JEN
DA
ELECTRONICS

System Central Interface SCI

SCI Specifications

3 independent parallel ports, each bit programmable input or latched output. Handshake signals for each port.

Serial port with RS-232, 20ma, 60ma current loops, speed selectable between 45 and 9600 baud.

High-speed biphase cassette interface with selectable speeds from 800 to 100,000 baud. Can also be used as a signal generator to test audio equipment.

Two on-board relays control two recorders. Three status lines for external recorder control.

Reset-jump: on reset, the CPU jumps to the SCI monitor.

2708 programmer.

256 bytes of RAM for stack and buffer space.

2k monitor in 2708's provides system control and I/O.

SCI requires the following S-100 bus signals:

+8 volts, +16 volts, -16 volts, circuit ground

$\overline{\text{PWR}}$, PDBIN, $\overline{\text{PHOLD}}$, $\overline{\text{PRESET}}$, PSYNC

SINP, SOUT

AO-A15, DIO-DI7, DOO-DO7

2 mhz clock

Power Requirements: (nominal)

+8 -	450 ma
+16-	85 ma
-16-	65 ma

Dimensions: 10.0 " wide x 5.1 " high (excluding edge connector)

SCI generates its own wait states, independent of the processor used, and can use 650ns 2708's.

SYSTEM CENTRAL INTERFACE

Table of Contents

SCI Assembly

A note of caution: This is an expensive interface. To protect your investment, and ensure a long operating life, exercise care in the construction of this board. Take your time, be certain of parts placement, don't force I.C.'s into their sockets, solder thoroughly, but sparingly, as a little solder goes a long way. Handle the I.C.'s carefully, making sure that the pins are straight before inserting them into their sockets. I.C. pins can break, and lead receptacles damaged, by careless insertion. Take the time to inspect and test the finished board before applying power. This avoids the smoke test.

Personal care: wear eye protection when cutting leads and soldering. This prevents foreign material from entering your eyes.

Use a metallic holder for the soldering iron. This prevents heat-damaged tables, possible fires, and carpet burns when soldering irons get loose. The soldering iron need only be 25 watts with a 1/8 inch spade tip. A welder is not necessary. An iron with a three-wire cord is preferable for electrical safety. An Ungar #135 is an excellent iron for p.c. work, and costs about $8.00. Rosin-core solder is supplied with the kit. If it's not enough, use 60/40 lead-tin rosin-core solder only. Acid-core solder will destroy the circuit and void the warranty.

All parts mount on the FRONT side, and solder on the BACK. Solder the parts after each step.

□ Check the parts list and the parts layout. These should agree with the parts supplied. Contact us immediately if anything is missing, as shipping takes time.

□ Locate the 7-pole DIP switch. Mount it in the upper left corner of the board. Mount it with the numbers toward the top of the board.

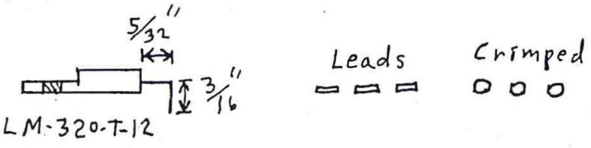
□ Note the orientation of the I.C.'s: For vertical I.C.'s, pin 1 is to the upper left, and for horizontal I.C.'s, pin 1 is to the upper right.

□ Install the three standard profile 24-pin sockets in pads 17,18, and 19. These are the sockets for the 2708's.

Note that I.C.pad 6 is the address jumper pad and does not have a socket.

□ Install the remaining sockets.

□ Install the four regulators and the two heat sinks. Line up the heat sinks over the areas marked on the p.c. board. Bend the leads of the regulators at right angles as shown on the parts list. The holes for the LM-320-T-12 were small on early production boards, and the leads of the regulator should be bent and cut approximately as shown, then crimp the leads slightly with pliers so they will fit the p.c. holes. Use 4-40 nuts and bolts to secure the regulators.



□ Next, install the capacitors. Observe the polarity of the 16 tantalum capacitors: the + lead must line up with the + shown in the layout and marked on the p.c. board.

The - lead for the tantalum capacitor on the right-hand side of the board must be bent inward, to avoid hitting the guide rails of the card rack.

□ Install the diodes, being careful to align the cathode bands with the markings on the p.c. board and shown in the layout drawing.

All resistors mount vertically:

□ Install the 1/2 watt resistors. These are about 50% larger than the 1/4 watt resistors. All are near IC-45. Note: There is an extra 51 ohm 1/2 watt, used in the 60 ma serial.

□ Install the remaining resistors.

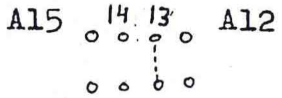
□ There are two types of small-signal transistors used: 2N3904 and 2N3906. Be sure to place them at the correct locations, and oriented as shown in the layout. Spread the leads slightly to fit the hole patterns.

□ The MJE-180 transistor in the upper right-hand corner of the board, has leads that must be crimped slightly to fit the mounting holes. Trim the leads, then crimp with pliers to fit.

□ Insert the LED's, noting that the flat edge is the cathode. Check the parts list for details.

Jumper Options

□ Strap the address jumpers for the board address you specified when ordering. If you didn't specify, D000 hex is the standard address in the firmware supplied by DAJEN.



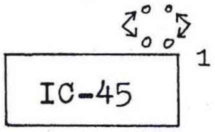
Address pad, D000 shown

ADDRESS	Jumper(s)	ADDRESS	Jumper(s)
8000 hex	A-12,13,14	C000 hex	A-12,13
9000	A-13,14	D000	A-13
A000	A-12,14	E000	A-12
B000	A-14	F000	None

Lower addresses (6000,7000) could be used, but are not recommended, as system RAM usually occupies the first 32 K of memory space, if not more.

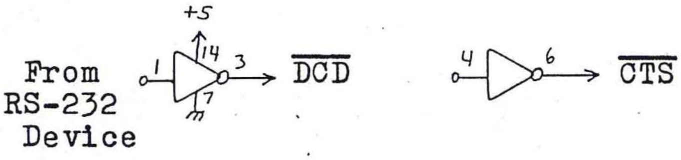
Note: I/O addresses use 16 consecutive ports, with the same byte as the board address. For a board address of D000, the I/O addresses are: D0,D1,D2,...DE,DF. For a board address of A000, the I/O addresses are: A0, A1, A2.. ..AE, AF.

□ The two jumpers in the upper right-hand corner next to pins 1 and 2 of IC pad 45, the serial port, should be installed if there are no immediate plans for connection to a Modem.



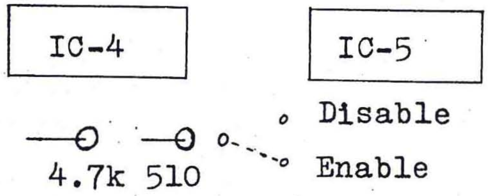
Jumpers for $\overline{CTS}$, $\overline{DCD}$

When used, $\overline{CTS}$ (Clear To Send) and $\overline{DCD}$ (Data Carrier Detect) must be converted to RS-232 levels (they are TTL compatible) using a 1489 or equivalent RS-232 line receiver.



Typical 1489 circuit

Reset-Jump Jumper



IC-4 and 5 are located in the lower left portion of the SCI board.

The reset-jump circuit takes control of the S-100 bus when the reset line (pin 75 on the bus) is active (low). No other jump circuits should be active, as they will probably conflict and the system will hang-up.

This circuit is enabled by placing a jumper from the reset-jump pad to the enable pad. It is disabled by placing a jumper to the disable pad.

The address to which the processor jumps is determined by the first three bytes of the 2708 in IC position 17 (PØ). With the SCI monitor, the jump address is directed to the initialization routine of the monitor, and is typically:

```
HØØØ  C3  Jump instruction for CPU
HØØ1  1E  Low-order jump address
HØØ2  HØ  High-order jump address
```

The manner in which the SCI jump circuit is set up requires that the CPU jump to a location within the first 1k of the SCI board address to reset the jump circuit. This design eliminates false reset of the reset-jump circuit when the SCI is addressed at FØØØ. Thus, if other than the SCI monitor is used, and a jump outside the first 1k of the SCI memory block is desired, a second jump instruction must be used, and could be placed immediately following the first jump instruction.

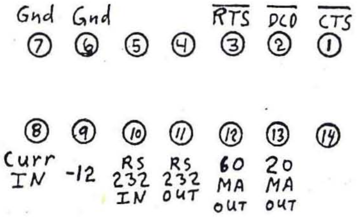
Example: SCI addressed for DØØØ-DFFF, desired jump address is to EØØØ:

```
DØØØ C3 Ø3 DØ first jump instruction
DØØ3 C3 ØØ EØ second jump instruction
```

If a Modem is to be used, the $\overline{CTS}$ and $\overline{DCD}$ (Clear to Send, Data Carrier Detect) from the 1489 will connect to pins 1 and 2 on the interface plug. If a Modem with CTS and DCD is to be used interchangably with a device not sending those signals, wire one I/O plug for the Modem, and another for the other device, connecting pins 1 and 2 to pin 6 on the plug for the other device.

RS-232 or Current Loop

Both an RS-232 device and a current-loop device can be connected to the SCI at the same time. Each keyboard can independently provide input, but each printer will print the same output. The devices must operate at the same baud rate. If only one device is used, the other input must be jumpered on the 14 pin connection plug.



Pad 45

Serial Port Connections
$\overline{RTS}$ is a TTL level output

If a current loop device is used, connect the printer + lead to pin 13 for a 20 ma loop, or to pin 12 for a 60 ma loop, and the other lead to pin 6. The keyboard contacts connect to pins 8 (+) and 7 . The RS-232 input must be jumpered from pin 10 to pin 9.

If an RS-232 device is used, connect the printer lead to pin 11, and the keyboard to pin 10, with common signal lead to pin 7. De-activate the current loop keyboard contact with a jumper from pin 8 to pin 7.

Remember, when only one keyboard device is used on the serial port, the other must be de-activated with an appropriate jumper.

It is recommended that coaxial cable such as RG-174 be used for connection to the serial devices, as the shielding prevents extraneous noise pick-up.

EIA RS-232 Connections

The following page lists the pin numbers used on the EIA RS-232 standard connections. RS-232 is normally used on Modems, and data terminals, for high-speed serial data transfers. The standard connector used is a 25 pin EIA designated type. The use of cables under 50 feet is recommended. If longer cables are used, the total capacitance of the cable and terminator must not exceed 2500 picofarads (micro-micro-farad). Normal cable capacitance is about 30 picofarads per foot.

EIA Connections

The following are the standard connections used with a 25-pin EIA port connector (RS-232).

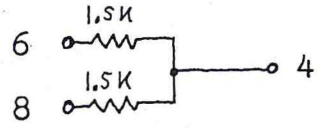
EIA pin	Signal	SCI pin on pad 45	Description
1	Frame ground	--	Connect to chassis ground (Tie to bolt of EIA socket).
2	Terminal Out	10	Data from terminal keyboard.
3	Terminal In	11	Data to terminal printer.
4	Request to Send	--	Request by terminal to send.
5	Clear to Send	--	Send data.
6	Data Set Ready	--	To terminal. If needed, tie to +12 volts through a 1.5k resistor.
7	Signal ground	7	ground, common signal.
8	Received line signal detector		If needed, tie to +12 volts via 1.5k resistor.
9-16	Not used		
17	TTY output	8	Keyboard of TTY serial loop. (Tie to pin 6 if not used.)
18-22	No connection		
24	TTY output	6	ground for TTY loop.
23	TTY input	13	20ma current loop to TTY printer.
25	TTY input	6	ground reference for loop.

If the RS-232 input is not used (SCI pin 10), tie to -12 volts on pin 9.

If the serial input to the SCI is not used, tie pin 8 to pin 6 or 7.

Resistor connections to +12 volts if Data Set Ready or Received Line Signal Detector are needed by terminal:

EIA pin SCI pin (connect to +12 with a jumper).



On SCI, jumper pin 4 (IC pad 45) to pin 8 of IC-38, the LM-358, or TL-082.

Keyboard and Parallel Port Wiring

All three parallel ports are wired identically, except that IC pad 44, the "Keyboard" port, has -12 volts supplied on pin 1, for keyboards that require a negative. bias. Otherwise the ports are identical, and the software determines which ports are input or output. The current available on the +5 volt supply is mainly a function of the regulator temperature. If forced air cooling is used, the regulator will remain cool, and a total of 500 ma is available. If convection cooling is used, it is recommended that total current be held to 100 ma.

If a keyboard is used that has only seven data lines, don't use D7, leave it open.

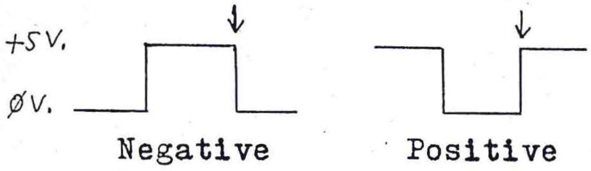
Gnd	D6	D7	SO	SI		-12 (44)
7	6	5	4	3	2	1
8	9	10	11	12	13	14
D5	D4	D3	D2	D1	DØ	+5

Parallel Ports, 42, 43, 44

DØ-D7= Data lines

SI= Strobe in SO= Strobe out

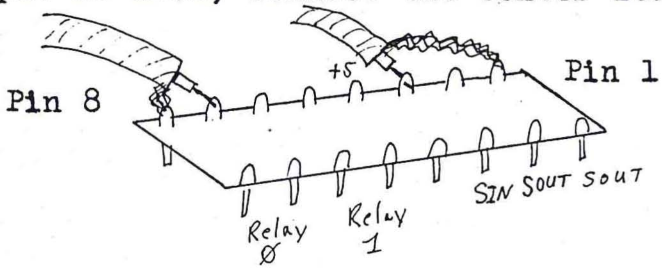
Strobe polarity is determined by software, and is normally negative. The 2k monitor supplied looks at DIP switch 7 to determine the strobe of the Keyboard port. If switch 7 is on, the monitor expects a negative strobe. If the switch is off or "Open", the monitor expects a positive strobe.



Typical Strobe Pulses

Cassette Cable Connections

□ Cut the audio cable into two equal lengths. These ends then solder to appropriate pins of a 16 pin DIP plug for record and playback connections. If a standard cassette recorder is being used, connect the lead from the speaker jack to pins 7 and 8 on the plug (pin 8, is the shield connection). If a high-impedance output from a deck is used, connect the center lead to pin 6. For a microphone input, connect to pins 1 and 2 (shield to pin 1). If an AUX input is used, connect the center lead to pin 3.



Cassette Connections
Spkr, AUX shown

Cassette Control

□ If you plan to use two recorders, connect the Read recorder remote control line to pins 9 and 10 on the plug. Connect the Write recorder remote line to pins 11 and 12. When reading a cassette tape, the Read recorder will be activated, and when writing a tape, the Write recorder will be active.

If you will be using one recorder for both reading and writing, wire pin 9 to pin 11, and pin 10 to pin 12. Then connect the recorder remote line to pins 11 and 12, and it will be active when reading and writing.

When using more sophisticated cassette control, pins 14,15, and 16, provide additional control and status lines. Normally, with the monitor supplied, pin 14 is set for status input , for turns counters, or other status info, and pins 15 & 16 output for additional control, connecting to a transistor relay driver or equivalent. These lines can be set up for other configurations, such as all outputs or two outputs and one input, by changing the initialization routine as desired. Complete control of an automatic deck, such as a Phi-Deck, can be easily implemented, to control forward, reverse, read, and write. One status output controls Fast-Forward, or Rewind, one status output controls Read-Write, one on-board relay turns power on and off to the drive motors, and the other relay controls the Playback-Record solenoid. The status input counts the turns of the tape reel to locate files. Further information is forth-coming.

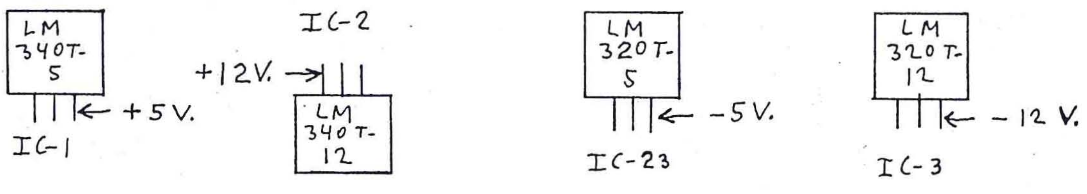
Check-out

Before inserting the I.C.'s or applying power, inspect the finished board. Are all connections soldered, are there any solder bridges ? Are any leads touching other leads ? Solder bridges are not likely to occur with the solder-masked p.c. board. If they do, draw the solder away with the tip of the iron. Compare the orientation of capacitors, diodes, transistors, and sockets, with the layout drawing. Correct any discrepancies. Excess solder flux can be removed with Miller-Stephenson Heavy Duty Flux Remover. Alternatively, isopropyl alcohol and a clean rag will remove flux. Soak a corner of the rag in the alcohol and wipe the flux off, in a direction away from the edge connector, as flux remaining on the connector fingers could cause intermittent connection.

After inspecting and cleaning the board, use a VOM (Volt Ohm Meter) set to resistance scale X 100, to check the resistance of all fingers of the edge connector to ground, pins 50 and 100, and to +5 volts on board (pin 16 of IC 5 or IC 7). This eliminates the destruction of bus drivers in the computer if a problem exists. Most fingers will have infinite resistance.. Zero, or low resistance indicates a problem. Only pins 2 (+16), 52 (-16), and Reset pin 75 will have any resistance indication. Even here, the resistance should be greater than 200 ohms. Of course, pins 1 and 51 the +8 supply will show resistance to ground and to the on-board +5 supply. If there is an abnormal indication, or short, follow the p.c. trace from that finger and look for a solder bridge or an etch problem. Repair the problem.

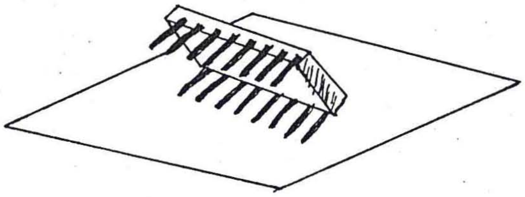
Caution: never insert or remove a board in the system bus with the power on. Always turn the power off, and wait half a minute for the power supply capacitors to discharge.

After a satisfactory resistance check, plug the SCI into the system bus and apply power. Measure the voltages out of the four regulators. The values should be within 10 % of the nominal value shown:



If any voltage measurement is not within range, there may be a problem with the regulator, associated bypass capacitor, a short in the p.c. power trace, or inadequate supply voltage. If the regulator is getting very hot, there is probably a short. Check that the associated tantalum bypass capacitors are correctly installed, the + lead is aligned with the + mark on the P.C. board. Or there is a short somewhere from the output trace to ground. If the regulator is cool to the touch, check the supply voltage to the regulator. If the supply is ok (+8,+16, or -16), the regulator is faulty. Contact DAJEN for a replacement.

With the regulators working correctly, turn off the power and remove the SCI. The next step is to insert the I.C.'s. Be careful when handling the I.C.'s. Avoid conditions that can generate static electricity: don't walk across carpets and then touch the I.C.'s, ground yourself to a metal object such as a desk or chair. Don't slide across plastic covered chairs. Leave the I.C.'s in the pink anti-static plastic bags until actually installing them. Place the p.c. board in front of you, the bag of I.C.'s to one side, and the layout picture to the other side. Remove one I.C. from the bag at a time and install it. Prior to inserting each I.C., straighten any bent pins, and squeeze them in slightly so they line up with the lead receptacles of the socket. One way to do this is to hold the I.C. between thumb and fore-finger, and with the leads resting on a table top, press down lightly. Forming the leads prevents damage to the I.C. pins and to the socket.



To insert the I.C., place the pins directly over the socket receptacles and with an even pressure press the IC into the socket until it seats firmly. The 24 pin I.C.'s will require a great deal of force to seat into the socket.

Note: the 2708 marked P0 plugs into the left-most standard profile socket. P1 plugs into the middle socket, and P2, when supplied, plugs into the right-hand socket(IC-19).

After all the I.C.'s have been installed, make a visual check of their orientation and location. For all vertical I.C.'s, the orientation mark faces toward the top of the board, and for all horizontal I.C.'s the orientation mark faces to the right of the board. Make sure all I.C.'s are in their correct position.

