Subsystem B

User's Manual

Processor Technology

Subsystem B User's Manual

ProcessorTechnology

7100 Johnson Industrial Way
Pleasanton, California 94566

First Printing, August, 1977

Manual No. 730013

SECTION	TITLE	PAGE

TABLES/FIGURES

1.0 PROCESSOR TECHNOLOGY SUBSYSTEM B

The Processor Technology Subsystem B is a set of five S-100 compatible modules which convert most microprocessor mainframe assemblies into computing systems comparable in power to the Sol 20. Subsystem B comprises two memory modules and three interface modules. The interface modules are designated VDM-1, 3P+S, and CUTS. The Subsystem B concept assumes that the user has an S-100 compatible microprocessor chassis with a CPU module, an ASCII keyboard, and a video monitor or modified TV. The system block diagram is shown in Fig. 1.

Subsystem B modules are delivered as kits or as assembled and tested components. Each module has its own manual complete with assembly and testing instructions. A brief description of each module follows:

1.1 3P+S PERIPHERAL INTERFACE MODULE

The 3P+S has two 8-bit parallel I/O ports, with full handshaking logic, plus a serial I/O port with a data rate that can be set anywhere between 35 and 9600 Baud.

One parallel output port can be used to set up control conditions for both parallel and serial ports, as well as for setting the serial I/O baud rate under program control. One parallel input port is available for polling Input Data flags and External Device flags, and for checking the serial I/O error flags. These provisions can implement full handshaking with both input and output peripherals.

Interfacing to the S-100 vectored interrupt bus is provided by a jumper selectable option which allows any of the UART (Universal Asynchronous Receiver Transmitter) error flags or handshaking signals to generate interrupts. (A vectored interrupt capability is required for this process.)

Module addressing is selectable to any of 64 four-address segments within the range of 256 I/O addresses. You may add another dimension of flexibility by using either the UART and control port, or the two parallel ports, to occupy the lower two relative addresses.

The 3P+S will send and receive 1.5 stop bits, required by the old (and less expensive) model teletypes, such as the 15 Model or 28, Teletypes.

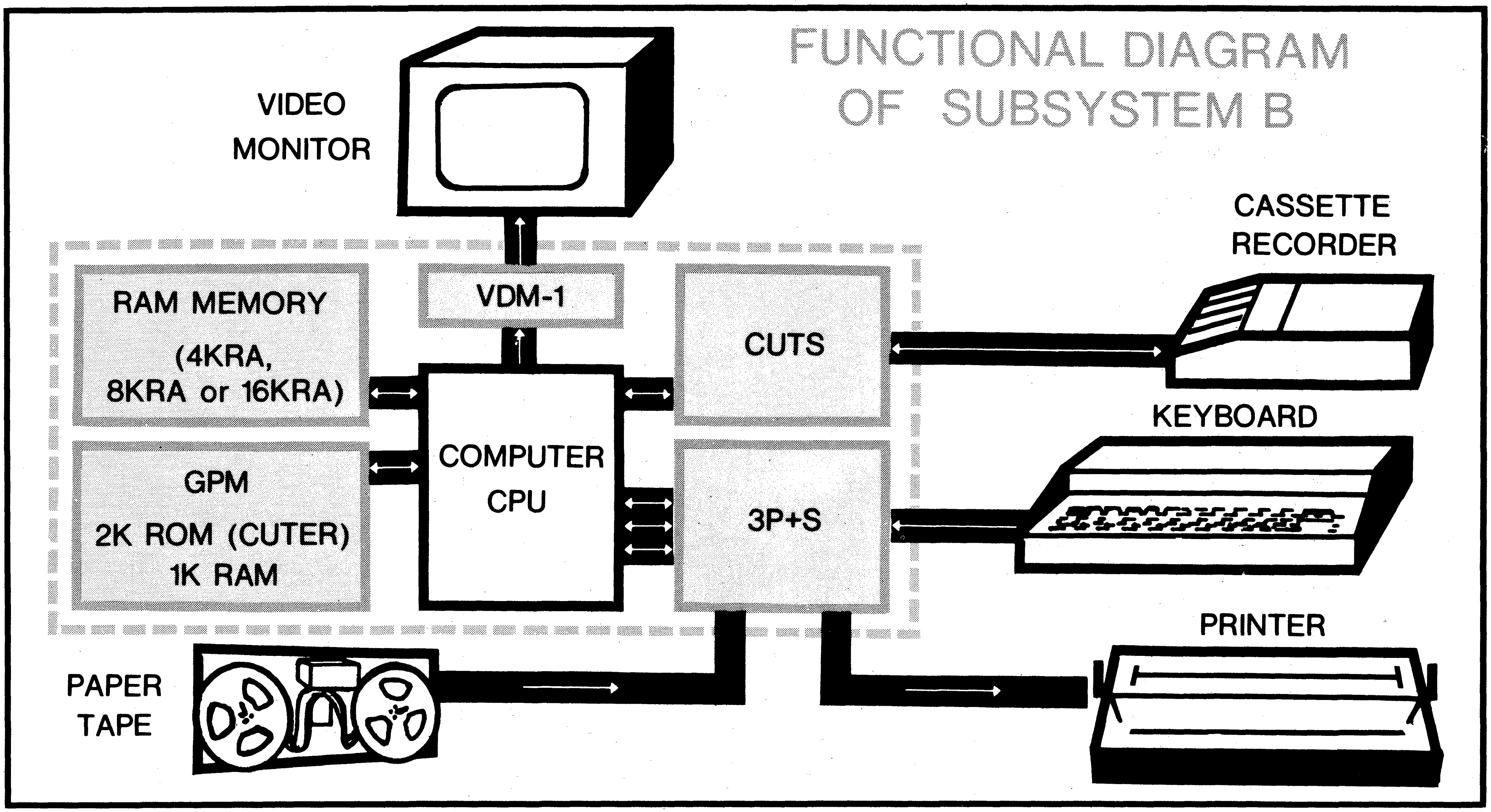

FUNCTIONAL DIAGRAM
OF SUBSYSTEM B
VIDEO
MONITOR
CASSETTE
RECORDER
RAM MEMORY
(4KRA,
8KRA or 16KRA)
VDM-1
CUTS
KEYBOARD
COMPUTER
CPU
GPM
2K ROM (CUTER)
1K RAM
3P+S
PRINTER
PAPER
TAPE

Figure 1.

1.2 VDM-1 VIDEO DISPLAY MODULE

The VDM-1 generates sixteen 64-character lines in a large easy-to-read font with both upper and lower case letters. It contains 1K (1024) bytes of random access memory, to which the processor can read or write, just as though the memory were an integral part of the system. As the information is written in, contents of the on-card memory are displayed instantly without interrupting the operation of the processor.

The VDM-1 can scroll its display upwards or downwards. A built-in timer allows scrolling at about four lines per second eliminating complicated timing program routines. At top speed, the display scrolls through a dump of 65K of memory in two minutes.

Within each text line, the VDM-1 can be set to respond to two control characters: Carriage Return (CR) will cause the rest of the line to be blanked, and Vertical Tab (VT) will cause the remainder of the line and the rest of the screen to be blanked. This is a useful feature when writing over old text still in the memory.

Multiple programmable cursor circuitry is built in. All 1024 cursors can be displayed at one time or begin anywhere in the display. Thus, the VDM-1 can display white-on black or black-on-white. The VDM-1 also features EIA Video output for any standard video monitor, with instructions for the conversion of a TV into a video monitor.

1.3 SUBSYSTEM B MEMORY: 4KRA, 8 KRA, or 16KRA

Three versions of Subsystem B are available; they are identical except for the memory module supplied. The B70 version uses the 4KRA module with a capacity of 4096 eight-bit bytes; the B110 version uses the 8KRA module with a capacity of 8192 bytes; and the B190 uses the 16KRA module with a capacity of 16,384 bytes.

All three memory modules have provisions for simple battery back-up to prevent loss of data during power interruption or loss. Each has starting addresses set by means of DIP switches. The 4KRA has a switch array which can set the starting address for the entire 4096 byte block at any of 16 addresses; the 8192 block in the 8KRA may set at any of 64 starting addresses. The 16,384 byte capacity of the 16KRA is divided into four 4096 byte blocks, each of which may be separately addressed to any of 16 possible starting addresses.

The 4KRA and 8KRA use low power ICs which operate in static mode, and are available as kits or fully assembled and tested. The 16KRA uses RAM ICs which operate in dynamic mode, with "invisible" refresh circuitry, and is available assembled and tested only.

1.4 CUTS: COMPUTER USER'S TAPE SYSTEM

CUTS, the Computer User's Tape System is a high speed, simple to use audio cassette interface. The recording technique is asynchronous Manchester encoding at 1200 or 2400 Hz.

