National Multiplex

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 $\overline{Read}$ for use by the UART and USART and pins (2 and 13 invert the $\overline{PWR}$ signal before it is applied to IC "A" to provide PWR.

In terms of logic:

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

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

All that we are saying here is that if A_1 is high (The USART is selected), you cannot obtain $\overline{SWE}$ or $\overline{RDE}$, 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.

(C000 Hex) H H L L
1 1 0 0 0 0 X X X X X X X X

examine

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. 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 tomatically and the monitor returns with a '?' If a parity error occurred during the read, '02' will be inted 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.

XX P [ ] 6 letter File Name [ ] Loading Address C/R
This command is used for phase encoded tapes where external clocking from a phase encoding interface is provided. It is otherwise the same as the S command. The USART receives different set up codes which are set for PE 2400 (1 stop bit, even parity, 1/1 clock).

XXI Control P [ ] 4 Letter File Name [ ] Loading Address C/R
This is the Phase Encoded (PE2400) repeat search Command. It is the same as the Control S command except for the USART commands which are for 1/1 clock, 1 stop bit, even parity.

XXII Control E [ ] Start Address [ ] Stop Address/file name.
This command formats a file in the PE2400 mode. It is similar to the Control D command except for the USART commands; (1/1 clock, 1 stop bit even parity). Note that the slash must follow the last character of the stop address. If a space or carriage return is placed here, the file will be unformatted. The file name should be six characters, but spaces may be part of the six.

XXIII E [ ] Start Address C/R
This command loads unformatted tape in the PE2400 mode. It is similar to the Control L handler used in the NRZ program.

XXIV GC3F7 Clear All of Memory to Zeroes.
This is an endless loop which writes zeroes to all of memory. The computer must be stopped from the front panel and the monitor reinitialized by examining location C000 and running about 2 seconds after typing this command.

* Note PE2400 is the name given to the new "Phase Encoded 2400 Baud, 1 stop bit, even parity" system. It can be used with audio cassettes and a suitable audio or phase Encoded interface such as the National Multiplex AC1 or the PerCom interface. However, the method is not limited to 2400 baud. The USART can operate in the 40-50 kilobaud region.

EXTERNAL CLOCK OPERATION

When using the 2SIO(R) with self-clocking systems such as the Tarbell, K.C.Standard, PE2400, Bi-Sync, or other encoded systems, follow the instructions below. Note that these systems require external cassette interfaces or software.

Just below the USART are two pads marked "T" and "R". On the opposite side of the board is a short foil jumper which connects these two points. Cut this foil. The R pad now goes directly to Connector I, pin j. A clock from the recorder or reconstructed clock is brought in via pin j. Remove any baud rate jumper to R above IC "F" and jump the baud rate to "T" on the USART. The USART now receives its transmit clock from the baud generator and its receive clock from an external source.

To get a clock signal out to an encoder board, use pin B Connector I. Cut the foil to this pin and jump it to the baud rate generator. TXD-TTL is also available on pin F so the loss of Pin B is of no importance.

STANDARD and TARBELL

There are too many variables with these systems to give a full description. Study the 8251 set-up codes and match these to what you have. For example, to load 7 bit even parity and two stop bits, use mode FA and command 31. To play it back, use mode FA, command 16. These codes used with the K and Control K commands will load teletype tape equivalent codes from a KC standard tape. But if the tape is 8 bit data, no parity and two stop bits, you will have to change the code. Knowing the bit pattern on KC Standard tapes is an absolute necessity. Do not use the synchronous mode for KC Standard.

Bi-Sync or Tarbell tapes require software in RAM. The two sync bits (bi-sync) are output to the 8251 as shown on the data sheets and the 8251 does the rest. Use the synchronous mode for Tarbell with a 1/1 clock. Tarbell runs at 1500 baud, which is sufficiently close to 4800 baud divided by 3 to enable a phase encoder board to be built using 4800/3 clock, or a free running PLL or 555 timer can be used. There are no start, stop or parity bits with Bi-Sync, so you must use a checksum.

You may use any external clock source such as that derived from a 555 timer, a K.C. Standard board, the clock track on the CC-7 or the National Multiplex Audio cassette board.

To use the CC-7 with a 1/1 clock, remove the white wire in the cable from its normal RS232 Output point and connect it to pins 3,4($\overline{Q}$) of the 7401 on the P.C. board. Record the clock via the blue lead, recover it from the white lead. Clock gain is best left full on. The CC-7 will record a 2400 baud clock, but not a 4800 baud clock. At 4800 baud you will have to run asynchronous or use a 2/1 divided clock.

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.

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 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.

OAD - 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.

CLOAD

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 LO 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.

NOTE 4 - Switches A_{14} and A_{15} must be down before you type GØ. This sets up the terminal option in th Basic for ports 0,1. If you do not do this, the Basic will not respond.

NOTE 5 - MITS has many versions of Basic in circulation. The location of the CSAVE and CLOAD commands thus varies among versions and issues.

MODIFICATIONS TO ALTAIR BASIC FOR 2SIO(R)

8K Basic (3.2)

Address	DATA BITS
1A4	36
1A6	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 E0 C1
1061	CD E0 C1
106D	CD E0 C1

1A00 is upper end.

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.

8K Basic (3.1)

Address	DATA BITS
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

I M S A I S O F T W A R E

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 Ø. Dump from Ø to 1ØØØ Hex.

IMSAI 8080 - Self-Contained System I
Assembler, Revision I

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
```

* This loader will also accept the Imsai SCS2, but not Imsai Basic. Use the loader in the Basic book with the last line altered as shown here.

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

