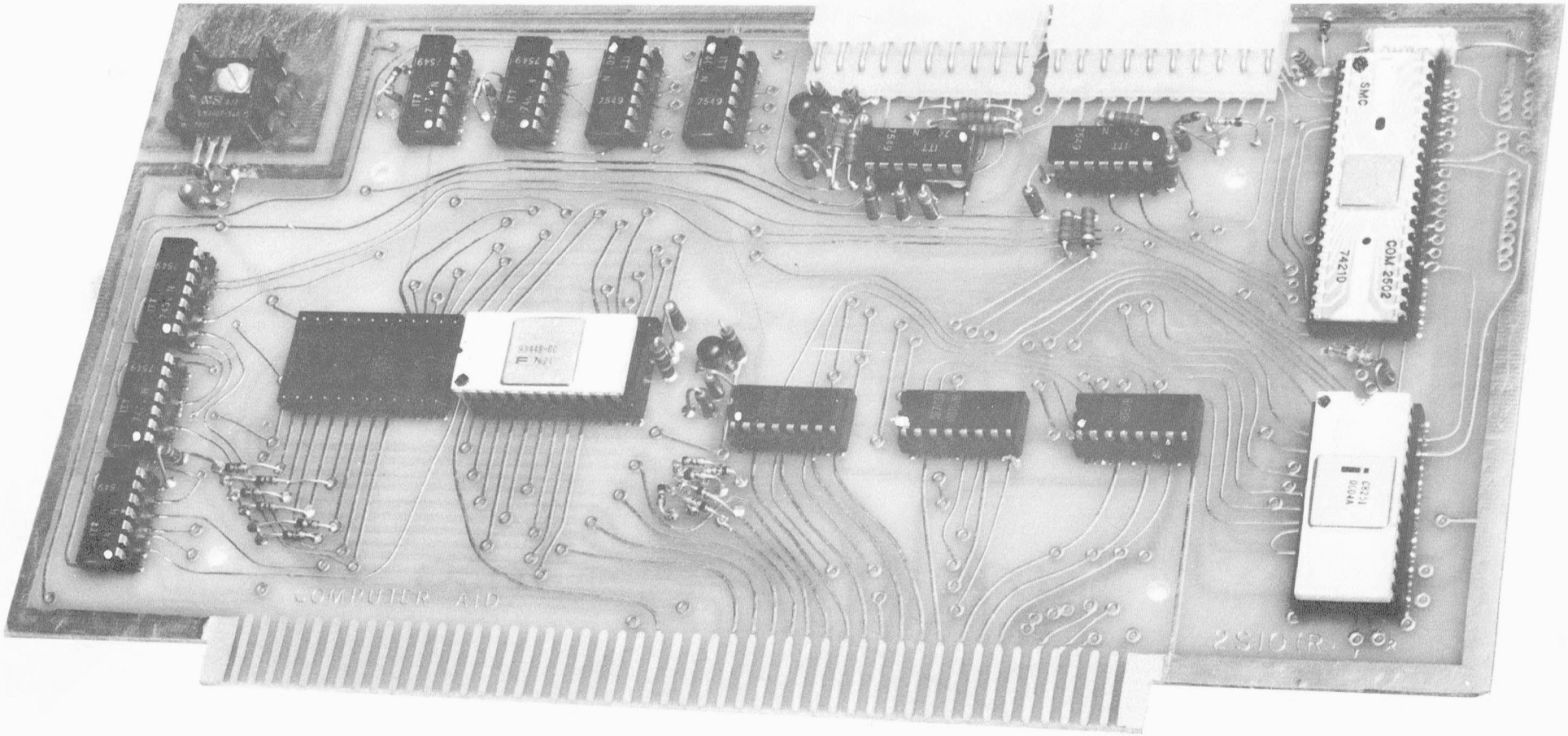
NATIONAL MULTIPLEX COMPUTER OPERATING SYSTEM



2SIO(R)

2 S I O (R) B O A R D
GENERAL INFORMATION

The 2SIO(R) Board is a plug in I/O board compatible with MITS and IMSAI 8080 type computers. It will control one terminal such as a Teletype, Video Terminal or other serial device and one or two cassettes under full or partial software control. In addition, it offers "Turnkey" operation of the computer since all basic monitor or housekeeping functions are programmed in ROM. While intended primarily for use with the National Multiplex cassette or cartridge units, it will operate other cassettes such as the Phi-Deck.

The user turns on his computer, examines address C000 (Hex) and goes to run. The terminal will answer with a question mark. The user then types in what he wants done. The ROM program enables him to load memory (L), Dump Memory (D), Load a file from Cassettes (Control L), Dump a formatted file to cassettes (Control D), Search for a file and store it in designated memory (S) and to search for and store all repeated file entries such as bookkeeping entries of a given class (Control S). It also permits (with the aid of a second board* which bolts to the master 2SIO(R) Board) the full keyboard control of cassette systems having full software control capability. This function uses the DC1, DC2, DC3 and DC4 commands. (Control Q, R,S,T). This board is not provided since there are too many possible control situations. We can provide design assistance to those requesting it. When used with the 3M1 and 3M3 recorders with the two ROM set, these functions are provided without the second board.

The ROM program automatically formats a file so that the user need only give it a six letter name. This includes putting the name at the start and the necessary stop character at the end.

The E and P functions in software* allow the use of the computer as a word processor when suitable editing type terminals are employed. With a suitable printer these functions will enable one to type error free letters and do automatic mailing lists. Combined with string functions in basic or even the multiple file search on the ROM, you can do selected mailing lists. The E and P functions utilize CALL's on the ROM, but must be typed in to a suitable memory location by the user of a 1 ROM set.*

Since one high density cartridge such as the 3M 300 Cartridge used in the 3M3 model can store 2.5 mega bytes and a reel to reel recorder several times that, the 2SIO(R) with a suitable tape storage medium is the equal of most disc systems except for speed.

* E, P and other programs are provided in RAM software for the 1 ROM set and on the ROM with the two ROM set.

T H E O R Y O F O P E R A T I O N

I SWITCHING CIRCUITS The switching circuits are responsible for selecting what function is to take place - such as clear, read and write. Together with the addressing circuits they control the flow of information on the board.

Refer to Figure 1. This shows the three IC's labelled A, B and C. IC "A" controls the read, write and clear (reset) functions. Pins 1 and 2 receive Power on clear and front panel clear signals which then leave the area to reset the UART and USART at the opposite side of the board. SINP enters via a jumper to pin 4, where mixed with PDBIN it generates the Read signal. This signal (0 on read) is used to gate the bi-directional gates and control the UART and USART read functions. SOUT and PWR combine to give the write signal via pins 9 and 10. Both the read and write signals have outputs only when addressed by the address decoder. This address signal is applied via pins 3 and 11.

IC "B" is a Hex inverter used as the name implies. Pins 1 and 2 are part of the address circuitry to be discussed later. Pin 3 receives an A_0 signal which is inverted to give $\overline{A_0}$. This is combined with $\overline{A_1}$ in IC "C" to give a status enable signal to the UART (SWE). $\overline{A_0}$ is inverted to restore A_0 which is used to control the UART read function (RDE) and the USART Control/Data function. (C/$\overline{D}$). Pin 8 receives A_9 which is inverted for the ROM to select ROM 1 or ROM 2. Pins 11 and 10 invert the $\overline{\text{Read}}$ for use by the UART and USART and pins (2 and 13 invert the $\overline{\text{PWR}}$ signal before it is applied to IC "A" to provide PWR.

In terms of logic:

PDBIN + SINP + Proper address = $\overline{\text{Read}}$. SOUT + PWR + Proper address = $\overline{\text{Write}}$

IC "C" performs three functions. (1) It is used to invert A_1 which is used to select the USART. It also performs the following logic functions.

(2) Read + $\overline{A_1}$ + $\overline{A_0}$ = $\overline{\text{SWE}}$ (3) Read + $\overline{A_1}$ + A_0 = $\overline{\text{RDE}}$

All that we are saying here is that if A_1 is high (The USART is selected), you cannot obtain $\overline{\text{SWE}}$ or $\overline{\text{RDE}}$, and that which of the above you obtain depends on whether A_0 is a "1" or zero. (Zero for control status read and 1 for data read). This selects port 00 for control and 01 for data.

II ADDRESSING Diode logic is used for addressing. Since the only addresses used by the UART and USART are 0, 1, 2 and 3, address lines 2, 3, 4, 5, 6 and 7 must all be zero. Diodes connect these addresses to the input of one section of IC "B", a hex inverter. This input is held low by a 330 ohm resistor. If any of the above lines goes to a "1", the output of this inverter shuts off the USART and UART.

Figure 1 shows the circuits involved. Similar logic is used to select the ROM's. Having selected C000 as the ROM address, A_{15} and A_{14} must be high, but A_{10}, A_{11}, A_{12} and A_{13} must be low. (Addresses A_0 - A_9 are used by the ROM).