Two separate tape transport control outputs and two common audio inputs and outputs are provided to drive one or two recorders. In addition, CUTS has provision for selecting a high-level audio output signal for driving the auxiliary input to an audio recorder, or a 5-volt peak-to-peak squarewave output for driving a digital recorder.

1.5 SUBSYSTEM B GPM (GENERAL PURPOSE MEMORY)

The preprogramed CUTER ROM of the GPM contains the software which integrates the overall functions of the Subsystem B.

CUTER ties together the internal functions of the CPU as well as a cassette recorder, keyboard, video display and other peripheral eqiupment. It also brings standard commands to the system such as ENTER, DUMP, GET, EXECUTE, and CATALOG as well as custom commands. Furthermore, CUTER makes all Processor Technology and Software Technology programs compatible with your own system. CUTER takes up 2K of memory space. GPM also has 1K of RAM which is used as a scratch pad by CUTER.

Reserved space is available on the GPM board for later addition of a powerful ROM-resident assembler, the ALS-8. To simulate 8080 programs without running them in real time, the SIM-1 interpretive simulator can also be added. With SIM-1 comes the TXT-2 text editor, designed to make file editing quick and easy. Alternately, this reserved space may be used to install 8K of 2708 EPROM or other types of ROM or EPROM.

Figure 2, GPM Memory Map, shows how CUTER, the VDM-1 and various blocks of memory are addressed within the 65K addressing capability of the 8080 microprocessor.

GPM Memory Map

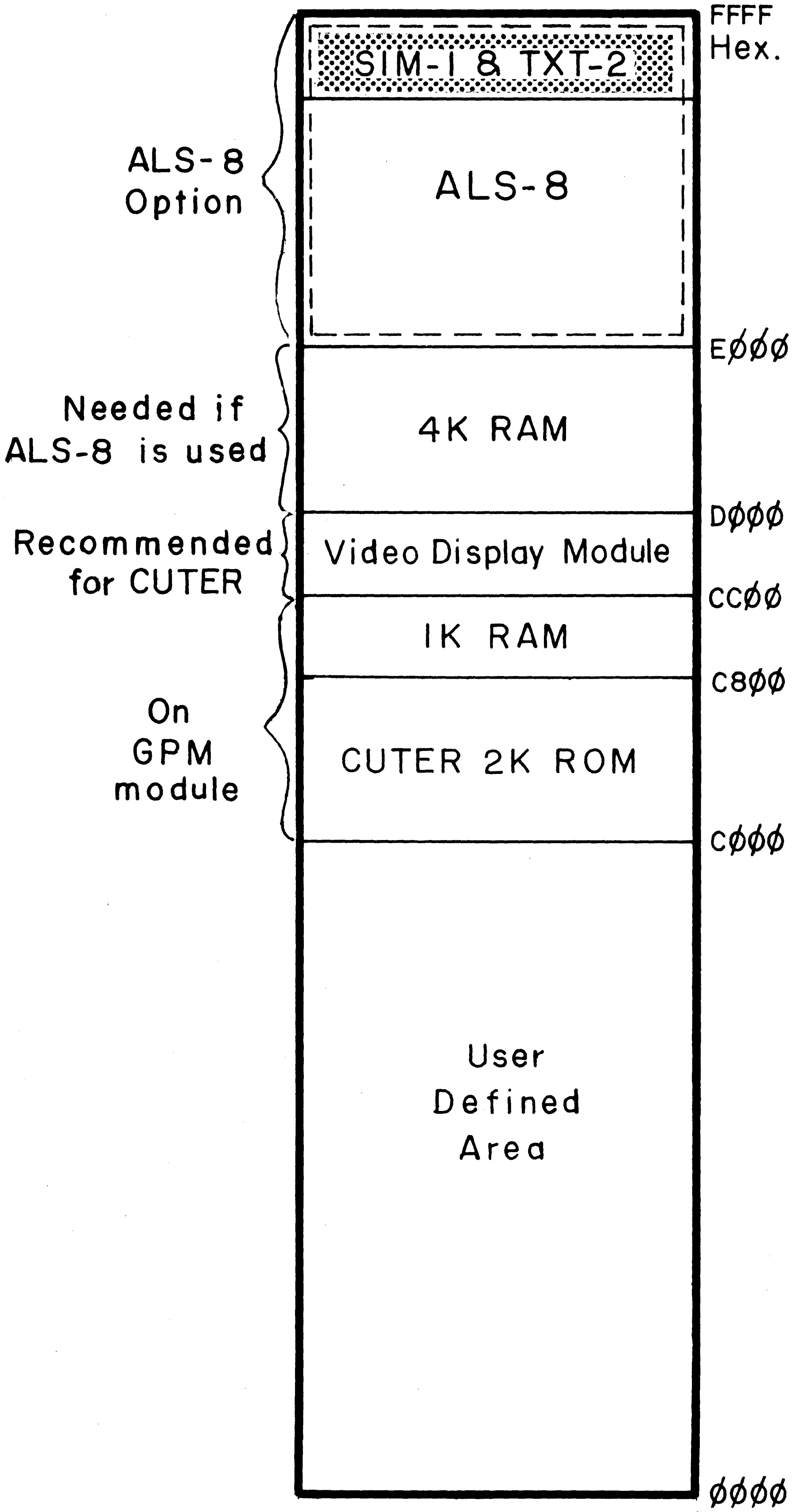


Figure 2.

2.0 SUBSYSTEM B AND OPTIONS

The subsystem B is offered in a variety of configurations. Table 1 shows these options.

MODEL NO.	B70	B110	B190
MODULES	4KRA GPM CUTS VDM-1 3P+s	8KRA GPM CUTS VDM-1 3P+S	16KRA GPM CUTS VDM-1 3P+S
SOFTWARE	BASIC/5 CASSETTE	BASIC/5 CASSETTE	BASIC/5 CASSETTE
MANUALS	CUTER Source Subsystem B 4KRA GPM CUTS VDM-1 3P+S BASIC/5 SOLOS/CUTER	CUTER Source Subsystem B 8KRA GPM CUTS VDM-1 3P+S BASIC/5 SOLOS/CUTER	CUTER Source Subsystem B 16KRA GPM CUTS VDM-1 3P+S BASIC/5 SOLOS/CUTER

Table 1 Subsystem B Options

2.1 UNPACKING AND AUDITING SHIPMENT RECEIVED

The shipping containers used to ship your Subsystem B were made to give more than adequate protection to the contents. (They are useful, in fact, for moving, shipping or storing the system at any time so you may want to keep them.) Accidents can occur in transit beyond the expectations of either shipper or carrier. As you unpack the shipment, inspect the contents for damage. Then refer to Table 1 (Subsystem B Options) for a list of the components and manuals of the various Subsystem B options. If damage has occurred, or your shipment is not complete, report it immediately to Processor Technology.

If your Subsystem B modules are factory assembled, proceed to Section 3. If your Subsystem B is in kit form check the packages containing the parts for each module against the parts list. It may be helpful to lay the components out on each assembly drawing. As you check off each part against the parts list, also check the assembly drawing. Observe how the part is oriented and verify that you have correctly identified it. Most failures of newly assembled boards can be traced to components which are installed in the wrong location, or are incorrectly oriented.

2.2 PROGRESSION FOR ASSEMBLING KITS

Begin assembly of your Subsystem B with the 3P+S Module. It gives you the means to interface with peripherals.

After the 3P+S, build the VDM-1 to give your system display. Third, build the memory board . Fourth, build CUTS. Lastly, build the GPM Module.

> WARNING
>
> Synthetic clothing or bench covers generate static electricity which can damage MOS IC's. Keep ICs away from nylon, orlon, and all other such synthetic fabrics.

The modules are tested as they are assembled. Follow the test procedure in each manual. Some of the modules are tested at various stages of assembly, notably, the VDM-1. The 8KRA Memory cannot be tested until all other modules have been assembled.

You are cautioned to read each manual thoroughly before beginning assembly. Assembly and Test Instructions are found in Section 2 of each manual except for the 16KRA Memory which is supplied assembled only.

2.3 PRINTED CIRCUIT BOARD ORIENTATION AND LEGEND

There are two surfaces to a printed circuit board (PCB). On one side is a printed legend. The legend consists of silk-screened component location information as well as component identification information. Each component is installed over its respective outline. The side on which the legend is printed will be referred to as the component side of the board. The other surface of the PCB is called the solder side. Components are not installed on the solder side of the board. To work on a board, lay it flat on the work bench with the component side up. The legend should be readable just as a page of text, and oriented the same as the assembly drawing.