```
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
```

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 and status bits plus C2 instead of CA on the jumps.

IMSAI 8K BASIC 1.3

Modify the following Addresses to:

ØØA6	DB	ØØ	
ØØA8	E6	Ø1	
ØØAA	CA	B3	ØØ
18F5	DB	ØØ	
18F7	E6	Ø1	
18F9	C2	F5	18
1928	DB	ØØ	
192A	B7		
192B	FA	28	19
192E	F1		
192F	D3	Ø1	
1AØE	DB	ØØ	
1A1Ø	E6	Ø1	
1A12	CØ		
1A13	DB	Ø1	

0049	ØØ	ØØ
004B	ØØ	ØØ
004D	ØØ	ØØ

Courtesy of Computer Mart of New Jersey, Colonia, N.J.

POLYMORPHIC

Using the 2SIO(R) Board with the Polymorphic System is possible, but presents some duplication problems. Polymorphic ROM is located at 0000, which prevents the use of most 8080 software. The 2MHZ clock is actually 1.8432 MHZ and the ÷ 13 counter on the 2SIO(R) must be altered to ÷ 12.

PTCo & GODBOUT SOFTWARE

The ALS8 and the 2SIO(R) can be used together. Send for modification to the 2SIO(R) involving a switch to change from 6,7 high to 0,7 low status bits. This applies to all PTCo and Godbout ROM boards and software that have not been modified to change the status ports.

IMSAI Self Contained System II

A special ROM set located at FØØØ is available for use with this system. Alter the following locations when loaded. Use the IMSAI checksum loader on page 5, with line 1Ø5Ø altered for the UART. Leave the cassette turned off while paper tape loading.

BØC6	DB	ØØ	
BØC8	E6	8Ø	
BØCA	C2	C6	BØ
BØCD	78		
BØCE	D3	Ø1	
BØD1	DB	ØØ	
BØD3	E6	Ø1	
BØD5	C2	D1	BØ
BØD8	DB	Ø1	

Fill Addresses BØ84 thru BØ93 with NOP's (ØØ)

Note that there is no command comparable to the L command in the 2SIO(R). To alter specific memory bytes use EXEC FØØØ then load from the ROM. To return GBØØ3. If you go to BØØØ on reentry, you will wipe out your files.

SERVICE NOTES SUMMARY

1. The incidences of blown IC's seem to have been limited to Signetics 1458's with a 1974 date code. There have been no reported incidents with Motorola or Raytheon that have occurred since we changed to Raytheon in assembly.
2. Some instances of reset circuit failure have been reported. This is due to the motor switch (S4) arcing during initial pressing of the play button. It does not occur under software control. If you are getting errors on the first few bytes, this is the problem. Use a 47 microfarad instead of a 4.7 microfarad condenser as shown in Figure 15. This has been changed in all units shipped after July 1, 1976.
3. Sorry about the Software. One user called and said he couldn't make it run. We checked, and sure enough it won't run. Some character in the jump table was wrong in the Hexadecimal software dump.
   After July 1, 1976, all units were shipped with a new 8080 Assembler. We'll test our master periodically to make sure this one keeps running. Someday soon we hope to have a 6800 assembler, but it is not available yet. Unfortunately, some bad ones have gone out. We now use the software as the cassette operating check.
4. A problem with the software control system and TTY use has come up. If you use the software control circuit (Figure 15) with a TTY - looped in and out of the CC-7, the playback amplifier is active during record and stray signals cause the output flip flop to shift. This causes the print magnet to run wild. To record in and out of a CC-7 on TTY, you will have to go back to the original reset circuit shown on the main schematic. Add a 47 mf condenser to the purple lead (to -6 volts) to prevent arcing from S4 during start from causing a false set and initial errors. Alternately, don't leave the output to the print magnet connected while recording.

5. Failure to erase clean causing errors when a tape is re-recorded is not common but it occurred several times so that it is worth mentioning. If your tape still has noise on it after you run it through in record with no signal input, you have an off tolerance head. Change the 3.3K resistor in the record circuit to a 2.7K or 2.2K. If you don't want to remove the board, tack a 4.7K resistor across it on the foil side.
6. Data Track Shorting - Sends 100% errors on data track -- The gain control for the clock track gets pressed down by the cover so that a sharp wire from the data track punctures the insulation on the back of the control and shorts one side of the IC4 output to ground. If you get all errors, remove the back, lift this control slightly and try again. We now cut this sharp tip down. Cut it down as close as you can and add some additional insulation. A piece of cardboard will do in a pinch.

7. TAPES - While we praise Memorex MRX2 highly and have still to find a better low cost generally available tape - it is not computer tape. We buy it in thousand lots, use it to test machines and send it on. About .3 - .4% of what we buy has excessive defects. If you have lots of errors - 10% or more - check the machine. 1 in 10,000 check the tape. We have rejected machines on inspection only to find that they are O.K. when tested with computer tape or a fresh MRX2 cassette.

   CERTIFIED WABASH COMPUTER cassettes are available frmm us at $5.00 each. MRX2 - C30 is 3 for $5.00.

8. NEW CC-7A Units - The latest batch of CC-7A motors have very tight bearings. When first turned on, they draw excessive current. For this reason we have been running in all CC-7A's for a while until the current drain drops to normal. If by some chance we didn't run them in long enough, you may have to reset the speed later as it will tend to creep up in use. After about 10 hours it appears stable.

   We are embarrassed not to have discovered it sooner, but the heads are not all wired the same, some have the phase reversed. This does not effect a single user, but can be disastrous for group users who must exchange tapes. We must have your test tape to check phase on re-orders as well as speed.
   To reverse the phase, the unit must be disassembled and the leads changed at the head. Any common grounds must be removed and each head treated as a separate item.
   Any group users experiencing this problem can return the unit to us and we will reverse the leads at no charge, but send us the test tape from the unit that you want the repaired unit to match.

10. We strongly recommend that you return your CC-7 if you buy a 2SIO(R) and let us put the automatic start/stop in and update your unit. Some very old units may get new P.C. boards. We will also clean the switches and check the motor for noise and wear. (At no charge to you and 1-2 day turn around).

SERVICE NOTE
CC-7, CC-7A and 2SIO(R) in Combination

A The 2SIO(R) uses an Intel USART which is a tough task master on data distortion. Our normal automatic checker will pass 100% of the recorders we ship, but the 2SIO(R) board will reject 50% of them. This has meant we have had to abandon the automatic test set-up and test each one individually on a computer using the 2SIO(R). The items noted below are the principal causes for rejection:

   (1) Spring tension on back of the cassette cartridge. Twist this flat spring clockwise so that it presses on the side of the cassette nearest the pinch roller. Next, put a cassette in and feel the pressure. A properly adjusted spring will exert a very light pressure, just barely enough to hold the cassette forward. If it is too loose, the cassette will flop around in the enclosure. If it is too tight, the cassette will bind. Adjust this spring with your fingers, pushing it back and forth until the right tension is achieved.
   (2) The 4.7 mfd condenser shown in Figure 15 of the Cassette Manual should be a 47 mfd.
   (3) This only effects some units. There is a mechanical resonance from the flywheel or motor that causes excessive wow and flutter. The 3.2"/second we picked for the CC-7A is probably the worst choice we could have made. 3.5" is generally free from this effect. If you suspect this, try increasing the speed about 15%.

B We are happy to report that the first 100 2SIO(R) boards did not turn up any major circuit design changes needed for the next run. We did have to add two 2.2K resistors to the 7493's to stabilize the counter, but this was caught early in the run. The second ru n is now out, with some improvements in layout, but the same circuit. If you have had any trouble with your 2SIO(R), return it for checkout at no charge.

C ROM 2 is expected to be available about 15 October. This will control the present cassette (CC-7) or the new 3M design.

   If you use a Z80 MPU board with variable clock, you will have to change the clocking input on the 2SIO(R). The first 120 boards obtained this from pin 25 (Ø1) which was easily accessible for P.C. layout. With the Z80 board, the Ø1 clock may vary so the 2SIO(R) clock will have to be moved to pin 49 (2MHZ clock). Cut the foil as it leaves pin 25 and ru n a new jumper to 49. On later boards this has been changed.

E Pin E on Connector II is not used now, but is reserved for use with our new model 3M1 and 3M3 recorders. It is jumped to "Ext Clear" pin 54 to reset the USART at the end of the tape (EOT/BOT).

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 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.

IX N - Puts 11 seconds of leader on the tape. Use with CC-7 and CC-7A units to put 11 seconds of erased leader at the start of a cassette.

X F Fast Forward - Used with 3M1 and 3M3 Cartridge units to advance the tape at approximately 25" per second. Stop the cassette or cartridge drive by hitting the space bar.

XI R Rewind - Used to rewind the 3M1 and 3M3 cartridge units at approximately 25" per second. Stop by hitting the space bar.

XII 0 Overshoot. Used to relocate an interrecord gap after a rewind overshoot. Causes the tape to advance at normal speed to locate the interrecord gap by means of sound or LED indicator.

* NOTE - When using F, R or 0, the cartridge units are in the play mode and inter record gaps may be seen on the LED indicator. This signal is also returned to the USART via the DSR lead and may be polled in software.

For F, R, N and 0 do not use C/R. F, R and 0 are stopped by any new character coming in. N will time itself out.

XIII W [ ] Start Address C/R
Enables you to enter ASCII characters into memory from the keyboard. Backspace and type over are recognized. This is a Word Processing command, or word storage system.

XIV Control W [ ] Start Address [ ] Stop Address C/R.

This command prints out the memory in ASCII. Form letters or mailing labels can be printed from the computer utilizing magnetic tape storage.

XV H [ ] 5 letter String [ ] Loading Address C/R

A formatted file with five letters or numbers embedded will be loaded. For example, a mailing label can be picked out by means of the first five letters of the Addresses's last name or by means of the five numbers in the zip code. The entire file is loaded, even though the string occurs at the center or end. This differs from the S or Control S where the string must be at the start.

XVI T [ ] Load Address C/R
Reads from binary paper tape. Leaading rubouts are ignored, so load the tape in the tape reader so as to start in the rubout region.

XVII Control T [ ] Start Address [ ] End Address C/R
Punches paper tape in binary format with the data between the two addresses. Eight rubouts are placed at the start and end of the data to demark the data limits.

EXTERNAL CLOCK TAPE FORMATS

XVIII K [ ] Start Address [ ] USART Code C/R
The normal S or Control L program assumes a fixed USART Code. With this command you can vary the USART code to accept 16/1 or 1/1 clocks, and varying data lengths, stop bits and parity. Use this with Kansas City Standard tapes.(1) The USART may be clocked by the baud rate generator or from the cassette. These tapes are unformatted.

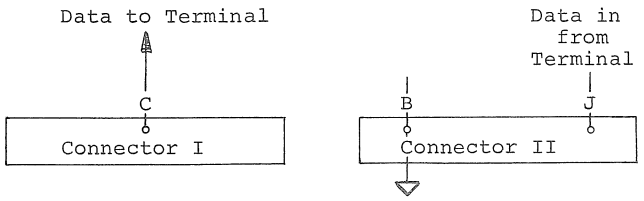
XIX Control K [ ] Start Address [ ] Stop Address [ ] USART Code C/R
This command dumps to tape in any USART compatible format. It operates the same as the Control D command without slash in the fixed NRZ format. Three Control X's are added at the end, but no file names can be used.

(1) See note on KC Standard Tapes.

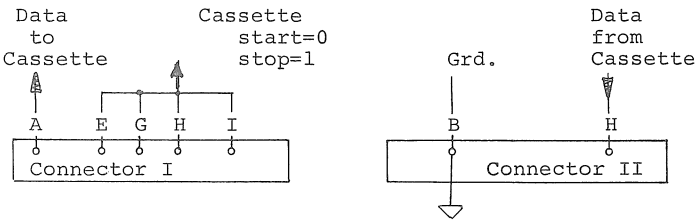
INTERFACING THE 2SIO(R)

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.

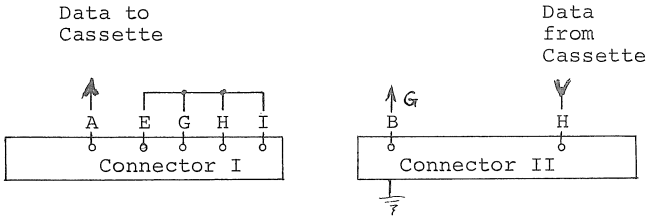
(A) RS232 Terminal to UART



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



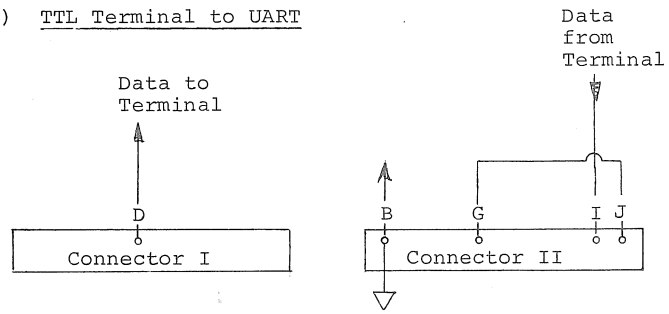
(C) RS 232 Cassette (Single) Manual Start Stop



(D) RS 232 Terminal to USART

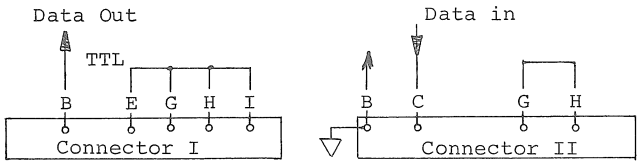
Same as (C) - but jump RTS-CTS and DTR-DSE on EIA plug at terminal. Otherwise you must connect E, G, H, I appropriately. Note that these reverse going from unit to unit.

(E) TTL Terminal to UART

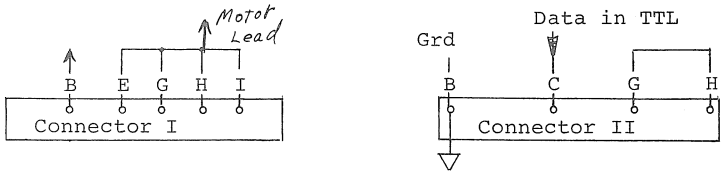


(F) TTL Cassette to UART - Same as E

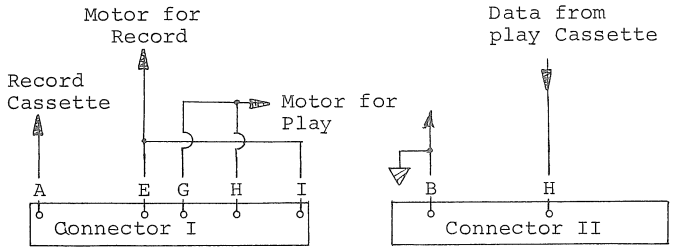
(G) TTL to USART



(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)

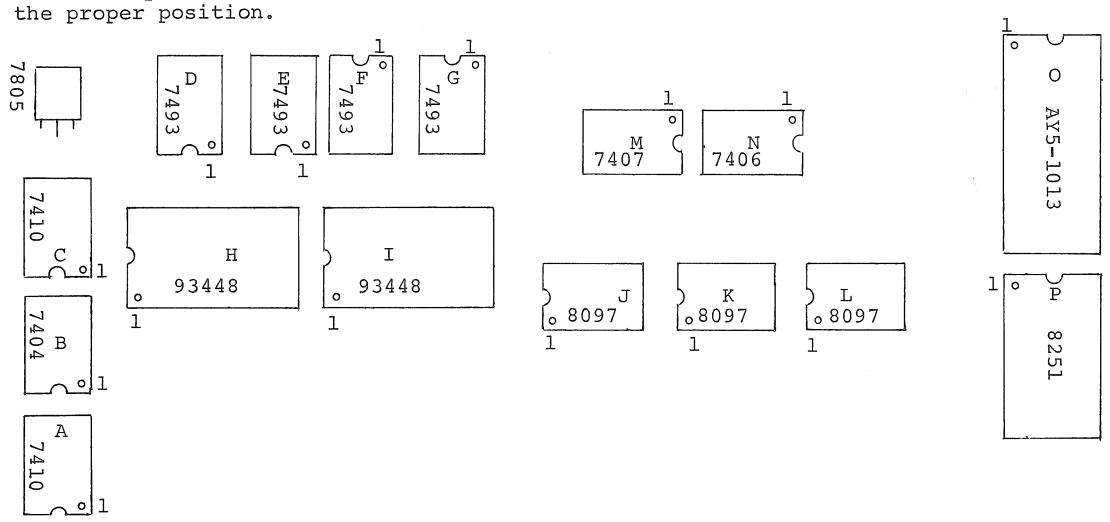
TELETYPE While you can use a teletype on the USART, we recommend that you use the UART only. There are many different teletype connections and we can cover only a few here. Many teletypes use RS232 in/out hence the circuit is the same as that in (A) above.

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.

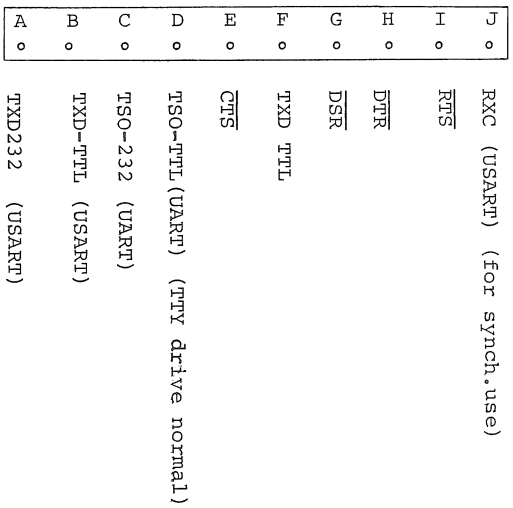
There are two pads on the I/O board above pins F and G on Connector II. Use one for the positive resistor, the other for the negative. Since these pads are hidden by the plug, put the resistors on the side opposite that of the plug.

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

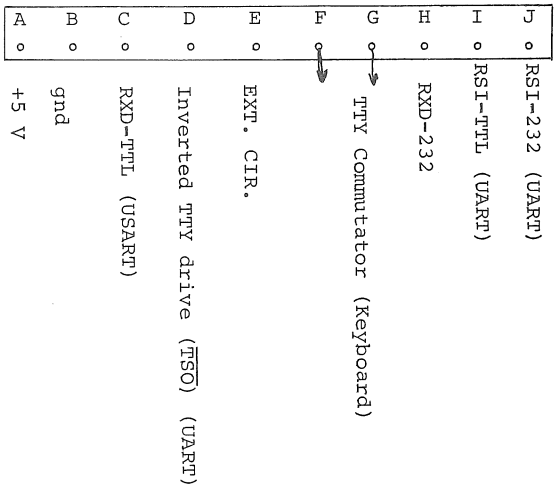


CONNECTOR I

Jumpers Required

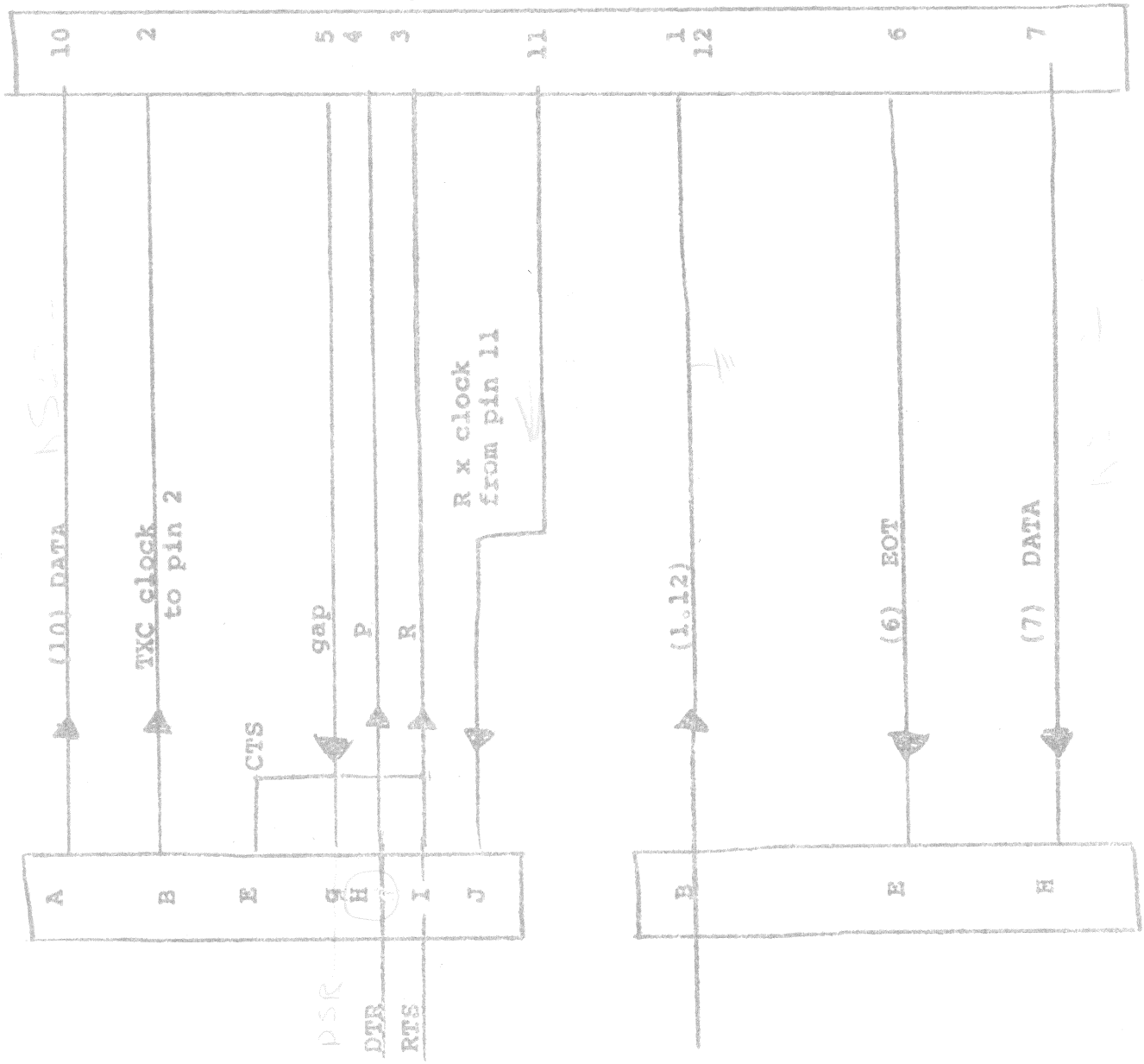


CONNECTOR II



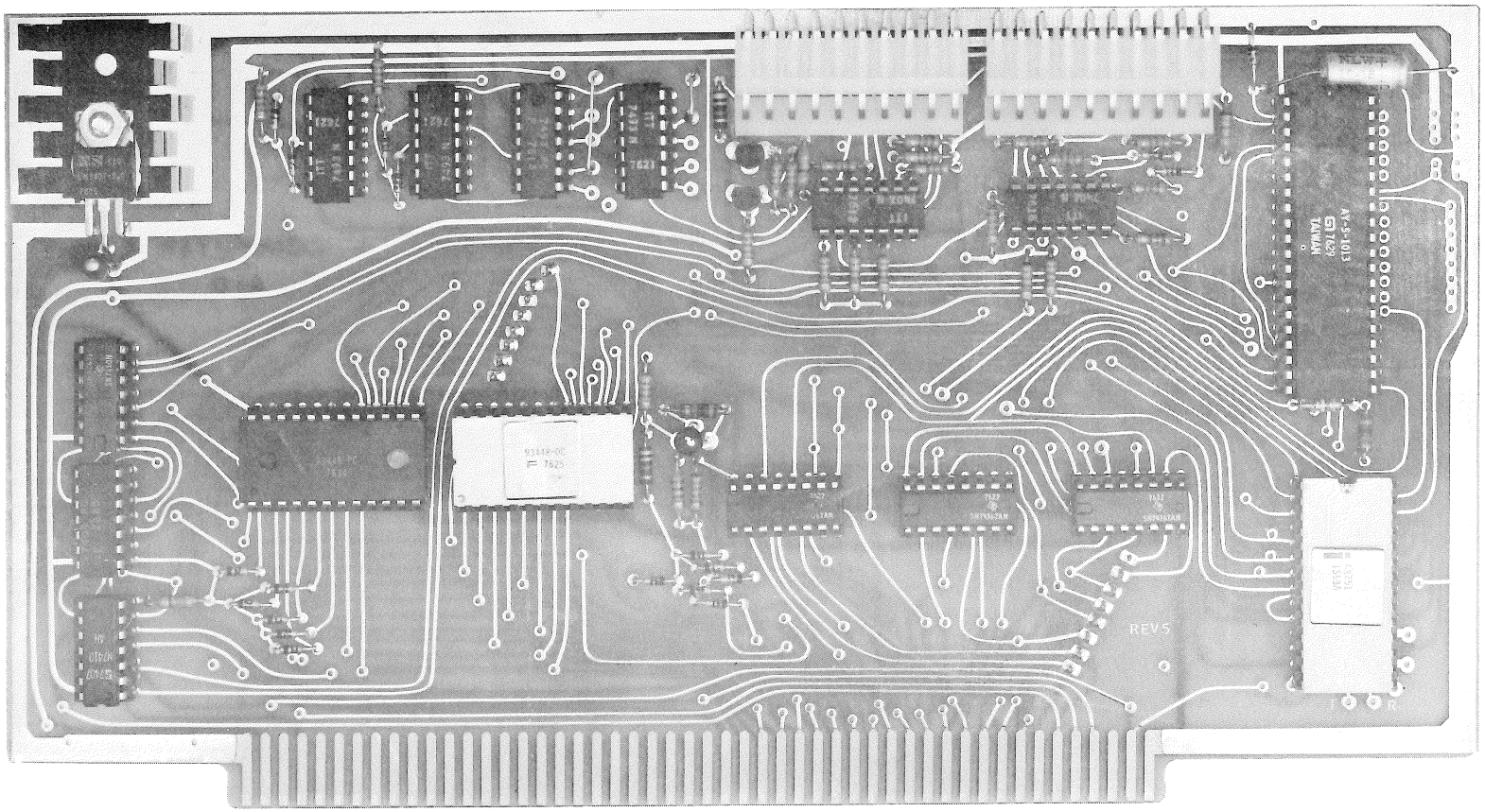
* Be sure the UART Jumpers are compatible with your terminal on data bits, stop bits, and parity bits. If not, the unit will not work.

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

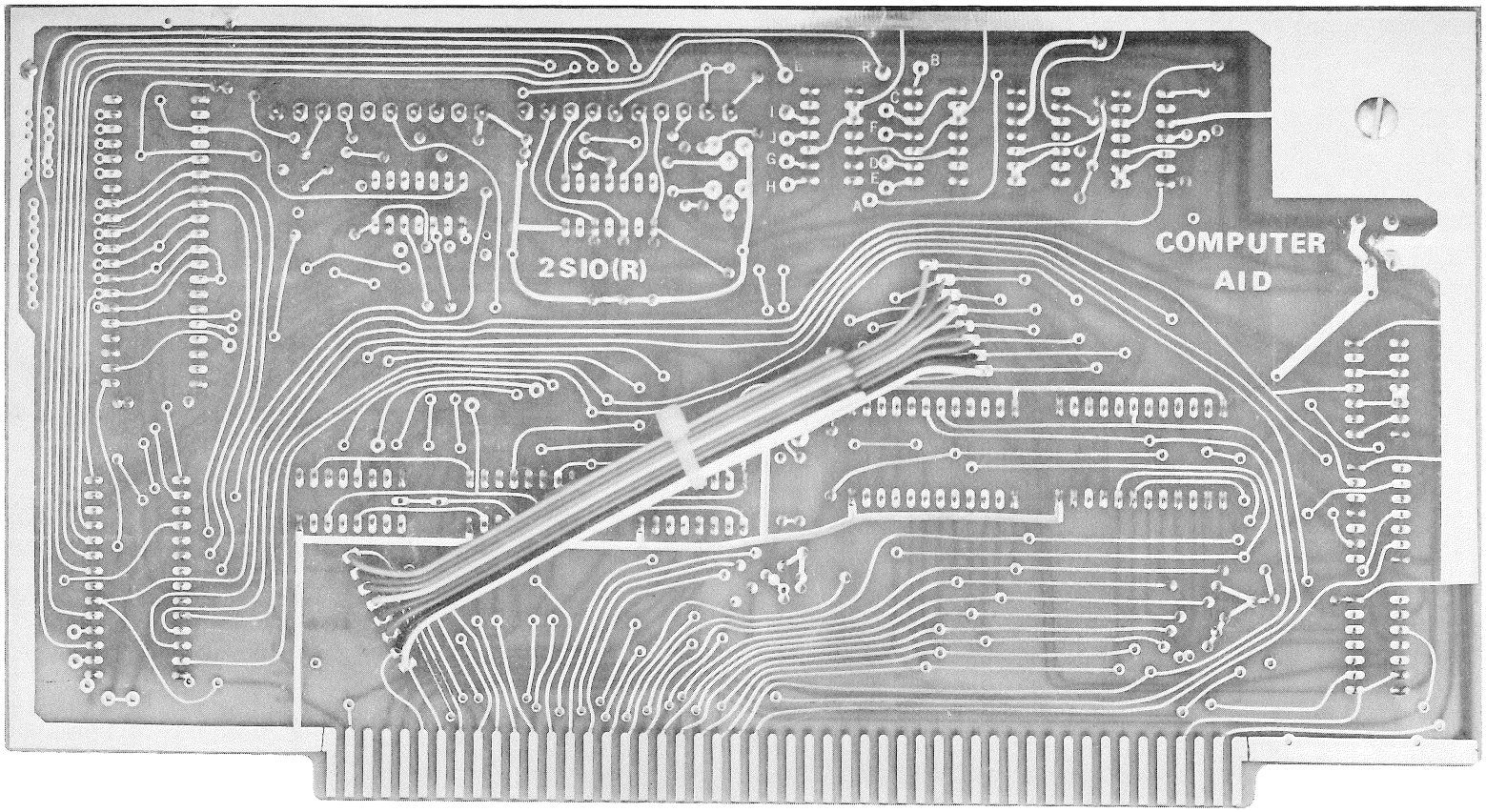


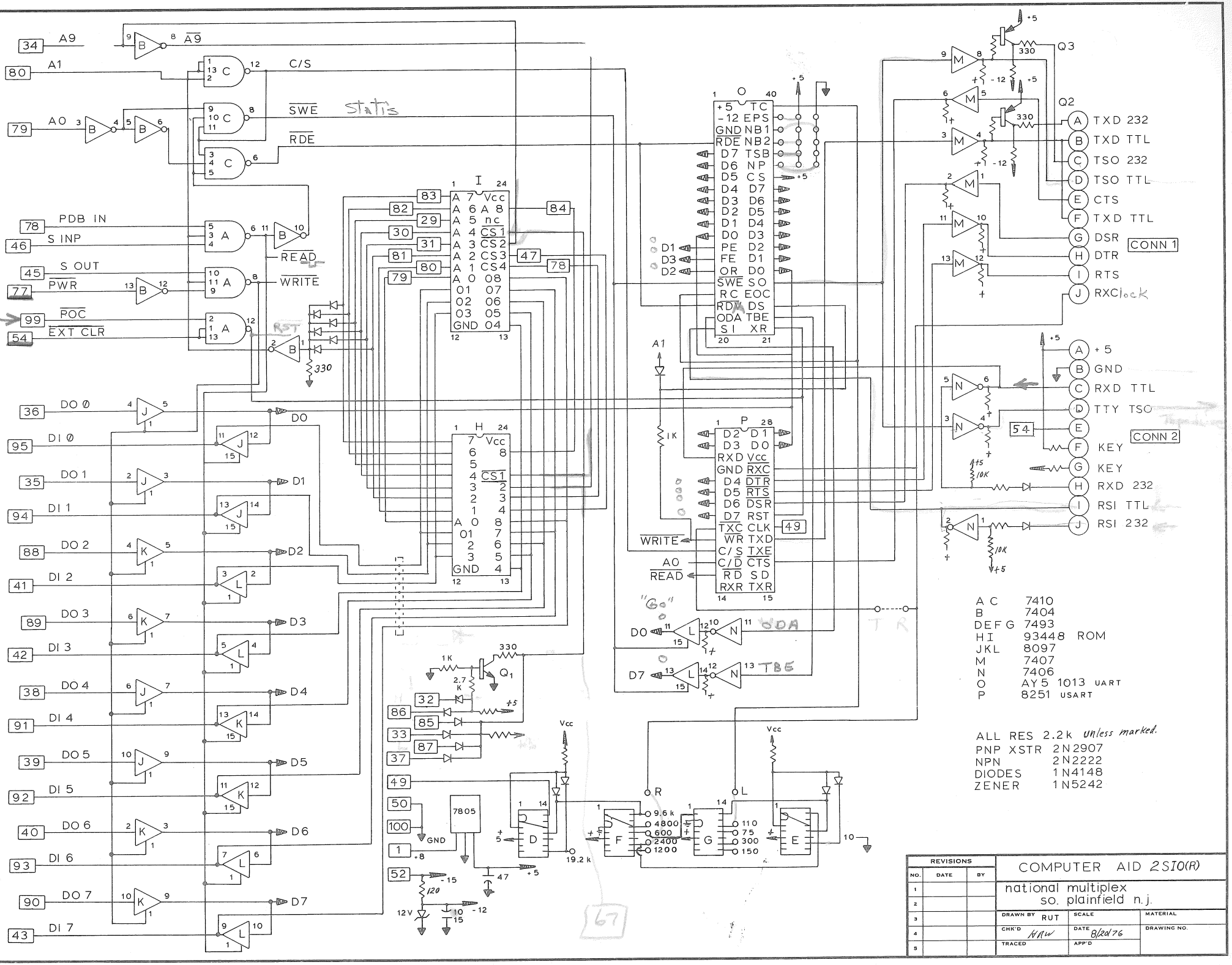
I. Normal connection using 1 to 1 computer clock requires 2SIO(R) BOARD alteration.

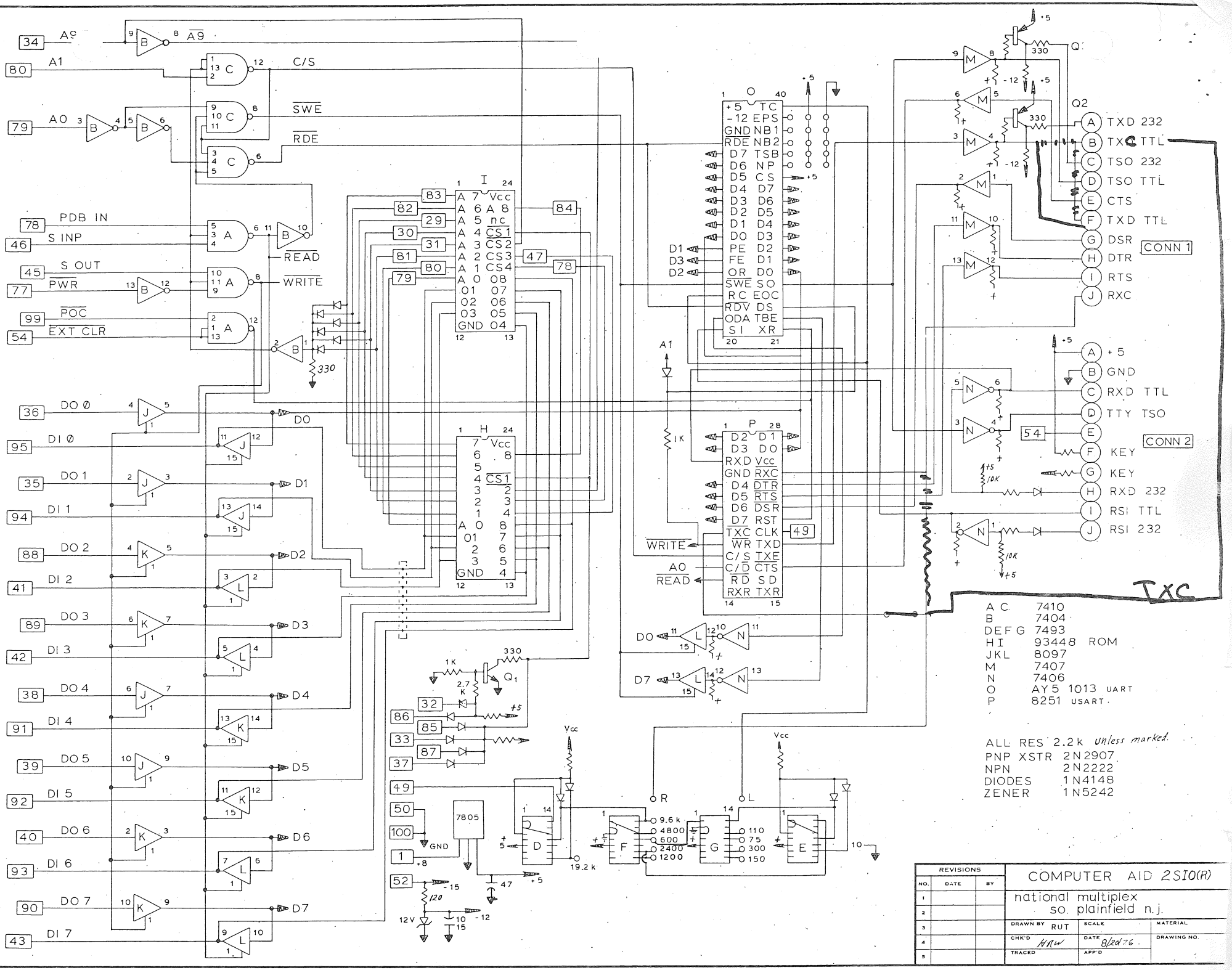
NATIONAL MULTIPLEX COMPUTER OPERATING SYSTEM



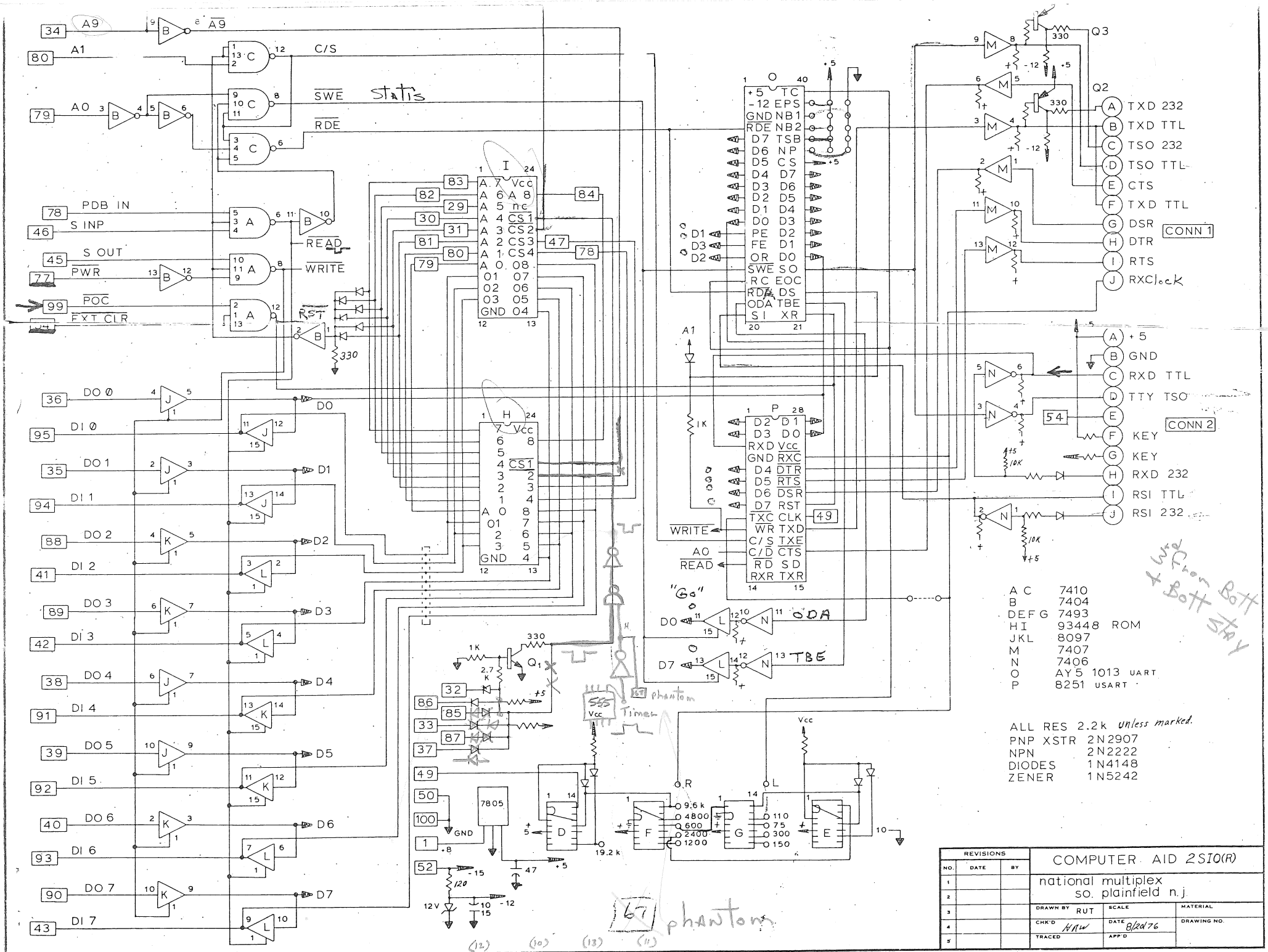
2SIO(R)


2SIO(R)
COMPUTER
AID


A9
A1
A0
C/S
SWE
RDE
PDB IN
S INP
S OUT
PWR
POC
EXT CLR
READ
WRITE
RST
330
DO Ø
DI Ø
DO 1
DI 1
DO 2
DI 2
DO 3
DI 3
DO 4
DI 4
DO 5
DI 5
DO 6
DI 6
DO 7
DI 7
D0
D1
D2
D3
D4
D5
D6
D7
I
H
P
O
1K
2.7 K
Q1
7805
GND
12V
19.2 k
9.6 k
4800
600
2400
1200
110
75
300
150
Vcc
Q2
Q3
TXD 232
TXD TTL
TSO 232
TSO TTL
CTS
TXD TTL
DSR
DTR
RTS
RXClock
CONN 1
+5
GND
RXD TTL
TTY TSO
KEY
KEY
RXD 232
RSI TTL
RSI 232
CONN 2
"Go"
TBE
A C 7410
B 7404
DEFG 7493
HI 93448 ROM
JKL 8097
M 7407
N 7406
O AY 5 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
CHK'D NAW
TRACED
SCALE
DATE 8/20/76
APP'D
MATERIAL
DRAWING NO.
REVISIONS
NO. DATE BY


A9
A1
A0
C/S
SWE
RDE
PDB IN
S INP
READ
S OUT
PWR
WRITE
POC
EXT CLR
330
DO Ø
DI Ø
DO 1
DI 1
DO 2
DI 2
DO 3
DI 3
DO 4
DI 4
DO 5
DI 5
DO 6
DI 6
DO 7
DI 7
D0
D1
D2
D3
D4
D5
D6
D7
I
H
P
O
1K
2.7 K
Q1
7805
GND
12V
19.2 k
9.6 k
4800
600
2400
1200
110
75
300
150
Vcc
R
L
Q1
Q2
TXD 232
TXD TTL
TSO 232
TSO TTL
CTS
TXD TTL
DSR
DTR
RTS
RXC
CONN 1
+5
GND
RXD TTL
TTY TSO
KEY
KEY
RXD 232
RSI TTL
RSI 232
CONN 2
10K
A1
A0
READ
WRITE
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
TXC
REVISIONS
NO. DATE BY
COMPUTER AID 2SIO(R)
national multiplex
so. plainfield n.j.
DRAWN BY RUT
SCALE
MATERIAL
CHK'D HRW
DATE 8/20/76
DRAWING NO.
TRACED
APP'D

1 - 894-1120



A C	7410
B	7404
DEF G	7493
HI	93448 ROM
JKL	8097
M	7407
N	7406
O	AY 5 1013 UART
P	8251 USART

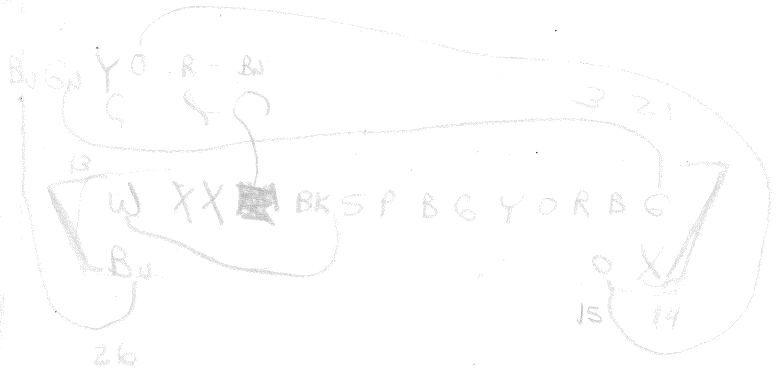
ALL RES 2.2k *unless marked.*

PNP XSTR	2N2907
NPN	2N2222
DIODES	1N4148
ZENER	1N5242

REVISIONS			COMPUTER AID 2SIO(R)		
NO.	DATE	BY	national multiplex so. plainfield n.j.		
1			DRAWN BY RUT	SCALE	MATERIAL
2			CHK'D HAW	DATE 8/20/76	DRAWING NO.
3			TRACED	APP'D	
4					
5					

67 phantom

(12) (10) (13) (11)
33 37 85 87 Low select Rom
32 86 High
(14) (15)


Bu
W
Y
O
R
Bu
3
2
1
13
W
XX
BK
S
P
B
G
Y
O
R
B
G
Bu
15
14
26