The diode transistor logic here says if 14 or 15 go low, the ROM is off, or if 10, 11, 12, or 13 go high the ROM is off. This cutoff signal is applied to the ROM via one of the four gating inputs where A_9 or $\overline{A_9}$ plus MEMR and PDBIN are mixed to select the output reading from the ROM.

III BAUD RATE TIMING IC's "D, E, F and G" are 7493 counters which may be programmed to count from 2 to 16. IC "D" is set by means of diode feedback to divide by 13. This counter divides the 2 mhz clock to obtain a 154 Khz signal which is 16 times the 9600 baud clock. A signal taken from its D output is labelled by a silk screened letter. The "C" output of this counter (labelled A) gives a series of pulses which compares to a 19,200 baud clock x 16. There will be a position error of a few percent here, but this is of no consequence. Successive divide by 2 counts give the baud rates from 9600 to 75. The 2400 baud output is divided by 11 in IC "E" to get 220, which is then divided by 2 to get 110 baud.

BAUD RATE TABLE

	x 16	x 1
A	19,200	
B	9,600	
C	4,800	
D	2,400	38,400
E	1,200	19,200
F	600	9,600
G	300	4,800
H	150	2,400
I	110	1,750
J	75	1,200

IV ROM

The ROM used is the Fairchild 93448 which is a fused link (NOT REPROGRAMMABLE) ROM with tri-state outputs. When all four of the inputs are correct, the ROM outputs its data to the DI buss. In order for the ROM to be "on" MEMR, PDBIN, A_9 or $\overline{A_9}$ and the address logic must all be at the proper voltage level.

One or two ROMs are provided with the Monitor program. A socket for a second ROM is provided for future expansion of the 1 ROM set. The ROM provided is addressed at C000 (Hex) and is selected by switches A_{14}, A_{15} up, all others down and examine. The output of these ROM's is connected to the DI buss by means of a 8 color coded jumper.

NOTE: These jumpers are left long so that you can cut them to install the ROM blowing circuit should you later wish to blow ROM's.

ROM's with your special programs are available. If you have a special program developed, contact us for instructions on ordering. We also provide a blow it yourself circuit so that you can automatically blow your own in the computer - which is the reason for the ribbon lead.

ROM sockets are designated H and I. Use the Right hand socket for ROM I and the left-hand socket for ROM II.

Three 8097, 74367 or 8T97 bidirectional switches (IC's J, K,L) are used to determine whether the bidirectional buss is used for read or write functions. The selection of these switches is done by IC's A,B, C. Figure 1 shows the sections used and their pin numbers. The appropriate edge card connector (main buss) is also shown.

INTERFACING Two buffer IC's M and N are used to buffer between the UART, USART and the cassettes and terminal.

IC "N" buffers the inputs for RS232 and one possible TTY output. Connector II from the area above it has 10 pins which connect to this IC. Pin J is for RS232 from the Terminal to the UART. A diode and resistor in series between this pin and the 7406 limit the current and voltage applied to pin 1. At pin 2 the output is inverted and goes to the UART. A connection from this point to the connector, pin I serves a dual function. It gives a TTL output from Pin J and serves as a TTL input for the UART.

Pin 3 receives the TTL level signal from the UART and inverts it for magnet or cassette drive use. (See Interfacing Instructions). Pin 5 accepts the RS232 input from Pin H for the USART. Its output on pin C is a TTL inversion. Pin C of the connector also serves as a TTL input for the USART. Two other sections of the 7406 are used to invert TBE and ODA from the UART. These are added to the Status buss inverted from their polarity leaving the UART so that a "Go" condition on the status buss is all zeros. One section is not used.

IC "M" is non-inverting. Signals to and from the USART at TTL levels are passed through this 7407 to buffer it from outside inputs and provide current drive for turning cassettes on or off. These signals are RTS, CTS, DTR and DSR. One section is used to provide non-inverted, i.e. TTL magnet drive for the UART. Another section provides a TTL output from the USART to cassettes. Both of these TTL outputs are also connected to RS232 driving transistors for RS232 outputs (+5 to -15 volts).

USART AND UART

The USART is programmed by means of commands sent from the ROM. It cannot use other commands without a ROM change. One factor in its use has been noticed. The program requires that it always start from a reset mode. If the cassette should fail to start and stop on keyboard command, stop the computer, press "External Clear" and then restart. DO NOT hit "Reset." The ROM program calls for the program on the cassettes to be recorded 8 bits, even parity and 1 stop bit.

OPERATING INSTRUCTIONS - R O M 1

The NATIONAL MULTIPLEX PROM MONITOR is a 512 BYTE program stored in Read Only MEMORY. When used with a serial terminal, it allows the user to free himself from the front panel of an 8080 based computer. It also interfaces to any NATIONAL MULTIPLEX DATA RECORDER, giving the capability to automatically store and retrieve tape files. It may also be used with other recorders having RS232 or TTL interfaces and software start/stop.

Since the Monitor requires a small amount of read-write memory, it will automatically search through the address space until it locates a page of RAM. It will automatically locate and use the top 30 decimal bytes in the highest page of implimented memory. Thus, anything the user stores in these locations will be lost when the monitor is entered.

All files stored on tape by the monitor are in the NATIONAL MULTIPLEX format. This is shown below.

```
FILE  )  ------------FILE NAME xxx.............xxx, EOF ---------------
FORMAT)  4 sec leader          File (Binary)          1 sec trailer
```

This format consists of a leader (blank), the file name, file, three control x's as an EOF, and a trailer (blank). Note that because of the EOF code, no more than two consecutive control x's (18 Hex) can be in a file. The monitor searches for three control x's to use as a cassette turn off signal. These control x's are inserted automatically and should not be typed in by the user.

When using the monitor, it is necessary for the operator to manually load and rewind tapes. He must also depress the PLAY button before a SEARCH and the record buttons before a DUMP. The first ten seconds or so of the tape should not be used as it usually contains dropouts. Also, a READ (i.e. search) should not be started in the middle of recorded data since a parity error will occur, spoiling the READ. Normally it is not necessary for the operator to bother with these problems if he allows the monitor to position the tape itself. He only need rewind to just before the start of tape and let the computer do the rest. If not starting at the beginning, start at a silent spot. Always avoid stopping and starting tape in the middle of a data block.

A listing of the monitor has been included to allow the use of the subroutines it contains. These are documented on the listing. Also, the source for some additional commands is included. They can be assembled into RAM the user has and executed via the "G" command. They will also be available later in an optional PROM to plug into the second socket on the I/O board.

The following is a description of the commands recognized by the monitor and how to use them. In the format of each, the following conventions are used:

[]	Delimiter - any non-hex character except control x (space will do)
bbb	Variable number of blanks
Opt.	Optional
Cntl.	Control
Control x	Return to command mode

As an example of the monitor commands. See No.2 below. Type D, a space is optional, then the four character start address in Hex (for example 0000) space, then the stop address (for example 2000) and another character. This time try carriage return although space will also do.

A control function does not print. Hence a space or non-Hex character here can be a good idea to let you see what you did.

COMMANDS:

I-LOAD HEX L [] START ADDRESS [] BYTE 1 [] BYTE 2 [].... [] BYTE 'n' [] CNTLx *
Opt.

The "L" command allows the user to load memory from the terminal. Both the starting address and bytes to be loaded are in hexadecimal. It is not necessary to type leading zeros, but a ØØ must be entered as at least one zero. In the event more than the required number of digits are entered, the computer will use only the four on the right for address or two on the right for data. This is invaluable in correcting errors. The [] represent delimiters which may be any non-hex character except CNTL x. Control x is used to return to the command mode.

* NOTE that the final delimiter is required before the CNTL x if the last byte is to be stored.

The computer will automatically line feed and carriage return as required, printing the present loading address each time it does so when the last Hex character in the address is 0.

II DUMP HEX D [] START ADDRESS [] END ADDRESS []
Opt.

This command dumps memory in hexadecimal to the terminal, from the starting address to the ending address. Both addresses are read in standard Hex format. The output will be displayed with a four-digit address starting each line, followed by the two-digit bytes in the successive memory locations. The ADDRESS corresponds to the left most data byte on the printed line.

III Control D - Binary Dump (No slash)

CNTL D [] START ADDRESS [] END ADDRESS - CR
Opt. (Do not use a slash at the end)

This command dumps memory to the cassette recorder in binary from Start address to stop address. (non-formatted). Two addresses are read in under the standard hexadecimal input format. As soon as the end address is terminated by a CR, the recorder is started and the computer waits 4 sec. before dumping the memory. After the dump it will put approximately one second of trailer on the tape, and then shut off the recorder.

NOTE: The record buttons on the Recorder must be depressed before using this command.

IV Control D - Formatted Binary Dump (with slash)

(Note that a slash is used here.)
CNTL D [] START ADDRESS [] END ADDRESS/ bbb FILE NAME *[CR]
Opt. Opt. Opt.