Power-on Test

Plug the completed board into the system bus. Connect the various peripherals (Tape recorder, keyboard, teletype, etc). If you are using a video monitor as the principal console output device, connect to the video interface board in the bus, and turn on the monitor. If you specified a teletype or serial device as the main console output, turn it to "Line". Apply power to the computer. If you have a front panel, hit the "Reset" switch. The front panel lights should indicate address 0000, and the data bus should show C3 (bits 7,6,1,0 on). Hit "Run", the data lights should all glow, and the high-order address lights (A-15,14,13,12) should indicate the address of the SCI board (For example, for address D000, A-15,14, and 12 on, A-13 off). If you don't have a front panel, hit "Reset". When running, the SCI monitor program clears the video monitor screen, outputs a prompt character (>) to the main console output device, and waits for a command from the main console input device. Characters typed on the input device are echoed to the output device. This action confirms that 90 % of the SCI is functional, so proceed to the Cassette Checkout.

If SCI fails to perform up to here, there is a problem. A few tests can be performed with a voltmeter. To know where to begin, analyze what is happening: when "Reset" is pressed the reset-jump circuit is activated. SCI grabs the bus and outputs a jump instruction to the processor (the jump instruction is the first three bytes in Prom 0). The CPU reads the jump data and jumps to the beginning of the SCI monitor. At this time the reset-jump circuit is reset and the SCI monitor initializes the LSI circuits on board, clears the video screen, outputs the prompt character, and waits for input. Troubleshooting determines where the chain of events is interrupted. This then identifies the area of the faulty component. Start with the reset-jump circuit. With a VOM measure the voltage to circuit ground (pin 7, IC-4) of pin 6 on IC-4, while holding the reset button pressed. This voltage should be less than 0.3 volt. Still holding reset, check the voltage at pin 6 of IC-13, which should be low (0.3), check pin 6 of IC-27, also low, and finally, check the voltage on the DI pins of the edge connector. DI-7,6,1,0, should be high (3 volts or so), and DI-5,4,3,2 should be low. Use the schematic to identify the pin numbers of the DI lines. A problem at this point isolates the reset-jump circuit (IC-4,13,27) or the bus driver IC-10. Tests beyond this point require a front panel and an oscilloscope to trace signals. Contact us immediately.

Programmer

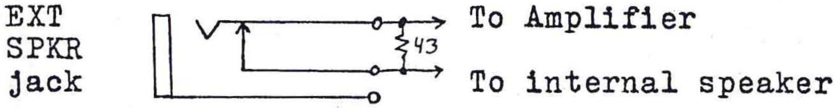
The on-board programmer power supply is active when the switch marked "P" is off or open. Place this switch off. Measure the output voltage at E of the 2N3906 transistor next to IC-29. It should be about 27 volts or slightly higher. Place the switch back in the on position.

When programming 2708's, switch "P" must be in the off position, otherwise leave it in the on position.

Cassette Check-out

With SCI functioning and responding to commands, insert the supplied tape into the cassette recorder. Turn the recorder volume control to about 1/3 of full range, and the tone control to mid-range. Hit the reset button and type in the command: SR 03E8(CR) (CR=Carriage Return). This sets the cassette read speed to 1500 baud. Place the recorder in the playback mode and type RC(CR) on the keyboard. The recorder should start playing and when the leader tone on the tape is reached, the "Level" light on SCI should glow. Adjust the level of the recorder to illuminate the "Level" light. When the sync code is received, the "Sync" light will glow. If it doesn't, change switch "R" to the opposite of its present position and repeat the entire procedure from the beginning.

For recorders that can't monitor the tape with a plug in the external speaker jack, add a 43 ohm resistor across the amp and speaker leads as shown:



With the Read circuit working, fast forward the tape to a blank area. Record the digital counter reading. Hit the reset button and use the Z command as follows: Z 0 800 E6(CR). This writes sync bytes into the first 2k of memory. Place the recorder in the record mode and type in this command: WC 0 800(CR). The recorder should start, and then stop after about 10 seconds. Rewind the tape to the beginning of this section, place the recorder in the playback mode, and read in the data. The sync light should glow. If it doesn't glow with sync bytes coming from the recorder, try reversing the "R" switch, hitting reset, and read in the sync bytes again. If they read in this time, return the "R" switch to its original position and flip the "X" switch to the opposite of its present position. This reverses the phase of the data recorded on the tape so it will read in.

Try recording some random memory blocks to check for errors on playback. If there are errors, try adjusting the tone control on the recorder to eliminate them. On some recorders there is an optimum setting of tone for best results.

Tape

We recommend Scotch "Highlander" or AVC series tape, which is low noise, high density, and costs around one to two dollars per 60 minute cassette. With this tape, we estimate an error rate of about one in ten million. We have never had a non-recoverable error (re-reading fails to bring correct data).

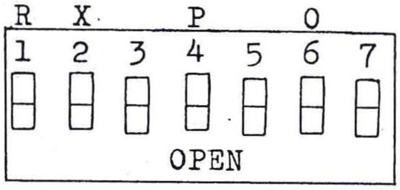
Recorders

We believe you get what you pay for. A medium-priced recorder will provide better reliability, capability, and much longer life. For a moderately priced recorder, we recommend the Panasonic RQ-430S, at about $50-$60. For best price-performance, we have found the Superscope C-104 at $80-$100 unbeatable. This recorder can routinely record reliably at 5000 baud (624 bytes per second). We have one that's seen continuous service for a year and a half and still provides reliable, fast data recording.

Characteristics to look for in a recorder: a digital tape counter to keep track of program locations, automatic level to eliminate guess-work about recording, freedom from line-noise pick-up. This last requirement can only be determined by actual use with your system. When you buy a recorder, buy it on condition that you can return it in exchange for one that works with your computer system. Most dealers are willing to comply with your desires. If yours won't, find one who deserves your business.

Maintenance is essential for reliable performance. Keep the tape heads and capstan pressure roller clean. A Q-tip soaked in isopropyl alcohol is a good tool to get to those hard-to-reach places in the recorder. The dirt you see on the cotton swab can cause drop-outs and lost data. Keeping the capstan roller clean will help keep wow and flutter down. Although the SCI cassette interface can tolerate a 33% change in speed, a dirty pressure roller can cause that amount of speed change and consequent loss of data. We routinely record 24k data blocks, and following the above precautions, we almost <u>never</u> have errors.

DIP Switch Functions



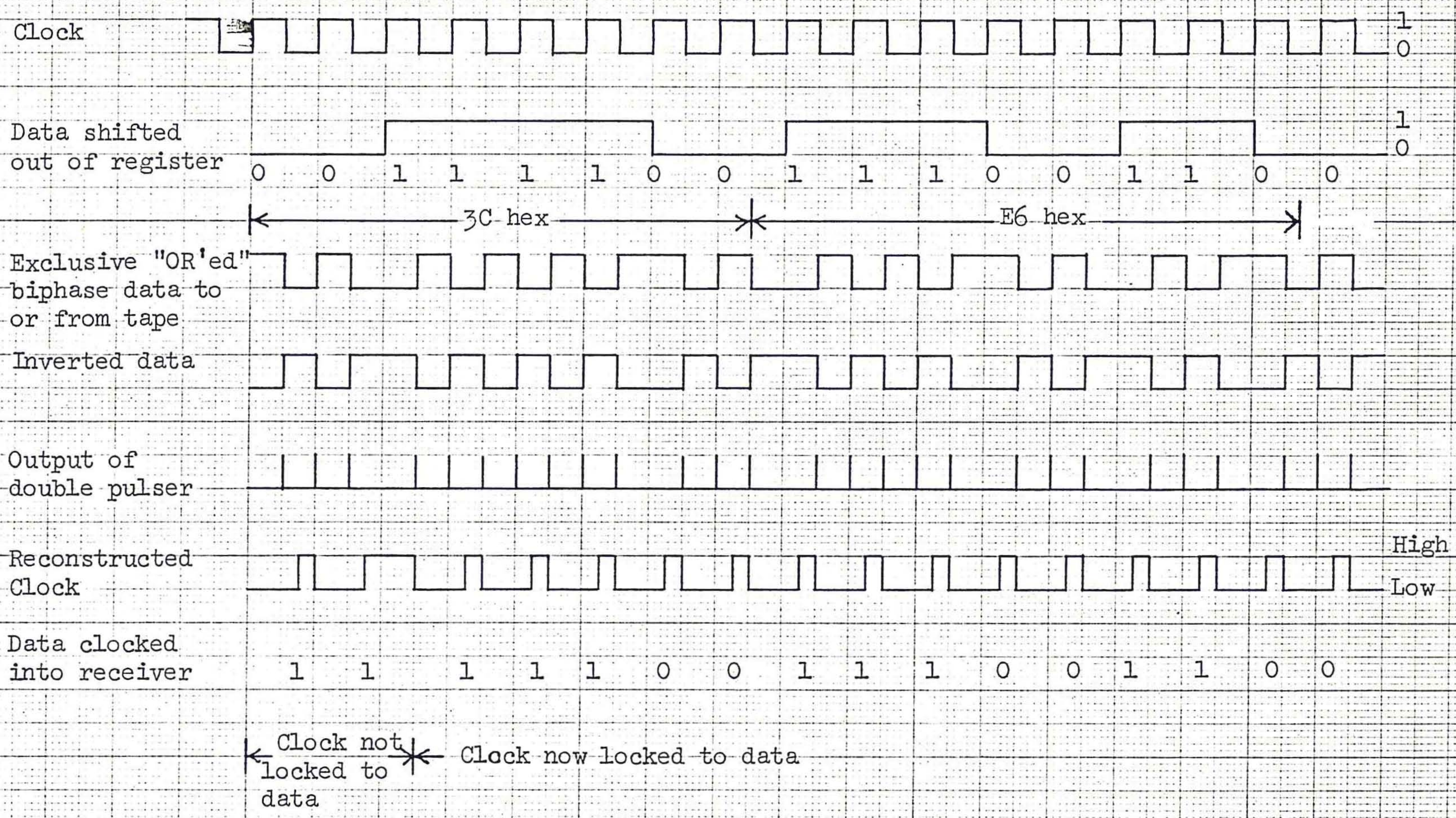
DIP Switch in IC pad 39

Switches 1 and 2 control the data inversion in the biphase cassette interface. Switch 1, "R", inverts the received data from the tape recorder. Switch 2, "X", inverts the data transmitted to the recorder. These switches compensate for different data inversions in different recorders.

Switches 3 and 5 are not used, and are available to the user if so desired.

Switch 4, "P", is the Programmer enable switch. When "OPEN" or "OFF", the on-board 2708 programmer is active. Normally, except when programming 2708's, this switch is "ON" to prevent inadvertent programming of the 2708 in IC pad 19 (3rd standard profile socket on the right.).

Switches 6 and 7 control options selected by the SCI monitor program. In current versions of the SCI software, switch 6, on initial start-up or reset, selects the console devices used for input and output. When switch 6 is "ON", a keyboard connected to IC pad 44 is assumed to be the console input device, and a video interface is assumed for output. When switch 6 is "OPEN" or "OFF", a serial device connected to IC pad 45 is assumed for both input and output.



Biphase data encoding and decoding

Theory of Biphase Cassette Operation

Biphase recording is a means of recording digital data in a high-speed, densely-packed format, on cassette tape. Because each cycle of the recording process represents one data bit, there is no wasted space. A standard cassette recorder operating well within its capabilities can record data at 2500 baud error-free. This is 2500/8, or 312 bytes per second. In comparison, Kansas-City recording, operating with shifting frequencies, runs at 300 baud, or 37 bytes per second. Thus biphase is faster, and can as a consequence store more data on a given cassette. A 30 minute side of a standard Phillips cassette can hold 500 k-bytes of data.

To understand how biphase works, look at the figure on page 12 titled "Biphase Data Encoding and Decoding". Look at the top drawing. This is the basic cassette clock which sets the data transmission speed, or baud rate (one baud is one bit per second). This clock is generated by dividing the crystal-controlled system clock through a series of dividers. Contained within the cassette interface is a shift register which is loaded with one byte of data to be output. Each positive transition of the cassette clock shifts one bit out ot the shift register. Data in a Tarbell-type system is shifted most significant bit first (bit 7 first, then bit 6, 5, ...1, 0). When all eight bits of the present byte are shifted out, the interface requests another byte.

The data shifted out of the register is then exclusive-OR'ed with the cassette clock. The logic states of a two-input exclusive-OR gate are as follows:

Inputs	A	⊕	B	=	Y	Output
	Ø		Ø		Ø	
	Ø		1		1	
	1		Ø		1	
	1		1		Ø	

Note: when both inputs are the same, the output is low.

The third line of the figure on page 12 illustrates the data after it has been exclusive-OR'ed with the clock. This is then fed to the recorder input. Notice that when the data is unchanging (a string of zeroes or ones), the output to the recorder is the cassette clock frequency, and when the data changes from a one to a zero, or a zero to a one, an extra one-half clock cycle is added to the data output. These extra half-cycles are the basis of biphase. When they occur, the phase changes and the decoded data changes to the opposite state.

Now consider the decoding process. For the decoder presented, the input data must be inverted to provide the correct data output.

After being amplified and squared by a comparator, the data from the tape recorder is fed to a receiver shift register and a "double pulser". The "double pulser" puts out a pulse every time the incoming data changes state.

This output triggers a digital timer which is set equal to 75% of the time required for one complete cassette clock cycle. The output of this timer is the "reconstructed clock" shown in the figure. When triggered, this timer output goes low until it times out and returns to the high state. As it returns, the timer output pulse clocks the present data bit into the receiver shift register. When eight bits of data have entered the shift register, the cassette interface signals the computer that a data byte is available.

Notice that the first two data bits clocked into the receiver are 1's, where 0's were sent originally. The interface, started at random, has not locked to the data stream. However, after the first data transition from a 0 to a 1, the clock is now locked to the data stream, and the correct bits are entered into the receiver register. That first byte of data (3C hex) is called the clock synchronizing byte, because it synchronizes the reconstructed clock with the data stream.

The byte following the clock sync byte, is the data synchronizing byte (E6 hex). Up to the time it is received, the receiver shift register has been reset, not signalling the reception of bytes to the computer. When the receiver recognizes the data "sync" byte, it knows the next 8 bits will be a data byte, and readies the data flag to signal the computer that a data byte is available.

Analyze what would happen if the original, non-inverted data had been input to the receiver: all the data would be inverted. This is why the receiver has a data inversion switch. If the tape recorder doesn't invert the data it receives, then the interface must.

The reason there is a transmitted data inversion switch is to allow easy software interchange among biphase-system owners. Each interface is shipped with a pre-recorded cassette tape. The system owner adjusts his interface and recorder to be compatible with that tape and he is then able to easily exchange tapes with other owners.

Parts List

	Capacitors
3	39 pf
2	.001 uf
4	.0047uf
3	.01 uf
4	.05 uf
16	4.7uf 35 VDC

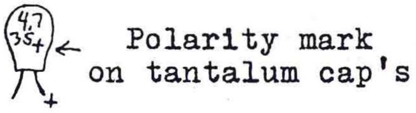
	Resistors	1/4 watt, unless noted
4	4.7 ohm	(Yellow-Violet-Gold)
3	20 1/2 watt	(Red-Black-Black)
1	51 1/2 watt	(Green-Brown-Black)
2	43	(Yellow-Orange-Black)
2	300	(Orange-Black-Brown)
1	300 1/2 watt	(Orange-Black-Brown)
6	510	(Green-Brown-Brown)
1	510 1/2 watt	(Green-Brown-Brown)
10	2k	(Red-Black-Red)
28	4.7k	(Yellow-Violet-Red)
5	20k	(Red-Black-Orange)
2	39k	(Orange-White-Orange)
3	100k	(Brown-Black-Yellow)
1	1Meg	(Brown-Black-Green)

6	1N4148 diodes
2	MV-5075 LED lights
4	2N3904 transistor
3	2N3906 transistor
1	MJE 180 transistor
1	7-pole DIP switch
4	14 pin DIP plugs
1	16 pin DIP plug

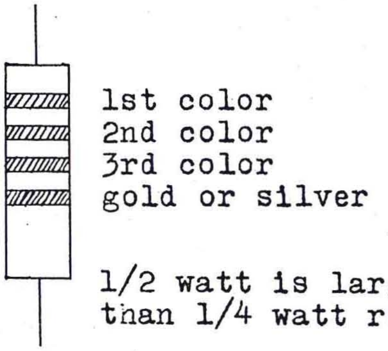
	Sockets
3	8 pin
15	14 pin
10	16 pin
3	20 pin
3	24 pin low profile
3	24 pin standard profile
2	40 pin

2 Heat Sinks

	I.C.'s		
2	74LS00	1	LM-320-T5
2	74LS04	1	LM-320-T 12
2	74LS11	1	LM-340-T 5
1	74LS32	1	LM-340-T 12
2	74LS155	1	TL-082
2	74LS221	1	LM-380
2	74LS367	2	LM-2903
1	74LS386		
2	2112		
2	2708	1	Audio Cable
2	6820	1	Cassette tape with Sync code
1	6850		
1	6852	1	P.C. Board
1	8131	4	4-40 bolts and nuts
3	81LS97	15	feet of solder
1	8253		



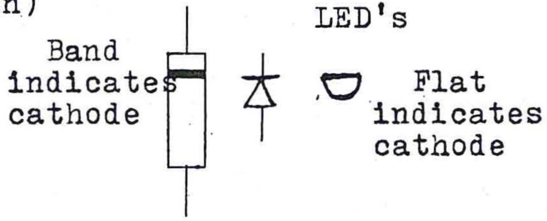
Resistors



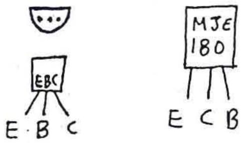
1/2 watt is larger than 1/4 watt res.

