. power bracket	1	SC68-7090
Power cord	1	SC71-2328
Power cord retaining ring	1	SC74-5050
Power switch	1	SC72-3052
2K1 line filter	1	SC90-3000
1 amp fuse	1	SC60-6037
1 amp fuseholder	1	SC60-6038
#09-50-3061 6-pin Molex famale	1	SC61-8043
#08-50-0106 pins for 6-pin Molex	6	SC61-8044
Machine phil pan 6-32 x 3/8" screw	8	
6-32 Kep nut	8	
Shrink tubing		

General Mechanical Parts

Part	Qty/Sorcerer	Exidy Part #
Top & bottom casing (Main Unit)	1	SC91-4001
ROM PAC guide	1	SC91-4002
Top rear heat vent screen	1	SC68-7070B
Top side heat vent screen	1	SC68-7071
Bottom front heat vent screen	2	SC68-7080B
Bottom side heat vent screen	1	SC68-7081
Tape cassette cable	2	SC71-2327
Video cable	1	SC71-2326
RCA jack, female (phono pin)	3	
Keyboard case overlay — front	1	SC89-2000
Keyboard case overlay — back	1	SC89-2001
Logo overlay	1	SC89-2002
Write-on label	1	SC89-2004
Keyboard panel overlay	1	SC89-2005
Serial # label	1	
1/4 x 6/32 mach. screw	21	
6-32 x 5/8 (9/16) nylon (phenolic) standoff	4	
1/2 x 6/32 rd. head Allen — black	5	
6/32 kep nut	10	
#6 washer nylon	4	
6/32 x 1/2 metal standoff	6	
6/32 x 1/4 plastic screw	11	
6/32 x 1/2 plastic thread stock	1	
4-40 x 5/8” screw machine ph. pan hd	2	
6-32 x 3/8 phil pan machine screw	9	
4-40 Kep nut	2	
Rubber bands		

Table 9

Hexadecimal — Decimal Conversion

	4		3		2		1		
K	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	
	0	0	0	0	0	0	0	0	0 0 0 0
4	1	4096	1	256	1	16	1	1	0 0 0 1
8	2	8192	2	512	2	32	2	2	0 0 1 0
12	3	12288	3	768	3	48	3	3	0 0 1 1
16	4	16384	4	1024	4	64	4	4	0 1 0 0
20	5	20480	5	1280	5	80	5	5	0 1 0 1
24	6	24576	6	1536	6	96	6	6	0 1 1 0
28	7	~~29672~~ 28672	7	1792	7	112	7	7	0 1 1 1
32	8	32768	8	2048	8	128	8	8	1 0 0 0
36	9	36864	9	2304	9	144	9	9	1 0 0 1
40	A	40960	A	2560	A	160	A	10	1 0 1 0
44	B	45056	B	2816	B	176	B	11	1 0 1 1
48	C	49152	C	3072	C	192	C	12	1 1 0 0
	D	53248	D	3328	D	208	D	13	1 1 0 1
	E	57344	E	3584	~~D~~ E	224	E	14	1 1 1 0
	F	61440	F	3840	F	240	F	15	1 1 1 1
	Bits 4-7		Bits 0-3		Bits 4-7		Bits 0-3		
	HIGH ORDER BYTE				LOW ORDER BYTE				

Clock Time

2.1 MHz = 476 nanosec /cycle.

4.76×10^{-1} μsec /cycle ≃ $\boxed{0.5 \text{ μsec/cycle}}$

4.76×10^{-4} msec/cycle or 2.1×10^{3} cycles/msec

4.76×10^{-7} sec/cycle

BAUD (bits/sec)	sec/bit	sec/10 bit char	Max cycles/per bit
300	.0033	33.3×10^{-3}	6.9 k
1200	8.3×10^{-4}	8.3×10^{-3}	1.7 k
2400	4.2×10^{-4}	4.2×10^{-3}	0.88 k
9600	1.04×10^{-4}	1.04×10^{-3}	0.2 2k

Of there are avg 10 cycles per instruct ; @ 1200 baud could do 170 instructions between bits

LIMITED WARRANTY

EXIDY warrants to the registered owner for a period of 90 days from the date of purchase that the computer components described herein shall be free from defects in material and workmanship under normal use and service. This warranty shall be void if the computer enclosure is opened or if the unit is altered or returned as outlined in Customer Service Information. (Refer to A GUIDED TOUR OF PERSONAL COMPUTING, p. 5.) Customer's sole and exclusive remedy in the event of defect is expressly limited to the correction of the defect by adjustment, repair or replacement at EXIDY's election and there shall be no obligation to replace or repair items which by their nature are expendable. No representation or other affirmation of fact, including but not limited to statements regarding capacity, suitability for use, or performance of the equipment, shall be deemed to be a warranty or representation by EXIDY, for any purpose, nor give rise to any liability or obligation of EXIDY what-so-ever.

EXCEPT AS SPECIFICALLY PROVIDED IN THIS AGREEMENT, THERE ARE NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND IN NO EVENT SHALL EXIDY BE LIABLE FOR LOSS OF PROFITS OR BENEFITS, INDIRECT, SPECIAL, CONSEQUENTIAL OR OTHER SIMILAR DAMAGES ARISING OUT OF ANY BREACH OF THIS WARRANTY OR OTHERWISE.

NOTICE

ALL EXIDY SOFTWARE IS DISTRIBUTED ON AN "AS IS" BASIS WITHOUT WARRANTY

Exidy shall have no liability or responsibility to customers or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by computer equipment or software sold or distributed by EXIDY, including but not limited to any interruption of services, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such computer or software.

$11.95

Exidy inc.

Reorder No. DP5003