This command dumps a block of memory to the cassette recorder in the NATIONAL MULTIPLEX file format. The start and end addresses are read in under the standard hexadecimal format, but the end address must be terminated with a slash. The file name consists of six characters (no restrictions except not a CNTL x), and leading blanks will not be included. However, imbedded and trailing blanks must be entered. This command is used for both single and repetitive search (S and Control S).

NOTE 1) The record buttons on the Recorder must be depressed before using this command.

NOTE 2) The dump starts as soon as the sixth letter of the file name is typed. Therefore, if you need to set up your recorder or make any other arrangements, do so before typing the sixth character.

Three Control X's are automatically added at the end of the dump and need not be inserted by the user. This enables the cassette to stop at the end of the file when search is used.

For repetitive file use, the first four characters must be entered. For example, the file name is JOHN. We enter JOHN01, JOHN02, JOHN03 etc. The repetitive search will look for and store all JOHN files. If you want only JOHN03, ask for it in single search.

V S(Search)

S bbb FILE NAME [] LOADING ADDRESS []
Opt. Opt.

This command searches a tape for the specified file, and if found, loads it into memory starting at the loading address. The loading address is in the standard Hex format. The file name is six characters (No Control X's), with leading blanks ignored. Imbedded and trailing blanks must be entered, however. When the last character is entered, the recorder is automatically started and the search begun. When the file is found, the bell on the terminal is rung and loading started. When the EOF mark is found, the recorder stops automatically and the monitor returns with a '?' If a parity error occurred during the read, '02' will be printed across the terminal. The user must hit any key to exit this condition. (This key will be interpretted as a command). Note that since 3CNTL x's are used as an EOF mark, no file may contain more than 2 consecutive CNTL X's since this would prematurely terminate loading.

NOTE: The play button on the Recorder must be depressed before using this command. In general, the recorder or recorders must be in a standby (button down condition) for either load or dump sequences in automatic from the ROM to function.

VI CONTROL S (Repetitive Search)

CNTL S bbb FILE NAME [] LOADING ADDRESS []
Opt. Opt.

This command allows loading of all files on a tape which have file names containing the same first four characters. The file name is entered as the four characters on which the multiple search is based, with leading blanks ignored. Imbedded or trailing blanks must be included however. This command allows you to separate out all files (check data or customer transactions) with a common four letter heading.

For repetitive file use, the first four characters are used for the search, but six file name characters must be entered. For example, the file name is JOHN. We enter JOHN01, JOHN02, JOHN03, etc. The repetitive search will look for and store all JOHN files. If you want only JOHN03, ask for it in single search.

VII CONTROL L (Unformatted Binary Load) CNTL L [] LOADING ADDRESS []
Opt. A space between Control L and the address is optional.

This command is provided to allow loading of unformatted tapes into memory. When the load address is terminated, the Recorder is started and reading begun. Information on the tape is loaded as binary characters, starting at the loading address with the first character on the tape. Loading will continue until a key on the terminal is pressed. This stops the Recorder and executes the command the key indicates. A parity error on read causes 02's to be printed across the terminal, until the user types a key.

NOTE: The play key on the Recorder must be pressed before using this command.

VIII GoTO G [] Go To Address []
Opt.

This command starts execution at the specified GoTo Address. The Address is in standard Hex format, and the processor will jump to this address as soon as it is terminated by the ending non-Hex character, i.e., space - CR or period.

ERROR CHECK

When programs are loaded from the tape, the software checks for parity errors. If a parity error is found, the software outputs an 02 via the terminal continuously until the USART is reset.

If this occurs, hit any key on the keyboard to break the load loop and return to the monitor. Clean the head in the recorder and rewind the tape. Then try again.

If you start the recorder in the middle of a data block, the first 60-70 characters will all give parity errors and set the parity error latch on the USART. This will cause a constant string of 02's to be printed until you exit the command. For this reason always locate a silent spot on the tape before starting. In normal operation the machine stops between data batches and is up to speed before data starts. For this reason, you should always use file names and auto starts if at all possible. If you have unlabelled files, it is a good idea to reload them with labels. At any rate you must start loading from a silent spot or leader section. This also protects the tape since leaving the capstan engaged for long periods on one spot can cause stretching.

When starting and stopping a tape under manual control, the motor switch causes static or arcing noise which can start the USART off on the wrong foot. If at all possible, use the auto start. The schematics given on page 12 of the CC-7 Manual show how this is done. The 4.7 mfd condenser shown has been changed to 33 mfd to prevent this noise on manual starts. On automatic (software controlled) starts the only noise or false signal comes from the tape or transients.

All CC-7A units shipped after June 15, 1976, have these circuits installed. All CC-7 units shipped after that date have the automatic reset but not the motor start transistors installed.

NOTE: To start a tape from the beginning with a clean 15 second leader, use the following routine -

With the cassette player on record, type in S space AAAAAA space (CR)
This will start the recorder searching for a non-existent file, but recording a space condition on the tape. After 10-15 seconds, hit the spaee bar to stop the recorder.
You are now ready to dump files using the Control D routines.

H A N D L I N G M I T S B A S I C

Enter the basic to your computer the first time by the method you normally use. DO NOT, however, use two I/O boards on Ports 0,1 or 2,3 at the same time. If you enter by TTY, use the 2SIO(R). If you use the ACR and ports 6,7 there is no conflict.

Once entered, modify the addresses shown for your basic to the data shown in the boxes, i.e., your dump of these addresses should read the same as ours. Use the L Command and D commands.

This changes the CSAVE and CLOAD routines to call on the 2SIO(R) monitor for cassette dump and load. Once this is done, dump the Basic to a cassette with a formatted heading - say BASIC8 - or BASICE. Then you can reload it with the S command. Dump it 3 or 4 times to make sure you have a good error free copy in case one dump gets a crimp or break.

NOTE: This dump should be uninitialized, i.e., once it is in, do not respond to requests but dump it as it is.

CSAVE - To use this command, insert a tape and depress the record buttons (The Recorder should not start at this time). Type in CSAVE followed by the letter you wish to name the file. (For example, CSAVEA). Then hit return, and the recorder will start. When Basic comes back with an 'ok', press the external clear switch on the computer front panel. This will shut off the Recorder. The file has now been dumped to tape. Try to get 10-15 seconds of leader before dumping to avoid using the front end of the tape which usually has glitches.

CLOAD - To Load a file off tape, insert the tape and depress the play button. Type in CLOAD followed by the file name (1 letter) - (For example, CLOADA). Hit return which will start the Recorder and begin the search. When Basic has found the file, it will ring the terminal bell and type an 'OK' when it is loaded. Press the external clear switch on the computer front panel to stop the recorder.

NOTE 1 - The reading is done under the I/O Board's Prom Monitor which checks for parity errors. Should an error occur, 02's will be printed across the terminal. Pressing any key will exit this condition, but will enter the Monitor. Simply type GØ (zero) to reenter Basic and try the LOAD again.

NOTE 2 - The tape files are done in MITS format so that they can be used on other systems.

NOTE 3 - You can only use CSAVE and CLOAD when BASIC is in the machine already.

MODIFICATIONS TO ALTAIR BASIC FOR 2SIO(R)

ADDRESS	DATA BITS

8K Basic (3.2)

```
01A4    36
01A6    07
1007    E5 CD 4B C1 3E 21 D3 03 CD
1010    AC C1 3E D3 CD BA C1 CD BA C1 CD BA C1 E1 E5 7E
1020    CD BA C1 2A 43 02 EB 2A 45 02 1A 13 CD BA C1 E7
1030    C2 2A 10 E1 D7 C9 F5 CD 4B C1 00 00 00 00 00 F1

1055    CD EO C1
1061    CD EO C1
106D    CD EO C1          1A00 is upper end.
```

8K Basic (3.1)

```
01A4    15
01A6    E0

0FE0    E5 CD 4B C1 3E 21 D3 03 CD AC C1 3E D3 CD BA C1
0FF0    CD BA C1 CD BA C1 E1 E5 7E CD BA C1 2A 43 02 EB
1000    2A 45 02 1A 13 CD BA C1 E7 C2 03 10 CD BA C1 CD
1010    BA C1 E1 D7 C9 F5 CD 4B C1 00 00 F1

1024    CD E0 C1
1033    CD E0 C1
103F    CD E0 C1
```

Extended 3.2

```
0200    C7
0202    98

1598    E5 CD 4B C1 3E 21 D3 03
15A0    CD AC C1 3E D3 CD BA C1 CD BA C1 CD BA C1 E1 E5
15B0    7E CD BA C1 2A F6 03 EB 2A F8 03 1A 13 CD BA C1
15C0    E7 C2 BB 15 E1 D7 C9 F5 CD 4B C1 00 00 00 00 00
15D0    F1

15E6    CD E0 C1
15F2    CD E0 C1
15FE    CD E0 C1          Ends 2AC0
```

* CD E0 C1 is a call routine to the USART software.