Diodes

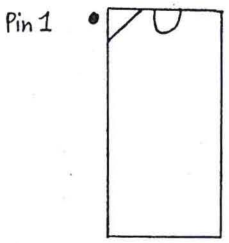
LED's



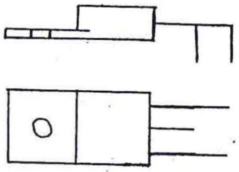
Transistors



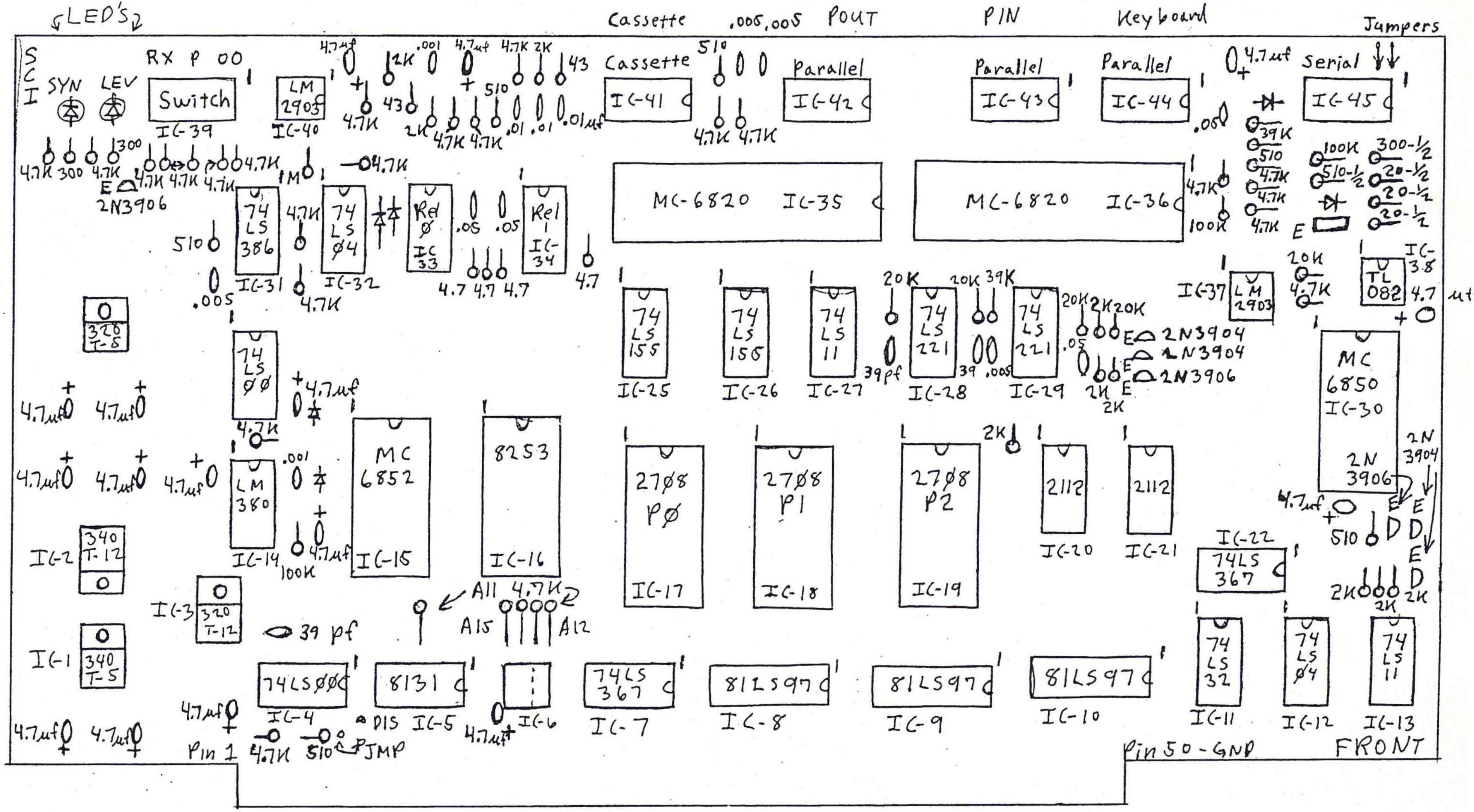
I.C.'s and sockets



A notch or dot or U section locates pin 1



Typical Regulator Installation



B2
Parts Layout

Baud Rate Table

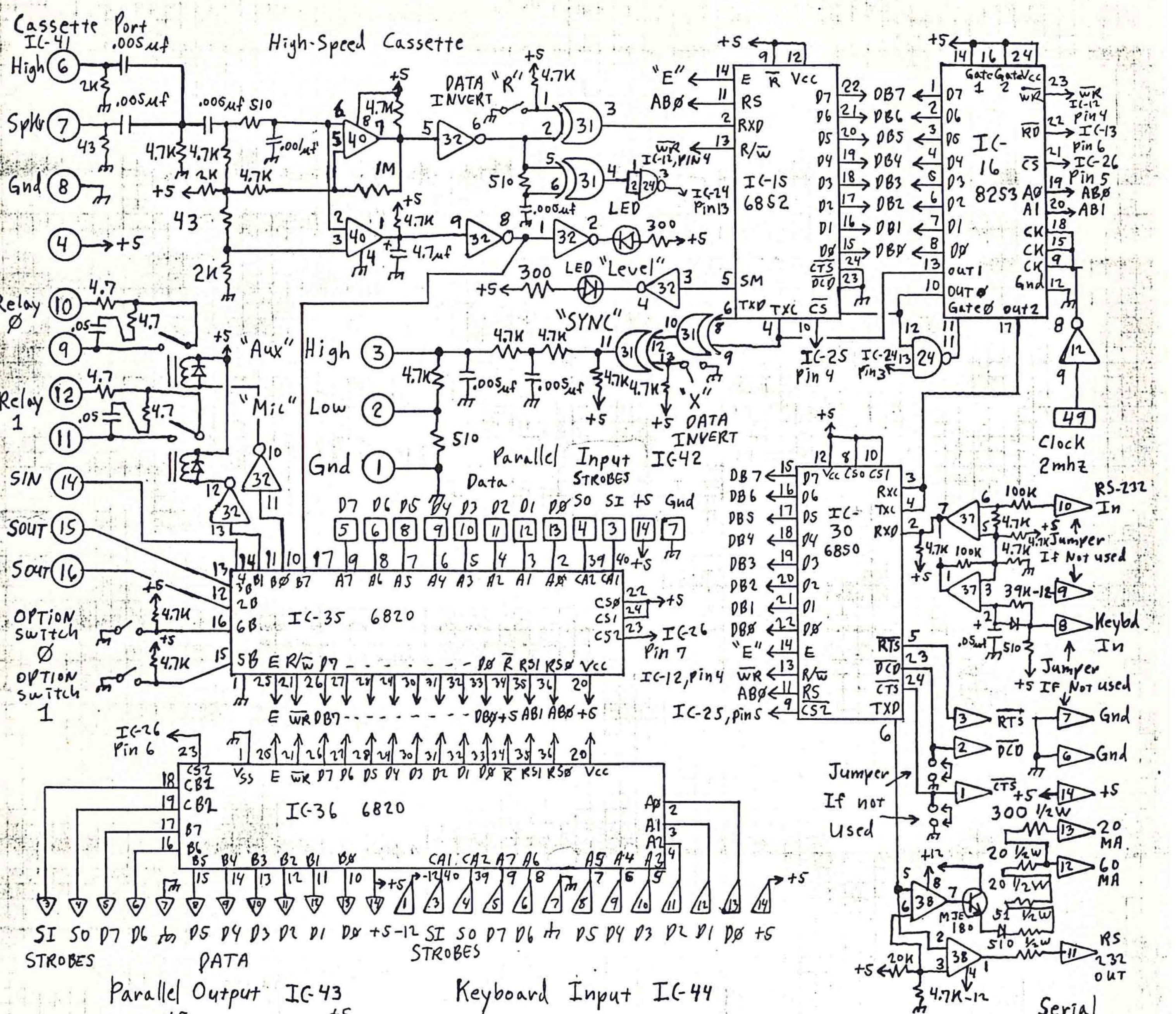
This table lists some of the baud rates which the cassette and serial ports can generate. Use the appropriate command (SR for Read, SS for serial, SW for write) to set the desired baud rate by inserting the hex number listed in the table, after the command letters. For example, to set the Read speed for 3600 baud, the command is: SR 01A0 CR (Carriage Return). Remember that when reset, or started cold, the various speeds are set by the pre-determined values in the initialization routine. For speeds not listed in the table, the appropriate hex number can be calculated as follows:

For Write: N=2,000,000/Baud Rate, N must be expressed in hex. Example: 2500 baud desired. N=2,000,000/2500=800 decimal A hex number is N3 N2 N1 NO, where N3 has a value of 4096, N2 256, N1 16, NO ones. 800 has 0-4096, so N3=0;800 has 3-256, so N2=3. This leaves 800-3X256=32. N1=2 (2X16=32), and NO=0 because nothing remains. Thus the hex number is 0320. The command is: SW 0320CR (Carriage Return).

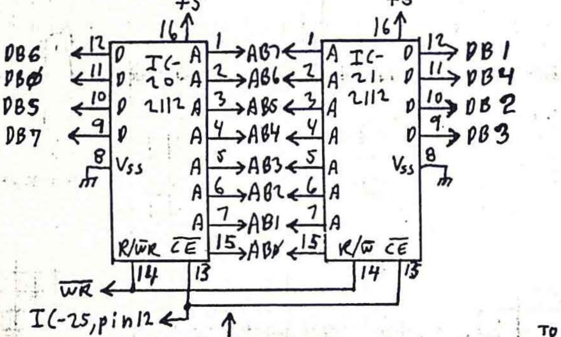
Read speed: N=1,500,000/Baud Rate

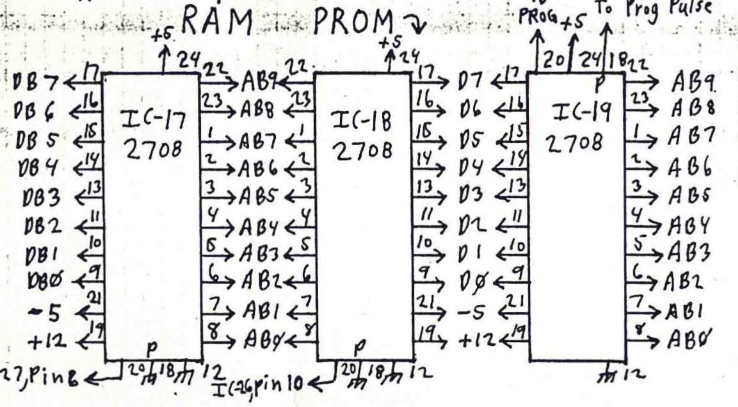
Serial speed: N=125,000/Baud Rate

	Cassette System		Serial Port	
Baud Rate	Write	Read	Serial	
45.45	ABE4	80EB	0ABE	60 WPM, Baudot
75	682A	4E20	0682	
75.75	6722	4D59	0672	110 WPM, Baudot
110	4705	3544	0470	
150	3415	2710	0341	
300	1A0A	1388	01A0	
600	0D05	09C4	00D0	
900	08AE	0682	008A	
1200	0682	04E2	0068	
1500	0535	03E8	0053	Tarbell cassette
1800	0457	0341	0045	
2200	038D	02A9	0038	
2400	0341	0271	0034	
2500	0320	0258	0032	DAJEN Cassette
3000	029A	01F4	0029	
3600	022B	01A0	0022	
4000	01F4	0177	001F	
4800	01A0	0138	001A	
5400	0172	0115	0017	
6000	014D	00FA	0014	
7200	0115	00D0	0011	
8000	00FA	00BB	000F	
9600	00D0	009C	000D	
10400	00C0	0090	000C	
12000	00A6	007D	000A	
14000	008E	006B	0008	
18000	006F	0053	0006	
19200	0068	004E	0006	
31000	0040	0030		
45000	002C	0021		
70000	001C	0015		
100000	0014	000F		



Serial Port IC-39





Resistance in	IC-1	LM-340T-5	IC-15	6852		
ohms, K = 1000,	IC-2	LM-340T-12	IC-16	8253		
M = 1,000,000	IC-3	LM-320T-12	IC-17,18,19	2708	IC-37,40	LM-2903
Capicitance	IC-4,24	74LS00	IC-20,21	2112	IC-38	LM-358
ufz microfarad	IC-5	8131	IC-23	LM-320T-5	IC-39	DIP Switch
pf= picofarad	IC-6	Address Pad	IC-25,26	74LS155	IC-41	Cassette
DB= Buffered Data	IC-7,22	74LS367	IC-28,29	74LS221	IC-42	Parallel
AB= Buffered Address	IC-8,9,10	81LS97	IC-30	6850	IC-43	Parallel
	IC-11	74LS32	IC-31	74LS386	IC-44	Parallel
	IC-12,32	74LS04	IC-33,34	Relay	IC-45	Serial Port
	IC-13,27	74LS11	IC-35,36	6820		
	IC-14	LM-380				

□ = Edge Connector

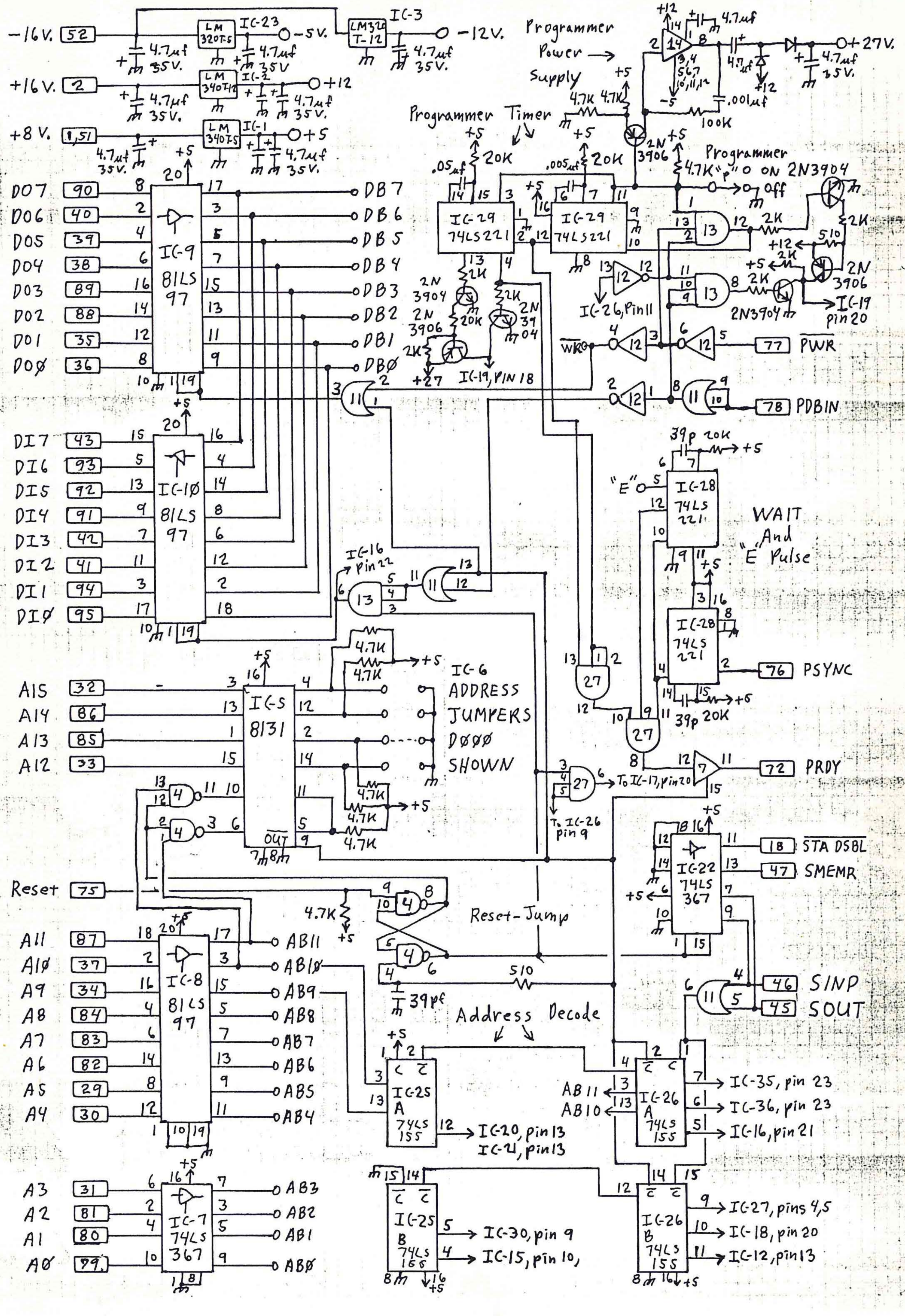
○ Cassette Port

▽,△,□ Parallel Ports

▷ Serial Port

DAJEN ELECTRONICS

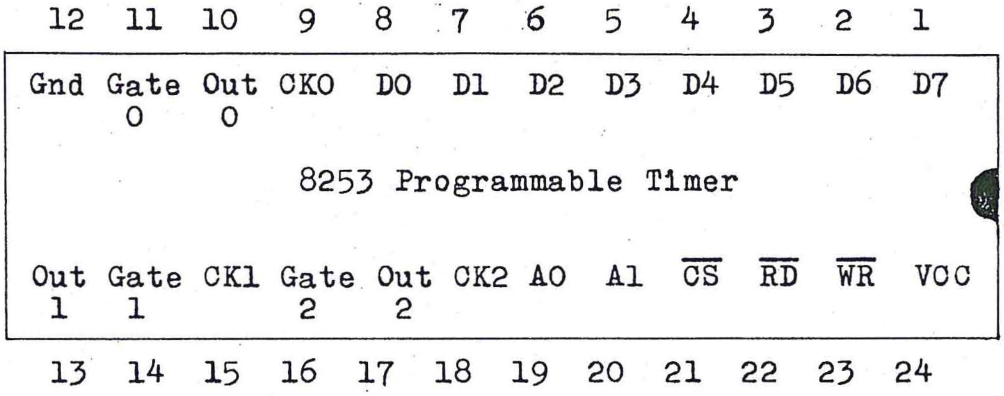
SCI



-16V. 52
4.7uf 35V.
LM 320T-5
IC-23
-5V.
LM320 T-12
IC-3
-12V.
4.7uf 35V.
+16V. 2
LM 340T-12
IC-2
+12
4.7uf 35V.
+8 V. 1,51
LM 340T-5
IC-1
+5
4.7uf 35V.
Programmer Power Supply
14
4.7uf
+12
27V.
4.7uf 35V.
3,4 5,6,7 10,11,12
-5
.001uf
100K
4.7K 4.7K
2N 3906
Programmer Timer
.05uf
20K
.005uf
20K
IC-29 74LS221
IC-29 74LS221
4.7K
Programmer "P" O ON 2N3904 Off
2K
510
2N 3906
2N3904
IC-19 pin 20
2N 3904
2N 3906
2N 3904
+27
IC-19, PIN 18
IC-26, Pin 11
WR
PWR 77
PDBIN 78
DO7 90
DO6 40
DO5 39
DO4 38
DO3 89
DO2 88
DO1 35
DOØ 36
IC-9 81LS 97
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DBØ
DI7 43
DI6 93
DI5 92
DI4 91
DI3 42
DI2 41
DI1 94
DIØ 95
IC-1Ø 81LS 97
IC-16 pin 22
39p 20K
"E"
IC-28 74LS 221
WAIT And "E" Pulse
IC-28 74LS 221
PSYNC 76
39p 20K
A15 32
A14 86
A13 85
A12 33
IC-5 8131
4.7K
IC-6 ADDRESS JUMPERS DØØØ SHOWN
OUT
PRDY 72
To IC-17, pin 20
To IC-26 pin 9
IC-22 74LS 367
STA DSBL 18
SMEMR 47
Reset 75
Reset-Jump
4.7K
510
39pf
SINP 46
SOUT 45
A11 87
A1Ø 37
A9 34
A8 84
A7 83
A6 82
A5 29
A4 30
IC-8 81LS 97
AB11
AB1Ø
AB9
AB8
AB7
AB6
AB5
AB4
Address Decode
IC-25 A 74LS 155
IC-20, pin 13
IC-21, pin 13
IC-26 A 74LS 155
AB11
AB10
IC-35, pin 23
IC-36, pin 23
IC-16, pin 21
A3 31
A2 81
A1 80
AØ 79
IC-7 74LS 367
AB3
AB2
AB1
ABØ
IC-25 B 74LS 155
IC-30, pin 9
IC-15, pin 10,
IC-26 B 74LS 155
IC-27, pins 4,5
IC-18, pin 20
IC-12, pin 13

8253 Programmable Timer

Pin Configuration



12	11	10	9	8	7	6	5	4	3	2	1
Gnd	Gate 0	Out 0	CK0	D0	D1	D2	D3	D4	D5	D6	D7
Out 1	Gate 1	CK1	Gate 2	Out 2	CK2	A0	A1	$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	VCC
13	14	15	16	17	18	19	20	21	22	23	24

D0-D7	Data lines	$\overline{CS}$	Chip Select
A0,A1	Address lines	$\overline{RD}$	Read
CK0,1,2	Clock inputs	$\overline{WR}$	Write

The 8253 consists of three independent 16-bit programmable timers, with a maximum count rate of 2 mhz. Each counter is programmed by writing a mode control word into the control word register. Depending on the mode, the count is loaded into the appropriate count register.

Control Word:	D7	D6	D5	D4	D3	D2	D1	D0
Bit assigned:	SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Where: SC=Select Counter to which control word applies

SC1	SC0	Counter
0	0	0
0	1	1
1	0	2
1	1	Illegal

BCD= 0= Binary counter
1= 4-decade BCD counter

RL=Read/Load command

RL1	RL0	
0	0	Latch counter for read operation
0	1	Read/Load least significant byte only
1	0	Read/Load most significant byte only
1	1	Read/Load least sig.byte first, then most sig.byte.

Mode:

M2	M1	M0	Mode
0	0	0	0
0	0	1	1
X	1	0	2
X	1	1	3
1	0	0	4
1	0	1	5

where X= don't care

Bus Address:

A1	A0	Device Selected
0	0	Count Reg. 0
0	1	Count Reg. 1
1	0	Count Reg. 2
1	1	Control Reg.

8253 Programmable Timer

Mode Definitions

Mode 0: Out will be low at start of count, goes high and remains high at the end of the count. The Gate input inhibits counting when low, and enables counting when high. Out goes low when count register is re-loaded. Re-loading during counting: 1st load byte stops the count, 2nd load byte starts the new count.