```
C000                    0000 *  NATIONAL MULTIPLEX PROM MONITOR
C000                    0010 *       FOR THE 2SIO(R) BOARD
C000                    0020 *
C000                    0030 *  PROGRAMMED  6/28/76  BY GREG EPLAND
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 75 C0           0230 PEMON      CALL    CRLF
C00F 3E 3F              0240            MVI     A,'?'
C011 CD E8 C0           0250            CALL    WRCHR
C014 CD D9 C0           0260            CALL    RDCHR
C017 FE 4C              0280            CPI     'L'
C019 CA 3A C0           0290            JZ      LOAD
C01C FE 44              0300            CPI     'D'
C01E CA 4B C0           0310            JZ      DUMP
C021 FE 47              0320            CPI     'G'
C023 CA 70 C0           0330            JZ      GO
C026 FE 53              0340            CPI     'S'
C028 CA F4 C0           0350            JZ      ESS
C02B FE 45              0380            CPI     'E'
C02D CA 59 C1           0390            JZ      CNTLD
C030 21 00 C2           0420            LXI     H,FNDSTK+512
C033 7E                 0430            MOV     A,M
C034 FE 55              0440            CPI     55H
C036 C2 0C C0           0450            JNZ     PEMON
C039 E9                 0460            PCHL
C03A                    0479 *
C03A                    0500 *  THIS IS THE LOAD HEX ROUTINE. NUMBERS
C03A                    0510 * ARE READ IN VIA RDNUM AND STORED IN
C03A                    0520 * MEMORY AS EACH NUMBER IS TERMINATED.
C03A                    0530 * EACH TIME DURING LOADING THE LSD (HEX)
C03A                    0540 * OF THE ADDRESS IS 0 THE ADDRESS IS
C03A                    0550 * PRINTED OUT AND A NEW LINE BEGUN.
C03A                    0560 * CONTROL X IS USED TO EXIT THE ROUTINE
C03A                    0570 * BACK TO AWAIT A NEW COMMAND. HOWEVER,
C03A                    0580 * CONTROL X CANNOT BE USED AS A TERMINATOR
C03A                    0590 * FOR THE LAST ENTRY SINCE THAT ENTRY
C03A                    0600 * WILL NOT BE STORED.
C03A                    0610 *
C03A CD 84 C0           0620 LOAD       CALL    RDNUM
C03D 42                 0630            MOV     B,D
C03E 4B                 0640            MOV     C,E
C03F CD 84 C0           0650 HERE       CALL    RDNUM
C042 7B                 0660            MOV     A,E
C043 02                 0670            STAX    B
C044 03                 0680            INX     B
C045 CD 61 C0           0690            CALL    TSTAUT
C048 C3 3F C0           0720            JMP     HERE
C04B                    0730 *
C04B                    0740 *  THIS IS THE DUMP COMMAND PROCESSOR.
C04B                    0750 * IT DUMPS IN HEX BETWEEN THE TWO
C04B                    0760 * SPECIFIED ADDRESSES, WHICH ARE READ
C04B                    0770 * IN VIA TWOADR. THE LSD OF THE ADDRESS
C04B                    0780 * EQUAL TO 0 SIGNALS A NEW LINE TO BE
C04B                    0790 * BEGUN.
C04B                    0800 *
C04B CD 7F C0           0820 DUMP       CALL    TWOADR
C04E CD 65 C0           0830            CALL    ADROUT
C051 0A                 0840 BEGN       LDAX    B
C052 CD BB C0           0850            CALL    PBHEX
C055 21 00 C0           0855            LXI     H,FNDSTK
C058 CD 9B C0           0860            CALL    IPCMP
C05B CD 61 C0           0880            CALL    TSTAUT
C05E C3 51 C0           0890            JMP     BEGN
C061                    0895 *
C061                    0897 *  THE FOLLOWING SUBROUTINE CHECKS FOR
C061                    0900 * THE LSD OF THE ADDRESS EQUAL TO 0.
C061                    0910 * IF NOT, THEN THE ROUTINE IS EXITED.
C061                    0920 * OTHERWISE THE ADDRESS IS PRINTED ON
C061                    0930 * ON A NEW LINE AND THEN THE ROUTINE IS
C061                    0940 * DONE. THE ADDRESS IS ASSUMED TO BE IN
C061                    0950 * THE BC REGISTER PAIR.
C061                    0960 *
C061 79                 0970 TSTAUT     MOV     A,C
C062 E6 0F              0980             ANI     0FH
C064 C0                 0990            RNZ
C065 CD 75 C0           1000 ADROUT     CALL    CRLF
C068 78                 1010            MOV     A,B
C069 CD C2 C0           1020            CALL    PNTHEX
C06C 79                 1025            MOV     A,C
C06D C3 C2 C0           1030            JMP     PNTHEX
C070                    1040 *
C070                    1060 *  THIS IS THE GO COMMAND ROUTINE. IT
C070                    1070 * BEGINS EXECUTION AT THE GIVEN ADDRESS.
C070                    1080 *
C070 CD 84 C0           1090 GO         CALL    RDNUM
C073 EB                 1100            XCHG
C074 E9                 1110            PCHL
C075                    1120 *
C075                    1430 *  THIS SUBROUTINE DOES A CARRIAGE
C075                    1440 * RETURN AND LINE FEED.
C075                    1450 *
C075 3E 0D              1460 CRLF       MVI     A,0DH     CR
C077 CD E8 C0           1470            CALL    WRCHR
 07A 3E 0A              1480            MVI     A,0AH     LF
 07C C3 E8 C0           1490            JMP     WRCHR
C07F                    1500 *
C07F                    1510 *  THE FOLLOWING SUBROUTINE READS IN
C07F                    1520 * TWO 16 BIT NUMBERS VIA RDHEX. THE
C07F                    1530 * FIRST IS PLACED IN BC, THE SECOND
C07F                    1540 * IN DE. THIS IS INTENDED TO BE USED
C07F                    1550 * FOR INPUTING ADDRESSES, BUT RDNUM
C07F                    1560 * CAN ALSO BE USED EAD IN HEX TO
C07F                    1570 * DE.
C07F                    1580 *
C07F CD 84 C0           1590 TWOADR     CALL    RDNUM
C082 42                 1600            MOV     B,D
C083 4B                 1610            MOV     C,E
C084 21 00 00           1620 RDNUM      LXI     H,0
C087 CD A2 C0           1630            CALL    RDHEX
C08A DA 84 C0           1640            JC      RDNUM
C08D 29                 1650 RDNXT      DAD     H
C08E 29                 1660            DAD     H
C08F 29                 1670            DAD     H
C090 29                 1680            DAD     H
C091 B5                 1690            ORA     L
C092 6F                 1700            MOV     L,A
C093 CD A2 C0           1710            CALL    RDHEX
C096 D2 8D C0           1720            JNC     RDNXT
C099 EB                 1730            XCHG
C09A C9                 1740            RET
C09B                    1750 *
C09B                    1760 *  THE NEXT SUBROUTINE INCREMENTS
C09B                    1770 * THE BC REGISTER PAIR AND THEN
C09B                    1780 * COMPARES IT TO THE DE PAIR. IF
C09B                    1790 * BC<=DE THEN THE SUBROUTINE IS
C09B                    1800 * EXITED. OTHERWISE, CONTROL IS
C09B                    1810 * PASSED TO THE ADDRESS IN HL.
C09B                    1820 *
C09B 03                 1830 IPCMP      INX     B
C09C 7B                 1840            MOV     A,E
C09D 91                 1850            SUB     C
C09E 7A                 1860            MOV     A,D
C09F 98                 1870            SBB     B
C0A0 D0                 1880            RNC
C0A1 E9                 1890            PCHL
C0A2                    1900 *
C0A2                    1910 *  THIS SUBROUTINE READS IN HEX
C0A2                    1920 * NUMBERS FROM THE TERMINAL. IF
C0A2                    1930 * THE CHARACTER IS NOT A MEMBER
C0A2                    1940 * OF THE HEX GROUP, IT IS RETURNED
C0A2                    1950 * IN THE ACC WITH THE CARRY BIT
C0A2                    1960 * SET. OTHERWISE THE HEX VALUE
C0A2                    1970 * IS LEFT IN THE LSD OF THE ACC
C0A2                    1980 * AND THE CARRY IS CLEARED.
C0A2                    1990 *
C0A2 CD D9 C0           2000 RDHEX      CALL    RDCHR
C0A5 E6 7F              2010            ANI     7FH
C0A7 FE 30              2020            CPI     '0'
C0A9 D8                 2030            RC
C0AA FE 3A              2040            CPI     '9'+1
C0AC DA B8 C0           2050            JC      NUMB
C0AF FE 41              2060            CPI     'A'
C0B1 D8                 2070            RC
C0B2 FE 47              2080            CPI     'F'+1
C0B4 3F                 2090            CMC
C0B5 D8                 2100            RC
C0B6 D6 07              2110            SUI     07H
C0B8 D6 30              2120 NUMB       SUI     '0'
C0BA C9                 2130            RET
C0BB                    2140 *
C0BB                    2150 *  THIS SUBROUTINE PRINTS OUT THE
C0BB                    2160 * ACCUMULATOR IN HEX ON THE TERMINAL
C0BB                    2170 * WITH AN OPTIONAL LEADING BLANK.
C0BB                    2180 *
C0BB F5                 2190 PBHEX      PUSH    PSW
C0BC 3E 20              2200            MVI     A,' '
C0BE CD E8 C0           2210            CALL    WRCHR
C0C1 F1                 2220            POP     PSW
C0C2 F5                 2230 PNTHEX     PUSH    PSW
C0C3 1F                 2240            RAR
C0C4 1F                 2250            RAR
C0C5 1F                 2260            RAR
C0C6 1F                 2270            RAR
C0C7 CD CB C0           2280            CALL    DIGOUT
C0CA F1                 2290            POP     PSW
C0CB E6 0F              2300 DIGOUT     ANI     0FH
C0CD C6 30              2310            ADI     '0'
C0CF FE 3A              2320            CPI     '9'+1
C0D1 DA E8 C0           2330            JC      WRCHR
C0D4 C6 07              2340            ADI     'A'-'0'-10
C0D6 C3 E8 C0           2350            JMP     WRCHR
C0D9                    2360 *
C0D9                    2370 *  THE FOLLOWING SUBROUTINE READS
C0D9                    2380 * AND WRITES CHARACTERS FROM THE
C0D9                    2390 * TERMINAL. THE READ ROUTINE AUTO-
C0D9                    2400 * MATICALLY ECHOS THE INPUT. A CNTL
C0D9                    2410 * X ON INPUT RETURNS CONTROL TO THE
C0D9                    2420 * MONITOR.
C0D9                    2430 *
C0D9 DB 00              2440 RDCHR      IN      0
C0DB 1F                 2450            RAR
C0DC DA D9 C0           2460            JC      RDCHR
C0DF DB 01              2470            IN      1
C0E1 E6 7F              2475            ANI     7FH
C0E3 FE 18              2480            CPI     18H     CNTL X
C0E5 CA 00 C0           2490            JZ      FNDSTK
C0E8 F5                 2500 WRCHR      PUSH    PSW
C0E9 DB 00              2510            IN      0
C0EB E6 80              2520            ANI     80H
C0ED C2 E9 C0           2530            JNZ     WRCHR+1
C0F0 F1                 2540            POP     PSW
C0F1 D3 01              2550            OUT     1
C0F3 C9                 2560            RET
C0F4                    2600 *
```