ASSEMBLER: The IMSAI assembler is loaded by means of an INTEL checksum loader. The first time this loader is used, via the 2SIO(R), type in the loader with the changes indicated below. Once the Assembler is in the computer, alter the locations shown so that it now responds to terminal ports 00 and 01. Then dump the Assembler on a cassette using the heading ASMBLR and it can be retrieved the next time you use it using S ASMBLR.

IMSAI basic is handled the same way. Locate the I/O routines in the Basic and alter them the same way the Assembler I/O routine has been altered here. Change I/O Ports, status bits and jump addresses plus C2 instead of CA.

IMSAI 8080 - Self-Contained System
Assembler, Revision 1

FOR USART - Normal IMSAI loader.

```
1ØØØ 3E CE D3 Ø3 3E 17 D3 Ø3 3E 11 D3 Ø2 31 67 1Ø CD
1Ø1Ø 51 1Ø FE 3A C2 ØF 1Ø CD 37 1Ø B7 CA 36 1Ø 47 CD
1Ø2Ø 37 1Ø 67 CD 37 1Ø 6F CD 37 1Ø CD 37 1Ø 77 23 Ø5
1Ø3Ø C2 2A 1Ø C3 ØF 1Ø 76 CD 44 1Ø 87 87 87 87 57 CD
1Ø4Ø 44 1Ø B2 C9 CD 51 1Ø FE 3A FA 4E 1Ø C6 Ø9 E6 ØF
1Ø5Ø C9 DB Ø3 E6 Ø2 CA 51 1Ø DB Ø2 E6 7F C9
```

FOR UART - Change last line to:

```
1Ø5Ø C9 DB ØØ E6 Ø1 C2 51 1Ø DB Ø1 E6 7F C9
```

Omit all but last four bytes in line 1.

To load the assembler (from the ROM) type G1ØØØCR, then start the tape. When the tape loading ends, stop and examine ØØ5Ø. Go to Run. Nothing will happen. If the assembler is in, it will echo the keyboard. Type DUMP ØØ5Ø ØØ6Ø CR. If the assembler is running, it will dump between the addresses. Note errors in assembler listed in IMSAI manual. Then make the following changes.

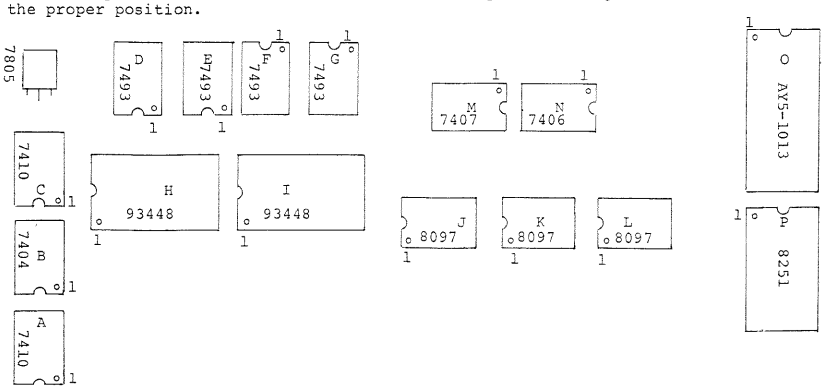
Since the USART has been replaced by the UART for the assembler under keyboard control, initializing the USART is not required. Then change the start to:

```
ØØ5Ø C3 9Ø ØØ
```

Then change the following addressed because of I/O port differences.

INPUT		OUTPUT	
Ø129	DB ØØ	Ø136	DB ØØ
Ø12B	E6 Ø1	Ø138	E6 8Ø
Ø12Ø	C2 29 Ø1	Ø13A	C2 36 Ø1
Ø13Ø	DB Ø1	Ø13D	78
Ø132	E6 7F	Ø13E	D3 Ø1
Ø134	47	Ø14Ø	C9
Ø135	C9		

Layout of IC's; use caution not to bend pins and to get the IC's inserted with pin 1 in the proper position.



CONNECTOR I

Jumpers Required

A	B	C	D	E	F	G	H	I	J
TXD232 (USART)	TXD-TTL (USART)	TSO-232 (UART)	TSO-TTL(UART) (TTY drive normal)	$\overline{CTS}$	TXD TTL	DSR	$\overline{DTR}$	$\overline{RTS}$	RXC (USART) (for synch.use)

CONNECTOR II

A	B	C	D	E	F	G	H	I	J
+5 V	gnd	RXD-TTL (USART)	Inverted TTY drive ($\overline{TSO}$) (UART)	EXT. CIR.		TTY Commutator (Keyboard)	RXD-232	RSI-TTL (UART)	RSI-232 (UART)

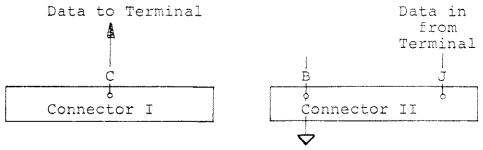
* Be sure the UART Jumpers are compatible with your terminal on data bits, stop bits, and parity bits. If not, the unit will not work.

* If the teletype spaces continuously when connected to pin D of connector I, connect it to pin D of connector II.

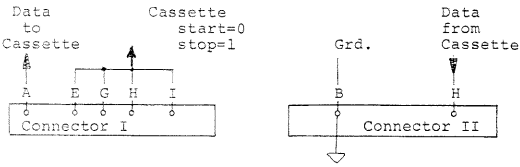
INTERFACING THE 2SIO(R)

The 2SIO(R) is intended to control one or two cassettes and one terminal. However, it can be used many ways. The connections given below are representative.

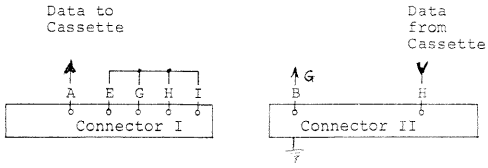
(A) RS232 Terminal to UART



(B) RS232 Cassette (Single) ROM Controlled Start Stop



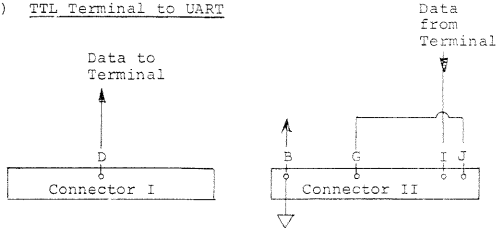
(C) RS 232 Cassette (Single) Manual Start Stop



(D) RS 232 Terminal to USART

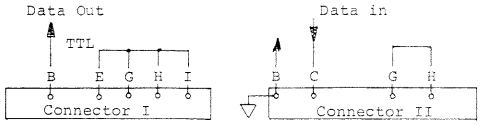
Same as (C) - but jump RTS-CTS and DTR-DSE on EIA plug at terminal. Otherwise you must connect E, G, H, I appropriately. Note that these reverse going from unit to unit.

(E) TTL Terminal to UART

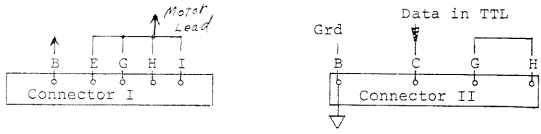


(F) TTL Cassette to UART - Same as E

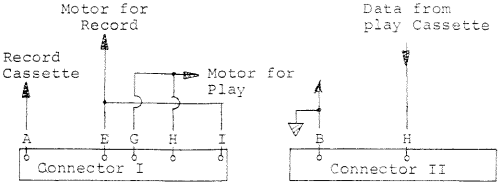
(G) TTL to USART



(H) Single Cassette (TTL) to USART



(I) Dual Cassettes (Read/Write) RS232



Dual Cassettes (REad/Write) TTL

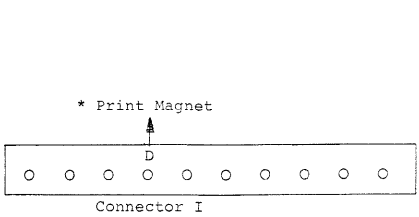
Motor Controls same as (I); data in out as in (G)

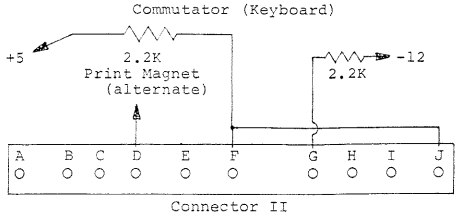
TELETYPE While you can use a teletype on the USART, we recommend that you use the UART only. There are many different teletype connections and we can cover only a few here. Many teletypes use RS232 in/out hence the circuit is the same as that in (A) above.

For current loop systems, typically the magnet driver (Print magnet) input is held at +5 volts in the "mark" state, going to 0 for space. This is the same as a TTL output. MITS uses this with MITS Teletype call units.

The commutator, which is only a switch, is generally connected through two resistors to cause an RS232 voltage swing.