Mode 1: Programmable One-shot. Out goes low following the rising edge of the Gate input. Out goes high at the end of the count. If a new count value is loaded during the counting, the current count will not be affected. The One-shot is retriggerable and Out will remain low for the full count after the last trigger.

Mode 2: Rate Generator. Divides by N (the count). Out is low for one period of the input clock after each N counts. Gate when low inhibits counting and holds Out high. When Gate goes high, the counter starts counting for the full count. Gate will synchronize the counter. If the count register is reloaded, only subsequent counts will be affected. When Mode 2 is set, Out remains high until the count is loaded.

Mode 3: Square wave generator. Similar to mode 2, but the output is a square wave. If the count is odd, Out is high for (N+1)/2 counts, and low for (N-1)/2. If the count register is reloaded, the new count is effective immediately after the next output transition. Gate controls counting.

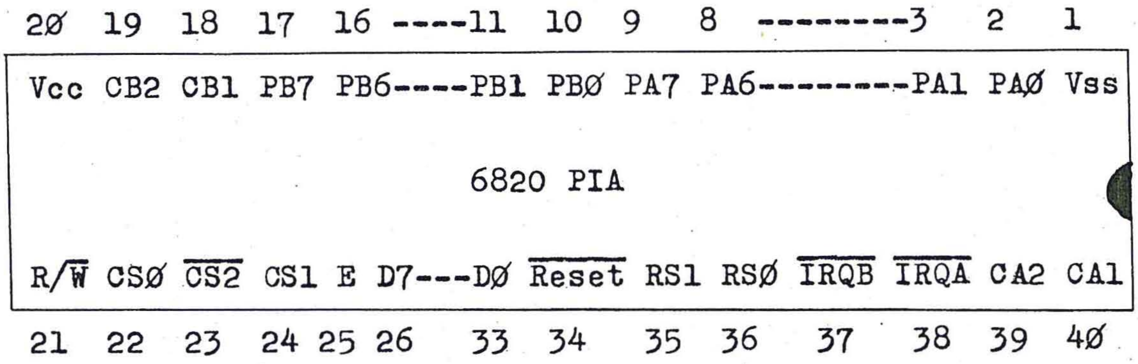
Mode 4: Software triggered strobe. Out is high after the mode is set. When the count is loaded, counting begins. Gate low inhibits counting. On reaching the count, Out goes low for one clock period. Reloading the count register while counting does not affect the present count. Reloading the count register restarts counting. Gate low inhibits counting.

Mode 5: Hardware triggered strobe. The counter starts counting after the rising edge of the Gate input, and Out goes low for one clock period when the count is reached. The counter is retriggerable.

The loading of the count register does not have to immediately follow the loading of the Mode register. The sequence of bytes specified in the mode word must be followed. All counters are down counters (count down from the count loaded).

To read the counter while counting, issue the RL Ø Ø mode addressed to the correct counter in a mode word. Then reading the counter will produce a latched count. The mode of reading must then be issued, and either one or two reads made to input the least significant byte, then the most significant byte of the count. The entire reading procedure must be completed. If two bytes are to be read, then two bytes must be read, otherwise the WR command can't be sent to the counter.

6820 Peripheral Interface Adapter


2Ø 19 18 17 16 ----11 10 9 8 --------3 2 1
Vcc CB2 CB1 PB7 PB6----PB1 PBØ PA7 PA6--------PA1 PAØ Vss
6820 PIA
R/$\overline{W}$ CSØ $\overline{CS2}$ CS1 E D7---DØ $\overline{Reset}$ RS1 RSØ $\overline{IRQB}$ $\overline{IRQA}$ CA2 CA1
21 22 23 24 25 26 33 34 35 36 37 38 39 4Ø

CA1, CA2	Control lines for parallel port A
CB1, CB2	Control lines for parallel port B
CSØ, CS1,	CS2 Chip select inputs
DØ-D7	Data bus lines
E	Enable pulse, clocks valid data, conditions control lines.
IRQA, IRQB	Interrupt flag lines for use in interrupt-driven systems.
PAØ-PA7	Parallel Port A
PBØ-PB7	Parallel Port B
R/W	Read/Write control
Reset	Resets control registers and flags
RSØ, RS1	Address lines to select internal registers.

The 682Ø Peripheral Interface Adapter consists of two independent parallel ports, each with its own handshake signals and interrupt request line, each bit programmable as input or latched output, and the necessary control registers to implement operation.

Register Addressing

RS1	RSØ	Control Register bit	Location selected
Ø	Ø	CRA-2= 1	Data Register A
Ø	Ø	CRA-2= Ø	Direction Register A
Ø	1		Control Register A
1	Ø	CRB-2= 1	Data Register B
1	Ø	CRB-2= Ø	Direction Register B
1	1		Control Register B

The Data Direction registers determine the function of each bit in the associated parallel port. A Ø bit sets the corresponding peripheral line to an input mode. A 1 results in a latched output mode.

The Control Registers set the operation of the control lines, CA-1,2, CB-1,2, and enable the interrupts. Bits Ø-5 of the control register may be written and read, while bits 6 and 7 are read-only.

Control Word Format for Control Registers

Bit 7	6	5 4 3	2	1 Ø
IRQ1	IRQ2	CA-2 Control CB-2 Control	Data or Direction Register	CA-1 Control CB-1 Control

6820 continued

Bit 2 of each control word allows access to either the data or direction register. See the register addressing on page E-3.

Bits 6 and 7 are read-only flags, and are set by signals on the Control lines. They are reset after read operations, by an E pulse, when the PIA is not selected.

Control lines CA-1,2, and CB-1,2, can be programmed for a variety of input-output functions. The following tables summarize the options available:

Interrupt inputs CA-1 and CB-1

Bit 1	Bit Ø	Interrupt Input	Interrupt Flag Bit 7	Interrupt Request $\overline{IRQA}$ or $\overline{IRQB}$
Ø	Ø	↓ Active	set high	Disabled-$\overline{IRQ}$ stays high
Ø	1	↓ Active	set high	Low when flag goes high
1	Ø	↑ Active	set high	Disabled-$\overline{IRQ}$ stays high
1	1	↑ Active	set high	Low when flag goes high

CA-2 and CB-2 as interrupt inputs

Bit5	Bit4	Bit3	Interrupt Input	Flag Bit 6	Interrupt request $\overline{IRQA}$ or $\overline{IRQB}$
Ø	Ø	Ø	↓ Active	set high	Disabled-$\overline{IRQ}$ stays high
Ø	Ø	1	↓ Active	set high	Low when flag goes high
Ø	1	Ø	↑ Active	set high	Disabled-$\overline{IRQ}$ stays high
Ø	1	1	↑ Active	set high	Low when flag goes high

CB-2 configured as an output

Bit5	Bit4	Bit3	CB-2 Cleared	CB-2 Set
1	Ø	Ø	Low on positive transition of first E pulse after a write B operation.	High when flag bit 7 is set by active transition of CB-1.
1	Ø	1	Low on positive transition of first E pulse after a write B operation.	High on pos. edge of first E pulse after E pulse which occurred when not selected.
1	1	Ø	Low when CR bit 3 goes low after write into Control Register B.	Low as long as CRbit 3 is low. High when CR bit 3 is written high.
1	1	1	same	same

CA-2 configured as an output

1	Ø	Ø	Low on negative transition of E after a read operation of reg A.	High when flag goes high from active transition on CA-1
1	Ø	1	same	High after first E pulse when part not selected

6820 continued

Bit5	Bit4	Bit3	CA-2 Cleared	CA-2 Set
1	1	Ø	Low when CR bit 3 is set low. ('A' Control Register)	High when CR bit 3 is set high ('A' Control Register)
1	1	1	same	same

Parallel Port A has internal pull-up resistors that represent about one TTL load. When configured as outputs, the voltages at the port pins must be above 2.0 volts, or erroneous data will be seen when trying to read what was written.

Parallel Port B when configured as output, can drive up to 1 ma of current, and the output voltage does not have to be above 2.0 volts for the output to be read correctly after a write operation. Thus, Port B can drive the base of a transistor switch directly.

The E pulse conditions the interrupt control lines. At least one E pulse must occur from the inactive edge to the active edge of the input signal to condition the edge sense network. In other words, at least one E pulse (about 2 us apart) must occur between strobe inputs from an external device. Then the interrupt flag will be set on the next active transition of the strobe.

The interrupt flags are cleared after a read of the particular data register. They can not be set again until an E pulse occurs while the part is deselected.

Interrupts

When using interrupts, $\overline{\text{IRQA}}$ or $\overline{\text{IRQB}}$ is tied to the appropriate Vectored Interrupt pin on the S-100 bus, or they may be tied together to $\overline{\text{Int}}$ (pin 73), as they are open-drain outputs. Then when an interrupt occurs, the appropriate flags can be checked in the Control Registers to determine the requesting device.

6850 Asynchronous Communications Interface Adapter, or ACIA

Pin-outs

1	Vss	$\overline{\text{CTS}}$	24
2	RxData	$\overline{\text{DCD}}$	23
3	RxCK	DØ	22
4	TxCk	D1	21
5	$\overline{\text{RTS}}$	D2	20
6	TxData	D3	19
7	$\overline{\text{IRQ}}$ 6850	D4	18
8	CSØ	D5	17
9	$\overline{\text{CS2}}$	D6	16
10	CS1	D7	15
11	RS	E	14
12	Vdd	R/$\overline{\text{W}}$	13

Definition of pins

CSØ,CS1,$\overline{\text{CS2}}$ Chip-select inputs. When CSØ and CS1 are high, and CS2 is low, the 6850 is selected by the bus.

$\overline{\text{CTS}}$ Clear to Send input from a modem, when low indicates that the data channel is ready for transmission. When a modem is not used, must be tied to ground.

$\overline{\text{DCD}}$ Data Carrier Detect input from a modem, when low the data carrier is present. If high, the data carrier is lost, and the 6850 will generate an interrupt to reflect the DCD status in the status register. The receive interrupt must be enabled for this to happen. Must be tied to ground when not used.

DØ-D7 Eight bidirectional data lines to the central bus.

E Enable input. Clocks data to and from the ACIA and enables the input/output bus drivers.

$\overline{\text{IRQ}}$ TTL compatible, open drain, active low output that requests an interrupt when the ACIA interrupts are enabled. It may be tied together with other open-drain interrupt outputs in a polled-interrupt system where the interrupt routine checks the status of each device to determine which one requested service.

RS Register Select. Tied to address line AO. When low, the status/control registers are selected. When high, the receive/transmit data registers are selected.

$\overline{\text{RTS}}$ Ready to Send output. Signals the modem that the ACIA is ready to send data. Active low.

R/$\overline{\text{W}}$ Read/Write signal from the bus.

RxCK Receiver clock, normally 16 times the actual data rate.

RxData Received data, serial input, least significant bit first.

TxCK Transmitter clock, normally 16 times the data rate.

TxData Transmitter data, serial output, least significant bit sent first.

Vss System ground.

Vdd +5 volt supply, nominally 5%.

The ACIA transmits and receives serial data in an asynchronous format, with one start bit, seven or eight data bits, and one or two stop bits, with or without parity. As used in the SCI, the ACIA is set up for one start bit, eight data bits, no parity, two stop bits, and divides the clock by a factor of 16.

Status Register Bit Assignments:

The status register is selected when RS is low, and R/W is high, and the ACIA is selected.

Bit 7	6	5	4	3	2	1	Ø
IRQ	PE	OVRN	FE	CTS	DCD	TDRE	RDA

CTS Indicates the state of the CTS pin. When high, TDRE is disabled.
DCD Indicates the state of the DCD pin. Remains high after DCD goes low, until reset by reading the status register then the data register, or reset by Master Reset.

FE Framing error which indicates a lack of the proper stop bit, which can be the result of a poor transmission, or the break condition (open loop).
IRQ Interrupt request status, reflects the state of the IRQ pin. High indicates interrupt. Cleared by a read or write data operation.
OVRN Indicates a receiver over-run. The last received data was not read by the time the new data was ready.
PE Parity Error, when parity is selected, indicates that the parity of the received data did not match the parity of the data transmitted, which results from a lost data bit.
RDA Received data available in the receiver buffer.
TDRE Transmit Data Register Empty, ready for new data to be entered for transmission.

Control Register Bit Assignment:

The control register is selected when RS is low, R/W is low, and the ACIA is selected.

Bit 1	Bit Ø	Select Clock divide ratio, or reset
Ø	Ø	Divide by 1
Ø	1	Divide by 16 (SCI)
1	Ø	Divide by 64
1	1	Master Reset

Bit 4	3	2	Word Length, Parity, Stop bits
Ø	Ø	Ø	7 data bits, even parity, 2 stop bits
Ø	Ø	1	7 bits, odd parity, 2 stop bits
Ø	1	Ø	7 bits, even parity, 1 stop bit
Ø	1	1	7 bits, odd parity, 1 stop bit
1	Ø	Ø	8 bits, 2 stop bits (SCI)
1	Ø	1	8 bits, 1 stop bit
1	1	Ø	8 bits, even parity, 1 stop bit
1	1	1	8 bits, odd parity, 1 stop bit

Bit 6	5	State of RTS	Transmit control
0	0	low	transmit interrupt disabled
0	1	low	interrupt enabled
1	0	high	interrupt disabled
1	1	low	interrupt disabled, break level (open loop) transmitted.

Bit 7	Receive interrupt control
0	disabled
1	enabled

Reference information

For further information on the 8253 Programmable Counter, see the Intel Data Catalog

Intel Corporation
3065 Bowers Ave.
Santa Clara, Ca 95051

For information on the 6820, 6850, and 6852, see the M6800 Microcomputer System Design Data from:

Motorola Semiconductor Products, Inc
Box 20912
Phoenix, Arizona 85036

A good source of data for TTL IC's is:

The TTL Data Book for Design Engineers

Texas Instruments, Inc.
P.O. Box 5012
Dallas, Texas 75222

SCI 2K Monitor version 1.6

This monitor provides the routines that initialize and interface the LSI circuits on board the System Central Interface. Other monitor functions are a video driver for a video display interface, commands to display and alter memory, high-speed cassette routines to store and load blocks of memory or data in a check-summed format (Automatically starting and stopping the tape recorders), assign I/O devices, and program 2708's.

Monitor Operation

When reset is activated, or on initial start-up, the monitor configures the I/O ports, sets the speeds for cassette and serial port operation, and turns off the relays. The monitor then erases the screen of the video display, clears the command buffer area, outputs a prompt character (>) to the main console output device, and waits for a command.

Commands consist of one or two letters followed by hexadecimal addresses where appropriate. Command letters must be followed by a space, then addresses as necessary. If a mistake is made when typing in a command, hit rubout or control H (backspace) to delete the last character typed. Addresses can be from one to four hex characters (Ø-9,A-F). If more than four hex characters are typed without any intervening space, only the last four characters will be used. For example, if AC1ØØ is typed, only C1ØØ is seen by the monitor. To read 16 bytes of a tape, and place them at ØØØØ hex, the following is sufficient: RS Ø 1Ø (CR).

All command strings are terminated by a carriage return. At this time, a line feed and carriage return are output, and the command is executed. If the command is erroneous, or there is an error in the command string, ERROR is output and the monitor waits for a new command. After the command has been executed, an appropriate remark is output and the monitor awaits another command.

Escape

All command routines contain an escape. To return to the command mode, just hit the Escape (Esc) key on the main console input device. The routine being executed will abort, and the monitor will wait for a command input. When escape is used to exit cassette routines, the cassette relays are left activated. This allows the operator to rewind, or fast forward the cassette. Before using the cassette routines again, the reset switch must be activated, or a command to go to the start of the monitor. This re-initializes the cassette ports and turns the relays off.

Video Driver Software

This driver assumes that the video interface is a 1k block of memory located at CCØØ to CFFF hex, and that CCØØ is the upper left-hand corner of the monitor screen. A cursor is displayed on-screen to indicate where the next character will be written. Control commands available are: carriage return, which moves the cursor to the left-hand margin; line feed, which moves the cursor down one line; control H, a backspace, which moves the cursor back one position (when the cursor is at the left-hand margin, backspace moves it to the right-hand margin, one line up); rubout, does the same thing as backspace; control K, a vertical tab, moves the cursor to the upper left-hand corner of the screen (useful for live-action games, the program can use the upper left corner as a reference point to the rest of the screen when changing parts of the display); control L, form feed, which erases the entire screen and places the cursor at the upper left corner of the screen.

When the last line of the display is filled, the screen is scrolled up one line. The speed at which this occurs can be varied by typing Ø-9 during the scrolling process. Ø is the slowest speed, about 2 seconds per line, and 9 is the fastest, about 1Ø lines per second. Typing a space will stop the scroll. Typing any other character will start it again.

There are differences among the video interfaces offered on the market. The P-Tech VDM requires that bit 7 be low for characters output to the screen. The Poly VTI requires bit 7 be high. The areas where bit 7 is critical are illustrated in the video driver section of the assembly listing.

Jump Locations

At the beginning of the monitor are 12 jump instructions. The first jump simply jumps to the start of the monitor. The following jumps are the solution to easy interfacing with any program. Just call the appropriate routine and the SCI monitor takes over to do all the interface work. No need to worry about port locations, status data, or cassette routines. Each call (3 bytes) replaces at least 10 bytes in a typical status check,data handling routine, and the 12 bytes of cassette functions replace at least 38 bytes of code, in addition to providing much more advanced functioning. So program modification is simple, fast, and saves memory space.

Data in the routines is always handled in the accumulator (A). H is the high-order nibble of the SCI memory location (example, DØØØ, EØØØ, FØØØ, etc.).

HØØ3 Jumps to the input routine and returns to the calling program with the character from the console input device. Bit 7 is not masked.

HØØ6 Jumps to the console output routine, outputs the character, then returns.

HØØ9 Cassette input. Calling this location starts the recorder, waits for a level indication, initializes the cassette circuits, and returns with a data byte. Each subsequent call returns with a byte of data.

HØØC End cassette read operation. Call this location when all the data has been read. This routine then waits for an inter-record gap (a blank spot), then turns off the recorder.

HØØF Cassette write. This routine turns on the recorder, writes a leader on tape, sets the baud rate, outputs the sync bytes then the data. Subsequent calls will write data to tape.

HØ12 Cassette write end, writes a brief trailer, followed by an inter-record gap, turns off the recorder, and returns to the calling program.

HØ15 Parallel input from the second parallel input port. (Port HØ). Gets the data byte and returns. Useful for paper tape readers or analog-to-digital converters.

HØ18 Parallel output, port H6. Outputs a byte, then a strobe pulse to the peripheral, and returns.

HØ1B Input status check. The status of the currently-selected input device is polled. If data is not available, the zero flag is set. If data is available, the zero flag is not set, and the accumulator will return with the data byte. This is useful for escape routines, data input, and is directly compatible with Processor Technology software. Bit 7 is masked low on all data (E6 7F).

HØ1E Input masked. This routine is identical to HØØ3, but masks bit 7 low for all data. Saves two bytes in the user's program.

HØ21 Write B. This routine outputs the data byte passed in the B register. Directly compatible with Processor Technology software.

HØ24 Call Escape, check input character, return

For details on the use of these jump locations, and interfacing the SCI to popular software, see appendix H, "Patches to Popular Software".

Cassette Format

The cassette tapes written and read by this monitor have the following format: a leader of tone, a clock synchronizing byte (3C hex), a data synchronizing byte (E6 hex), a low-order starting address byte (where the data will be stored), a high-order starting address byte, a high-order block-length byte, a low-order block-length byte (the number of data bytes), the data, and a checksum which is the 8-bit result of adding all the bytes (excluding the clock and data sync bytes), and finally a trailer tone and an inter-record gap (a blank spot).

When an external program is calling cassette functions via the jump locations, the sync bytes and the leader and trailer tones with an inter-record gap are generated by the SCI. The header and data information are the responsibility of the calling program.

Monitor Commands

The SCI monitor will accept both upper and lower case characters. In the command mode, bit 7 is ignored.

In the following explanations, CR means Carriage Return, or simply Return. ADDR means a one to four character hex address. Commands, their operands, and hex addresses must be separated by one space.