3.0 INSTALLING SUBSYSTEM B

Installing the Subsystem B entails installing jumpers and setting switches on each board. No attempt will be made to describe all the ways the system can be jumpered to accomplish particular ends. The procedures below give you the means for setting up your system to the standard configuration required by the CUTER program, but do not define option limits. The flexibility of Subsystem B and its capacity for expansion become increasingly obvious as you proceed. Use #24 solid insulated wire to make jumpers.

3.1 JUMPERING 3P+S

Figures 3 and 4 illustrate jumpering for two serial interface options: RS-232-C interface at 300 baud, and the 20-milliampere current loop interface at 110 baud. Both illustrations also include jumpering for an ASCII keyboard as the console device on pseudo-port Ø. With component side of the 3P+S Module turned up and connectors J1 and J2 pointed away, note that the board has alphabetically designated areas to indicate where jumpering is possible.

3.1.1 JUMPERING 3P+S FOR KEYBOARD CONSOLE AND RS232-C SERIAL INTERFACE (Fig 3)

( ) In Area A, A2 through A7 are jumpered to ground.

Areas B, C and D, located to the left of Area A, each have one jumper.

( ) In Area B, the left pad (L) is jumpered to the right pad (R).

( ) In Area C, the center pad (C) is jumpered to the right pad (R).

( ) In Area D, the center pad (C) is jumpered to the right pad (R).

( ) In Area E, install 12 jumpers as illustrated in Fig. 3.

wrong

( ) In Area J, jumper pad 3 to the first row of pads beneath.

Jumper Area G is located to the right of Area E. Five jumpers are installed in Area G.

( ) Pad RIN is jumpered to pad 1 at the top of Area G.

( ) Pad FA is jumpered to pad C1.

( ) Pad FB is jumpered to pad CØ.

( ) Jumper pad RDA to pad C6.

( ) Jumper pad TBE to pad C7.

3.1.2 JUMPERING 3P+S FOR KEYBOARD CONSOLE AND CURRENT LOOP SERIAL INTERFACE (Fig 4)

Jumpering of Areas A, B, C, and D for using a teletype is identical to that just described for the RS-232-C interface.

( ) In Area E, install 12 jumpers as illustrated in Fig. 4.

Area G is jumpered in four places:

( ) Pad RDA to C6, pad TBE to C7, and pad FB to CØ, pad FA to Cl.

There is one remaining jumper to be installed to make the 3P+S ready to interface with a teletype:

( ) The pad of the collector of transistor Q5 is jumpered to the pad beside the emitter of Q4. Q4 and Q5 are located to the right of Area E.

3.2 VDM-1 JUMPERING AND OPTION SELECTION

Figure 5 shows jumpering for the VDM-1 module in the Subsystem B. This illustration is identical with the diagram on p. II-19 of the VDM-1 manual. Be sure you have placed the jumpers as shown.

Video display options are selected by means of the 7-switch DIP array. The switches are set as shown in Fig 6:

( ) 1 on, 2 off, 3 on, 4 off, 5 on and 6 on, (7 is a spare).

This setting provides control characters, black letters on a white field, a non-blinking cursor, and control characters are displayable. Text blanking from CR to end of line and VT to end of screen is disabled. Refer to p. III-9 and 10 of the VDM-1 Manual for more detail.

3.3 CUTS JUMPERING AND OPTION SELECTION

Figure 7 shows the CUTS jumper connection which is used in the Subsystem B configuration. With the CUTS board oriented component side up and 100-pin connector edge toward you, the jumper field is located to the right. The pads designated A,B,C, and D comprise the jumper field. Augat pins are installed on these pads to simplify option selection.

( ) Jumper A to B. This selects an audio output level compatible with a tape recorder's auxiliary input.

The CUTS module utilizes an 8-switch DIP array by which to select port addresses (Ø through FA). Figure 8 shows how these switches are set for Subsystem B:

( ) 1 on, 2 off, and 3 through 8 all on.

3.4 GPM SWITCH SELECTION

In the upper right corner of the GPM board is an 8-switch DIP array which configures the module for various operation options. Figure 9 illustrates the settings for the switches. Table 2 describes each switch setting and its purpose. These settings are:

( ) 1 on, 2 off, 3 off, 4 off, 5 off, 6 on, 7 off, and 8 on.