There are two pads on the I/O board above pins F and G on Connector II. Use one for the positive resistor, the other for the negative. Since these pads are hidden by the plug, put the resistors on the side opposite that of the plug.



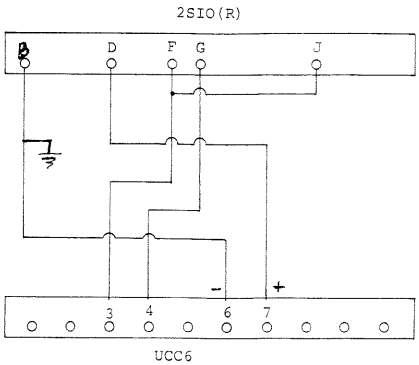


The MITS style connection is shown above.

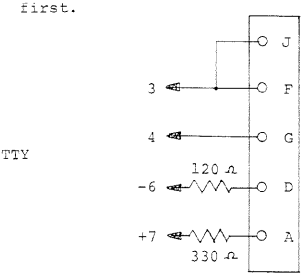
* If for some reason your teletype requires a reversed drive polarity on the print magnet, connect to pin D on Connector II. This will be indicated by continuous running (spacing) when connected as shown above.

The 2.2K resistors shown above are on the back side of the board (i.e.) opposite side from the components.

The connections below apply to the UCC6 line local call control unit. Pins 6,7 must effectively be shorted to stop spacing. If unit does not print, reverse 6,7.



Some users report this will not work for them. They recommend the 3 P+S recommended connection. This is the more or less standard 20 mil. loop connection. If you don't have a MITS teletype, or make your connections directly to the print magnet as shown in the CC-7 manual, try this connection first.

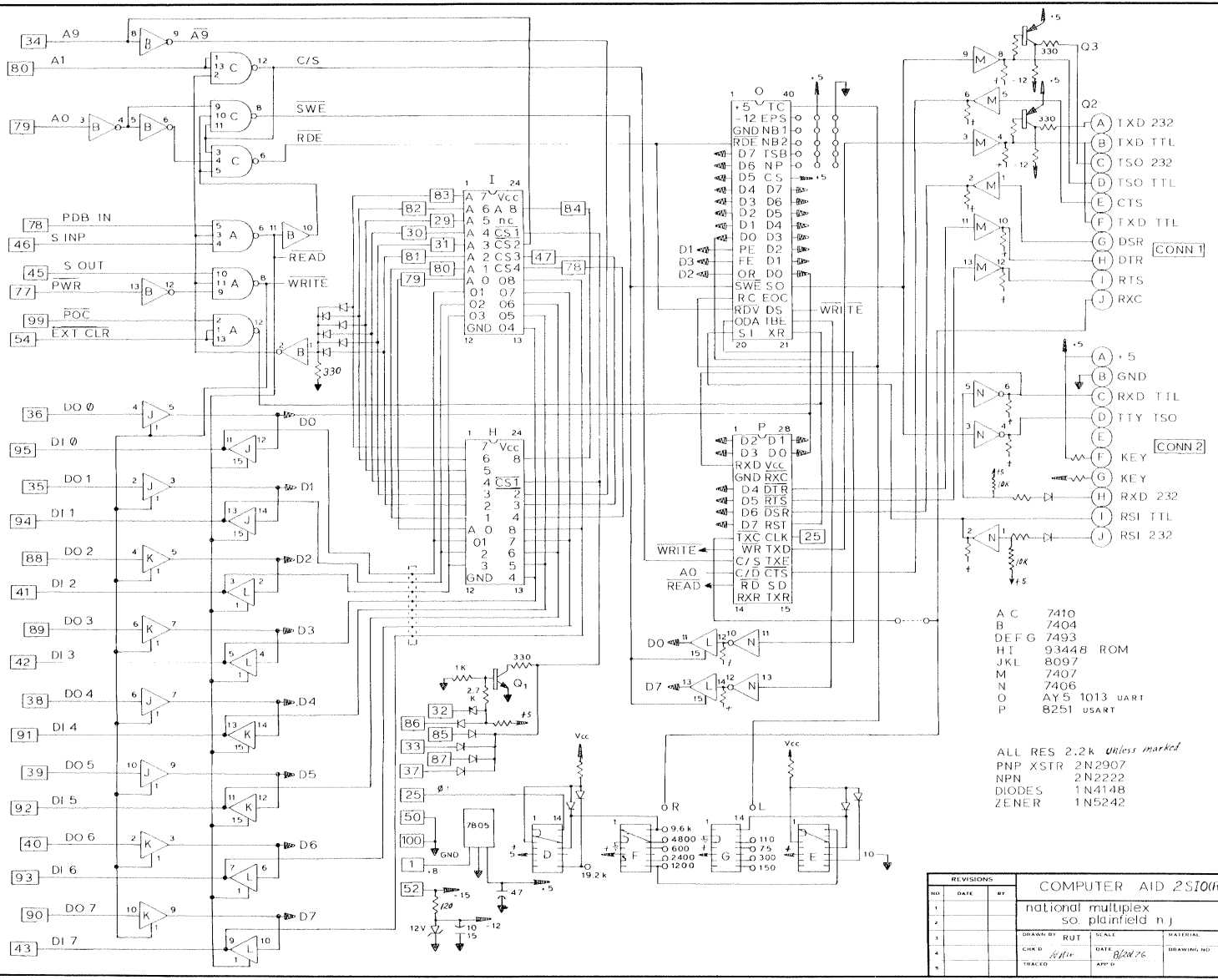


If you wish to change terminal bits, stop bits and parity, use the following table to set up the UART (AY5 -1013).

35	No Parity	NP	A logic "1" on this lead will eliminate the parity bit from the transmitted and received character (no PE indication). The stop bit (a) will immediately follow the last data bit. If not used, this lead must be tied to a logic "0".
36	Number of Stop Bits	TSB	This lead will select the number of stop bits, 1 or 2, to be appended immediately after the parity bit. A logic "0" will insert 1 stop bit and a logic "1" will insert 2 stop bits.
37-38	Number of Bits/Character	NB2, NB1	These 2 leads will be internally decoded to select either 5,6,7 or 8 data bits/character.

NB2	NB1	Bits/Character
0	0	5
0	1	6
1	0	7
1	1	8

39	Odd/Even Parity Select	EPS	The logic on this pin selects the type of parity which will be appended immediately after the data bits. It also determines the parity that will be checked by the receiver. A logic "0" will insert odd parity and a logic "1" will insert even parity.