AI K, P, or S(CR) Assign Input device, either Serial, Parallel, or the Keyboard port at IC-44. When reset is activated, input returns to the device specified by the DIP switch (see page 11).

AO P, S, or V(CR) Assign Output device, either Parallel, Serial, or Video interface using the SCI video software driver.

D ADDR(CR) Dump hex data from memory. Dumps one line of 16 hex bytes, starting at ADDR. To dump additional data, hit the space bar. To return to the command mode, hit any other character.

DA ADDR ADDR(CR) Dump ASCII; displays memory data in hex bytes, and the ASCII equivalent. Useful for locating tables and comments in programs.

E ADDR(CR) Enter memory. The memory address and its content are displayed. If a new byte is desired, type it in and it will be entered into memory. The byte actually stored will be displayed, and the next memory location will be displayed. If the content is to remain unchanged, typing a space will bring the next byte. Typing a non-hex character will terminate the routine. If you want to move backward in memory, hit Control H, a backspace.

EM ADDR(CR) Enter Memory. Similar to the E command, but no data is read from memory. Sixteen bytes may be entered per line of type. This routine saves paper when using a printer for console output. Use the DA or D command to verify correct entry of the data into memory.

G ADDR(CR) Go. Transfers execution to that address.

H ADDR ADDR(CR) Hex arithmetic. Calculates the sum and difference of the two hex numbers. The first number printed is the sum, the second is the result of subtracting the first number from the second.

I NN(CR) Input from port NN. The byte at the specified port is displayed. To repeat the input, hit the space bar. To return to the monitor, hit any other key.

M ADDR ADDR ADDR(CR)Move memory from the block specified by the first two addresses, to a block starting at the third. The data is moved and then verified. Any errors that occurred in the move are listed. Can be used as a simple memory check, see appendix G.

O NN MM(CR)

Output the hex byte MM to the port NN. To repeat the function, type a space. To return to the monitor, hit any other key.

P ADDR ADDR(CR)

Program a 2708 with data from memory beginning at the first address, to the 2708 at the second address, which for the SCI is H8ØØ (D8ØØ, E8ØØ, etc). When the 2708 is programmed according to Intel specifications, the data is checked, and any errors listed. Allow about 2 1/2 minutes for execution.

RC(CR)

Read Cassette. Reads a cassette tape of the format specified earlier. When finished, writes COMPLETE. If the checksum read in doesn't agree with the checksum on the tape, an error has occurred. In this case, the monitor writes TAPE ERROR.

RS ADDR ADDR(CR)

Read Specify. If no address is given, reads all information in the file and places it in memory, including the header, beginning at ØØØØ. If one address is given, the information is placed in memory starting at that address. If two addresses are given, that block of memory is filled with the first part of the file. The recorder is stopped when it reaches an inter-record gap. The checksum is calculated to check for errors, except where two addresses are given. In that case, TAPE ERROR always appears.

RV(CR)

Read Verify. Reads a file and compares it with memory. Useful for checking files as they are written to make sure there are no errors.

S ADDR ADDR N1 N2..N9

Search the memory block specified by the two addresses, for the hex character string N1 N2...N9. The address of the start of each string found is printed. Up to 9 bytes may be specified, and a minimum of one, or the routine will loop forever. If all memory is searched, allow 50 seconds for execution.

SR NNNN(CR)

Set Read speed for the cassette. The appropriate hex number, NNNN, is given in the Baud Rate table, page C-1. When reset is hit, the SCI returns to the read speed specified in the initialization routine.

SS NNNN(CR)

Set Serial speed for the serial port. Same requirements as SR.

SW NNNN(CR)

Set Write speed for the cassette. The requirements are the same as for SR.

V ADDR ADDR ADDR(CR)

Verify a block of memory specified by the first two addresses, with a block of memory starting at the third address. Useful for checking the programming of 2708's, or changing bits in different memory blocks.

WC ADDR ADDR(CR)

Write Cassette, the memory block specified by the two addresses, in the format defined earlier. The routine says WRITTEN when done. Automatically starts and stops the recorder as do all tape commands.

Z ADDR ADDR NN(CR)

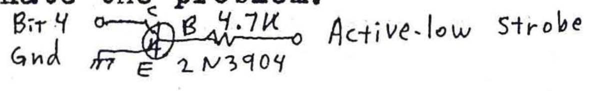
Zero memory from ADDR to ADDR. If NN is not given, zeroes are written. If NN is given, the memory block is filled with that hex byte. Makes a simple means to fill memory with sync bytes to write a sync tape for recorder set-up. Or fill memory with FF to check the erasure of 2708's.

Hitting Escape will cause an exit from all routines, and return to the command mode. If used during a cassette routine, reset must be hit before trying to use the cassette routine again, or Go to the start of the monitor (H000), or output 00 to port H2 the cassette port control.

Cassette commands turn the relays on and off to control two recorders, one for read and one for write. If only one recorder is used, wire the two sets of relay contacts together in parallel, as outlined on page 6, Cassette Control.

Keyboard problems

Keyboards with unlatched outputs may have problems with Escape functions. Increasing the keyboard strobe pulse to 20 ms will allow the strobe and escape character to be recognized, while the monitor is executing a program. Also, some encoders are noisy on the data lines when not active. A false escape character may be occasionally generated which the SCI monitor will interpret as a legitimate escape. Connecting a transistor such as a 2N3904 to bit 4 of the keyboard data lines, and driving it with an active-low strobe will eliminate the problem.



Using the 8253 for a sweep generator.

Using the cassette output pin (2 of IC pad 41 for low level, and pin 3 for high level) as the signal source, and the following program, the 8253 can be programmed to sweep any range of frequencies. This range can start as low as 30 hz, and go as high as 1 mhz (to use the 1 mhz band, the output capacitors of the cassette system must be bypassed, so get the signal direct from pin 11 of IC-31, the 74LS386).

The program as written produces an electronic siren, which provides interesting results when piped through a stereo. For a slowly changing tone that can be used to check the response of an audio amplifier, change the following lines:

```
0000 21 00 30  this changes the start frequency
000A 1E 80     this changes the total range
0013 01 00 E0  this changes the magnitude of the step
```

Program to sweep the 8253

```
0000 21 00 0C  set the starting frequency to 651 hz
0003 01 A0 FF  set B to subtract 60hex each freq change
0006 3E 76     set up 8253
0008 D3 DB     SCI address set for D000. change to suit
000A 1E 10     set number of steps
000C 7D        output L to 8253 counter
000D D3 D9     change D9 for other address of SCI
000F 7C        output H to 8253 counter
0010 D3 D9     change D9 for other address of SCI
0012 C5        Push B to save
0013 01 00 FA  set up a delay
0016 03        increment BC
0017 78        MOV A,B
0018 B1        ORA C    all zeroes yet?
0019 C2 16 00  JNZ      if not, loop
001C C1        POP B    restore BC
001D 09        DAD B    add B to HL
001E 1D        DCR C    decrement step counter
001F C2 0C 00  JNZ      loop until done
0022 C3 00 00  JMP      start all over
```

Try other variations of locations 0000, 000A, and 0013 for different sound effects.

Simple Memory Checker

Using the E command and the M command, a simple check of a block of memory can be made to look for hard errors (Always bad bits). This is a simple, fast memory check. Since it only executes once, it is not always able to find elusive errors, but will pinpoint hard memory failure.

Using the E command, to check and verify entry, enter the following sixteen bytes of data in the first sixteen locations of the block of memory to be tested:

```
ØØ 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF
```

Hit return and using the Move command, enter:

```
M ADDR1 ADDR2 ADDR3(CR)
```

ADDR1 is the beginning address of the sixteen bytes of data, ADDR2 is the ending address of the block of memory, minus 16, and ADDR3 is ADDR1 plus 16.

Example:

A block of unknown memory at 2ØØØ hex to 2FFF is to be tested. First, enter the sixteen test bytes:

```
E 2ØØØ(CR)
2ØØØ NN ØØ ØØ
2ØØ1 NN 11 11
2ØØ2 NN          and so on until all 16 bytes are correctly
entered.  NN above is the present content of the memory
location.
```

Then use the Move command :

```
M 2ØØØ 2FEF 2Ø1Ø(CR)
```

This command then fills the block of memory with the sixteen bytes of test data and checks for any errors. If any errors do show up, the memory is unusable and should be tested thoroughly to locate the problems.

Interfacing the SCI Monitor to ESP-1

ESP-1 is an Assembler, Editor, and Monitor for an 8080, written for the 8080 by Michael Shrayer. The following patches are for the SCI Monitor version 1.5.

Location	Code before	for SCI	
6C52		CD Ø6 HØ	Output
6CAD		CD Ø6 HØ	Output
7ØC6	DB ØØ	C3 1E	Input
7ØC8	E6 Ø2	HØ	
7ØCA	CA C6 7Ø		
7ØCD	DB Ø1	C3 24 HO	Escape Routine
7ØCF	E6 7F		
7ØD1	C9		
7ØDF	DB ØØ	ØØ ØØ	Output
7ØE1	E6 Ø1	ØØ ØØ	"
7ØE3	CA DF 7Ø	ØØ ØØ ØØ	"
7ØE6	78	78	"
7ØE7	D3 Ø1	C3 Ø6 HØ	"
7ØE9	C9		
77E8	26	29	Cassette Read
77EC	26	29	" "
77FØ	26	29	" "
77FA	26	29	" "
7815	26	29	" "
781D	CD ED 7Ø	CD ØC HØ	" "
782Ø	21 ØØ 6Ø	CD ED 7Ø	" "
7823	C3 DE 73	21 ØØ 6Ø	" "
7826	CD BD 7Ø	C3 DE 73	" "
7829	DB 6E	CD BD	" "
782B	E6 1Ø	7Ø C3	" "
782D	C2 26 78	Ø9 HØ 78	" "
776C	CD A7 77	C3 74 77	Cassette Write
7776	A7	AA	" "
777A	A7	AA	" "
777E	A7	AA	" "
7782	A7	AA	" "
778C	A7	AA	" "
779A	A7	AA	" "
77A1	21 5F 78	CD 12 HØ	" "
77A4	C3 E5 76	21 5F 78	" "
77A7	F5	C3	" "
77A8	CD BD 7Ø	E5 76 F5	" "
77AB	DB 6E	CD BD	" "
77AD	E6 2Ø	7Ø F1	" "
77AF	C2 A8 77	C3 ØF HØ	" "

In the Escape routine, if ESC is typed the SCI monitor will take over. This is not a standard character for ESP-1.

In the cassette read routine, if ESP-1 detects an error, the SCI monitor must be reset to turn off the recorder, and reset the cassette port.

The write cassette routine takes full advantage of the SCI monitor to write a tape and control the recorder.

In the above, H stands for the 4k memory block of the SCI (DØ, EØ, etc.).

Patches for Cromemco 3k Control Basic

Cromemco 3k Control Basic is a simple Basic modelled after Palo Alto Tiny Basic. It has integer arithmetic, and can call its programs located anywhere in memory. Its greatest utility is in control applications, such as Data Aquisition, and process control.

Location	Code Before	Code for SCI	
EE9D	DB ØØ	F1 E5	Output routine
EE9F	E6 8Ø	F5 E6	"
EEA1	CA 9D EE	7F CD 06	"
EEA4	F1	HØ	"
EEA5	E5	ØØ	"
EEA6	F5	ØØ	"
EEA7	E6 7F	ØØ ØØ	"
EEA9	D3 Ø1	ØØ ØØ	"
EEEE	DB ØØ	CD 1B	Input routine
EEFØ	E6 4Ø	HØ ØØ	"
EEF2	C8	C8	"
EEF3	DB Ø1	ØØ ØØ	"

The SCI input routine checks status of the selected input device, and returns with the zero flag set if no data is available. If data was available, the zero flag is not set, and the accumulator contains the data input.

In the above, HØ represents the memory block of the SCI monitor, such as DØ, EØ, etc.

MICROCHESS by MicroWare Limited

An aggressive game of chess, a challenge for any player.

```
Location Original Code       SCI

ØDF1 DB ØØ                   CD Ø3    Input Routine
ØDF3 E6 4Ø                   HO ØØ
ØDF5 CA F1 ØD                ØØ ØØ ØØ
ØDF8 DB Ø1                   ØØ ØØ


ØDE6 DB ØØ                   C3 21    Output Routine, Character
ØDE8 E6 8Ø                   HØ 8Ø    is in B register.


Ø1D7 C3 ØØ EØ                C3 ØØ HØ Return linkage to monitor
```

The game of CHASE

```
Ø1A9 DB ØØ                   CD Ø3     Input Routine
Ø1AB E6 4Ø                   HØ ØØ
Ø1AD CA A9 Ø1                ØØ ØØ ØØ
Ø1BØ DB Ø1                   ØØ ØØ
```

Output is always to the video display memory located at CCØØ to CFFF hex. (Normally a VDM-1)

TREK-80

The best game of STAR TREK to hit the market. A live-action game, it's a challenge to win. Output is always to a video display memory which must be equivalent to a VDM-1. TREK-80 is from Processor Technology.

```
Ø1BØ DB ØØ                   CD 1B       Input Routine
Ø1B2 ØØ                      HØ
Ø1B3 E6 4Ø                   ØØ ØØ
Ø1B5 C8                      C8
Ø1B6 DB Ø1                   ØØ ØØ
Ø1B8 E6 7F                   E6 7F
```