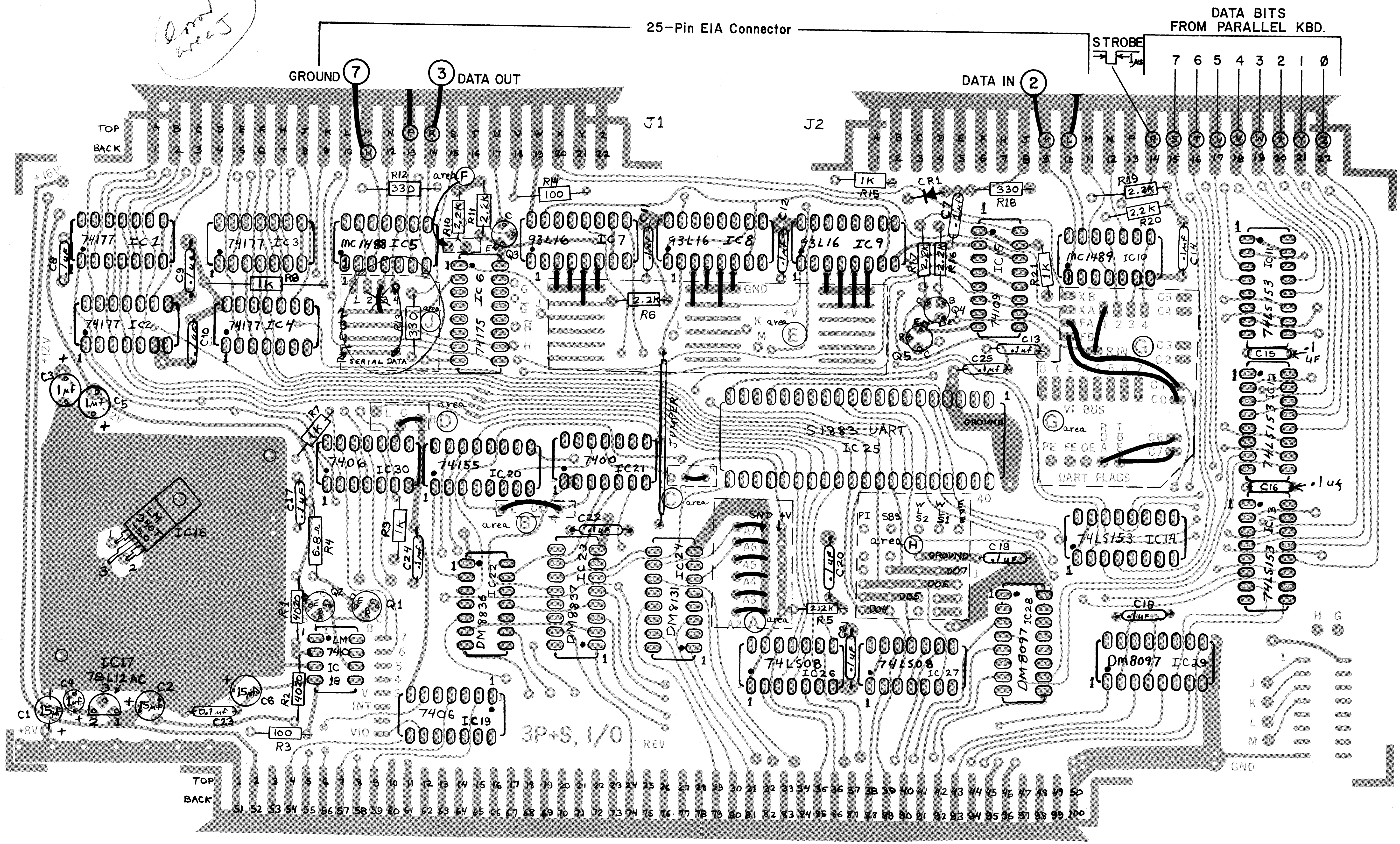


Figure 3. Jumpering 3P+S for Keyboard Console and RS232-C Serial Interface

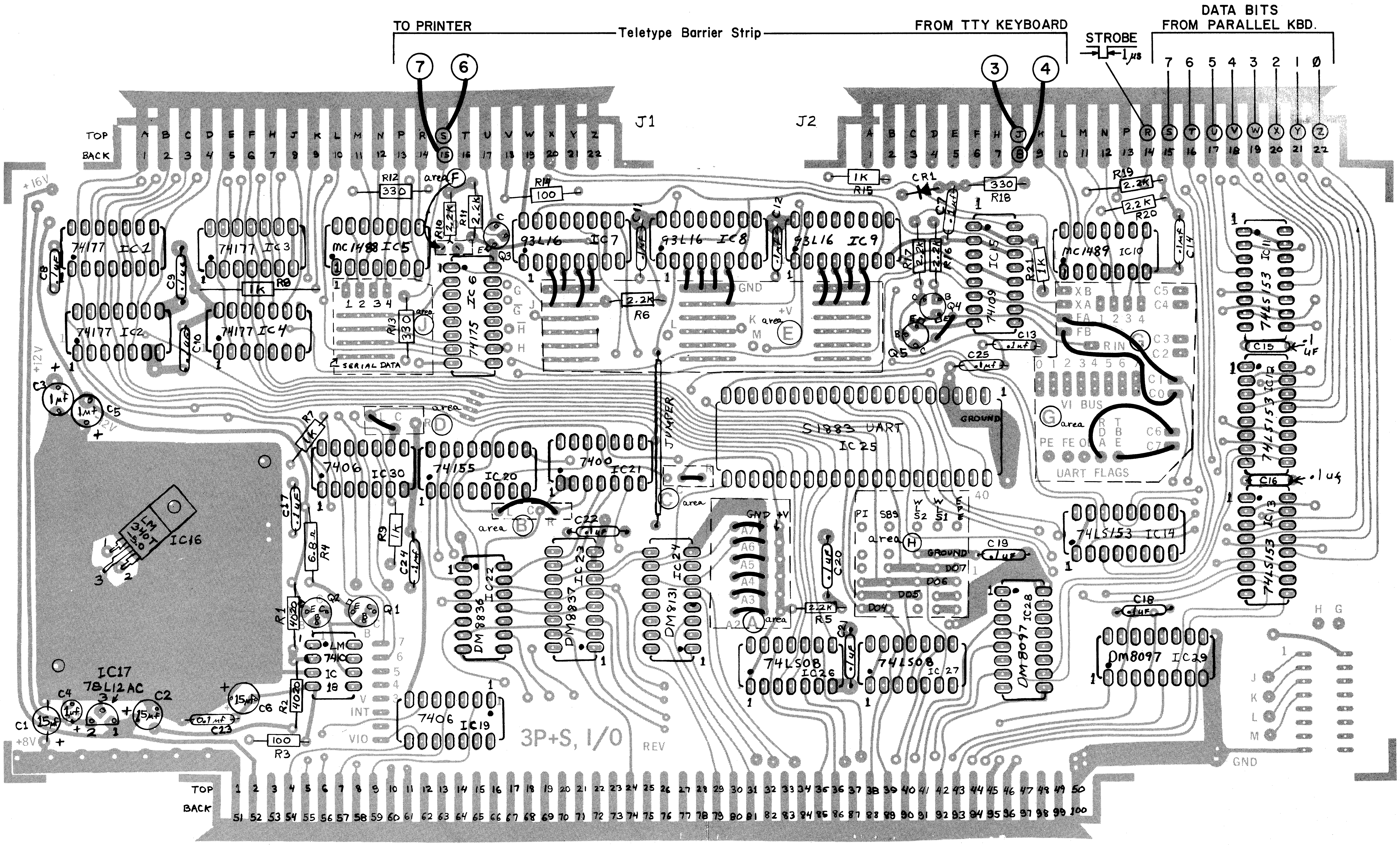


Figure 4. Jumpering 3P+S for Keyboard Console and Current Loop Serial Interface

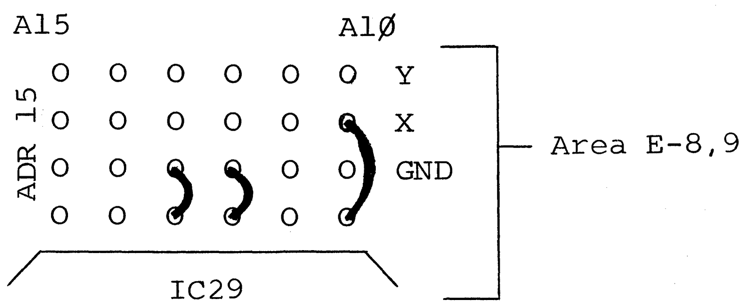


Figure 5. VDM-1 Jumpers

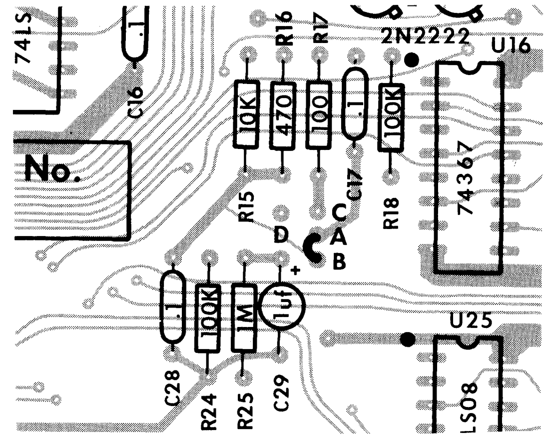


Figure 7. CUTS Jumper

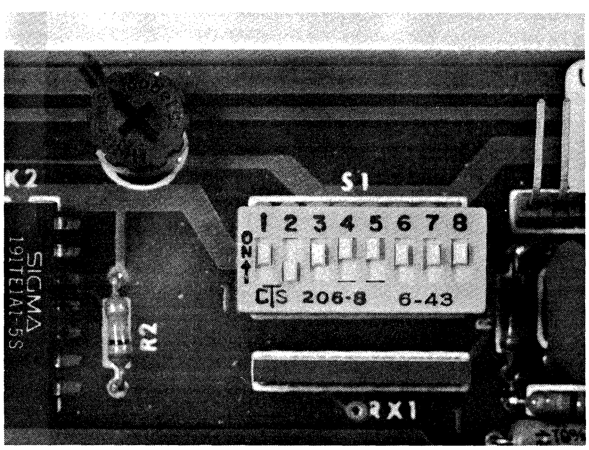


Figure 8. CUTS Switches

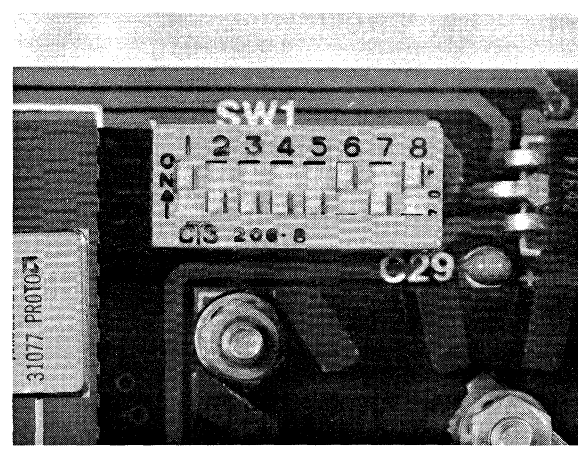


Figure 9. GPM Switches

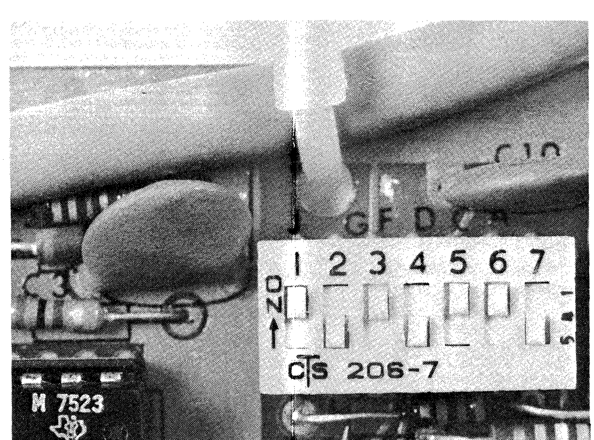


Figure 6 VDM-1 Switches

SWITCH	SETTING	PURPOSE
SW-1 * SW-3	ON OFF	Auto-start at CØØØ. (CUTER command mode.)
OR		
SW-1 SW-3	OFF ON	Auto-start at EØØØ. (First Address in PROM, e.g. ALS-8)
SW1-2	OFF	Allows memory to run at full speed CØ wait states
SW1-4	OFF	To interface GPM with non-Sol system
SW1-5	OFF	Because this description assumes IMSAI system
SW1-6	ON	Selects GPM autostart
SW1-7	OFF	To interface GPM with non-Sol interface
SW1-8	ON	Connects phantom signals from autostart to S-100 bus

Table 2 GPM DIP Switch Settings

Section 3 of the GPM manuals explains in greater detail each option selection provided by the module.