A C 7410
B 7404
DEF G 7493
HI 93448 ROM
JKL 8097
M 7407
N 7406
O AY5 1013 UART
P 8251 USART
ALL RES 2.2k Unless marked
PNP XSTR 2N2907
NPN 2N2222
DIODES 1N4148
ZENER 1N5242
COMPUTER AID 2SIO(R)
national multiplex
so. plainfield n j
DRAWN BY RUT
CONN 1
CONN 2
TXD 232
TXD TTL
TSO 232
TSO TTL
CTS
DSR
DTR
RTS
RXC
RXD TTL
TTY TSO
KEY
RXD 232
RSI TTL
RSI 232
WRITE
READ
7805
GND
Vcc

```
C000                    0000 *  NATIONAL MULTIPLEX PROM MONITOR
C000                    0010 *       FOR THE 8SIOCR1 BOARD
C000                    0020 *
C000                    0030 *  PROGRAMMED  6/28/78  BY GREG ERLAND
C000                    0040 *
C000                    0050 *  THE BEGINNING OF THIS PROGRAM LOCATES
C000                    0060 * A PAGE OF RAM TO USE AS A STACK AREA.
C000                    0070 * IT WILL FIND THE HIGHEST PAGE OF RAM IN
C000                    0080 * THE USERS SYSTEM AND START THE STACK
C000                    0090 * AT THE TOP OF THAT PAGE. IN NO CASE WILL
C000                    0100 * THE STACK EXCEED 30 DECIMAL BYTES. THUS
C000                    0110 * THE USER IS FREE TO USE UP TO PPE1
C000                    0120 * WHERE PP IS THE HIGHEST PAGE OF IMPLI-
C000                    0130 * MENTED RAM.
C000                    0140 *
C000 3E 55              0150 FNDSTK     MVI     A,55H
C002 21 FF 00           0160            LXI     H,0FFH
C005 25                 0170 DOAGN      DCR     H
C006 77                 0180            MOV     M,A
C007 BE                 0190            CMP     M
C008 C2 05 C0           0200            JNZ     DOAGN
C00B F9                 0210            SPHL
C00C                    0220 *
C00C                    0221 *  THIS THE COMMAND DECODER
C00C                    0225 *
C00C CD 7F C0           0230 PEMON      CALL    CRLF
C00F 3E 3F              0240            MVI     A,'?'
C011 CD F2 C0           0250            CALL    WRCHR
C014 CD E3 C0           0260            CALL    RDCHR
C017 FE 4C              0280            CPI     'L'
C019 CA 44 C0           0290            JZ      LOAD
C01C FE 44              0300            CPI     'D'
C01E CA 55 C0           0310            JZ      DUMP
C021 FE 47              0320            CPI     'G'
C023 CA 7A C0           0330            JZ      GO
C026 FE 53              0340            CPI     'S'
C028 CA FE C0           0350            JZ      ESS
C02B FE 13              0360            CPI     13H
C02D CA 46 C1           0370            JZ      CNTLS
C030 FE 04              0380            CPI     4
C032 CA 5C C1           0390            JZ      CNTLD
C035 FE 0C              0400            CPI     0CH
C037 CA F5 C1           0410            JZ      CNTLL
C03A 21 00 C2           0420            LXI     H,0C200H
C03D 7E                 0430            MOV     A,M
C03E FE 55              0440            CPI     55H
C040 C2 0C C0           0450            JNZ     PEMON
C043 E9                 0460            PCHL
C044                    0479 *
C044                    0500 *  THIS IS THE LOAD HEX ROUTINE. NUMBERS
C044                    0510 * ARE READ IN VIA RDNUM AND STORED IN
C044                    0520 * MEMORY AS EACH NUMBER IS TERMINATED.
C044                    0530 * EACH TIME DURING LOADING THE LSD (HEX)
C044                    0540 * OF THE ADDRESS IS 0 THE ADDRESS IS
C044                    0550 * PRINTED OUT AND A NEW LINE BEGUN.
C044                    0560 * CONTROL X IS USED TO EXIT THE ROUTINE
C044                    0570 * BACK TO AWAIT A NEW COMMAND. HOWEVER,
C044                    0580 * CONTROL X CANNOT BE USED AS A TERMINATOR
C044                    0590 * FOR THE LAST ENTRY SINCE THAT ENTRY
C044                    0600 * WILL NOT BE STORED.
C044                    0610 *
C044 CD 8E C0           0620 LOAD       CALL    RDNUM
C047 42                 0630            MOV     B,D
C048 4B                 0640            MOV     C,E
C049 CD 8E C0           0650 HERE       CALL    RDNUM
C04C 7B                 0660            MOV     A,E
C04D 02                 0670            STAX    B
C04E 03                 0680            INX     B
C04F CD 6B C0           0690            CALL    TSTAUT
C052 C3 49 C0           0720            JMP     HERE
C055                    0730 *
C055                    0740 *  THIS IS THE DUMP COMMAND PROCESSOR.
C055                    0750 * IT DUMPS IN HEX BETWEEN THE TWO
C055                    0760 * SPECIFIED ADDRESSES, WHICH ARE READ
C055                    0770 * IN VIA TWOADR. THE LSD OF THE ADDRESS
C055                    0780 * EQUAL TO 0 SIGNALS A NEW LINE TO BE
C055                    0790 * BEGUN.
C055                    0800 *
C055 CD 89 C0           0820 DUMP       CALL    TWOADR
C058 CD 6F C0           0830            CALL    ADROUT
C05B 0A                 0840 BEGN       LDAX    B
C05C CD C5 C0           0850            CALL    PBHEX
C05F 21 00 C0           0855            LXI     H,FNDSTK
C062 CD A5 C0           0860            CALL    IPCMP
C065 CD 6B C0           0880            CALL    TSTAUT
C068 C3 5B C0           0890            JMP     BEGN
C06B                    0895 *
C06B                    0897 *  THE FOLLOWING SUBROUTINE CHECKS FOR
C06B                    0900 * THE LSD OF THE ADDRESS EQUAL TO 0.
C06B                    0910 * IF NOT, THEN THE ROUTINE IS EXITED.
C06B                    0920 * OTHERWISE THE ADDRESS IS PRINTED ON
C06B                    0930 * ON A NEW LINE AND THEN THE ROUTINE IS
C06B                    0940 * DONE. THE ADDRESS IS ASSUMED TO BE IN
C06B                    0950 * THE BC REGISTER PAIR.
C06B                    0960 *
C06B 79                 0970 TSTAUT     MOV     A,C
C06C E6 0F              0980            ANI     0FH
C06E C0                 0990            RNZ
C06F CD 7F C0           1000 ADROUT     CALL    CRLF
C072 78                 1010            MOV     A,B
C073 CD CC C0           1020            CALL    PNTHEX
C076 79                 1025            MOV     A,C
C077 C3 CC C0           1030            JMP     PNTHEX
C07A                    1040 *
C07A                    1060 *  THIS IS THE GO COMMAND ROUTINE. IT
C07A                    1070 * BEGINS EXECUTION AT THE GIVEN ADDRESS.
C07A                    1080 *
C07A CD 8E C0           1090 GO         CALL    RDNUM
C07D EB                 1100            XCHG
C07E E9                 1110            PCHL
C07F                    1120 *
C07F                    1430 *  THIS SUBROUTINE DOES A CARRIAGE
C07F                    1440 * RETURN AND LINE FEED.
C07F                    1450 *
C07F 3E 0D              1460 CRLF       MVI     A,0DH     CR
C081 CD F2 C0           1470            CALL    WRCHR
C084 3E 0A              1480            MVI     A,0AH     LF
C086 C3 F2 C0           1490            JMP     WRCHR
C089                    1500 *
C089                    1510 *  THE FOLLOWING SUBROUTINE READS IN
C089                    1520 * TWO 16 BIT NUMBERS VIA RDHEX. THE
C089                    1530 * FIRST IS PLACED IN BC. THE SECOND
C089                    1540 * IN DE. THIS IS INTENDED TO BE USED
C089                    1550 * FOR INPUTTING ADDRESSES, BUT RDNUM
C089                    1560 * CAN ALSO BE USED TO READ IN HEX TO
C089                    1570 * DE.
C089                    1580 *
C089 CD 8E C0           1590 TWOADR     CALL    RDNUM
C08C 42                 1600            MOV     B,D
C08D 4B                 1610            MOV     C,E
C08E 21 00 00           1620 RDNUM      LXI     H,0
C091 CD AC C0           1630            CALL    RDHEX
C094 DA 8E C0           1640            JC      RDNUM
C097 29                 1650 RDNXT      DAD     H
C098 29                 1660            DAD     H
C099 29                 1670            DAD     H