This is a typical Processor Technology input routine and is applicable to most of their software.

```
D000                0005 *******SCI MONITOR VERSION 1.6*******
D000                0010 PORT    EQU   0D0H          PORT LOCATION
D000                0015 STACK   EQU   0DFC0H        STACK LOCATION
D000                0020 STK0    EQU   STACK+4       INP DEV CODE
D000                0025 STK1    EQU   STACK+5       OUT DEV CODE
D000                0030 STK2    EQU   STACK+6       SCROLL SPEED
D000                0035 STK3    EQU   STACK+7       CASS READ SPD
D000                0040 STK5    EQU   STACK+9       CASS WRITE SPD
D000                0045 STK7    EQU   STACK+11      SERIAL SPEED
D000                0050 STK9    EQU   STACK+13      TEMP STORE
D000                0055 STK10   EQU   STACK+14      VDM CHARACTER
D000                0060 STK11   EQU   STACK+15      VDM POINTER
D000                0065 STK13   EQU   STACK+17      TEMP STORE
D000                0070 STK15   EQU   STACK+19      TEMP STORE
D000                0075 STK16   EQU   STACK+20      TEMP STORE
D000                0080 STK17    EQU   STACK+21      INPUT BUFFER
D000                0085 PORT0   EQU   PORT+0        PAR INP
D000                0090 PORT1   EQU   PORT+1        CONTROL REG
D000                0095 PORT2   EQU   PORT+2        CASSETTE CONTROL
D000                0100 PORT3   EQU   PORT+3        CONTROL
D000                0105 PORT4   EQU   PORT+4        KEYBD IN
D000                0110 PORT5   EQU   PORT+5        CONTROL
D000                0115 PORT6   EQU   PORT+6        PAR OUT
D000                0120 PORT7   EQU   PORT+7        CONTROL
D000                0125 PORT8   EQU   PORT+8        CASS READ CLOCK
D000                0130 PORT9   EQU   PORT+9        CASS WRITE CLOCK
D000                0135 PORTA   EQU   PORT+10       SERIAL CLOCK
D000                0140 PORTB   EQU   PORT+11       CONTROL REG
D000                0145 PORTC   EQU   PORT+12       UART CONTROL
D000                0150 PORTD   EQU   PORT+13       UART DATA
D000                0155 PORTE   EQU   PORT+14       CASS SSDA CONT
D000                0160 PORTF   EQU   PORT+15       CASS SSDA DATA
D000                0165 **CODE STARTS HERE**
D000 C3 27 D0       0170 SCI     JMP   INIT          INIT PORTS
D003 C3 64 D2       0175         JMP   INPUT         INPUT
D006 C3 99 D2       0180         JMP   WRITE         WRITE
D009 C3 D1 D3       0185         JMP   CASR0         CASS INPUT
D00C C3 D6 D4       0190         JMP   END           TURN OFF RELAY
D00F C3 9D D5       0195         JMP   CASW00         CASS WRITE
D012 C3 EE D5       0200         JMP   CASW2         END OF WRITE
D015 C3 7C D2       0205         JMP   PIN           PAR INPUT
D018 C3 B0 D2       0210         JMP   POUT          PAR OUTPUT
D01B C3 52 D2       0215         JMP   ISTAT         INP STATUS
D01E C3 5E D2       0220         JMP   INPM          INP MASKED
D021 C3 98 D2       0225         JMP   WRITB         B OUT
D024 C3 58 D2       0230         JMP   ICHAR         ESCAPE
D027 31 C0 DF       0235 INIT    LXI   SP,STACK      SET STACK
D02A 3E 0C          0240         MVI   A,0CH         ERASE VIDEO
D02C CD BC D2       0245         CALL  VDM+1
D02F 3E 30          0250         MVI   A,30H         INIT PARALLEL PORTS
D031 D3 D1          0255         OUT   PORT1
D033 D3 D3          0260         OUT   PORT3
D035 D3 D5          0265         OUT   PORT5
D037 D3 D7          0270         OUT   PORT7
D039 D3 DB          0275         OUT   PORTB         CLOCK REG
D03B 97             0280         SUB   A             SET INPUT PORTS
D03C D3 D4          0285         OUT   PORT4
```

```
D03E D3 D0                 0290         OUT   PORT0          PAR IN
D040 3E 00                 0295         MVI   A,00           RESET VDM
D042 D3 C8                 0300         OUT   0C8H
D044 00 00                 0305         DW    0              CUSTOM
D046 00 00                 0310         DW    0
D048 00 00                 0315         DW    0
D04A 3E FF                 0320         MVI   A,0FFH         OUT PORT
D04C D3 D6                 0325         OUT   PORT6
D04E 3E 0F                 0330         MVI   A,0FH          CASS CONTROL
D050 D3 D2                 0335         OUT   PORT2
D052 3E 2C                 0340         MVI   A,2CH          ENABLE PORTS
D054 D3 D1                 0345         OUT   PORT1          2C=NEG STROBE
D056 D3 D3                 0350         OUT   PORT3          2E=POS STROBE
D058 D3 D7                 0355         OUT   PORT7
D05A 11 C4 DF              0360 INIT1   LXI   D,STK0         SW ON=K AND V
D05D 06 07                 0365         MVI   B,07H          KEYBD-VID
D05F DB D2                 0370         IN    PORT2          SW OFF=SERIAL
D061 07                    0375         RLC                  IN AND OUT
D062 07                    0380         RLC
D063 07                    0385         RLC
D064 67                    0390         MOV   H,A            SAVE A
D065 E6 02                 0395         ANI   02             MASK SWITCH
D067 12                    0400         STAX  D              INP DEVICE
D068 13                    0405         INX   D
D069 12                    0410         STAX  D              OUT DEVICE
D06A 7C                    0415         MOV   A,H            GET A
D06B 07                    0420         RLC   GET            OTHER SW
D06C E6 02                 0425         ANI   02H            MASK SW
D06E C6 2C                 0430         ADI   2CH            ADD FOR STR
D070 D3 D5                 0435         OUT   PORT5          SET STROBE POL
D072                       0440 **STROBE. SW ON=NEG , SW OFF=POS**
D072 21 B6 D7              0445         LXI   H,TABL2        SET OTHER OPTIONS
D075 13                    0450 INIT2   INX   D
D076 7E                    0455         MOV   A,M            GET  DATA
D077 12                    0460         STAX  D
D078 23                    0465         INX   H
D079 05                    0470         DCR   B
D07A C2 75 D0              0475         JNZ   INIT2
D07D 97                    0480         SUB   A              TURN OFF RELAYS
D07E D3 D2                 0485         OUT   PORT2
D080 3E 03                 0490         MVI   A,03H          SET UP SERIAL PORT
D082 D3 DC                 0495         OUT   PORTC
D084 3E 11                 0500         MVI   A,11H
D086 D3 DC                 0505         OUT   PORTC
D088 3E B6                  0510         MVI   A,0B6H          SET UP CLOCKS
D08A D3 DB                 0515         OUT   PORTB
D08C 3E 70                 0520         MVI   A,70H
D08E D3 DB                 0525         OUT   PORTB
D090 1B                    0530         DCX   D
D091 1A                    0535         LDAX  D
D092 D3 DA                 0540         OUT   PORTA
D094 13                    0545         INX   D
D095 1A                    0550         LDAX  D
D096 D3 DA                 0555         OUT   PORTA
D098 21 D5 DF              0560 STRT    LXI   H,STK17          INITIALIZE INPUT BUFFER
D09B 36 0D                 0565         MVI   M,0DH
D09D 23                    0570         INX   H
D09E 97                    0575         SUB   A
D09F BD                    0580         CMP   L
```

```
D0A0 C2 9E D0      0585        JNZ   STRT+3
D0A3 31 C0 DF      0590        LXI   SP,STACK
D0A6 21 E7 D7      0595        LXI   H,CMNTC
D0A9 CD 31 D2      0600        CALL  COMNT         INITIAL COMMENT
D0AC 0E 28         0605 COMMD  MVI   C,28H         BUFF LENG
D0AE 21 D5 DF      0610        LXI   H,STK17
D0B1 CD 5E D2      0615        CALL  INPM          GET COMMAND
D0B4 FE 08         0620        CPI   08H           A BACKSPACE?
D0B6 CA D4 D0      0625        JZ    DELET
D0B9 FE 7F         0630        CPI   7FH           A RUBOUT?
D0BB CA D4 D0      0635        JZ    DELET         IF SO, ERASE IT
D0BE FE 1B         0640        CPI   1BH           ESCAPE CHAR ?
D0C0 CA D4 D0      0645        JZ    DELET         GET RID OF IT
D0C3 FE 0D         0650        CPI   0DH           A CARRIAGE RETURN?
D0C5 CA FA D0      0655        JZ    CMD1          IF SO, EXECUTE COMMAND
D0C8 77            0660        MOV   M,A           PUT INTO BUFFER
D0C9 CD 99 D2      0665        CALL  WRITE         ECHO CHARACTER
D0CC 23            0670        INX   H
D0CD 0D            0675        DCR   C             END OF INPUT BUFFER?
D0CE C2 B1 D0      0680        JNZ   COMMD+5       NO, LOOP
D0D1 C3 CD D4      0685        JMP   ERROR         YES, TOO MUCH
D0D4 79            0690 DELET  MOV   A,C
D0D5 FE 28         0695        CPI   28H           END OF BUFFER?
D0D7 CA AC D0      0700        JZ    COMMD         YES, RETURN
D0DA 0C            0705        INR   C             NO, INCREMENT C
D0DB 2B            0710        DCX   H             DECREMENT POINTER
D0DC 36 0D         0715        MVI   M,0DH         ERASE CHARACTER
D0DE 3E 08         0720        MVI   A,08H         A BACKSPACE
D0E0 CD 99 D2      0725        CALL  WRITE
D0E3 C3 B1 D0      0730        JMP   COMMD+5
D0E6 7E            0735 CHCK   MOV   A,M           GET CHARACTER
D0E7 23            0740        INX   H             POINT NEXT
D0E8 D6 20         0745        SUI   20H           A SPACE ?
D0EA C8            0750        RZ
D0EB C6 20         0755        ADI   20H           RESTORE
D0ED CD 7E D1      0760        CALL  CONV4         CONVERT
D0F0 DA CD D4      0765        JC    ERROR         CARRY SET
D0F3 D6 0A         0770        SUI   0AH           LETTER ?
D0F5 DA CD D4      0775        JC    ERROR         NO
D0F8 3C            0780        INR   A
D0F9 C9            0785        RET
D0FA 21 D5 DF      0790 CMD1   LXI   H,STK17       START OF INPUT BUFFER
D0FD CD 8B D3      0795        CALL  CRLF          RET AND LINE FEED
D100 CD E6 D0      0800        CALL  CHCK          CHECK FIRST LETTER
D103 17            0805        RAL                 MULTIPLY BY 8
D104 17            0810        RAL
D105 17            0815        RAL
D106 47            0820        MOV   B,A           SAVE IT IN B
D107 CD E6 D0      0825        CALL  CHCK          CHECK 2ND LETTER
D10A CA 0E D1      0830        JZ    $+4           IT WAS A SPACE
D10D 23            0835        INX   H
D10E 80            0840        ADD   B             ADD FIRST LETTER VALUE
D10F 22 D1 DF      0845        SHLD  STK13         SAVE POINTER
D112 0E E4         0850        MVI   C,0E4H        COMD COUNT
D114 21 62 D7      0855        LXI   H,TABL1       COMMAND LOOK-UP TABLE
D117 BE            0860 CMD2   CMP   M             FIND IT?
D118 CA 25 D1      0865        JZ    CMD3          YES
D11B 23            0870        INX   H             NO
D11C 23            0875        INX   H
```

```
D11D 23          0880          INX   H
D11E 0C          0885          INR   C          INCREMENT COMMAND COUNTE
D11F CA CD D4    0890          JZ    ERROR      COULDN'T FIND IT
D122 C3 17 D1    0895          JMP   CMD2       LOOP
D125 23          0900 CMD3     INX   H          GET ADDRESS OF COMMAND
D126 5E          0905          MOV   E,M        SAVE IN D AND E
D127 23          0910          INX   H
D128 56          0915          MOV   D,M
D129 EB          0920          XCHG  PLACE      DE IN HL
D12A E9          0925          PCHL  PLACE      HL IN PC AND GO
D12B E5          0930 CONV0    PUSH  H          SAVE HL
D12C 11 00 00    0935          LXI   D,0000H    ZERO DE
D12F 2A D1 DF    0940          LHLD  STK13      GET POINTER TO BUFFER
D132 7E          0945          MOV   A,M
D133 FE 0D       0950          CPI   0DH        WAS IT A CARRIAGE RET?
D135 CA 70 D1    0955          JZ    CONV2+8    YES
D138 7E          0960          MOV   A,M
D139 FE 20       0965          CPI   20H        A SPACE?
D13B CA 47 D1    0970          JZ    CONV1      YES
D13E FE 0D       0975          CPI   0DH        A CARRIAGE RET?
D140 CA 47 D1    0980          JZ    CONV1      YES
D143 23          0985          INX   H          INCREMENT POINTER
D144 C3 38 D1    0990          JMP   CONV0+13   LOOP
D147 23          0995 CONV1    INX   H          INCREMENT POINTER
D148 22 D1 DF    1000          SHLD  STK13      SAVE POINTER
D14B 2B          1005          DCX   H          GET CHARACTER
D14C CD 68 D1    1010          CALL  CONV2      GET LOWER NIBBLE
D14F 5F          1015          MOV   E,A        SAVE IN E
D150 CD 68 D1    1020          CALL  CONV2      GET NIBBLE
D153 07          1025          RLC              ROTATE
D154 07          1030          RLC
D155 07          1035          RLC
D156 07          1040          RLC
D157 83          1045          ADD   E          GET 1ST HALF
D158 5F          1050          MOV   E,A        NOW HAVE 1ST BYTE
D159 CD 68 D1    1055          CALL  CONV2      GET 2ND BYTE
D15C 57          1060          MOV   D,A
D15D CD 68 D1    1065          CALL  CONV2
D160 07          1070          RLC
D161 07          1075          RLC
D162 07          1080          RLC
D163 07          1085          RLC
D164 82          1090          ADD   D
D165 57          1095          MOV   D,A
D166 E1          1100          POP   H
D167 C9          1105          RET
D168 2B          1110 CONV2    DCX   H          GET DATA
D169 7E          1115          MOV   A,M
D16A FE 20       1120          CPI   20H        DONE?
D16C C2 72 D1    1125          JNZ   CONV3      NO
D16F E1          1130          POP   H          GET RID OF LAST CALL
D170 E1          1135          POP   H
D171 C9          1140          RET
D172 CD 7E D1    1145 CONV3    CALL  CONV4      CONVERT
D175 DA CD D4    1150          JC    ERROR      CARRY SET
D178 FE 10       1155          CPI   10H        A NUMBER ?
D17A D2 CD D4    1160          JNC   ERROR      NO
D17D C9          1165          RET
D17E D6 30       1170 CONV4    SUI   30H        ASC BIAS
```

```
D180 D8          1175       RC               NOT LET OR NUM
D181 FE 0A       1180       CPI   0AH
D183 3F          1185       CMC
D184 D0          1190       RNC              A NUMBER
D185 FE 2B       1195       CPI   2BH        A LOW CASE ?
D187 D4 92 D1    1200       CNC   LCL
D18A 3F          1205       CMC
D18B D8          1210       RC               NO GOOD
D18C FE 11       1215       CPI   11H
D18E D8          1220       RC               NO GOOD
D18F D6 07       1225       SUI   07H        A LETTER
D191 C9          1230       RET
D192 D6 20       1235 LCL   SUI   20H        CONV LOW TO UP
D194 FE 2B       1240       CPI   2BH        NOT LET ?
D196 C9          1245       RET
D197 CD 2B D1    1250 GO    CALL  CONVC      GET ADDRESS
D19A EB          1255       XCHG             PUT IN HL
D19B E9          1260       PCHL             PUT IN PC AND GO
D19C CD 2B D1    1265 EM0   CALL  CONVC      GET ADDRESS
D19F EB          1270       XCHG             SAVE IN HL
D1A0 CD B9 D3     1275      CALL   RTHL       WRITE ADDRESS
D1A3 7E          1280       MOV   A,M
D1A4 CD 9C D3    1285       CALL  RTHEX      WRITE BYTE
D1A7 CD 96 D3    1290       CALL  SPACE
D1AA CD 5E D2    1295       CALL  INPM       GET CHAR
D1AD CD 99 D2    1300       CALL  WRITE      ECHO IT
D1B0 FE 20       1305       CPI   20H        A SPACE ?
D1B2 CA BD D1    1310       JZ    EM1        YES
D1B5 FE 08       1315       CPI   08H        A BACKSPACE?
D1B7 CA C8 D1    1320       JZ    EM2        YES
D1BA CD E9 D1    1325       CALL  EM4
D1BD 7E          1330 EM1   MOV   A,M        GET MEM BYTE
D1BE CD 9C D3    1335       CALL  RTHEX      WRITE IT
D1C1 23          1340       INX   H
D1C2 CD 8B D3    1345       CALL  CRLF
D1C5 C3 A0 D1    1350       JMP   EM0+4
D1C8 2B          1355 EM2   DCX   H          DECREMENT HL
D1C9 C3 C2 D1    1360       JMP   EM1+5
D1CC CD 2B D1    1365 EM3   CALL  CONVC      GET ADDR
D1CF EB          1370       XCHG  PUT        IN HL
D1D0 0E 10       1375       MVI   C,10H      BYTE COUNT
D1D2 CD B9 D3    1380       CALL   RTHL
D1D5 CD 5E D2    1385       CALL  INPM
D1D8 CD 99 D2    1390       CALL  WRITE
D1DB CD E9 D1    1395       CALL  EM4
D1DE 23          1400       INX   H
D1DF 0D          1405       DCR   C          DEC COUNT
D1E0 C2 D5 D1    1410       JNZ   EM3+9
D1E3 CD 8B D3    1415       CALL  CRLF
D1E6 C3 D0 D1    1420       JMP   EM3+4
D1E9 CD 72 D1    1425 EM4   CALL  CONV3      CONVERT BYTE
D1EC 07          1430       RLC   ROTATE     NIBBLE
D1ED 07          1435       RLC
D1EE 07          1440       RLC
D1EF 07          1445       RLC
D1F0 47          1450       MOV   B,A        SAVE IN B
D1F1 CD 5E D2    1455       CALL  INPM
D1F4 CD 99 D2    1460       CALL  WRITE
D1F7 CD 72 D1    1465       CALL  CONV3
```

```
D1FA 80          1470          ADD   B           GET NIBBLE
D1FB 77          1475          MOV   M,A         STORE
D1FC C3 96 D3    1480          JMP   SPACE
D1FF CD 2B D1    1485  PROG0   CALL  CONVD       PROGRAM 2708
D202 EB          1490          XCHG              SAVE IN HL
D203 CD 2B D1    1495          CALL  CONVD
D206 3E A0       1500          MVI   A,0A0H      DO 160X
D208 32 CD CF    1505  PROG1   STA   STK9        SAVE A
D20B D5          1510          PUSH  D
D20C E5          1515          PUSH  H
D20D 01 00 04    1520          LXI   B,0400H
D210 CD C5 D3    1525          CALL  ESCAP       WANT TO QUIT?
D213 7E          1530  PROG2   MOV   A,M         GET BYTE
D214 12          1535          STAX  D           PLACE IN 2708
D215 13          1540          INX   D
D216 CD 2B D2    1545          CALL  DONE
D219 C2 13 D2    1550          JNZ   PROG2       LOOP
D21C E1          1555          POP   H           GET HL
D21D D1          1560          POP   D           GET DE
D21E 3A CD CF    1565          LDA   STK9        GET COUNT BYTE
D221 3D          1570          DCR   A           DECREMENT IT
D222 C2 08 D2    1575          JNZ   PROG1       LOOP
D225 01 00 04    1580  VERO    LXI   B,0400H
D228 C3 26 D5    1585          JMP   VERFY+6     VERIFY DATA
D22B 23          1590  DONE    INX   H
D22C 0B          1595          DCX   B
D22D 97          1600          SUB   A
D22E B0          1605          ORA   B
D22F B1          1610          ORA   C
D230 C9          1615          RET
D231 7E          1620  COMNT   MOV   A,M         GET BYTE
D232 CD 99 D2    1625          CALL  WRITE       WRITE IT
D235 23          1630          INX   H
D236 B7          1635          ORA   A           BIT 7 HIGH ?
D237 F2 31 D2    1640          JP    COMNT       NO, LOOP
D23A C9          1645          RET
D23B CD 3E D2    1650  DELA2   CALL  DELAY       TWO TIMES
D23E C5          1655  DELAY   PUSH  B           SAVE B
D23F 01 00 70    1660          LXI   B,7000H
D242 03          1665          INX   B
D243 97          1670          SUB   A
D244 B0          1675          ORA   B
D245 B1          1680          ORA   C           DONE ?
D246 C2 42 D2    1685          JNZ   DELAY+4
D249 C1          1690          POP   B           RESTORE BC
D24A C9          1695          RET
D24B C5          17000 DELAS   PUSH  B
D24C 01 00 B8    1705          LXI   B,0B800H    HALF AS MUCH DELAY
D24F C3 42 D2    1710          JMP   DELAY+4
D252 CD 86 D2    1715  ISTAT   CALL  CHKST       GET STATUS
D255 E6 80       1720          ANI   80H         BIT 7
D257 C8          1725          RZ                Z= NO DATA
D258 CD 6A D2    1730  ICHAR   CALL  INPUT+6     GET CHAR
D25B E6 7F       1735          ANI   7FH         MASK BIT 7
D25D C9          1740          RET
D25E CD 64 D2    1745  INPM    CALL  INPUT
D261 E6 7F       1750          ANI   7FH         MASK BITS 0-6
D263 C9          1755          RET
D264 CD 52 D2    1760  INPUT   CALL  ISTAT       CHECK STATUS
```

```
D267  CA 64 D2     1765         JZ     INPUT      LOOP TIL READY
D26A  3A C4 DF     1770         LDA    STK0       INPUT DEVICE
D26D  B7           1775         ORA    A
D26E  CA 78 D2     1780         JZ     PIN0       PRIMARY PARALLEL INPUT
D271  3D           1785         DCR    A
D272  CA 82 D2     1790         JZ     PIN1       OTHER PARALLEL INPUT
D275  DB DD        1795         IN     PORTD      INPUT FROM SERIAL
D277  C9           1800         RET
D278  DB D4        1805 PIN0    IN     PORT4
D27A  00           1810         NOP               CMA IF INV DATA
D27B  C9           1815         RET
D27C  DB D1        1820 PIN     IN     PORT1      CHECK STATUS
D27E  B7           1825         ORA    A          CHECK BIT 7
D27F  F2 7C D2     1830         JP     PIN        LOOP
D282  DB D0        1835 PIN1    IN     PORT0
D284  00           1840         NOP               CMA IF INV DAT
D285  C9           1845         RET
D286  3A C4 DF     1850 CHKST   LDA    STK0       CHECK STATUS
D289  B7           1855         ORA    A
D28A  CA 95 D2     1860         JZ     PINS0      PAR STAT
D28D  3D           1865         DCR    A
D28E  DB D1        1870         IN     PORT1      STATUS PORT
D290  C8           1875         RZ
D291  DB DD        1880         IN     PORTD      SERIAL STATUS
D293  0F           1885         RRC               GET BIT 0
D294  C9           1890         RET
D295  DB D5        1895 PINS0   IN     PORT5      STATUS PORT
D297  C9           1900         RET
D298  78           1905 WRITB   MOV    A,B        GET DATA
D299  F5           1910 WRITE   PUSH   PSW        SAVE A
D29A  3A C5 DF     1915         LDA    STK1       OUT DEVICE
D29D  B7           1920         ORA    A
D29E  CA BB D2     1925         JZ     VDM        VIDEO ROUTINE
D2A1  3D           1930         DCR    A
D2A2  CA B1 D2     1935         JZ     POUT+1
D2A5  DB DD        1940 SOUT    IN     PORTD
D2A7  E6 02        1945         ANI    02H        READY ?
D2AA9 CA A5 D2     1950         JZ     SOUT       LOOP
D2AC  F1           1955         POP    PSW        GET A
D2AD  D3 DD        1960         OUT    PORRTD
D2AF  C9           1965         RET
D2B0  F5           1970 POUT    PUSH   PSW        SAVE A
D2B1  DB D7        1975         IN     PORT7      CHECK STATUS
D2B3  B7           1980         ORA    A          BIT 7 HIGH ?
D2B4  F2 B1 D2     1985         JP     POUT+1
D2B7  F1           1990         POP    PSW        GET A
D2B8  D3 D6        1995         OUT    PORT6
D2BA  C9           2000         RET
D2BB  F1           2005 VDM     POP    PSW        GET A
D2BEC C5           2010         PUSH   B          SAVE REG'S
D2BD  D5           2015         PUSH   D
D2BE  E5           2020         PUSH   H
D2BF  F5           2025         PUSH   PSW
D2C0  2A CF DF     2030         LHLD   STK11      GET VDM POINTER
D2C3  FE 0C        2035         CPI    0CH        ERASE SCREEN ?
D2C5  CA 44 D3     2040         JZ     FF         YES
D2C8  FE 0D        2045         CPI    0DH        CARRIAGE RETURN ?
D2CA  CA 54 D3     20500        JZ     CR
D2CD  FE 0A        2055         CPI    0AH        LINE FEED ?
```

```
D2CF CA 6E D3      2060           JZ     LF
D2D2 FE 08         2065           CPI    08H         A BACKSPACE ?
D2D4 CA 7D D3      2070           JZ     BS
D2D7 FE 7F         2075           CPI    7FH         A RUBOUT ?
D2D9 CA 7D D3      2080           JZ     BS
D2DC FE 0B         2085           CPI    0BH         CTRL K,HOME ?
D2DE CA 83 D3      2090           JZ     HOME
D2E1 E6 7F         2095           ANI    7FH         POLY-E6 80
D2E3 77            2100           MOV    M,A          PUT ON SCREEN
D2E4 23            2105           INX    H
D2E5 CD 3C D3      2110 VD1       CALL   VD3          END OF SCREEN?
D2E8 DA 32 D3      2115           JC     VD2
D2EB 3A C6 DF      2120 SC1       LDA    STK2        GET SCROLL SPEED
D2EE 57            2125           MOV    D,A
D2EF 1E FF         2130           MVI    E,0FFH
D2F1 13            2135           INX    D
D2F2 97            2140           SUB    A
D2F3 B2            2145           ORA    D
D2F4 B3            2150           ORA    E           DONE ?
D2F5 C2 F1 D2      2155           JNZ    SC1+6       LOOP
D2F8 CD 52 D2      2160 SC2       CALL   ISTAT       CHANGE SPEED?
D2FB CA 19 D3      2165           JZ     SC3         NO
D2FE FE 20         2170           CPI    20H         A SPACE?
D300 C2 06 D3      2175           JNZ    $+3
D303 CD 5E D2      2180           CALL   INPM        WAIT FOR CHAR
D306 FE 3A         2185           CPI    3AH         GREATER THAN 9 ?
D308 D2 19 D3      2190           JNC    SC3         YES
D30B D6 30         2195           SUI    30H         SUB ASCII BIAS
D30D DA 19 D3      2200           JC     SC3         NOT 0-9
D310 17            2205           RAL                MULTIPLY BY 16
D311 17            2210           RAL
D312 17            2215           RAL
D313 17            2220           RAL
D314 C6 6C         2225           ADI    6CH
D316 32 C6 DF      2230           STA    STK2        STORE SPEED
D319 21 40 CC      2235 SC3       LXI    H,0CC40H    START SCROLL
D31C 11 00 CC      2240           LXI    D,0CC00H    START OF SCREEN
D31F 7E            2245           MOV    A,M         GET BYTE
D320 12            2250           STAX   D           MOVE IT
D321 13            2255           INX    D
D322 23            2260           INX    H
D323 CD 3C D3      2265           CALL   VD3         DONE?
D326 DA 1F D3      2270           JC     SC3+6
D329 2B            2275           DCX    H
D32A 36 20         2280           MVI    M,20H       POLY-20
D32C 7D            2285           MOV    A,L
D32D FE C0         2290           CPI    0C0H        DONE ?
D32F C2 29 D3      2295           JNZ    SC3+16
D332 36 A0         2300 VD2       MVI    M,0A0H      POLY-0FFH
D334 22 CF DF      2305           SHLD   STK11       SAVE POINTER
D337 F1            2310           POP    PSW         RESTORE REG'S
D338 E1            2315           POP    H
D339 D1            2320           POP    D
D33A C1            2325           POP    B
D33B C9            2330           RET    DONE
D33C 7C            2335 VD3       MOV    A,H
D33D FE D0         2340           CPI    0D0H        C7-80X24
D33F C9            2345           RET                RC-80X24
D340 7D            2350           MOV    A,L         GET L
```

```
D341 FE 80           2355          CPI   80H            END OF LINE
D343 C9              2360          RET
D344 21 FF CF        2365 FF       LXI   H,0CFFFH       END OF SCREEN
D347 36 20           2370          MVI   M,20H          POLY-AC
D349 2B              2375          DCX   H
D34A 7C              2380          MOV   A,H
D34B FE CB           2385          CPI   0CBH           DONE ?
D34D C2 47 D3        2390          JNZ   FF+3
D350 23              2395          INX   H
D351 C3 32 D3        2400          JMP   VD2
D354 36 20           2405 CR       MVI   M,20H          POLY-AC
D356 7D              2410          MOV   A,L
D357 E6 C0           2415          ANI   0C0H
D359 6F              2420          MOV   L,A
D35A 7E              2425          MOV   A,M            SAVE BYTE
D35B 32 CE CF        2430          STA   STK10
D35E C3 32 D3        2435          JMP   VD2
D361                 2440 ***THE FOLLOWING  FOR 80X24***
D361 7A              2445          MOV   A,D
D362 9C              2450          SBB   H
D363 D1              2455          POP   D
D364 DA 5D D3        2460          JC    CR+9
D367 7E              2465          MOV   A,M
D368 32 CE CF        2470          STA   STK10          SAVE CHAR
D36B C3 32 D3        2475          JMP   VD2
D36E 3A CE CF        2480 LF       LDA   STK10          GET BYTE
D371 77              2485          MOV   M,A            PUT ON SCREEN
D372 11 40 00        2490          LXI   D,0040H
D375 19              2495          DAD   D              LINE-FEEDS HL
D376 7E              2500          MOV   A,M            GET BYTE
D377 32 CE CF        2505          STA   STK10
D37A C3 E5 D2        2510          JMP   VD1
D37D 36 20           2515 BS       MVI   M,20H          POLY-AC
D37F 2B              2520          DCX   H
D380 C3 E5 D2        2525          JMP   VD1
D383 36 20           2530 HOME     MVI   M,20H          POLY-AC
D385 21 00 CC        2535          LXI   H,0CC00H       START OF SCREEN
D388 C3 32 D3        2540          JMP   VD2
D38B 3E 0D           2545 CRLF     MVI   A,0DH          CARRIAGE RETURN
D38D CD 99 D2        2550          CALL  WRITE
D390 3E 0A           2555          MVI   A,0AH          LINE FEED
D392 CD 99 D2        2560          CALL  WRITE
D395 C9              2565          RET
D396 3E 20           2570 SPACE    MVI   A,20H          SPACE
D398 CD 99 D2        2575          CALL  WRITE
D39B C9              2580          RET
D39C F5              2585 RTHEX    PUSH  PSW
D39D 1F              2590          RAR                  GET UPPER NIBBLE
D39E 1F              2595          RAR
D39F 1F              2600          RAR
D3A0 1F              2605          RAR
D3A1 CD AF D3        2610          CALL  BINAS
D3A4 CD 99 D2        2615          CALL  WRITE
D3A7 F1              2620          POP   PSW            GET BYTE AGAIN
D3A8  CD AF D3        2625          CALL  BINAS
D3AB CD 99 D2        2630          CALL  WRITE
D3AE C9              2635          RET
D3AF E6 0F           2640 BINAS    ANI   0FH            BIN TO ASCII
D3B1 C6 30           2645          ADI   30H
```

```