```
C0F4                    2610 *  THIS IS THE ROUTINE THAT SEARCHES
C0F4                    2620 * FOR FILES ON TAPE. IT FIRST READS
C0F4                    2630 * IN THE FILE NAME AND LOADINNG ADDRESS
C0F4                    2640 * VIA TWO SUBROUTINES. THE NUMBER OF
C0F4                    2650 * CHARACTERS IN THE FILE NAME MUST BE
C0F4                    2660 * STORED IN REGISTER B BEFORE ENTERING
C0F4                    2670 * THIS ROUTINE. AFTER THE FILE NAME IS
C0                      2680 * FOUND CHARACTERS ARE LOADED DIRECTLY
C0                      2690 * INTO MEMORY VIA THE DE POINTER. THREE
C0                      2700 * CONTROL X CHARACTERS TERMINATE THE
C0F4                    2710 * LOAD.
C0F4                    2720 *
C0F4 06 06              2730 ESS        MVI     B,6
C0F6 CD C9 C1           2740            CALL    FRCHR
C0F9 E5                 2750            PUSH    H
C0FA CD 84 C0           2760            CALL    RDNUM
C0FD CD 3E C1           2770             CALL   STOPS
C100 CD 47 C1           2775             CALL   RSET
C103 E1                 2780 TRYAGN     POP     H
C104 48                 2790            MOV     C,B
C105 2E E1              2800            MVI     L,0E1H
C107 CD E4 C1           2810 LOOP1      CALL    TPRD
C10A BE                 2820            CMP     M
C10B C2 04 C1           2830            JNZ     TRYAGN+1
C10E 23                 2840            INX     H
C10F 0D                 2850            DCR     C
C110 C2 07 C1           2860            JNZ     LOOP1
C113 3E 07              2870 FOUND      MVI     A,7         BELL
C115 CD E8 C0           2880            CALL    WRCHR
C118 0E 03              2890 CONTIN     MVI     C,3
C11A CD E4 C1           2900            CALL    TPRD
C11D FE 18              2910            CPI     18H         CNTL X
C11F CA 27 C1           2920            JZ      MAYBE
C122 12                 2930            STAX    D
C123 13                 2940            INX     D
C124 C3 18 C1           2950            JMP     CONTIN
C127 12                 2960 MAYBE      STAX    D
C128 13                 2970            INX     D
C129 0D                 2980            DCR     C
C12A C2 1A C1           2990            JNZ     CONTIN+2
C12D 78                 3000            MOV     A,B
C12E FE 04              3010            CPI     4
C130 CA 04 C1           3020            JZ      TRYAGN+1
C133 FE FF              3030            CPI     0FFH
C135 CA 18 C1           3040            JZ      CONTIN
C138 CD 3E C1           3050 HALT        CALL   STOPS
C13B C3 00 C0           3070            JMP     ENDSTK
C13E                    3080 *
C13E                    4080 *  THIS ROUTINE INITIALIZES THE USART
C13E                    4090 * FOR AN INPUT OPERATION IN THE APPRO-
C13E                    4100 * PRIATE FORMAT (8 BITS, EVEN PARITY,
C13E                    4110 * 2 STOP BITS RECORD 1 STOP BIT PLAY)
C13E                    4120 *
C13E 3E BE              4130 STOPS       MVI    A,0BEH
C140 D3 03              4140             OUT    3
C142 3E 40              4150             MVI    A,40H
C144 D3 03              4160             OUT    3
C146 C9                 4170             RET
C1  E 7D  3E7D          4270 RSET        MVI    A,07DH
C1    03  D303          4280             OUT    3
C1    16  3E16          4290             MVI    A,16H
C14D D3 03              4300              OUT   3
C14F C9                 4310             RET
C150 3E FD              4320 TSET        MVI    A,0FDH
C152 D3 03              4330              OUT   3
C154 3E 31              4340             MVI    A,31H
C156 D3 03              4350             OUT    3
C158 C9                 4351              RET
C159                    4355 * THIS SUBROUTINE DUMPS MEMORY TO TAPE
C159                    4356 * IF THE END OF THE DUMP ADDRESS IS TERMINATED
C159                    4357 * WITH A '/' A FILE NAME IS ALSO APPENDED
C159                    4360 * AND THE MEMORY DUMPED IN FILE FORMAT.
C159                    4370 * THIS CONSISTS OF THE FILE NAME FOLLOWED
C159                    4380 * BY THE DUMP IN BINARY FOLLOWED BY THE
C159                    4390 * EOF (3 CNTL X'S)
C159                    4400 *
C159 CD 3E C1           4410 CNTLD      CALL    STOPS
C15C CD 7F C0           4415            CALL    TWOADR
C15F FE 2F              4420            CPI     '/'
C161 CA 88 C1           4430            JZ      FILE
C164 CD 50 C1           4440 DUMPER       CALL   TSET
C167 CD A4 C1           4460              CALL   WT
C16A 21 77 C1           4480 LKCS         LXI    H,ZILOG
C16D 0A                 4490 MORE       LDAX    B
C16E CD BE C1           4500            CALL    WRTP
C171 CD 9B C0           4510            CALL    IPCMP
C174 C3 6D C1           4520            JMP     MORE
C177 0E 03              4530 ZILOG      MVI     C,3
C179 3E 18              4540            MVI     A,18H       CNTL X
C17B CD BE C1           4550            CALL    WRTP
C17E 0D                 4560            DCR     C
C17F C2 7B C1           4570            JNZ     ZILOG+4
C182 CD A4 C1           4580            CALL    WT
C185 C3 38 C1           4590            JMP     HALT
C188 C5                 4600 FILE       PUSH    B
C189 06 06              4610            MVI     B,6
C18B CD C9 C1           4620            CALL    FRCHR
C18E 48                 4630            MOV     C,B
C18F CD 50 C1           4640            CALL    TSET
C192 CD A4 C1           4670            CALL    WT
C195 2E E1              4680            MVI     L,0E1H
C197 7E                 4690 FIN        MOV     A,M
C198 CD BE C1           4700            CALL    WRTP
C19B 23                 4710            INX     H
C19C 0D                 4720            DCR     C
C19D C2 97 C1           4730            JNZ     FIN
C1A0 C1                 4740            POP     B
C1A1 C3 6A C1           4750            JMP     MORE-3
C1A4 AF                 4760 WT         XRA     A
C1A5 [illegible]  2C    4770            INR     L
C1[illegible]  E3       4780            XTHL
C1[illegible]  3C       4790            XTHL
C1A8 C2 A5 C1           4820            JNZ     WT+1
C1AB 3C                 4830            INR     A
C1AC C2 A5 C1           4840            JNZ     WT+1
C1AF 3E 7F              4842 LEADER      MVI    A,7FH
C1B1 F5                 4844              PUSH   PSW
C1B2 3E FE              4846              MVI    A,0FEH
C1B4 CD BE C1           4848            CALL    WRTP
C1B7 F1                 4850             POP    PSW
C1B8 1F                 4852            RAR
C1B9 D0                 4854             RNC
C1BA C3 B1 C1           4856            JMP     LEADER+2
C1BD C9                 4858            RET
C1BE                    4860 *
C1BE                    4870 *  THIS SUBROUTINE WRITES THE ACC
C1BE                    4880 * ONTO THE TAPE.
C1BE                    4890 *
C1BE F5                 4900 WRTP       PUSH    PSW
C1BF DB 03              4910            IN      3
C1C1 1F                 4920            RAR
C1C2 D2 BF C1           4930            JNC     WRTP+1
C1C5 F1                 4940            POP     PSW
C1C6 D3 02              4950            OUT     2
C1C8 C9                 4960            RET
C1C9                    4970 *
C1C9                    4980 *  THIS SUBROUTINE READS IN FILE NAMES
C1C9                    4990 * FROM THE TERMINAL AND STORES THEM IN
C1C9                    5000 * THE STACK PAGE STARTING AT E1. THE
C1C9                    5010 * NNUMBER OF CHARACTERS TO BE READ IN
C1C9                    5020 * IS ASSUMED IN THE B REGISTER.
C1C9                    5030 *
C1C9 21 00 00           5040 FRCHR      LXI     H,0
C1CC 39                 5050            DAD     S
C1CD 2E E1              5060            MVI     L,0E1H
C1CF 48                 5070            MOV     C,B
C1D0 CD D9 C0           5080 LPR        CALL    RDCHR
C1D3 FE 20              5090            CPI     ' '
C1D5 CA D0 C1           5100            JZ      LPR
C1D8 0D                 5110            DCR     C
C1D9 77                 5120 NBLK       MOV     M,A
C1DA 23                 5130            INX     H
C1DB CD D9 C0           5140            CALL    RDCHR
C1DE 0D                 5150            DCR     C
C1DF C2 D9 C1           5160            JNZ     NBLK
C1E2 77                 5170            MOV     M,A
C1E3 C9                 5180            RET
C1E4                    5190 *
C1E4                    5200 *  THIS SUBROUTINE READS IN CHARACTERS
C1E4                    5210 * FROM THE TAPE. IT ALSO CHECKS TO SEE
C1E4                    5220 * IF THE KEYBOARD IS ACTIVE AND IF SO,
C1E4                    5230 * RETURNS FOR A NEW COMMAND. PARITY
C1E4                    5240 * ERROR ARE CAUGHT AND PRINTED ON THE
C1E4                    5250 * TERMINAL.
C1E4                    5260 *
C1E4 DB 00              5270 TPRD       IN      0
C1E6 1F                 5280            RAR
C1E7 D2 38 C1           5290            JNC     HALT
C1EA DB 03              5300            IN      3
C1EC 1F                 5310            RAR
C1ED 1F                 5320            RAR
C1EE D2 E4 C1           5330            JNC     TPRD
C1F1 E6 02              5340            ANI     2
C1F3 C4 F9 C1           5350            CNZ     ERROR
C1F6 DB 02              5360            IN      2
C1F8 C9                 5370            RET
C1F9 3E 58              5380 ERROR      MVI     A,'X'
C1FB D3 01              5390            OUT     01
C1FD C3 4B C1           5400            JMP     RSET+4
C200                    6000 PSW EQU 6
C200                    6010 S   EQU 6
```