C09A 29                 1680            DAD     H
C09B B5                 1690            ORA     L
C09C 6F                 1700            MOV     L,A
C09D CD AC C0           1710            CALL    RDHEX
C0A0 D2 97 C0           1720            JNC     RDNXT
C0A3 EB                 1730            XCHG
C0A4 C9                 1740            RET
C0A5                    1750 *
C0A5                    1760 *  THE NEXT SUBROUTINE INCREMENTS
C0A5                    1770 * THE BC REGISTER PAIR AND THEN
C0A5                    1780 * COMPARES IT TO THE DE PAIR. IF
C0A5                    1790 * BC<=DE THEN THE SUBROUTINE IS
C0A5                    1800 * EXITED. OTHERWISE, CONTROL IS
C0A5                    1810 * PASSED TO THE ADDRESS IN HL.
C0A5                    1820 *
C0A5 03                 1830 IPCMP      INX     B
C0A6 7B                 1840            MOV     A,E
C0A7 91                 1850            SUB     C
C0A8 7A                 1860            MOV     A,D
C0A9 98                 1870            SBB     B
C0AA D0                 1880            RNC
C0AB E9                 1890            PCHL
C0AC                    1900 *
C0AC                    1910 *  THIS SUBROUTINE READS IN HEX
C0AC                    1980 * NUMBERS FROM THE TERMINAL. IF
C0AC                    1930 * THE CHARACTER IS NOT A MEMBER
C0AC                    1940 * OF THE HEX GROUP, IT IS RETURNED
C0AC                    1950 * IN THE ACC WITH THE CARRY BIT
C0AC                    1960 * SET. OTHERWISE THE HEX VALUE
C0AC                    1970 * IS LEFT IN THE LSD OF THE ACC
C0AC                    1980 * AND THE CARRY IS CLEARED.
C0AC                    1990 *
C0AC CD E3 C0           2000 RDHEX      CALL    RDCHR
C0AF E6 7F              2010            ANI     7FH
C0B1 FE 30              2020            CPI     '0'
C0B3 D8                 2030            RC
C0B4 FE 3A              2040            CPI     '9'+1
C0B6 DA C2 C0           2050            JC      NUMB
C0B9 FE 41              2060            CPI     'A'
C0BB D8                 2070            RC
C0BC FE 47              2080            CPI     'F'+1
C0BE 3F                 2090            CMC
C0BF D8                 2100            RC
C0C0 D6 07              2110            SUI     07H
C0C2 D6 30              2120 NUMB       SUI     '0'
C0C4 C9                 2130            RET
C0C5                    2140 *
C0C5                    2150 *  THIS SUBROUTINE PRINTS OUT THE
C0C5                    2160 * ACCUMULATOR IN HEX ON THE TERMINAL
C0C5                    2170 * WITH AN OPTIONAL LEADING BLANK.
C0C5                    2180 *
C0C5 F5                 2190 PBHEX      PUSH    PSW
C0C6 3E 20              2200            MVI     A,' '
C0C8 CD F2 C0           2210            CALL    WRCHR
C0CB F1                 2220            POP     PSW
C0CC F5                 2230 PNTHEX     PUSH    PSW
C0CD 1F                 2240            RAR
C0CE 1F                 2250            RAR
C0CF 1F                 2260            RAR
C0D0 1F                 2270            RAR
C0D1 CD D5 C0           2280            CALL    DIGOUT
C0D4 F1                 2290            POP     PSW
C0D5 E6 0F              2300 DIGOUT     ANI     0FH
C0D7 C6 30              2310            ADI     '0'
C0D9 FE 3A              2320            CPI     '9'+1
C0DB DA F2 C0           2330            JC      WRCHR
C0DE C6 07              2340            ADI     'A'-'0'-10
C0E0 C3 F2 C0           2350            JMP     WRCHR
C0E3                    2360 *
C0E3                    2370 *  THE FOLLOWING SUBROUTINE READS
C0E3                    2380 * AND WRITES CHARACTERS FROM THE
C0E3                    2390 * TERMINAL. THE READ ROUTINE AUTO-
C0E3                    2400 * MATICALLY ECHOS THE INPUT. A CNTL
C0E3                    2410 * X ON INPUT RETURNS CONTROL TO THE
C0E3                    2450 * MONITOR.
C0E3                    2430 *
C0E3 DB 00              2440 RDCHR      IN      0
C0E5 1F                 2450            RAR
C0E6 DA E3 C0           2460            JC      RDCHR
C0E9 DB 01              2470            IN      1
C0EB E6 7F              2475            ANI     7FH
C0ED FE 18              2480            CPI     18H       CNTL X
C0EF CA 00 C0           2490            JZ      FNDSTK
C0F2 F5                 2500 WRCHR      PUSH    PSW
C0F3 DB 00              2510            IN      0
C0F5 E6 80              2520            ANI     80H
C0F7 C2 F3 C0           2530            JNZ     WRCHR+1
C0FA F1                 2540            POP     PSW
C0FB D3 01              2550            OUT     1
C0FD C9                 2560            RET
C0FE                    2600 *
C0FE                    2610 *  THIS IS THE ROUTINE THAT SEARCHES
C0FE                    2620 * FOR FILES ON TAPE. IT FIRST READS
C0FE                    2630 * IN THE FILE NAME AND LOADING ADDRESS
C0FE                    2640 * VIA TWO SUBROUTINES. THE NUMBER OF
C0FE                    2650 * CHARACTERS IN THE FILE NAME MUST BE
C0FE                    2660 * STORED IN REGISTER B BEFORE ENTERING
C0FE                    2670 * THIS ROUTINE. AFTER THE FILE NAME IS
C0FE                    2680 * FOUND CHARACTERS ARE LOADED DIRECTLY
C0FE                    2690 * INTO MEMORY VIA THE DE PRINTER. THREE
C0FE                    2700 * CONTROL X CHARACTERS TERMINATE THE
C0FE                    2710 * LOAD.
```

```
C0FE
C0FE 06 06
C100 CD C5 C1
C103 E5
C104 CD 8E C0
C107 CD 4B C1
C10A E1
C10B 48
C10C 2E E1
C10E CD E0 C1
C111 8E
C112 C2 0B C1
C115 23
C116 0D
C117 C2 0E C1
C11A 3E 07
C11C CD F9 C0
C11F 0E 03
C121 CD E0 C1
C124 FE 18
C126 CA 2E C1
C129 12
C12A 13
C12B C3 1F C1
C12E 18
C12F 13
C130 0D
C131 C2 21 C1
C134 78
C135 FE 04
C137 CA 0B C1
C13A FE FF
C13C CA 1F C1
C13F 3E 40
C141 D3 03
C143 C3 00 C0
C146
C146
C146
C146
C146
C146
C146 06 04
C148 C3 00 C1
C14B
C14B
C14B
C14B
C14B
C14B
C14B 3E FE
C14D D3 03
C14F 3E 40
C151 D3 03
C153 3E FE
C155 D3 03
C157 3E 16
C159 D3 03
C15B C9
C15C
C15C
C15C
C15C
C15C
C15C
C15C
C15C
C15C CD 89 C0
C15F FE 2F
C161 CA 8C C1
C164 CD 4B C1
C167 3E 21
C169 D3 03
C16B CD AC C1
C16E 21 7B C1
C171 0A
C172 CD BA C1
C175 CD A5 C0
C178 C3 71 C1
C17B 0E 03
C17D 3E 18
C17F CD BA C1
C182 0D
C183 C2 7F C1
C186 CD AC C1
C189 C3 3F C1
C18C C5
C18D 06 06
C18F CD C5 C1
C192 48
C193 CD 4B C1
C196 3E 21
C198 D3 03
C19A CD AC C1
C19D 2E E1
C19F 7E
C1A0 CD BA C1
C1A3 23
C1A4 0D
C1A5 C2 9F C1
C1A8 C1
C1A9 C3 6E C1
C1AC AF
C1AD 2C
C1AE E3
C1AF E3
C1B0 E3
C1B1 E3
C1B2 C2 AD C1
C1B5 3C
C1B6 C2 AD C1
C1B9 C9
C1BA
```

```
2720 *
2730 ESS        MVI     B,6
2740            CALL    FRCHR
2750            PUSH    H
2760            CALL    RDNUM
2770            CALL    RDINIT
2780 TRYAGN     POP     H
2790            MOV     C,B
2800            MVI     L,0E1H
2810 LOOP1      CALL    TPRD
2820            CMP     M
2830            JNZ     TRYAGN+1
2840            INX     H
2850            DCR     C
2860            JNZ     LOOP1
2870 FOUND      MVI     A,7         BELL
2880            CALL    WRCHR
2890 CONTIN     MVI     C,3
2900            CALL    TPRD
2910            CPI     18H         CNTL X
2920            JZ      MAYBE
2930            STAX    D
2940            INX     D
2950            JMP     CONTIN
2960 MAYBE      STAX    D
2970            INX     D
2980            DCR     C
2990            JNZ     CONTIN+2
3000            MOV     A,B
3010            CPI     4
3020            JZ      TRYAGN+1
3030            CPI     0FFH
3040            JZ      CONTIN
3050 HALT       MVI     A,40H
3060            OUT     3
3070            JMP     FNDSTK
3080 *
4000 *  THIS IS THE CONTROL S HANDLER.
4010 * IT LOADS B WITH THE NUMBER OF
4020 * CHARACTERS TO SEARCH FOR AND THEN
4030 * JUMPS INTO THE ESS ROUTINE.
4040 *