D3B3 FE 3A        2650       CPI   3AH        OK?
D3B5 D8           2655       RC    YES
D3B6 C6 07        2660       ADI   07H
D3B8 C9           2665       RET              NOW ITS DONE
D3B9 7C           2670 RTHL  MOV   A,H        WRITE HL
D3BA CD 9C D3     2675       CALL  RTHEX
D3BD 7D           2680       MOV   A,L
D3BE CD 9C D3     2685       CALL  RTHEX
D3C1 CD 96 D3     2690       CALL  SPACE
D3C4 C9           2695       RET
D3C5 CD 58 D2     2700 ESCAP CALL  ICHAR      GET CHAR
D3C8 FE 1B        2705       CPI   1BH        ESCAPE ?
D3CA C0           2710       RNZ
D3CB 21 BD D7     2715       LXI   H,CMNT6
D3CE C3 05 D5     2720       JMP   CMPLT+3
D3D1 DB D2        2725 CASR0 IN    PORT2      IS RELAY ON ?
D3D3 E6 01        2730       ANI   01H
D3D5 C2 1A D4     2735       JNZ   CASR1
D3D8 DB D2        2740       IN    PORT2      GET DATA AT PORT2
D3DA F6 01        2745       ORI   01H        TURN ON READ RELAY
D3DC D3 D2        2750       OUT   PORT2
D3DE 3E 32        2755       MVI   A,32H      SET UP READ CLOCK
D3E0 D3 DE        2760       OUT   PORTE
D3E2 3A C7 DF     2765       LDA   STK3       GET READ SPEED
D3E5 D3 D8        2770       OUT   PORT8
D3E7 3A C8 DF     2775       LDA   STK3+1
D3EA D3 D8        2780       OUT   PORT8
D3EC CD C5 D3     2785       CALL  ESCAP      WANT TO QUIT ?
D3EF DB D2        2790       IN    PORT2      LEVEL DETECTION
D3F1 B7           2795       ORA   A          BIT 7 HIGH ?
D3F2 F2 EC D3     2800       JP    CASR0+27   LOOP TIL TONE
D3F5 CD 3E D2     2805       CALL  DELAY
D3F8 DB D2        2810       IN    PORT2      CHECK LEVEL AGAIN
D3FA B7           2815       ORA   A          BIT 7 HIGH ?
D3FB F2 EC D3     2820       JP    CASR0+27   LOOP TIL TONE
D3FE 3E 80        2825       MVI   A,80H      SET UP SSDA
D400 D3 DE        2830       OUT   PORTE      RESET RECEIVER
D402 3E B8        2835       MVI   A,0B8H     8-BIT WORD,SM,1 BYTE
D404 D3 DF        2840       OUT   PORTF
D406 3E 82        2845       MVI   A,82H      WRITE TO CON REG 3
D408 D3 DE        2850       OUT   PORTE
D40A 3E 70        2855       MVI   A,70H      INT,ONE SYNC CLR STAT
D40C D3 DF        2860       OUT   PORTF
D40E 3E 81        2865       MVI   A,81H      WRITE TO SYNC REG
D410 D3 DE        2870       OUT   PORTE
D412 3E E6        2875       MVI   A,0E6H     SYNC CODE
D414 D3 DF        2880       OUT   PORTF
D416 3E 03        2885       MVI   A,03H      ENABLE X-MIT
D418 D3 DE        2890       OUT   PORTE
D41A CD C5 D3     2895 CASR1 CALL  ESCAP      WANT TO QUIT ?
D41D DB DE        2900       IN    PORTE      CHECK STATUS
D41F B7           2905       ORA   A          BIT 7 HIGH ?
D420 F2 1A D4     2910       JP    CASR1      NOT READY
D423 DB DF        2915       IN    PORTF      GET DATA
D425 C9           2920       RET
D426 CD D1 D3     2925 CASR2 CALL  CASR0      GET SA AND LEN
D429 06 00        2930       MVI   B,00H      ZERO CHECKSUM REG
D42B 6F           2935       MOV   L,A        PUT ADDRESS IN HL
D42C 47           2940       MOV   B,A
```

```
D42D CD 1A D4            2945         CALL  CASR1
D430 67                  2950         MOV   H,A
D431 80                  2955         ADD   B
D432 47                  2960         MOV   B,A
D433 CD 1A D4            2965         CALL  CASR1
D436 57                  2970         MOV   D,A      PUT BLOCK LEN IN DE
D437 80                  2975         ADD   B
D438 47                  2980         MOV   B,A
D439 CD 1A D4            2985         CALL  CASR1
D43C 5F                  2990         MOV   E,A
D43D 80                  2995         ADD   B
D43E 47                  3000         MOV   B,A
D43F C9                  3005         RET
D440 CD 1A D4            3010 CASR3   CALL  CASR1      GET DATA NOW
D443 77                  3015         MOV   M,A      PUT IN MEMORY
D444 80                  3020         ADD   B
D445 47                  3025         MOV   B,A
D446 23                  3030         INX   H
D447 1B                  3035         DCX   D
D448 97                  3040         SUB   A        ZERO  A
D449 B2                  3045         ORA   D        IS D ZERO ?
D44A B3                  3050         ORA   E        IS E ZERO ?
D44B C2 40 D4            3055         JNZ   CASR3      NO, LOOP
D44E CD 1A D4            3060         CALL  CASR1      GET CHECKSUM
D451 90                  3065         SUB   B        COMPARE WITH B
D452 57                  3070         MOV   D,A      SAVE RESULT IN D
D453 CD D6 D4            3075         CALL  END
D456 97                  3080         SUB   A
D457 BA                  3085         CMP   D        CHECKSUM OK ?
D458 21 DB C7            3090         LXI   H,CMNT2    TAPE ERROR
D45B CC 64 D4            3095         CZ    CASR4      TAPE OK
D45E CD 31 D2            3100         CALL  COMNT
D461 C3 98 D0            3105         JMP   STRT
D464 21 CA D7            3110 CASR4   LXI   H,CMNT4    COMPLETE
D467 C9                  3115         RET
D468 CD 2B C1            3120 CASR5   CALL  CONVC      READ AND SPECIFY
D46B EB                  3125         XCHG             SAVE IN H
D46C E5                  3130         PUSH  H
D46D 2A D1 DF            3135         LHLD  STK13
D470 7E                  3140         MOV   A,M
D471 FE 0D               3145         CPI   0DH      A CARRIAGE RET ?
D473 E1                  3150         POP   H
D474 C2 9A D4            3155         JNZ   CASR6      GET ANOTHER ADDRESS
D477 CD D1 D3            3160         CALL  CASRC
D47A 06 00               3165         MVI   B,00H    ZERO CHECKSUM REG
D47C 77                  3170         MOV   M,A      PLACE BYTE IN MEMORY
D47D 80                  3175         ADD   B        KEEP CHECKSUM
D47E 47                  3180         MOV   B,A
D47F 23                  3185         INX   H
D480 CD 1A D4            3190         CALL  CASR1
D483 77                  3195         MOV   M,A
D484 80                  3200         ADD   B
D485 47                  3205         MOV   B,A
D486 23                  3210         INX   H
D487 CD 1A D4            3215         CALL  CASR1
D48A 77                  3220         MOV   M,A
D48B 57                  3225         MOV   D,A
D48C 80                  3230         ADD   B
D48D 47                  3235         MOV   B,A
```

```
D48E 23           3240        INX   H
D48F CD 1A D4     3245        CALL  CASR1
D492 77           3250        MOV    M,A
D493 5F           3255        MOV   E,A
D494 80           3260        ADD   B
D495 47           3265        MOV    B,A
D496 23           3270        INX   H
D497 C3 40 D4     3275        JMP   CASR3
D49A CD EA D4     3280 CASR6  CALL  BKLEN+4        GET 2ND ADDR
D49D 50           3285        MOV   D,B
D49E 59           3290        MOV   E,C
D49F CD D1 D3     3295        CALL  CASRD
D4A2 06 00        3300        MVI   B,00H          ZERO CHECKSUM REG
D4A4 C3 43 D4     3305        JMP   CASR3+3
D4A7 CD 26 D4     3310 CASR7  CALL  CASR2          RC PROGRAM
D4AA C3 40 D4     3315        JMP   CASR3
D4AD CD 26 D4     3320 CASR8  CALL  CASR2          RV PROGRAM
D4B0 CD 1A D4     3325        CALL  CASR1
D4B3 BE           3330        CMP   M              MEMORY SAME AS TAPE?
D4B4 C2 C4 D4     3335        JNZ   VTERR          ERROR
D4B7 80           3340        ADD   B
D4B8 47           3345        MOV   B,A            KEEP CHECKSUM
D4B9 23           3350        INX   H
D4BA 1B           3355        DCX   D
D4BB 97           3360        SUB   A
D4BC B2           3365        ORA   D
D4BD B3           3370        ORA   E
D4BE C2 B0 D4     3375        JNZ   CASR8+3
D4C1 C3 4E D4     3380        JMP   CASR3+14
D4C4 CD D6 D4     3385 VTERR  CALL  END            SHUT OFF RELAY
D4C7 21 D4 D7     3390        LXI   H,CMNT3
D4CA CD 31 D2     3395        CALL  COMNT
D4CD CD 8B D3     3400 ERROR  CALL  CRLF
D4D0 21 E0 D7     3405        LXI   H,CCMNT1        ERROR
D4D3 C3 05 D5     3410        JMPP   CMPLT+3
D4D6 CD C5 D3     3415 END    CALL  ESCAP          QUIT ?
D4D9 DB D2        3420        IN    PORT2          CHECK LEVEL
D4DB B7           3425        ORA   A              BIT 7 HIGH ?
D4DC FA D6 D4     3430        JM    END            LOOP TIL NO LEVEL
D4DF DB D2        3435        IN    PORT2          A GAP
D4E1 E6 FE        3440        ANI   0FEH           MASK RELAY
D4E3 D3 D2        3445        OUT   PORT2          RESTORE
D4E5 C9           3450        RET
D4E6 CD 2B D1     3455 BKLEN  CALL  CONVD
D4E9 EB           3460        XCHG  SAVE           IN HL
D4EA CD 2B D1     3465        CALL  CONVD
D4ED 13           3470        INX   D
D4EE 7B           3475        MOV   A,E
D4EF 95           3480        SUB   L              CALCULATE LENGTH
D4F0 4F           3485        MOV   C,A
D4F1 7A           3490        MOV   A,D
D4F2 9C           3495        SBB   H
D4F3 47           3500        MOV   B,A
D4F4 C9           3505        RET
D4F5 CD E6 D4     3510 ZERO   CALL  BKLEN          ZERO A MEMORY
D4F8 CD 2B D1     3515        CALL  CONVD          NO ZERO ?
D4FB 73           3520        MOV   M,E
D4FC CD 2B D2     3525        CALL  DONE
D4FF C2 FB D4     3530        JNZ   ZERO+6         LOOP TIL DONE
```

```
D502 21 CA D7              3535 CMPLT  LXI   H,CMNT4        COMPLETE
D505 CD 31 D2              3540        CALL  COMNT
D508 C3 98 D0              3545        JMP   STRT
D50B CD E6 D4              3550 MOVE   CALL  BKLEN          MOVE A BLOCK OF MEM
D50E CD 2B D1              3555        CALL  CONVD          GET START ADDR
D511 7E                    3560        MOV   A,M            GET BYTE
D512 12                    3565        STAX  D              MOVE A BYTE
D513 13                    3570        INX   D
D514 CD 2B D2              3575        CALL  DONE
D517 C2 11 D5              3580        JNZ   MOVE+6         LOOP TIL DONE
D51A 21 D7 DF              3585        LXI   H,STK17+2
D51D 22 D1 DF              3590        SHLD  STK13
D520 CD E6 D4              3595 VERFY  CALL  BKLEN          VERIFY MEMORY
D523 CD 2B D1              3600        CALL  CONVD          GET ADDRESS
D526 CD C5 D3              3605        CALL  ESCAP          WANT TO QUIT ?
D529 1A                    3610        LDAX  D
D52A BE                    3615        CMP   M              COMPARE
D52B C2 38 D5              3620        JNZ   VMERR          ERROR
D52E 13                    3625        INX   D
D52F CD 2B D2              3630        CALL  DONE
D532 C2 26 D5              3635        JNZ   VERFY+6        LOOP
D535 C3 02 D5              3640        JMP   CMPLT
D538 E5                    3645 VMERR  PUSH  H              SAVE H
D539 21 D4 D7              3650        LXI   H,CMNT3
D53C CD 31 D2              3655        CALL  COMNT
D53F 21 E0 D7              3660        LXI   H,CMNT1
D542 CD 31 D2              3665        CALL  COMNT
D545 E1                    3670        POP   H
D546 CD B9 D3              3675        CALL  RTHL           WRITE ADDRESS
D549 7E                    3680        MOV   A,M            WRITE BYTE
D54A CD 9C D3              3685        CALL  RTHEX
D54D CD 96 D3              3690        CALL  SPACE
D550 CD 5D D5              3695        CALL  RTDE           WRITE OTHER BYTE
D553 1A                    3700        LDAX  D
D554 CD 9C D3              3705        CALL  RTHEX
D557 CD 8B D3              3710        CALL  CRLF
D55A C3 2E D5              3715        JMP   VERFY+14
D55D 7A                    3720 RTDE   MOV   A,D            WRITE DE
D55E CD 9C D3              3725        CALL  RTHEX
D561 7B                    3730        MOV   A,E
D562 C3 BE D3              3735        JMP   RTHL+5
D565 CD E6 D4              3740 WC0    CALL  BKLEN          WC WRITE CASS
D568 7D                    3745        MOV   A,L            WRITE ADDRESS
D569 16 00                 3750        MVI   D,00H          ZERO CHECKSUM REG
D56B CD 9DD D5              3755         CALL  CASW0
D56E 57                    3760        MOV   D,A            CHECKSUM IN D
D56F 7C                    3765        MOV   A,H
D570 CD 9D D5              3770        CALL  CASW0
D573 82                    3775        ADD   D
D574 57                    3780        MOV   D,A
D575 78                    3785        MOV   A,B            WRITE BLOCK LEN
D576 CD 9D D5              3790        CALL  CASW0
D579 82                    3795        ADD   D
D57A 57                    3800        MOV   D,A
D57B 79                    3805        MOV   A,C
D57C CD 9D D5              3810        CALL  CASW0
D57F 82                    3815        ADD   D
D580 57                    3820        MOV   D,A
D581 7E                    3825 WC1    MOV   A,M            GET MEMORY BYTE
```

```
D582 CD 9D D5          3830        CALL  CASW0       OUTPUT
D585 82                3835        ADD   D           ADD CHECKSUM
D586 57                3840        MOV   D,A         SAVE IN D
D587 CD 2E C2          3845        CALL  DONE
D58A C2 81 D5          3850        JNZ   WC1         LOOP TIL DONE
D58D 7A                3855        MOV   A,D
D58E CD 9D D5          3860        CALL  CASW0
D591 CD EE D5          3865        CALL  CASW2
D594 21 C3 D7          3870        LXI   H,CMNT5     WRITTEN
D597 CD 31 C2          3875        CALL  COMNT
D59A C3 98 C0          3880        JMP   STRT
D59D F5                3885 CASW0  PUSH  PSW         SAVE A
D59E DB D2             3890        IN    PORT2       IS WRITE RELAY ON
D5A0 E6 02             3895        ANI   02H
D5A2 C2 E0 D5          3900        JNZ   CASW1
D5A5 DB D2             3905        IN    PORT2       TURN RELAY ON
D5A7 F6 02             3910        ORI   02H
D5A9 D3 D2             3915        OUT   PORT2
D5AB 3E 40             3920        MVI   A,40H       SET UP TRANSMIT
D5AD D3 DE             3925        OUT   PORTE
D5AF 3E B8             3930        MVI   A,0B8H
D5B1 D3 DF             3935        OUT   PORTF
D5B3 3E 02             3940        MVI   A,02H       CONT REG 3
D5B5 D3 DE             3945        OUT   PORTE
D5B7 3E 70             3950        MVI   A,70H       CLEAR CTS TU
D5B9 D3 DF             3955        OUT   PORTF
D5BB 3E 03             3960        MVI   A,03H       ENABLE X-MIT
D5BD D3 DE             3965        OUT   PORTE
D5BF CD 3E C2          3970        CALL  DELAY
D5C2 3E 76             3975        MVI   A,76H
D5C4 D3 DB             3980        OUT   PORTB       SET UP CLOCK
D5C6 3A C9 DF          3985        LDA   STK5        GET SPEED
D5C9 D3 D9             3990        OUT   PORT9
D5CB 3A CA DF          3995        LDA   STK5+1
D5CE D3 D9             4000        OUT   PORT9
D5D0 CD 3B C2          4005        CALL  DELA2
D5D3 CD 3E C2          4010        CALL  DELAY
D5D6 3E 3C             4015        MVI   A,3CH       OUTPUT CLOCK BYTE
D5D8 CD 9D D5          4020        CALL  CASW0
D5DB 3E E6             4025        MVI   A,0E6H      OUTPUT SYNC BYTE
D5DD CD 9D D5          4030        CALL  CASW0
D5E0 CD C5 C3          4035 CASW1  CALL  ESCAP
D5E3 DB DE             4040        IN    PORTE       STATUS
D5E5 E6 40             4045        ANI   40H         READY ?
D5E7 CA E0 D5          4050        JZ    CASW1
D5EA F1                4055        POP   PSW         GET A
D5EB D3 DF             4060        OUT   PORTF       OUTPUT
D5ED C9                4065        RET
D5EE CD 4B C2          4070 CASW2  CALL  DELAS       WRITE A TRAILER
D5F1 3E 70             4075        MVI   A,70H       TURN OFF CLOCK
D5F3 D3 DB             4080        OUT   PORTB
D5F5 CD 3E C2          4085        CALL  DELAY       WRITE A GAP
D5F8 DB D2             4090        IN    PORT2       GET PORT DATA
D5FA E6 FD             4095        ANI   0FDH        TURN OFF RELAY
D5FC D3 D2             4100        OUT   PORT2
D5FE C9                4105        RET
D5FF 2A D1 DF          4110 AIO    LHLD  STK13       SET INPUT DEVICE
D602 CD E6 C0          4115        CALL  CHCK
D605 06 00             4120        MVI   B,00H
```

```
D607 FE 0B        4125         CPI    0BH       K FOR KEYBD
D609 CA 18 D6     4130         JZ     AI1
D60C 04           4135         INR    B
D60D FE 10        4140         CPI    10H       P FOR PAR PORT
D60F CA 18 D6     4145         JZ     AI1
D612 04           4150         INR    B
D613 FE 13        4155         CPI    13H       S FOR SERIAL
D615 C2 CD D4     4160         JNZ    ERROR     NO GOOD
D618 78           4165 AI1     MOV    A,B
D619 32 C4 DF     4170         STA    STK0      INPUT DEVICE CODE
D61C C3 98 D0     4175         JMP    STRT
D61F 2A D1 DF     4180 AO0     LHLD   STK13     ASSIGN OUTPUT DEVICE
D622 CD E6 D0     4185         CALL   CHCK
D625 06 00        4190         MVI    B,00H     ZERO B
D627 FE 16        4195         CPI    16H       V FOR VDM
D629 CA 38 D6     4200         JZ     AO1
D62C 04           4205         INR    B
D62D FE 10        4210         CPI    10H       P FOR PAR PORT
D62F CA 38 D6     4215         JZ     AO1
D632 04           4220         INR    B
D633 FE 13        4225         CPI    13H       S FOR SERIAL
D635 C2 CD D4     4230         JNZ    ERROR     NO GOOD
D638 78           4235 AO1     MOV    A,B
D639 32 C5 DF     4240         STA    STK1      OUTPUT DEVICE CODE
D63C C3 98 D0     4245         JMP    STRT
D63F CD 2B D1     4250 SR      CALL   CONVD     CASS READ SPEED
D642 EB           4255          XCHG   PUT       IN HL
D643 22 C7 DF     4260         SHLD   STK3
D646 C3 98 D0     4265         JMP    STRT
D649 CD 2B D1     4270 SW      CALL   CONVD     CASS WRITE SPEED
D64C EB           4275         XCHG   PUT       IN HL
D64D 22 C9 DF     4280         SHLD   STK5
D650 C3 98 D0     4285         JMP    STRT
D653 CD 2B D1     4290 SS      CALL   CONVD     SERIAL SPEED
D656 7B           4295         MOV    A,E
D657 D3 DA        4300         OUT    PORTA
D659 7A           4305         MOV    A,D
D65A D3 DA        4310         OUT    PORTA
D65C EB           4315         XCHG
D65D 22 CB DF     4320         SHLD   STK7
D660 C3 98 D0     4325         JMP    STRT
D663 CD E6 D4     4330 DA0     CALL   BKLEN     GET LENGTH
D666 CD 8B D3     4335         CALL   CRLF
D669 16 08        4340         MVI    D,08H     SET BYTE COUNTER
D66B CD B9 D3     4345         CALL   RTHL      WRITE ADDRESS
D66E CD 96 D3     4350         CALL   SPACE
D671 CD 96 D3     4355 DA1     CALL   SPACE
D674 7E           4360         MOV    A,M       GET MEMORY
D675 CD 9C D3     4365         CALL   RTHEX     WRITE IT.
D678 23           4370         INX    H
D679 15           4375         DCR    D
D67A C2 71 D6     4380         JNZ    DA1       LOOP
D67D 11 F8 FF     4385         LXI    D,0FFF8H  2'S COMP 9
D680 19           4390         DAD    D         SUB H BY 9
D681 16 08        4395         MVI    D,08H     SET BYTE COUNTER
D683 CD 96 D3     4400         CALL   SPACE
D686 CD 96 D3     4405 DA2     CALL   SPACE
D689 7E           4410         MOV    A,M       GET BYTE
D68A E6 7F        4415         ANI    7FH       MASK BIT 7
```

```
D68C FE 20              4420          CPI   20H
D68E DA A0 D6           4425          JC    SKIP        NOT CHAR
D691 FE 5E              4430          CPI   5EH
D693 DA A2 D6           4435          JC    PRINT
D696 FE 61              4440          CPI   61H
D698 DA A0 D6           4445          JC    SKIP        NOT CHAR
D69B FE 7B              4450          CPI   7BH
D69D DA A2 D6           4455          JC    PRINT       STILL ASCII
D6A0 3E 2E              4460 SKIP     MVI   A,'.'       OUTPUT A '.'
D6A2 CD 99 D2           4465 PRINT    CALL  WRITE       OUTPUT
D6A5 CD 2B D2           44700          CALL  DONE
D6A8 CA 02 D5           4475          JZ    CMPLT       DONE
D6AB 15                 4480          DCR   D           DECREMENT BYTE COUNT
D6AC C2 86 D6           4485           JNZ   DA2         LOOP
D6AF CD C5 D3           4490          CALL  ESCAP       WANT TO QUIT ?
D6B2 C3 66 D6           4495          JMP   DA0+3       CONTINUE
D6B5 CD E6 D4           4500 HEX      CALL  BKLEN       GET ADDR
D6B8 1B                 4505          DCX   D
D6B9 19                 4510          DAD   D
D6BA CD B9 D3           4515          CALL  RTHL
D6BD 0B                 4520          DCX   B
D6BE 60                 4525          MOV   H,B
D6BF 69                 4530          MOV   L,C
D6C0 CD B9 D3           4535          CALL  RTHL
D6C3 C3 98 D0           4540          JMP   STRT
D6C6 CD E6 D4           4545 SERCH    CALL  BKLEN
D6C9 EB                 4550          XCHG              PUT H IN D
D6CA 2A D1 DF           4555          LHLD  STK13       GET BUFF
D6CD E5                 4560          PUSH  H           PUT ON STACK
D6CE EB                 4565          XCHG              RESTORE H
D6CF 2B                 4570 CMPR0    DCX   H
D6D0 03                 4575          INX   B
D6D1 EB                 4580          XCHG              PUT H IN D
D6D2 E1                 4585          POP   H           GET BUFF
D6D3 22 D1 DF           4590          SHLD  STK13       STORE
D6D6 E5                 4595          PUSH  H           SAVE
D6D7 EB                 4600          XCHG              RESTORE H
D6D8 E5                 4605          PUSH  H
D6D9 D1                 4610          POP   D           PUT H IN D
D6DA 13                 4615          INX   D
D6DB CD 2B D2           4620 CMPR1    CALL  DONE
D6DE CA 02 D5           4625          JZ    CMPLT
D6E1 E5                 4630          PUSH  H
D6E2 2A D1 DF           4635          LHLD  STK13
D6E5 7E                 4640          MOV   A,M
D6E6 E1                 4645          POP   H           RESTORE H
D6E7  FE 0D              4650           CPI   0DH         END OF STRING
D6E9 CA F9 D6           4655          JZ    CMPR2       YES
D6EC D5                 4660          PUSH  D           SAVE D
D6ED CD 2B D1           4665          CALL  CONVO       GET CHAR
D6F0 7B                 4670          MOV   A,E
D6F1 D1                 4675          POP   D           RESTORE D
D6F2 BE                 4680          CMP   M           COMPARE
D6F3 CA DB D6           4685          JZ    CMPR1       GOOD
D6F6 C3 D1 D6           4690          JMP   CMPR0+2
D6F9 CD C5 D3           4695 CMPR2    CALL  ESCAP       QUIT ?
D6FC EB                 4700          XCHG              GET H
D6FD CD B9 D3           4705          CALL   RTHL        WRITE ADDR
D700 CD 8B D3           4710          CALL  CRLF
```

```
D703 EB                     XCHG                     RESTORE H
D704 C3 CF D6               JMP     CMPRO
D707 CD 2B C1         INPO  CALL    CONVC
D70A 21 E9 DF               LXI     H,STK17+20       BUFFER AREA
D70D 36 DB                  MVI     M,0DBH
D70F 23                     INX     H
D710 73                     MOV     M,E
D711 23                     INX     H
D712 36 C9                  MVI     M,0C9H
D714 CD E9 DF         INP1  CALL    STK17+20         EXECUTE IT
D717 CD 9C C3               CALL    RTHEX
D71A CD 3B D7               CALL    DOAGN
D71D C3 14 D7               JMP     INP1             DO AGAIN
D720 CD E6 D4         OUTO  CALL    EKLEN
D723 1D                     DCR     E
D724 4D                     MOV     C,L
D725 21 E9 DF               LXI     H,STK17+20       BUFFER AREA
D728 36 7E                  MVI     M,07EH           MOV A,E
D72A 23                     INX     H
D72B 36 D3                  MVI     M,0D3H
D72D 23                     INX     H
D72E 71                     MOV     M,C
D72F 23                     INX     H
D730 36 C9                  MVI     M,0C9H
D732 CD E9 DF         OUT1  CALL    STK17+20         EXECUTE IT
D735 CD 3B D7               CALL    DOAGN            DO AGAIN?
D738 C3 32 D7               JMP     OUT1
D73B CD 5E D2         DOAGN CALL    INPM
D73E FE 20                  CPI     20H              A SPACE ?
D740 C2 02 D5               JNZ     CMPLT             DONE
D743 CD 8B C3               CALL    CRLF
D746 C9                     RET
D747 CD 2B C1         DMO   CALL    CONVC
D74A EB                     XCHG
D74B CD B9 C3               CALL    RTHL
D74E 0E 10                  MVI     C,10H
D750 7E                     MOV     A,M
D751 CD 9C C3               CALL    RTHEX
D754 23                     INX     H
D755 CD 96 C3               CALL    SPACE
D758 0D                     DCR     C
D759 C2 50 D7               JNZ     DMO+9
D75C CD 3B D7               CALL    DOAGN            REPEAT?
D75F C3 4B D7               JMP     DMO+4
D762                  ***COMMAND LOOK-UP TABLE***
D762 11               TABL1 DB      11H              AI
D763 FF D5                  DW      AIO
D765 17                     DB      17H              AO
D766 1F D6                  DW      AOO
D768 20                     DB      20H              D
D769 47 D7                  DW      DMO
D76B 21                     DB      21H              DA
D76C 63 D6                  DW      DAO
D76E 28                     DB      28H              E
D76F 9C D1                  DW      EMO
D771 35                     DB      35H              EM
D772 CC D1                  DW      EM3
D774 30                     DB      30H              F
D775 03                     DB      03               FLOPPY BOOT
```

```
D776 C8                 5010           DB    PORT+08H
D777 38                 5015           DB    38H            G
D778 97 D1              5020           DW    GO
D77A 40                 5025           DB    40H            H
D77B 55 D6              5030           DW    HEX
D77D 48                 5035           DB    48H            I
D77E 07 D7              5040           DW    INPO
D780 68                 5045           DB    68H            M
D781 0B D5              5050           DW    MOVE
D783 6B                 5055           DB    6BH            MC
D784 CD D4              5060           DW    ERROR          NOT IMPLEMENTED
D786 78                 5065           DB    78H            O
D787 20 D7              5070           DW    OUTO
D789 80                 5075           DB    80H            P
D78A FF D1              5080           DW    PROGO
D78C 93                 5085           DB    93H            RC
D78D A7 D4              5090           DW    CASR7
D78F 94                 5095           DB    94H            RD
D790 CD D4              5100           DW    ERROR          NOT IMPLEMENTED
D792 96                 5105           DB    96H            RF
D793 CD D4              5110           DW    ERROR          NOT IMPLEMENTED
D795 A3                 5115           DB    0A3H           RS
D796 68 D4              5120           DW    CASR5
D798 A6                 5125           DB    0A6H           RV
D799 AD D4              5130           DW    CASR8
D79B 98                 5135           DB    98H            S
D79C C6 D6              5140           DW    SERCH
D79E AA                 5145           DB    0AAH           SR
D79F 3F D6              5150           DW    SR
D7A1 AB                 5155           DB    0ABH           SS
D7A2 53 D6              5160           DW    SS
D7A4 AF                 5165           DB    0AFH           SW
D7A5 49 D6              5170           DW    SW
D7A7 B0                 5175           DB    0B0H           V
D7A8 20 D5              5180           DW    VERFY
D7AA BB                 5185           DB    0BBH           WC
D7AB 65 D5              5190           DW    WCO
D7AD BC                 5195           DB    0BCH           WD
D7AE CD D4              5200           DW    ERROR          NOT IMPLEMENTED
D7B0 BE                 5205           DB    0BEH           WF
D7B1 CD D4              5210           DW    ERROR          NOT IMPLEMENTED
D7B3 D0                 5215           DB    0D0H           Z
D7B4 F5 D4              5220           DW    ZERO
D7B6                    5225 ***INITIAL MODE TABLE***
D7B6 F0                 5230 TABL2     DB    0F0H           SCROLL SPEED
D7B7 59                 5235           DB    59H            CASS READ SPEED
D7B8 02                 5240           DB    02H            2500 BAUD
D7B9 20                 5245           DB    20H            CASS WRITE SPEED
D7BA 03                 5250           DB    03H            2500 BAUD
D7BB 70                 5255           DB    700H            SERIAL SPEED
D7BC 04                 5260           DB    04H            110 BAUD
D7BD 45 53 43 41 50     5265 CMNT6     ASC   'ESCAP'
D7C2 C5                 5270           DB    0C5H
D7C3 57 52 49 54 54     5275 CMNT5     ASC   'WRITTE'
     45
D7C9 CE                 5280           DB    0CEH
D7CA 20 20 43 4F 4D     5285 CMNT4     ASC   '  COMPLET'
     50 4C 45 54
D7D3 C5                 5290           DB    0C5H
```

```
D7D4 56 45 52 49 46        5295 CMNT3  ASC   'VERIFY'
     59
D7DA A0                    5300        DB    0A0H
D7DB 54 41 50 45 20        5305 CMNT2  ASC   'TAPE '
D7E0 20 45 52 52 4F        5310 CMNT1  ASC   ' ERROR'
     52
D7E6 A0                    5315        DB    0A0H
D7E7 0D                    5320 CMNT0  DB    0DH
D7E8 0A                    5325        DB    0AH
D7E9 3E                    5330        DB    '>'
D7EA A0                    5335        DB    0A0H
D7EB                       5340 ***END OF PRESENT ASSEMBLY***
```