3.5 INTERCONNECT CABLES AND DIAGRAMS

Three of the modules require interconnect cabling. See Figure 4-1 in the CUTS manual for cabling to cassette recorders, and Figure 4-1 in the 3P+S manual for cables to J1 and J2. The cable from VDM-1 to the monitor depends on the type of monitor used.

* Two alternative settings for SW-1 and SW-3 are shown depending on desired auto-start address.

4.0 SUBSYSTEM B OPERATING PROCEDURES

Information in this section will help you to become familiar with the operation of your Subsystem B. Following brief explanations of the operating controls and basic operating modes, you can put your system through some simple operations. This should sufficiently acquaint you with your system so that you can use the system effectively.

If you are an experienced personal computer user you may wish to go directly to the SOLOS/CUTER manual for an explanation of the CUTER Operating System. The remainder of this section will give the first time user detailed information on the operation and use of CUTER.

With the CUTER ROM in the GPM, the computer is in the command mode when its program is entered at address CØØØ. Command mode is a sort of "home base" from which excursions may be made into other programs. An analysis of three levels of programs will make the concept of command mode more understandable.

At the first level of software are the instructions which the 8080 CPU (central processing unit), the brains of the computer, can understand and run. All programs must ultimately be reduced to this basic level to be operated on by the computer. Software written at this level is often referred to as "object code" or "machine language", since the "machine" or 8080 CPU understands it directly. The CUTER program contained on the GPM module is stored in this machine language form, and the computer can therefore run directly from this program. Since the CUTER program is contained in permanent ROM which is plugged into the GPM, the CUTER program is always available. There is provision for returning at all times to the command mode of CUTER. From the command mode other programs may be brought in for various operations or stored on cassette tape. The contents of the computer's memory may be displayed or changed. The command mode also performs "housekeeping" functions such as setting the rate at which data is read from tape, or the rate at which characters are displayed on the video monitor.

The command mode allows the utilization of the second level of software. This level includes higher-level language programs such as BASIC/5 or FOCAL in which complex application programs may be more easily written. These are called higher level languages because they permit the user to write programs in a form much closer to English. However, programs written in these languages must be translated into the more basic machine language before they can be run. Besides higher level languages, this second level of software includes programs such as the TREK 80 and GAMEPAC video games and the ALS-8 program (a software package used for developing programs), all of which are offered by Processor Technology Corporation. Through the facilities of the command mode, these second level programs are transferred (loaded) into memory from cassette tape or other storage media, and then "executed" (used). Some of these programs are also delivered in ROM or EPROM (erasable programmable ALS-8 memory which, like the CUTER program, is plugged into the GPM to make them instantly available. All first and second level programs are stored in the computer as binary object code.

Let us illustrate the concept of the second level of programs with an example, BASIC/5. Using the "XEQ" command available in the command mode, we load the BASIC/5 program into the computer's memory module from cassette tape through the CUTS module. With this command BASIC/5 is ready for use as soon as the tape has stopped moving. The control of the computer is now taken over by the BASIC/5 program just loaded in memory, and computer is no longer in the command mode. All the features of BASIC/5 language are now available to us, with a new set of commands and rules. Since the CPU of the computer only understands the machine language of the first level of software, the BASIC/5 program must translate the commands and data we enter to this lower level. BASIC/5 does this as we go. While we are using BASIC/5, we still have access to some of the commands and features of CUTER, although they have a modified form while we are in BASIC/5.

The third level of software consists of programs written using the higher languages of the second level programs. A program written in BASIC/5 is on this third level. This program only makes sense to the computer when the computer has BASIC/5 in memory and control has been transferred to the BASIC/5 program. Third level programs written in any high level language are often called "applications programs" since they are usually written in order to fit a specific application need.

4.1 BRINGING UP THE SYSTEM

If your computer has sense switches, set them as shown in Table 3. If not, see Section 2.6.4 of the GPM for instructions.

If you are starting your system for the first time, or if your system has been idle for an inordinate period, check the cabling to ascertain that all connections are correct and secured.

To bring up the system in CUTER command mode:

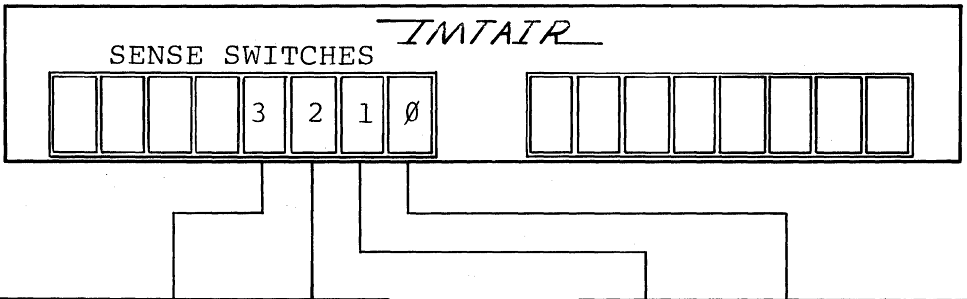
(1) Place the power switch ON.

(2) Push the RESET switch.

(3) Select the EXAMINE switch. Examine address CØØØ Hex.

(4) Refer to Table 3 for the desired sense switches setting.

(5) Run. A prompt and a rectangle >█ should appear in the upper left corner of the screen indicating that CUTER is in the command mode, ready to use. If it does not appear, adjust the VERTICAL and HORIZONTAL controls on the VDM-1 Module in order to move the cursor into the visible portion of the screen.)



Console Input Port Selection	SW 3	SW 2
KEYBOARD (PORT 3)	OFF	OFF
SERIAL INTERFACE (PORT 1)	OFF	ON
AUX. PARALLEL INTERFACE (PORT 2)	ON	OFF
CUSTOM INPUT	ON	ON

SW 1	SW Ø	Console Output Port Selection
OFF	OFF	VIDEO
OFF	ON	SERIAL
ON	OFF	PARALLEL
ON	ON	CUSTOM

Table 3 Sense-switch settings for CUTER Start-up. Most users will place the switches in the down or OFF position.

4.2 SUMMARY OF CUTER COMMANDS

The SOLOS/CUTER User's Manual supplied with Subsystem B contains a detailed guide to the commands and capabilities of CUTER. The discussion below is intended to provide the beginner with an introduction to CUTER in the context of Subsystem B.

CUTER is operated by means of four categories of commands: console, tape interface, set (tape and I/O), and special. They are given in Table 4.

CONSOLE COMMANDS
1. DUMP
2. ENTR
3. EXEC

TAPE CASSETTE INTERFACE COMMANDS
1. CAT
2. GET
3. XEQ
4. SAVE

SET COMMANDS	
TAPE	I/O
1. Set XEQ	1. Set I
2. Set TAPE	2. Set O
3. Set TYPE	3. Set N
4. Set CRC	4. Set CO
	5. Set CI
	6. Set S

SPECIAL COMMANDS
1. TERM
2. CUST

Table 4 Summary of CUTER Commands

4.3 CONSOLE COMMANDS

The CUTER operating system console commands allow you to accomplish many of the functions of a front panel using a keyboard and the video display instead of front panel switches and indicator lights. System control from a keyboard is much less error-prone and more convenient than using front panel switches. In the jargon of data processing, the keyboard-display combination which controls the system is referred to as the CONSOLE. The CUTER commands which perform front panel operations are referred to as the CONSOLE commands.

Our first interest is how we command CUTER to do as we wish.

CUTER is not ready to accept commands at all times. For instance, whenever CUTER is cataloging a cassette tape it ignores whatever we type on the keyboard with one exception. The exception is the key combination of CONTROL/SHIFT/@. Pressing all three of these keys simultaneously places the hexidecimal code ØØ4 on the keyboard data lines. When CUTER inputs this code, it terminates the operation in progress and places a prompt symbol on the screen, followed by the cursor: ▮

In this context, the > symbol is referred to as a prompt. The purpose of the prompt is to tell us that CUTER is now ready to accept a command from the keyboard. To the immediate right of the prompt is a rectangular block referred to as the cursor. The cursor indicates the position on the screen where the next displayable character will appear. If the key combination of CONTROL/SHIFT/@ does not result in a prompt with a cursor >█ it is then necessary to stop the computer and restart it at address CØØØ Hex, the starting address of the CUTER program. If you are not sure of the startup procedure, refer to Section 4.1, or consult the manual supplied with your computer for instructions pertaining to starting a program.

Now that we have a prompt indicating that CUTER is in the command mode, we may type in a command. Alphabetic characters such as A,B,C must always be entered in upper case. Let's say that we want to examine the contents of memory location ØØØØ. Typing in the command: DUMP ØØØØ will accomplish this. Spaces are used to separate a command from its argument.

What is an argument?

An argument is a numeric value which a command uses in some way to accomplish its task. The command DUMP without the argument ØØØØ would be ambiguous: Dump what?