4050 CNTLS      MVI     B,4
4060            JMP     ESS+2
4070 *
4080 *  THIS ROUTINE INITIALIZES THE USART
4090 * FOR AN INPUT OPERATION IN THE APPRO-
4100 * PRIATE FORMAT (8 BITS, EVEN PARITY,
4110 * 2 STOP BITS).
4120 *
4130 RDINIT     MVI     A,0FEH
4240            OUT     3
4250            MVI     A,40H
4260            OUT     3
4270            MVI     A,0FEH
4280            OUT     3
4290            MVI     A,16H
4300            OUT     3
4310            RET
4320 *
4330 *  THIS SUBROUTINE DUMPS MEMORY TO TAPE.
4340 * IF THE END OF DUMP ADDRRESS IS TERMINATE
4350 * WITH A '/' A FILE NAME IS ALSO READ IN
4360 * AND THE MEMORY DUMPED IN FILE FORMAT.
4370 * THIS CONSISTS OF THE FILE NAME FOLLOWED
4380 * BY THE DUMP IN BINARY FOLLOWED BY THE
4390 * EOF (3 CNTL X'S)
4400 *
4410 CNTLD      CALL    TWOADR
4420            CPI     '/'
4430            JZ      FILE
4440 DUMPER     CALL    RDINIT
4450            MVI     A,21H
4460            OUT     3
4470            CALL    WT
4480            LXI     H,ZILOG
4490 MORE       LDAX    B
4500            CALL    WRTP
4510            CALL    IPCMP
4520            JMP     MORE
4530 ZILOG      MVI     C,3
4540            MVI     A,18H       CNTL X
4550            CALL    WRTP
4560            DCR     C
4570            JNZ     ZILOG+4
4580            CALL    WT
4590            JMP     HALT
4600 FILE       PUSH    B
4610            MVI     B,6
4620            CALL    FRCHR
4630            MOV     C,B
4640            CALL    RDINIT
4650            MVI     A,21H
4660            OUT     3
4670            CALL    WT
4680            MVI     L,0E1H
4690 FIN        MOV     A,M
4700            CALL    WRTP
4710            INX     H
4720            DCR     C
4730            JNZ     FIN
4740            POP     B
4750            JMP     XORE-3
4760 WT         XRA     A
4770            INR     L
4780            XTHL
4790            XTHL
4800            XTHL
4810            XTHL
4820            JNZ     WT+1
4830            INR     A
4840            JNZ     WT+1
4850            RET
4860 *
```

```
C1BA
C1BA
C1BA
C1BA F5
C1BB DB 03
C1BD 1F
C1BE D2 BB C1
C1C1 F1
C1C2 D3 02
C1C4 C9
C1C5
C1C5
C1C5
C1C5
C1C5
C1C5
C1C5 21 00 00
C1C8 39
C1C9 2E E1
C1CB 48
C1CC CD E3 C0
C1CF FE 20
C1D1 CA CC C1
C1D4 0D
C1D5 77
C1D6 23
C1D7 CD E3 C0
C1DA 0D
C1DB C2 D5 C1
C1DE 77
C1DF C9
C1E0
C1E0
C1E0
C1E0
C1E0
C1E0
C1E0 DB 00
C1E2 1F
C1E3 D2 3F C1
C1E6 DB 03
C1E8 1F
C1E9 1F
C1EA D2 E0 C1
C1ED E6 02
C1EF C4 CC C0
C1F2 DB 02
C1F4 C9
C1F5
C1F5
C1F5
C1F5 CD 8E C0
C1F8 CD 4B C1
C1FB 06 FF
C1FD C3 1F C1
```

```
4670 *  THIS SUBROUTINE WRITES THE ACC
4880 * ONTO THE TAPE.
4890 *
4900 WRTP       PUSH    PSW
4910            IN      3
4920            RAR
4930            JNC     WRTP+1
4940            POP     PSW
4950            OUT     2
4960            RET
4970 *
4980 *  THIS SUBROUTINE READS IN FILE NAMES
4990 * FROM THE TERMINAL AND STORES THEM IN
5000 * THE STACK PAGE STARTING AT E1. THE
5010 * NNUMBER OF CHARACTERS TO BE READ IN
5020 * IS ASSUMED IN THE B REGISTER.
5030 *
5040 FRCHR      LXI     H,0
5050            DAD     S
5060            MVI     L,0E1H
5070            MOV     C,B
5080 LPR        CALL    RDCHR
5090            CPI     ' '
5100            JZ      LPR
5110            DCR     C
5120 NBLK       MOV     M,A
5130            INX     H
5140            CALL    RDCHR
5150            DCR     C
5160            JNZ     NBLK
5170            MOV     M,A
5180            RET
5190 *
5200 *  THIS SUBROUTINE READS IN CHARACTERS
5210 * FROM THE TAPE. IT ALSO CHECKS TO SEE
5220 * IF THE KEYBOARD IS ACTIVE AND IF SO,
5230 * RETURNS FOR A NEW COMMAND. PARITY
5240 * ERROR ARE CAUGHT AND PRINTED ON THE
5250 * TERMINAL.
5260 *
5270 TPRD       IN      0
5280            RAR
5290            JNC     HALT
5300            IN      3
5310            RAR
5320            RAR
5330            JNC     TPRD
5340            ANI     2
5350            CNZ     PNTHEX
5360            IN      2
5370            RET
5380 *
5390 *  THIS IS THE CONTROL L HANDLER.
5400 *
5410 CNTLL      CALL    RDNUM
5415            CALL    RDINIT
5420            MVI     B,0FFH
5430            JMP     CONTIN
```

```
DC000-C1FF:
C000 3E 55 21 FF 00 25 77 BE C2 05 C0 F9 CD 7F C0 3E
C010 3F CD F2 C0 CD E3 C0 FE 4C CA 4A C0 FE 4A CA 55
C020 C0 FE 47 CA 7A C0 FE 53 CA FE C0 FE 13 CA 46 C1
C030 FE 04 CA 5C C1 FE 0C CA F5 C1 21 00 C2 7E FE 55
C040 C2 0C C0 E9 CD 8E C0 42 4B CD 8E C0 7B 02 03 CD
C050 6B C0 C3 49 C0 CD 89 C0 CD 6F C0 0A CD C5 C0 21
C060 00 C0 CD A5 C0 CD 6B C0 C3 5B C0 79 E6 0F C0 CD
C070 7F C0 78 CD CC C0 79 C3 CC C0 CD 8E C0 E8 E9 3E
C080 0D CD F8 C0 3E 0A C3 F8 C0 CD 8E C0 42 4B 21 00
C090 00 CD AC C0 DA 8E C0 29 29 29 29 B5 6F CD AC C0
C0A0 D2 97 C0 EB C9 03 7B 91 7A 98 D0 E9 CD E3 C0 E6
C0B0 7F FE 30 D8 FE 3A DA C2 C0 FE 41 D8 FE 47 3F D8
C0C0 D6 07 D6 30 C9 F5 3E 20 CD F2 C0 F1 F5 1F 1F 1F
C0D0 1F CD D5 C0 F1 E6 0F C6 30 FE 3A DA F2 C0 C6 07
C0E0 C3 F2 C0 DB 00 DB 00 DB 01 E6 7F FE 18 CA
C0F0 00 C0 F5 DB 00 E6 80 C2 F3 C0 F1 D3 01 C9 06 06
C100 CD C5 C1 E5 CD 8E C0 CD 4B C1 E1 48 2E E1 CD E0
C110 C1 BE C2 0B C1 23 0D C2 0E C1 3E 07 CD F2 C0 0E
C120 03 CD E0 C1 FE 18 CA 2E C1 12 13 C3 1F C1 12 13
C130 0D C2 21 C1 78 FE 04 CA 0B C1 FE FF CA 1F C1 3E
C140 40 D3 03 C3 00 C0 06 04 C3 00 C1 3E FE D3 03 3E
C150 40 D3 03 3E FE D3 03 3E 16 D3 03 C9 CD 89 C0 FE
C160 2F CA 8C C1 CD 4B C1 3E 21 D3 03 CD AC C1 21 7B
C170 C1 0A CD BA C1 CD A5 C0 C3 71 C1 0E 03 3E 18 CD
C180 BA C1 0D C2 7F C1 CD AC C1 C3 3F C1 C5 06 06 CD
C190 C5 C1 48 CD 4B C1 3E 21 D3 03 CD AC C1 2E E1 7E
C1A0 CD BA C1 23 0D C2 9F C1 C1 C3 6E C1 AF 2C E3 E3
C1B0 E3 E3 C2 AD C1 3C C2 AD C1 C9 F5 DB 03 1F D2 BB
C1C0 C1 F1 D3 02 C9 21 00 00 39 2E E1 48 CD E3 C0 FE
C1D0 20 CA CC C1 0D 77 23 CD E3 C0 0D C2 D5 C1 77 C9
C1E0 DB 00 1F D2 3F C1 DB 03 1F 1F D2 E0 C1 E6 02 C4
C1F0 CC C0 DB 02 C9 CD 8E C0 CD 4B C1 06 FF C3 1F C1
?
```

```
C21C                    0200 *  THIS IS THE ROUTINE TO PRINT
C21C                    0210 * MEMORY IN ASCII ON THE TERMINAL.
C21C                    0220 *
C21C CD 89 C0           0230 PASCII     CALL    TWOADR
C21F CD 7F C0           0240            CALL    CRLF
C222 0A                 0250 HERE2      LDAX    B
C223 CD F2 C0           0260            CALL    WRCHR
C226 21 00 C0           0270            LXI     H,FNDSTK
C229 CD A5 C0           0280            CALL    HLCMP
C22C C3 22 C2           0290            JMP     HERE2
C22F                    0300 *
C22F                    0310 *  THIS IS THE ROUTINE TO ENTER
C22F                    0320 * ASCII CHARACTERS INTO MEMORY.
C22F                    0330 *
C22F CD 8E C0           0340 LASCII     CALL    RDNUM
C232 CD E3 C0           0350            CALL    RDCHR
C235 FE 5F              0360            CPI     5FH     BKARROW
C237 CA 3D C2           0370            JZ      ERASE
C23A 12                 0380            STAX    D
C23B 13                 0390            INX     D
C23C 13                 0400            INX     D
C23D 1B                 0410 ERASE      DCX     D
C23E C3 32 C2           0420            JMP     LASCII+3
```

To use the E and P functions in RAM, locate them at some convenient place. Let the calls to ROM alone. Change the jump addresses to fit your location (3 places).

To use E, manual examine the LASCII address, go to run, type starting address for your ASCII data storage and (CR). Then start typing.

To dump ASCII, examine PASCII address, go to run, enter start address, space, end address. Now if you don't want that on your copy, get everything set and hit return. It will print it out.

PASCII can also be used to punch 8 bit tape, but you will have to find the start and mark it. CR for example will appear ahead of your data dump.

These things will be corrected as the 2nd ROM Program grows.