```
?DC000 C1FF
C000 3E 55 21 FF 00 25 77 BE C2 05 C0 F9 CD 75 C0 3E
C010 3F CD E8 C0 CD D9 C0 FE 4C CA 3A C0 FE 44 CA 4B
C020 C0 FE 47 CA 70 C0 FE 53 CA F4 C0 FE 45 CA 59 C1
C030 21 00 C2 7E FF 55 C2 0C C0 E9 CD 84 C0 42 4B CD
C040 84 C0 7B 02 03 CD 61 C0 C3 3F C0 CD 7F C0 CD 65
C050 C0 0A CD BB C0 21 00 C0 CD 9B C0 CD 61 C0 C3 51
C060 C0 79 E6 0F C0 CD 75 C0 7B CD C2 C0 79 C3 C2 C0
C070 CD 84 C0 EB E9 3E 0D CD E8 C0 3E 0A C3 E8 C0 CD
C080 84 C0 42 4B 21 00 00 CD A2 C0 DA 84 C0 29 29 29
C090 29 B5 6F CD A2 C0 D2 8D C0 EB C9 03 7B 91 7A 98
C0A0 D0 E9 CD D9 C0 E6 7F FE 30 D8 FE 3A DA B8 C0 FE
C0B0 41 D8 FE 47 3F D8 D6 07 D6 30 C9 F5 3E 20 CD E8
C0C0 C0 F1 F5 1F 1F 1F 1F CD CB C0 F1 E6 0F C6 30 FE
C0D0 3A DA E8 C0 C6 07 C3 E8 C0 DB 00 1F DA D9 C0 DB
C0E0 01 E6 7F FE 18 CA 00 C0 F5 DB 00 E6 80 C2 E9 C0
C0F0 F1 D3 01 C9 06 06 CD C9 C1 E5 CD 84 C0 CD 3E C1
C100 CD 47 C1 E1 48 2E E1 CD E4 C1 BE C2 04 C1 23 0D
C110 C2 07 C1 3E 07 CD E8 C0 0E 03 CD E4 C1 FE 18 CA
C120 27 C1 12 13 C3 18 C1 12 13 0D C2 1A C1 78 FE 04
C130 CA 04 C1 FE FF CA 18 C1 CD 3E C1 C3 00 C0 3E BE
C140 D3 03 3E 40 D3 03 C9 3E 7D D3 03 3E 16 D3 03 C9
C150 3E FD D3 03 3E 31 D3 03 C9 CD 3E C1 CD 7F C0 FE
C160 2F CA 88 C1 CD 50 C1 CD A4 C1 21 77 C1 0A CD BE
C170 C1 CD 9B C0 C3 6D C1 0E 03 3E 18 CD BE C1 0D C2
C180 7B C1 CD A4 C1 C3 38 C1 C5 06 06 CD C9 C1 48 CD
C190 50 C1 CD A4 C1 2E E1 7E CD BE C1 23 0D C2 97 C1
C1A0 C1 C3 6A C1 AF 2C E3 E3 C2 A5 C1 3C C2 A5 C1 3E
C1B0 7F F5 3E FE CD BE C1 F1 1F D0 C3 B1 C1 C9 F5 DB
C1C0 03 1F D2 BF C1 F1 D3 02 C9 21 00 00 39 2E E1 48
C1D0 CD D9 C0 FE 20 CA D0 C1 0D 77 23 CD D9 C0 0D C2
C1E0 D9 C1 77 C9 DB 00 1F D2 38 C1 DB 03 1F 1F D2 E4
C1F0 C1 E6 02 C4 F9 C1 DB 02 C9 3E 58 D3 01 C3 4B C1
?
```

ROM I-b 3M3