If we make a mistake typing a line we can use the delete or rubout key to back up and erase the character. CUTER responds to the Hex code 7F with a rubout operation. If we want to backup without erasing characters, the key combination CONTROL/A (hex code Ø1) will move the cursor left until it reaches the left margin. If the cursor is at the left margin when the CONTROL/A key combination is struck, the cursor will "wrap around" to the right margin. We refer to this as a "wrap mode" of cursor control. CUTER responds to the hex code Ø1 with the cursor left operation. Once we have a command line which is correct, the RETURN key (hex code ØD) is used to indicate that the command should be executed, or processed, by CUTER. The RETURN key also has a function similar to that on a typewriter, i.e., it it moves the cursor to the beginning of a new line like the typewriter moves its carriage. When describing actual commands, RETURN is abbreviated CR (Carriage Return) and written in parentheses.

Now that we understand how commands are entered we can study in more detail each of the commands.

4.3.1 DUMP COMMAND

The DUMP command has been mentioned briefly. It is used to display the contents of memory. The command line consists of the command name DUMP, or just DU, followed by one or two arguments, with spaces separating the command name and the arguments. The first argument, following the command name, indicates to CUTER the memory address where the dump is to begin; the second argument is the last memory address to be dumped. If only one argument follows the command name DUMP, this single argument is considered both the starting and ending address. Here are several examples of the DUMP command:

Command: DU ØØØØ (CR)	ØØØØ is the starting and ending address. Just one memory location will be dumped.
Command: DU ØØØØ ØØØF (CR)	ØØØØ is the beginning address; ØØØF is the ending address. Sixteen memory locations will be dumped. Location ØØØØ through ØØØF.

Suppose we typed DUMP ØØØ3 ØØØ2. Would CUTER dump from location ØØØ3 all the way to location FFFF and then "wrap around" through Ø and on to ØØØ2 the last location? No. When a dump address is reached which is greater than the second argument, the dump operation ends and CUTER returns to the command mode. Thus, when we type the command line DUMP ØØØ3 ØØØ2, CUTER will dump address ØØØ3 then proceed to address ØØØ4; discovering that address ØØØ4 is greater than the ending address of ØØØ2, CUTER will return to the command mode.

Just for fun try the following command: DU ØØØØ WXYZ then hit the RETURN key. Note that the W has been replaced by a question mark and a new prompt has been placed on the screen; additionally; no DUMP operation was performed.

When CUTER encounters data in a command line it does not understand, it replaces the first character of that data with a question mark. The question mark (?) indicates that CUTER did not understand the argument WXYZ. This occurs because CUTER expects the argument to be a hexadecimal number.

4.3.2 ENTER COMMAND

The ENTER command is the equivalent of the front panel Deposit function. It is used to enter data into memory from the console. The ENTER command has one argument which tells CUTER where to begin loading data into memory. The command ENTR 1ØØØ directs CUTER to load data into memory from the console beginning at location 1ØØØ Hex. When the RETURN key is pushed after an ENTER command has been typed, CUTER will place a colon (:) on the next line. The colon indicates that CUTER is ready to accept data from the console keyboard. Had CUTER encountered something in the command line it did not understand, a prompt would have appeared on the next line instead of a colon. Data to be entered into memory consists of hexadecimal numbers in the range Ø to FF. For example, if we want to enter the value 5 into location 1ØØØ we would type Ø5, or simply 5, followed by a solidus or slash (/) and then a carriage return. The slash indicates that no more data follows. The carriage return tells CUTER to process the line of data and make the entries into memory. If we enter more than one data value separated from other data values by a space, each value will be entered into successive memory locations. If we type ENTR 1ØØØ and CUTER responds with a colon, we then type: 1 2 3 4 / (CR). CUTER will place a 1 in location 1ØØØ, a 2 location 1ØØ1, a 3 in location 1ØØ2, and a 4 in location 1ØØ3. We can enter more than a single line of values into memory by typing the first line of values, then typing a CR. CUTER will enter the data values on the first line into successive locations in memory until all the values on the line have been entered; then CUTER will place a colon on the next line. We now type another line of data values, and a CR. This second line is entered at addresses starting where the previous line left off. CUTER will continue to provide a new line with a colon until it encounters a line which contains the slash (/). CUTER then returns to the command mode when it has finished centering the values preceeding the slash. Try dumping for confirmation the entries just made. If no memory exists at the locations specified in the entry, the DUMP will show FF in such empty locations rather than the entered data.

4.3.3 EXECUTE COMMAND

The last console command which we must learn is the EXECUTE command. This command is the equivalent of the EXAMINE-and-RUN front-panel function. The EXECUTE command is used to run a program. If we want to run a program that begins at location ØØØØ, BASIC/5 for instance, we would type EXEC ØØØØ (CR).

4.4 CASSETTE INTERFACE COMMANDS

The Cassette Interface Commands allow us to control the operation of the CUTS cassette interface from the Subsystem B console using four commands. We can save programs, or data on tape, and then get them back into memory from tape. If we wish, we can save a program and the address of the first instruction of the program on tape. Then using the XEQ command we can get the program and transfer control to the first instruction in one operation. In addition, we can use the CAT command to list the programs and data on a cassette.

When a program or data is saved on tape, additional information is also recorded on the tape just ahead of the program, or data. This additional information, called the header, is used by CUTER when the program is loaded. It begins with the name used by CUTER to locate the program, or data, on the tape.

Following the name is a byte which indicates the type of data we are saving. Next, the header contains two bytes which are a count of the number of bytes in the data block which follows the header. Then there are two bytes which inform CUTER where to load the data into memory. Finally, there are two bytes which are used by the XEQ command. These bytes indicate to CUTER the location to which control should be transferred after the program has been loaded.

4.4.1 SAVE COMMANDS

To save a program, or a block of data, on tape, we use the SAVE command. The SAVE command is followed by three arguments which are separated by spaces. The first argument is the name we assign to the program. We can pick any name as long as it does not contain any spaces, and is between one and five characters long. When finally we type the (CR), CUTER will place this name into the header to identify this block of information. The second argrument is the address of the beginning byte of the block of information we want to save on tape. The third argument is the address of the last byte of information we want to save. These last two arguments are used by CUTER in the same way as the arguments for the DUMP command. Let's say we wanted to save a program we have written so that we can load it back into memory later. The first instruction of the program is at address Ø5ØØ Hex, and the end of the program is at address Ø7FF Hex. Furthermore, we want to name the program "PROG5". Here is the command line we would type:

SAVE PROG5 5ØØ 7FF (CR)

When we type the (CR) CUTER removes the cursor from the screen and automatically starts the tape recorder. In preparation for saving the program on tape we must use the controls of the tape recorder to position a blank section of tape at the tape head and also press the RECORD button so data will be recorded on tape. When the tape begins to move, CUTER will delay for a second or two, then it will write the header, and the block of memory containing our program. The last byte written is called the CRC Byte. The CRC Byte is used to check the block of data when it is read. The CRC Byte is calculated by CUTER in such a way that each block of data saved on tape has a unique CRC Byte. When the block is reloaded into memory, a CRC Byte is calculated from the data which is read from the tape. If this CRC Byte matches the one on the end of the block of data, CUTER knows that there were no errors when the tape was read. If the two CRC Bytes do not match, CUTER will indicate on the console that an error has occurred. After the CRC Byte has been written, CUTER will return to the command mode.

4.4.2 GET COMMANDS

In order to get the program we have just saved on tape back into memory, we now use the GET command. The tape should be rewound a point before the beginning of the program we have just saved, or in general, rewound all the way to the beginning of the tape. We must remember also to release the RECORD control so the recorder will no longer be recording on the tape, and to set the recorder to the PLAYBACK mode.

To use the GET command to load our program back into memory we type: GET PROG/5 (CR).

CUTER will start the tape in motion and read each header on the tape until it finds one with the name "PROG5". Then it will load our program into memory starting at the address indicated by the appropriate bytes in the header. If we want to load the program into memory at an address different from the address in the header, we could type GET PROG5 1ØØØ. This command line will load the program into memory starting at address 1ØØØ instead of address 5ØØ as indicated by the header data. If we type GET only with no argument, CUTER will get the next program in sequence on tape regardless of name. Once CUTER has loaded the program, it will respond on the console, following the GET command:

GET PROG5 PROG5 Ø5ØØ Ø2FF (CR)

The response repeats the name of the program to show that it got the information requested, and shows the starting address and program length rather than the last address. (The SAVE command's third argument is the last address.) If the CRC byte read from the program does not match the CRC byte calculated by CUTER, the tape motion stops and this response appears:

```
GET PROG5
ERROR PROG5 Ø5ØØ Ø2FF
```

When an error message has occurred, rewind and try again. If there is still an error, see the information in the CUTS manual relating to cassette recorders. Sometimes an error occurs at the beginning of the cassette where the leader joins the tape. Type the GET command again to resume playing back the tape.