SYMBOL TABLE

```
AI0   D5FF   AI1   D618   A00   D61F   A01   D638   BINAS D3AF   BKLEN D4E6
BS    D37D   CASR0 D3D1   CASR1 D41A   CASR2 D426   CASR3 D440   CASR4 D464
CASR5 D468   CASR6 D49A   CASR7 D4A7   CASR8 D4AD   CASW0 D59D   CASW1 D5E0
CASW2 D5EE   CHCK  D0E6   CHKST D286   CMD1  D0FA   CMD2  D117   CMD3  D125
CMNT0 D7E7   CMNT1 D7E0   CMNT2 D7DB   CMNT3 D7D4   CMNT4 D7CA   CMNT5 D7C3
CMNT6 D7BC   CMPLT D502   CMPR0 D6CF   CMPR1 D6DB   CMPR2 D6F9   COMMD D0AC
COMNT D231   CONV0 D12B   CONV1 D147   CONV2 D168   CONV3 D172   CONV4 D17E
CR    D354   CRLF  D38B   DA0   D663   DA1   D671   DA2   D686   DELA2 D23B
DELAS D24E   DELAY D23E   DELET D0D4   DM0   D747   DOAGN D73B   DONE  D22B
EM0   D19C   EM1   D1BD   EM2   D1C8   EM3   D1CC   EM4   D1E9   END   D4D6
ERROR D4CD   ESCAP D3C5   FF    D344   GO    D197   HEX   D6B5   HOME  D383
ICHAR D258   INIT  D027   INIT1 D05A   INIT2 D075   INPC  D707   INP1  D714
INPM  D25E   INPUT D264   ISTAT D252   LCL   D192   LF    D36E   MOVE  D50B
OUTC  D720   OUT1  D732   PIN   D27C   PIN0  D278   PIN1  D282   PINS0 D295
PORT  00D0   PORT0 00D0   PORT1 00D1   PORT2 00D2   PORT3 00D3   PORT4 00D4
PORT5 00D5   PORT6 00D6   PORT7 00D7   PORT8 00D8   PORT9 00D9   PORTA 00DA
PORTB 00DB   PORTC 00DC   PORTD 00DD   PORTE 00DE   PORTF 00DF   POUT  D2B0
PRINT D6A2   PROG0 D1FF   PROG1 D208   PROG2 D213   RTDE  D55D   RTHEX D39C
RTHL  D3B9   SC1   D2EB   SC2   D2F8   SC3   D319   SCI   D000   SERCH D6C6
SKIP  D6A0   SOUT  D2A5   SPACE D396   SR    D63F   SS    D653   STACK DFC0
STKC  DFC4   STK1  DFC5   STK10 DFCE   STK11 DFCF   STK13 DFD1   STK15 DFD3
STK16 DFD4   STK17 DFD5   STK2  DFC6   STK3  DFC7   STK5  DFC9   STK7  DFCB
STK9  DFCD   STRT  D098   SW    D649   TABL1 D762   TABL2 D7B6   VD1   D2E5
VD2   D332   VD3   D33C   VDM   D2BE   VER0  D225   VERFY D520   VMERR D538
VTERR D4C4   WC0   D565   WC1   D581   WRITB D298   WRITE D299   ZERO  D4F5
```