```
C200                    0010 *  THIS IS THE PROGRAM FOR THE EXPANSION
C200                    0020 * ROM IN THE 2S10(R) BOARD. BY H.R.WALKER
C200                    0021 * ASSISTED BY DENNIS P. DUPRE'
C200 55                 0030            DB      55H
C201 DB 01              0032            IN      1
C203 E6 7F              0034            ANI     7FH
C205 FE 17              0036            CPI     17H     CNTRL W
C207 CA 53 C2           0037            JZ      PASCII
C     FE 54             0075            CPI     'T'
      CA 38 C3          0076            JZ      READT
      FE 57             0080            CPI     'W'
C211 CA 66 C2           0082            JZ      LASCII
C214 FE 46              0091            CPI     'F'     FFWD
C216 CA D9 C2           0092            JZ      FFWD
C219 FE 52              0093            CPI     'R'     REWIND
C21B CA F5 C2           0094            JZ      FREWD
C21E FE 14              0096            CPI     14H     CNTRL T
C220 CA 1C C3           0097            JZ      PTAPE
C223 FE 4B              0102            CPI     'K'
C225 CA 4D C3           0103            JZ      KCLOAD
C228 FE 0B              0104            CPI     0BH     CNTRL K
C22A CA 63 C3           0105            JZ      KCSAVE
C22D FE 13              0106            CPI     13H
C22F CA 78 C2           0107            JZ      CNTLS
C232 FE 48              0110            CPI     'H'     HUNT
C234 CA 7D C2           0111            JZ      HUNT
C237 FE 4F              0115            CPI     'O'
C239 CA EC C2           0116            JZ      ORUN
C23C FE 49              0120            CPI     'I'
C23E CA 79 C3           0125            JZ      INTEL
C241 FE 00              0130            CPI     00H
C243 CA 00 C0           0135            JZ      FNDSTK
C246 FE 00              0140            CPI     00H
C248 CA 00 C0           0145            JZ      FNDSTK
C24B FE 00              0150            CPI     00H
C24D CA 00 C0           0155            JZ      FNDSTK
C250 C2 0C C0           0170            JNZ     PEMON
C253                    0200 *  THIS IS THE ROUTINE TO PRINT
C253                    0210 * MEMORY IN ASCII ON THE TERMINAL.
C253                    0220 *
C253 CD 7F C0           0230 PASCII     CALL    TWOADR
C256 CD 75 C0           0240            CALL    CRLF
C259 21 00 C0           0245            LXI     H,FNDSTK
C25C 0A                 0250 HERE2      LDAX    B
C25D CD E8 C0           0260            CALL    WRCHR
C260 CD 9B C0           0280            CALL    IPCMP
C263 C3 5C C2           0290            JMP     HERE2
C266                    0300 *
C266                    0310 *  THIS IS THE ROUTINE TO ENTER
C266                    0320 * ASCII CHARACTERS INTO MEMORY.
C266                    0330 *
C266 CD 84 C0           0340 LASCII     CALL    RDNUM
C269 CD D9 C0           0350            CALL    RDCHR
C26C FE 5F              0360            CPI     5FH     BKARROW
C26E CA 74 C2           0370            JZ      ERASE
C271 12                 0380            STAX    D
C272 13                 0390            INX     D
C273 13                 0400            INX     D
C274 1B                 0410 ERASE      DCX     D
C275 C3 69 C2           0420            JMP     LASCII+3
C278                    0430 *
                        0440 * THIS IS THE REPEAT SEACH ROUTINE
                        0441 * SEARCHING FOR FOUR CHARACTERS
C   06 04               0450 CNTLS      MVI     B,4
C27A C3 F6 C0           0455            JMP     ESS+2
C27D                    0530 *
C27D                    0600 * THIS ROUTINE SEARCHES FOR AND
C27D                    0610 * LOADS A FILE THAT CONTAINS A
C27D                    0620 * FIVE LETTER STRING
C27D                    0630 * SEQUENCE.
C27D                    0640 *
C27D 06 05              0650 HUNT       MVI     B,5
C27F CD C9 C1           0660            CALL    FRCHR
C282 E5                 0670            PUSH    H
C283 CD 84 C0           0680            CALL    RDNUM
C286 E1                 0690            POP     H
C287 D5                 0700            PUSH    D
C288 E5                 0710            PUSH    H
C289 CD 3E C1           0720            CALL    STOPS
C28C CD 47 C1           0721            CALL    RSET
C28F 2E 03              0730 RETRY      MVI     L,3
C291 48                 0740            MOV     C,B
C292 E3                 0750            XTHL
C293 2E E1              0760            MVI     L,0E1H
C295 E3                 0770            XTHL
C296 CD E4 C1           0780 LOOPS      CALL    TPRD
C299 12                 0790            STAX    D
C29A 13                 0800            INX     D
C29B E3                 0810            XTHL
C29C BE                 0820            CMP     M
C29D E3                 0830            XTHL
C29E C2 C0 C2           0840            JNZ     TSTCX
C2A1 E3                 0850            XTHL
C2A2 23                 0860            INX     H
C2A3 E3                 0870            XTHL
C2A4 0D                 0880            DCR     C
C2A5 C2 96 C2           0890            JNZ     LOOPS
C2A8 3E 07              0900            MVI     A,7             BELL
C2AA CD E8 C0           0910            CALL    WRCHR
C2AD 0E 03              0920 LOADER     MVI     C,3
C2AF CD E4 C1           0930            CALL    TPRD
C2B2 12                 0940            STAX    D
C2B3 13                 0950            INX     D
C2B4 FE 18              0960            CPI     18H             CNTL X
C2B6 C2 AD C2           0970            JNZ     LOADER
C2B9 0D                 0980            DCR     C
C2BA C2 AF C2           0990            JNZ     LOADER+2
C2BD C3 38 C1           1000            JMP     HALT
C2C0 FE 18              1010 TSTCX      CPI     18H             CNTL X
C2C2 C2 8F C2           1020            JNZ     RETRY
C2C5 2D                 1030            DCR     L
C2C6 C2 91 C2           1040            JNZ     RETRY+2
C2C   E1                1050 EOF        POP     H
      D1                1060            POP     D
      D5                1070            PUSH    D
C2C   E5                1080            PUSH    H
C2CD C3 8F C2           1090            JMP     RETRY
```