4.4.3 XEQ COMMAND

The XEQ command is identical to the GET command except for one important difference. The XEQ command will "GET" our program into memory, and then use information in the header to pass control to the first instruction of our program. Referring back to the description of the SAVE command, you will notice that none of its arguments specified the starting address to be used by the XEQ command. This starting address is specified by a special command, SET XEQ, discussed in Section 4.5.1.

4.4.4 CATALOG COMMAND

CATALOG is the final tape command we will consider. The purpose of the CATALOG command is to list on the console device the information in each of the headers on a cassette tape. This is very useful should we forget the name of a program we have "SAVED"; it is useful, too, if we want a listing of all the programs on a certain cassette. Further, since the CATALOG command starts the motor on the cassette recorder, it can be used to turn on the motor for a REWIND or FAST-FORWARD operation. Since CUTER has no way of knowing when it has read the last header on the tape or even when the end of the tape has been reached, CUTER will not automatically return to the command mode from a CATALOG operation. To return to the command mode from a CATALOG operation we must type CONTROL/SHIFT/ @ simultaneously, or restart the CUTER program at address CØØ3 hex.

4.5 SET COMMANDS

The CUTER SET commands are used to set up conditions within CUTER which will remain in effect until another set command changes these conditions, or until CUTER is restarted from address CØØØ. Restarting CUTER at CØØ3 will not change the conditions created by previous SET commands.

An important difference between other commands and the SET commands is that an equals sign (=) is used to separate the set modifier from its argument. For example, to set the speed at which the display driver moves characters onto the screen we would type:

SET S=5 (CR)

The S which follows the SET command is a modifier which informs CUTER which one of the conditions within CUTER to modify. The 5 is the argument which specifies the actual display rate. The argument for the SET S command may be any hexadecimal value between Ø and FF. The value Ø is the fastest speed, FF the slowest.

4.5.1 TAPE CASSETTE SET COMMANDS

Four set commands are used to control conditions within CUTER which affect the cassette interface commands.

The first of these commands is the SET XEQ. The SET XEQ command is used to set the address of the first instruction of a program into a special location within CUTER to be used when the header of a block is written on tape. That address is used by the XEQ command when the program is reloaded. If we wanted to load our program named PROG5 into memory and transfer control to it using the XEQ command, we must follow this procedure when we SAVE PROG5: First, we would use the SET XEQ command to put the address of the first instruction of PROG5 into the memory location reserved for it within CUTER. If the first instruction address of PROG5 is at 5ØØ hex we would type:

SET XEQ=5ØØ (CR)

Then we would type the SAVE command for our program SAVE PROG5 5ØØ 7FF (CR). When the SAVE command writes the header for PROG5, it will automatically include the XEQ address we entered using the SET XEQ command. Remember that the XEQ address we have placed in CUTER will remain there until we type another SET XEQ command, or restart at address CØØØ.

A unique feature of the CUTS cassette interface is that it can read and write data on tape using in the Kansas City/Byte format as well as the CUTS format. The Kansas City/Byte format reads and writes at a rate of about 30 bytes per second. It is useful for exchanging programs among friends because it is compatible with audio cassette interfaces from most manufacturers. The main limitation of the Kansas City/ Byte format is its slow speed. The CUTS format reads and writes at about 120 characters/second! When the CUTER program is started, it sets the Cassette Interface Module for the CUTS format. The SET TAPE command can be used to change back and forth between the low- and high-speed format. To change formats we type:

```
SET TAPE = 1 (CR) for low speed (Kansas City byte)
SET TAPE = Ø (CR) for high speed (CUTS)
```

If we try to read a tape with the format incorrectly set, we will, of course, be unable to read the tape correctly, and CUTER will indicate an error. Beware!

The SET TYPE command is used to store within CUTER an optional type byte which is placed in the header when a block of memory is SAVED. The type byte appears on the console output following the name when we get a program, or data, from tape. The argument of the SET TYPE command is somewhat unusual. It is a hexadecimal value from Ø to FF which is the ASCII code for the type character. For example, should we want the type character to be "P", the argument of the SET type command would be 50 hex. "A" would correspond to 41 hex etc. The ASCII Hex code table is provided in Appendix II, under 8080 operating codes, of most component manuals. If we want to save our program as type A we would type:

```
SET TYPE = 41 (CR)
```

Then we type the save command:

```
SAVE PROG5 5ØØ 7FF (CR)
```

The SET CRC command was included in CUTER because early versions of Processor Technology Program Cassettes used a different routine to calculate the CRC byte. These tapes cannot be loaded by CUTER unless the CRC error routine is disabled, by typing:

```
SET CRC = FF (CR)
```

To reenable the CRC error routine type:

```
SET CRC = Ø (CR)
```

When CUTER starts up, the CRC error routine is automatically enabled and under normal conditions should not be disabled. SET CRC to FF only if you are sure you are loading an old version Processor Technology Cassette Program. As soon as the program is loaded, SET CRC to Ø again.

4.5.2 I/O SET COMMANDS

The SET I command is used to control the port used when CUTER is inputting commands or data. The argument of the SET I command must be a number from Ø to 3. The number entered selects one of the four pseudo-ports controlled by CUTER. These I/O ports are called pseudo-ports because their port numbers do not correspond exactly to the actual 8080 I/O port numbers through which the input or output takes place. The pseudo-port numbers for input and the actual port selected are as follows:

- Ø Keyboard input (8080 Port 3)
- 1 Serial Input (8080 Port 1)
- 2 Parallel Input (8080 Port 2)
- 3 User-defined Input

In reality, the pseudo-port number directs CUTER to one of three routines within CUTER which actually communicates with the I/O port. If pseudo-port 3 is selected, CUTER is directed to a routine we have written which is loaded somewhere in memory. As an example, if we type the command:

```
SET I = 1 (CR)
```

CUTER will then expect to receive its input from the serial port, ignoring further commands typed on the keyboard.

We would then have to type:

```
SET I = Ø (CR)
```

on the keyboard of the device connected to the serial port to return control to the Pseudo-port Ø Keyboard routine where we started.

The SET O command is used to select the output routine CUTER uses when not in the command mode. That is, no matter what port is selected by the set out command, CUTER will always output to the console when CUTER is in the command mode. For instance, to DUMP the contents of memory to a device connected to the serial port we would type:

```
SET O = 1 (CR)
```

Then we would type the DUMP command followed by its arguments. The DUMP command and its arguments will appear on the console output; the actual dump will then take place through the device connected to the serial port. When the DUMP is finished, CUTER will send a prompt to the console device to indicate it is awaiting further commands. The pseudo-port arguments for the SET O command are:

- Ø = Video output
- 1 = Serial output (8080 port 1)
- 2 = Parallel output (8080 port 2)
- 3 = User defined output

Since we are able to direct CUTER to input or output through a routine we have written by selecting pseudo-port 3, we must also be able to direct CUTER to the first instructions of this routine. The SET CI command is used to direct CUTER to our own input routine. If we have written an input routine whose first instructions is at address 2ØØØ hex, we would type:

SET CI = 2ØØØ (CR)

Now when we select pseudo-port 3 for output by typing:

SET O = 3 (CR)

CUTER will transfer control to our routine at 2ØØØ hex whenever it needs to input a byte of data. We must write our input routine in such a way that when it has the byte of data which CUTER requested, it will then return to control to CUTER. Refer to the keyboard input routine in the CUTER listing for an example. The same procedure that applies to the SET CI command applies also to the SET CO command. The SET CO command is used to direct CUTER to the first instruction of an output routine which we have written. If the routine whose first instruction is at 3ØØØ hex is an output routine we would type:

SET CO = 3ØØØ (CR)

Then CUTER will transfer control to our routine whenever it has a character to output which should not go to the console driver.

4.5.3 SET NULL COMMAND

A characteristic of most common output devices which can be interfaced to the serial output of Subsystem B is that when these devices perform a carriage-return operation the computer must wait until the print mechanism of the device has actually moved to the left margin. Usually the computer sends one or more null characters which have no effect on the serial output device, but do require time to be transmitted to the device. As an example, Model 33 teletype (TTY) requires two null characters after each carriage return.

The SET N command is used to select the number of null characters sent to a serial output device after a CR. If we connect a Model 33 TTY to the serial interface of Subsystem B, it would then be necessary to use the SET N command to indicate to CUTER how many nulls to send after each CR. We would type:

SET N = 2 (CR)

The argument of the SET N command can be any hexidecimal value from Ø to FF.

4.6 SPECIAL COMMANDS

There are two CUTER commands which do not fit into any of the categories we have discussed. The first of these is the TERM command. When CUTER receives this command, it configures the Subsystem B to perform the functions of a serially interfaced, video terminal. The TERM command allows owners of Subsystem B to use their system for time-sharing applications. If we type:

TERM (CR)

CUTER will then move the data it receives from the serial interface to the video display, and send data typed on the console keyboard out the serial interface.

The last CUTER command, CU, allows us to manipulate a table of up to six custom commands. Each custom command table entry consists of a command name of two to five characters and the address of the first instruction of the routine which processes the command. If we type a custom command name following the command mode prompt, then type a CR, CUTER searches the custom command table for the same command name. When CUTER finds a custom command name that matches the one we have typed, it transfers control to the address associated with the command name in the custom command table. If CUTER cannot find the name we have typed, it replaces the first character of the name with a question mark (?).

Entering a new command into the custom command table is simple, for example:

CU NEW 1ØØØ (CR)

will enter the command name NE into the custom command table along with the address 1ØØØ hex. Note that only the first two characters of the command name are entered in the table.

Deleting a command name from the custom command table is a bit more tricky. When a name is deleted, it is replaced by an entry which indicates to CUTER that no more table entries exist. For example, if we enter five custom commands into the table then delete the second custom command, the third, fourth and fifth commands will also be deleted.

To delete a custom command we type the command followed by the name of the command we want to delete. If we type:

CU NEW (CR)

the custom command we previously entered will be deleted. From this discussion it is apparent that custom commands should be created and deleted judiciously.

This ends our discussion of the CUTER commands. You should now be quite proficient in the use of your Subsystem B. Continued experimentation and use will reveal features of CUTER which have not been described in this section. Some of these features such as the third address argument for the save command are mentioned in the SOLOS/CUTER manual. Other features will only be revealed to those who have become familiar with assembly language listings and spend some time studying the CUTER listing. There are innumerable possibilities using custom commands to access the CUTER routines in new and creative ways.

APPENDIX: SUBSYSTEM B COMPONENT SPECIFICATIONS

1. 3P+S SPECIFICATIONS

OUTPUTS: Two 8-bit parallel ports, standard TTL levels, relative addresses at 0 & 1, or 2 & 3. One Teletype 20mA current-loop output. Four EIA RS-232C outputs for serial transmit data and/or control signals. One peripheral interface control driver 50mA current source for paper-tape-reader control or cassette recorder control. Jumpers select the control port output bit.

INPUTS: Two 8-bit parallel ports, standard TTL voltage levels, input current is 0.36mA maximum. One Teletype 20mA current loop and four EIA RS-232C receivers for received serial data and/or control signals.

I/O CONTROL: One 8-bit output port relative card address selectable as 0 or 2. Lower four bits for baud rate control (35 to 9600 baud) and/or EIA control outputs and/or peripheral control driver. Upper four bits for UART control, i.e., word length, parity, and number of stop bits. Control conditions can be jumper controlled, or software controlled. One 8-bit status input port, relative card address selectable as 0 or 2. Bits selectable with jumpers to read UART flags, i.e., Receive Data Available, Transmitter Buffer Empty, parity, overrun, and framing errors, and/or EIA control inputs and/or Data Available flags for parallel input ports, and/or External Device Ready flags for parallel output ports.

INTERRUPT CONTROL: Any status flag may be jumpered to the Interrupt Bus Driver. Interrupt operation requires use of a vectored interrupt capability to gate the Restart instruction to the processor.

ACCESS & CYCLE TIMES:	1.0 microsecond worst case
BUS PINOUT:	Plug-in compatible with S-100 Bus
POWER REQUIREMENT:	+7.5 to +10VDC at 600mA max;-14 to -19VDC at 450mA max (with SIM-1 and TXT-2 options installed)
DIMENSIONS:	5.3in x 10.0in (13.46cm x 25.4cm)

2. VDM-1 SPECIFICATIONS

DISPLAY FORMAT: 16 lines of 64 characters, upper and lower case, with descenders. Control characters visible when on-card switch is thrown.

VIDEO OUTPUT: EIA composite video, 1VP-P nominal, 75 ohms 6.7 MHz.

INPUT: ASCII data written into RAM memory on card. Bit 7 sets cursor at character location. Processor may read contents of on-card RAM memory.

STATUS INPUTS: Single eight-bit word written by processor to output port on card. Comprised on two four-bit screen display parameters; Text Displacement and Text Address. Text Displacement (high-order bits); number of blanked lines on screen before display begins. Text Address (low-order four bits); line address within on-card RAM memory at which display begins.

STATUS OUTPUTS: Single bit read by processor from input port on card (DI Ø). Indicates that less than 0.25 sec has elapsed since last status parameter was output to card.

CURSOR: Solid video inversion block (black character on white background) superimposed over each character having bit 7 set to "1".

ADDRESSING: Any 1K page may be selected for memory address. Any I/O port address having the low-order two bits equal to "Ø" may be selected. Selection is performed by jumpers on card.

SCREEN ACCESS: One wait-state required for each processor access to on-card RAM memory. Processor always has priority over screen. Screen will blank during cycle in which access is performed.

SCREEN BLANKING: (a) During processor access cycle (PSYNC to PSYNC of following cycle). (b) From vertical sync pulse to beginning of text as specified by Text Displacement parameter. (c) From CR character (non-inclusive) to end of text line (may be disabled with on-card switch). (d) From VT character (non-inclusive) to bottom of display (may be disabled with on-card switch).

BLINKING: Cursors will flash at 0.5-second intervals if enabled with on-card switch.

VIDEO POLARITY: Polarity of display may be changed to black-on-white with on-card switch.

POWER REQUIREMENT: +8V/1.0 A max +16V/50 mA typical; -16V/30 mA typical.

DIMENSIONS: 5.3in x 10.0in (13.46cm x 25.4cm)

BUS PINOUT: S-100 Compatible.

3. Subsystem B Memories Specifications

	4KRA	8KRA	16KRA
Maximum Capacity (8-bit words)	4096	8192	16,384
RAMS used	91L02A	91L02A	Intel 2104 or Mostek 4096 Types
Operating Mode	Static	Static	Dynamic
Access and Cycle Time	520 nanoseconds worst case maximum. Typical 400 nanoseconds.	Same	400 nsec access 500 nsec cycle
Bus Pinout	S-100	Same	Same
Power Operating	+7.5 to 10 VDC @ 1.0A max (0°C), 0.8A typical at 25°C. 0.8A typical, 1A max	+7.5 to 10 VDC @ 1.4A typical (25° C); 1.9A max (0°C to 70°C)	+7.5 to 10 VDC @ 0.4A typical, 0.8A max +15 to +18 @100mA max -15 to -18 VDC @20mA max
Power Standby	+1.6 to 2.5 VDC at 0.5A max worst case 0.4A typical	+1.6V to 2.5 VDC typical; 0.9A max. (power connector provided for battery connection)	
Address Selection	Dual in line switches	Dual in line switch at top of PC board allows manual selection of any 8K segment on 1K increments	Each 4096-byte page addressable with dual in-line switches at top edge of PC board
Dimensions	5.3in x 10.0 in (13.46cm x 25.4cm)	5.4in x 10.0in	5.4in x 10.0in

4. CUTS SPECIFICATIONS

BAUD RATE: 300 bps or 1200 bps data rate. Baud rate may be changed under program control.

CODING: Asynchronous Manchester encoding, CUTS/Byte/ Kansas City Standard compatible.

INPUTS AND OUTPUTS: Two audio input and two audio output jacks are provided, with common circuitry. Two separate tape transport control outputs. Can be used with two cassette recorders at once.

PORT ADDRESS: Switch selectable to one of 130 possible port addresses, from Ø through FA (hexadecimal).

VOLTAGE REQUIREMENTS: +8, +16, and -16 VDC unregulated. On-board voltage regulators are included for each voltage.

5. GPM SPECIFICATIONS

MEMORY SIZE:	8192 bytes ROM addressed EØØØ to FFF (hex.) for firmware.
	2048 bytes ROM normally addressed cØØØ to C3ØØ for CUTER program.
	1024 bytes RAM addressed C8ØØ to CBFF in GPM version as CUTER scratch-pad, or ØØØØ-Ø3FF in GPM-Sol version.
ACCESS & CYCLE TIMES:	1.0 microsecond worst case
BUS PINOUT:	Plug-in compatible with S-100 Bus
POWER REQUIREMENT:	+7.5 to +10VDC at 600mA max;-14 to -19VDC at 450mA max (with SIM-1 and TXT-2 options installed)
DIMENSIONS:	5.3in x 10.0in (13.46cm x 25.4cm)