```
C2D0                    1203 * THIS ROUTINE CAUSES THE CARTRIDGE TO
C2D0                    1204 * RUN FULL SPEED AND PLAY WHILE RUNNING
C2D0                    1205 * TO SEARCH FOR INTER RECORD GAPS.
C2D0 3E BE              1220 FAST       MVI     A,0BEH
C2D2 D3 03              1221            OUT     3
C2D4 3E 32              1222            MVI     A,32H
C2D6 D3 03              1225            OUT     3
C2D8 C9                 1230            RET
C2D9                    1235 * THIS IS THE FAST FORWARD ROUTINE.
C2D9 CD D0 C2           1240 FFWD       CALL    FAST
C2DC 3E 16              1270            MVI     A,16H
C2DE D3 03              1280            OUT     3
C2E0 DB 00              1290            IN      STS
C2E2 E6 01              1292            ANI     01H
C2E4 C2 E0 C2           1295            JNZ     FFWD+7
C2E7 DB 02              1297            IN      02
C2E9 C3 38 C1           1300            JMP     HALT
C2EC CD 3E C1           1305 ORUN       CALL    STOPS
C2EF CD 47 C1           1306            CALL    RSET
C2F2 C3 E0 C2           1307            JMP     FFWD+7
C2F5                    1315 * THIS IS THE REWIND ROUTINE.
C2F5 CD D0 C2           1320 FREWD      CALL    FAST
C2F8 3E 34              1350            MVI     A,034H
C2FA D3 03              1360            OUT     3
C2FC DB 00              1370            IN      STS
C2FE E6 01              1372            ANI     01H
C300 C2 FC C2           1373            JNZ     FREWD+7
C303 DB 02              1375            IN      02
C305 C3 38 C1           1380            JMP     HALT
C308                    1400 * THIS ROUTINE PROVIDES 8 RUBOUTS
C308                    1401 * FOR THE BINARY TAPE DUMP
C308 3E 7F              1410 RUBOUT     MVI     A,7FH
C30A F5                 1445            PUSH    PSW
C30B DB 00              1450 LUUP       IN      00H
C30D E6 80              1460            ANI     80H
C30F C2 0B C3           1470            JNZ     LUUP
C312 3E FF              1480            MVI     A,0FFH
C314 D3 01              1485            OUT     01H
C316 F1                 1490            POP     PSW
C317 1F                 1492            RAR
C318 D0                 1494            RNC
C319 C3 0A C3           1495            JMP     RUBOUT+2
C31C                    1505 * THIS ROUTINE PUNCHES TAPE IN
C31C                    1506 * THE BINARY FORMAT WITH 8 RUBOUTS
C31C                    1507 * AT THE BEGINNING AND END.
C31C CD 7F C0           1510 PTAPE      CALL    TWOADR
C31F CD 08 C3           1520            CALL    RUBOUT
C322 21 28 C3           1525            LXI     H,PTAPE+12
C325 CD 5C C2           1530            CALL    HERE2
C328 CD 08 C3           1540            CALL    RUBOUT
C32B C3 00 C0           1550            JMP     FNDSTK
C32E DB 00              1575 CHRRD      IN      STS
C330 E6 01              1578            ANI     01H
C332 C2 2E C3           1580            JNZ     CHRRD
C335 DB 01              1583            IN      1
C337 C9                 1584            RET
C338                    1585 * THIS ROUTINE READS PAPER TAPE IGNORING
C338                    1586 * RUBOUTS AT START.
C338 CD 84 C0           1590 READT      CALL    RDNUM
C33B 62                 1600            MOV     H,D
C33C 6B                 1610            MOV     L,E
C33D CD 2E C3           1620            CALL    CHRRD
C340 FE FF              1623            CPI     0FFH
C342 CA 3D C3           1626            JZ      READT+5
C345 77                 1630            MOV     M,A
C346 23                 1640            INX     H
C347 CD 2E C3           1645            CALL CHRRD
C34A C3 45 C3           1650            JMP     READT+13
```

ROM II - a 3M3

```
C34D                    1651 * THESE ROUTINES ENABLE THE USER
C34D                    1652 * TO SET UP THE USART IN ANY MODE
C34D                    1653 * AND COMMAND TO LOAD PROGRAMS IN
C34D                    1654 * FORMATS SYNCHRONOUSLY OR
C34D                    1655 * ASYNCHRONOUSLY--SUCH AS THE
 34D                    1656 * KANSAS CITY STANDARD TAPES WITH
 34D                    1657 * AUDIO CASSETTE INTERFACE.
C34D CD 3E C1           1660 KCLOAD    CALL  STOPS
C350 CD 84 C0           1680           CALL  RDNUM
C353 D5                 1690           PUSH  D
C354 CD 84 C0           2010           CALL  RDNUM
C357 7A                 2020           MOV   A,D
C358 D3 03              2030           OUT   3
C35A 7B                 2040           MOV   A,E
C35B D3 03              2050           OUT   3
C35D D1                 2080           POP   D
C35E 06 FF              2090           MVI   B,0FFH
C360 C3 18 C1           2100           JMP   CONTIN
C363 CD 3E C1           2110 KCSAVE    CALL  STOPS
C366 CD 7F C0           2130           CALL  TWOADR
C369 C5                 2140           PUSH  B
C36A D5                 2160           PUSH  D
C36B CD 84 C0           2180           CALL  RDNUM
C36E 7A                 2190           MOV   A,D
C36F D3 03              2200           OUT   3
C371 7B                 2210           MOV   A,E
C372 D3 03              2220           OUT   3
C374 D1                 2240           POP   D
C375 C1                 2245           POP   B
C376 C3 6A C1           2260           JMP   LKCS
C379 CD D9 C0           2300 INTEL     CALL  RDCHR
C37C FE 3A              2310           CPI   ':'
C37E C2 79 C3           2320           JNZ   INTEL
C381 CD A1 C3           2330           CALL  GTBIT
C384 B7                 2340           ORA   A
C385 CA C9 C2           2350           JZ    EOF
C388 47                 2360           MOV   B,A
C389 CD A1 C3           2370           CALL  GTBIT
C38C 67                 2380           MOV   H,A
C38D CD A1 C3           2390           CALL  GTBIT
C390 6F                 2400           MOV   L,A
C391 CD A1 C3           2410           CALL  GTBIT
C394 CD A1 C3           2420 LOOP2     CALL  GTBIT
C397 77                 2430           MOV   M,A
C398 23                 2440           INX   H
C399 05                 2450           DCR   B
C39A C2 94 C3           2460           JNZ   LOOP2
C39D C3 79 C3           2470           JMP   INTEL
C3A0 76                 2480 ENDF      HLT
C3A1 CD AE C3           2490 GTBIT     CALL  INDIG
C3A4 87                 2500           ADD   A
C3A5 87                 2510           ADD   A
C3A6 87                 2520           ADD   A
C3A7 87                 2530           ADD   A
C3A8 57                 2540           MOV   D,A
C3A9 CD AE C3           2550           CALL  INDIG
C3AC B2                 2560           ORA   D
 3AD C9                 2570           RET
 3AE CD D9 C0           2580 INDIG     CALL  RDCHR
C3B1 FE 3A              2590           CPI   '9'+1
C3B3 FA B8 C3           2600           JM    INDI
C3B6 C6 09              2610           ADI   9
C3B8 E6 0F              2620 INDI      ANI   0FH
C3BA C9                 2630           RET
C3BB                    9000 FNDSTK    EQU   0C000H
C3BB                    9010 TWOADR    EQU   FNDSTK+127
C3BB                    9014 STS       EQU   00H
C3BB                    9015 TBE       EQU   80H
C3BB                    9020 CRLF      EQU   FNDSTK+117
C3BB                    9030 WRCHR     EQU   FNDSTK+232
C3BB                    9040 PEMON     EQU   FNDSTK+12
C3BB                    9050 IPCMP     EQU   FNDSTK+155
C3BB                    9060 RDNUM     EQU   FNDSTK+132
C3BB                    9070 RDCHR     EQU   FNDSTK+217
C3BB                    9080 WT        EQU   FNDSTK+420
C3BB                    9090 HALT      EQU   FNDSTK+312
C3BB                    9100 TPRD      EQU   FNDSTK+484
C3BB                    9110 PSW       EQU   6
C3BB                    9120 FRCHR     EQU   FNDSTK+457
C3BB                    9130 CONTIN    EQU   FNDSTK+280
C3BB                    9140 FIN       EQU   FNDSTK+407
C3BB                    9150 TRYAGN    EQU   FNDSTK+259
C3BB                    9160 LKCS      EQU   FNDSTK+362
C3BB                    9170 ESS       EQU   FNDSTK+244
C3BB                    9180 STOPS     EQU   FNDSTK+318
C3BB                    9190 RSET      EQU   FNDSTK+327
C3BB                    9200 TSET      EQU   FNDSTK+336
```

```
?DC200 C3FF
C200 55 DB 01 E6 7F FE 17 CA 53 C2 FE 54 CA 38 C3 FE
C210 57 CA 66 C2 FE 46 CA D9 C2 FE 52 CA F5 C2 FE 14
C220 CA 1C C3 FE 4B CA 4D C3 FE 0B CA 63 C3 FE 13 CA
C230 78 C2 FE 48 CA 7D C2 FE 4F CA EC C2 FE 49 CA 79
C240 C3 FE 00 CA 00 C0 FE 00 CA 00 C0 FE 00 CA 00 C0
C250 C2 0C C0 CD 7F C0 CD 75 C0 21 00 C0 0A CD E8 C0
C260 CD 9B C0 C3 5C C2 CD 84 C0 CD D9 C0 FE 5F CA 74
C270 C2 12 13 13 1B C3 69 C2 06 04 C3 F6 C0 06 05 CD
C280 C9 C1 E5 CD 84 C0 E1 D5 E5 CD 3E C1 CD 47 C1 2E
C290 03 48 E3 2E E1 E3 CD E4 C1 12 13 E3 BE E3 C2 C0
C2A0 C2 E9 23 E3 0D C2 96 C2 3E 07 CD E8 C0 0E 03 CD
C2B0 F4 C1 12 13 FE 18 C2 AD C2 0D C2 AF C2 C3 38 C1
C2C0 FE 18 C2 8F C2 2D C2 91 C2 E1 D1 D5 E5 C3 8F C2
C2D0 3E BE D3 03 3E 32 D3 03 C9 CD D0 C2 3E 16 D3 03
C2E0 DB 00 E6 01 C2 E0 C2 DB 02 C3 38 C1 CD 3E C1 CD
C2F0 47 C1 C3 E0 C2 CD D0 C2 3E 34 D3 03 DB 00 E6 01
C300 C2 FC C2 DB 02 C3 38 C1 3E 7F F5 DB 00 E6 80 C2
C310 0B C3 3E FF D3 01 F1 1F D0 C3 0A C3 CD 7F C0 CD
C320 08 C3 21 28 C3 CD 5C C2 CD 08 C3 C3 00 C0 DB 00
C330 E6 01 C2 2E C3 DB 01 C9 CD 84 C0 62 6B CD 2E C3
C340 FE FF CA 3D C3 77 23 CD 2E C3 C3 45 C3 CD 3E C1
C350 CD 84 C0 D5 CD 84 C0 7A D3 03 7B D3 03 D1 06 FF
C360 C3 18 C1 CD 3E C1 CD 7F C0 C5 D5 CD 84 C0 7A D3
C370 03 7B D3 03 D1 C1 C3 6A C1 CD D9 C0 FE 3A C2 79
C380 C3 CD A1 C3 B7 CA C9 C2 47 CD A1 C3 67 CD A1 C3
C390 6F CD A1 C3 CD A1 C3 77 23 05 C2 94 C3 C3 79 C3
C3A0 76 CD AE C3 87 87 87 87 57 CD AE C3 B2 C9 CD D9
C3B0 C0 FE 3A FA B8 C3 C6 09 E6 0F C9 00 00 00 00 00
C3C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C3D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C3E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C3F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
?
```

DETAILED OPERATION DESCRIPTION

General

The complete functional definition of the 8251 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize the 8251 to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD PARITY etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251 is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251 is ready to receive a character. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251. On the other hand, the 8251 receives serial data from the MODEM or I/O device, upon receiving an entire character the RxRDY output is raised "high" to signal the CPU that the 8251 has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU read operation.

The 8251 cannot begin transmission until the TxEN (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send (CTS) input. The TxD output will be held in the marking state upon Reset.

Programming the 8251

Prior to starting data transmission or reception, the 8251 must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251 and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

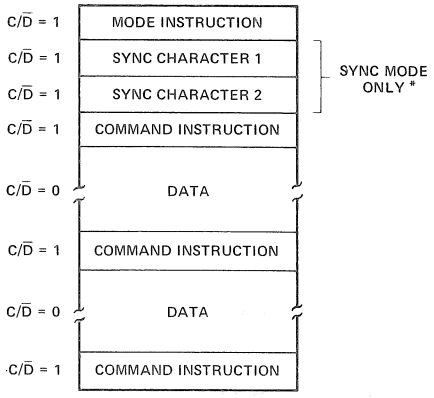
This format defines the general operational characteristics of the 8251. It must follow a Reset operation (internal or external). Once the Mode instruction has been written into the 8251 by the CPU, SYNC characters or Command instructions may be inserted.

Command Instruction

This format defines a status word that is used to control the actual operation of the 8251.

Both the Mode and Command instructions must conform to a specified sequence for proper device operation. The Mode Instruction must be inserted immediately following a Reset operation, prior to using the 8251 for data communication.

All control words written into the 8251 after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251 at any time in the data block during the operation of the 8251. To return to the Mode Instruction format a bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251 back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.



*The second SYNC character is skipped if MODE instruction has programmed the 8251 to single character Internal SYNC Mode. Both SYNC characters are skipped if MODE instruction has programmed the 8251 to ASYNC mode.

Typical Data Block

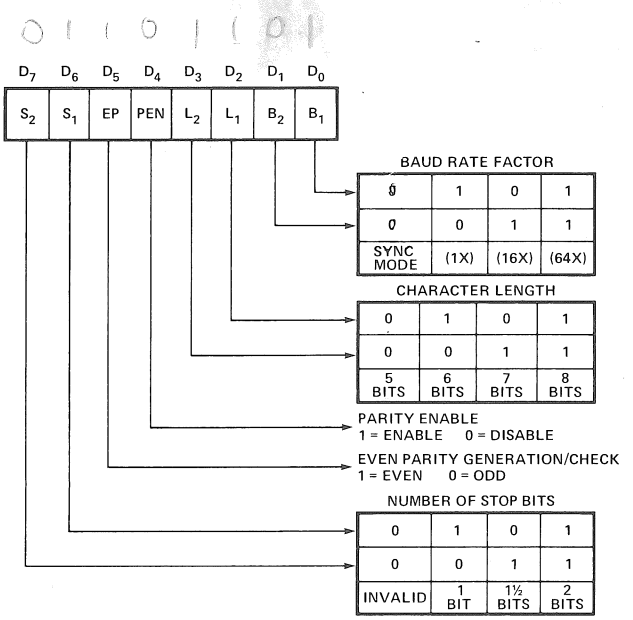
Mode Instruction Definition

The 8251 can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251 the designer can best view the device as two separate components sharing the same package. One Asynchronous the other Synchronous. The format definition can be changed "on the fly" but for explanation purposes the two formats will be isolated.

Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251 automatically adds a Start bit (low level) and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

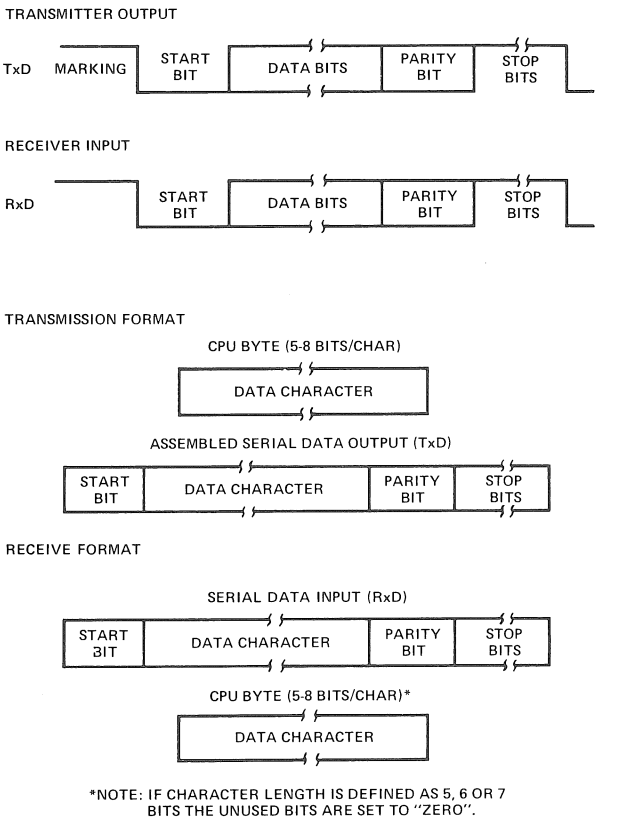
When no data characters have loaded into the 8251 the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.



Mode Instruction Format, Asynchronous Mode

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center. If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. This character is then loaded into the parallel I/O buffer of the 8251. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN flag is raised (thus the previous character is lost). All of the error flags can be reset by a command instruction. The occurrence of any of these errors will not stop the operation of the 8251.



Asynchronous Mode

Synchronous Mode (Transmission)

The TxD output is continuously high until the CPU sends its first character to the 8251 which usually is a SYNC character. When the $\overline{CTS}$ line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251 with a character before the 8251 becomes empty, the SYNC characters (or character if in single SYNC word mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin is raised high to signal that the 8251 is empty and SYNC characters are being sent out. The TxEMPTY pin is internally reset by the next character being written into the 8251.

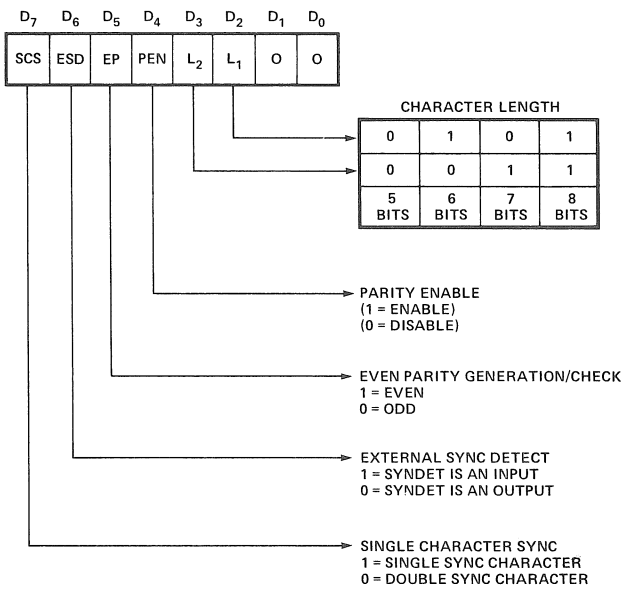
Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the internal SYNC mode has been programmed, the receiver starts in a HUNT mode. Data on the RxD pin is then sampled in on the rising edge of $\overline{RxC}$. The content of the Rx buffer is continuously compared with the first SYNC character until a match occurs. If the 8251 has been programmed for two SYNC characters, the subsequent received character is also compared; when both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ.

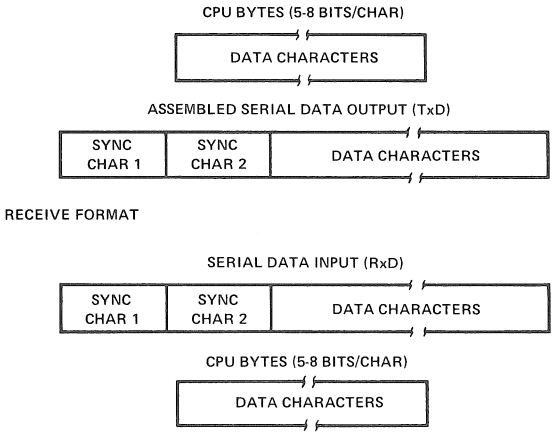
In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin. The high level can be removed after one $\overline{RxC}$ cycle.

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost.



Mode Instruction Format, Synchronous Mode

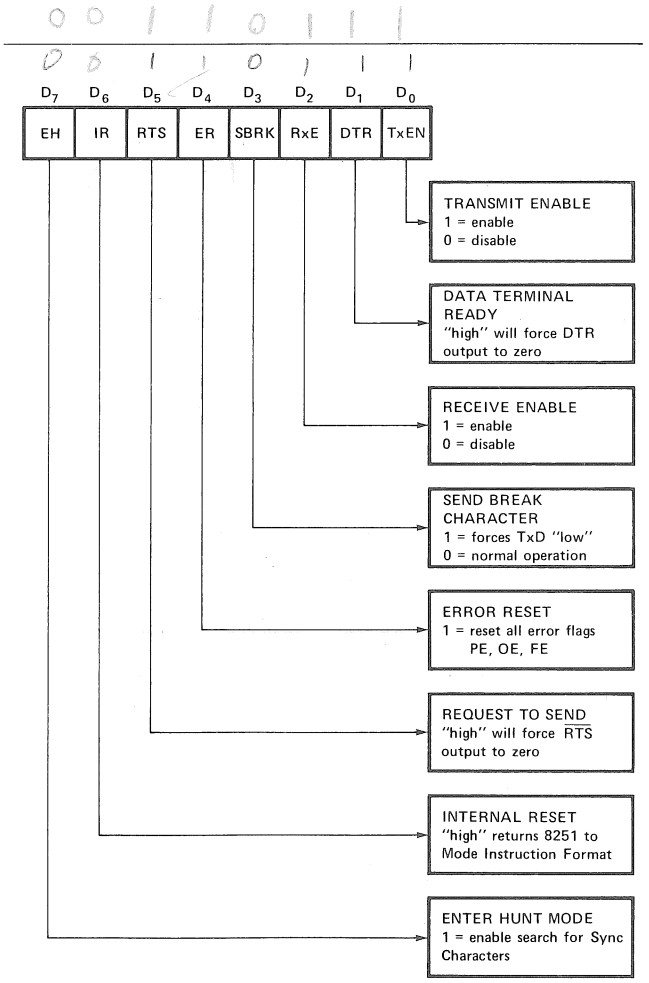


Synchronous Mode, Transmission Format

COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251 has been programmed by the Mode Instruction and the Sync Characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251 and Sync characters inserted, if necessary, then all further "control writes" (C/$\overline{D}$ = 1) will load the Command Instruction. A Reset operation (internal or external) will return the 8251 to the Mode Instruction Format.



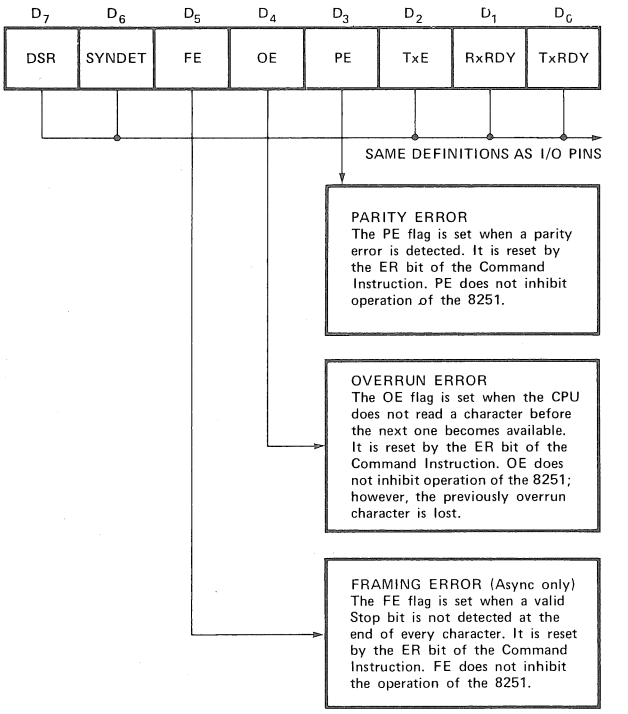
Command Instruction Format

STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251 has facilities that allow the programmer to "read" the status of the device at any time during the functional operation.

A normal "read" command is issued by the CPU with the C/D input at one to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251 can be used in a completely Polled environment or in an interrupt driven environment.



Status Read Format