CHAPTER 2

INSTALLATION

PDOS can be configured for a host of different system configurations. These range from small, TMS9995 based STD modules to the 128K byte, TMS9900 based TM990/102 CPU module. This chapter begins by walking a new user through a typical boot procedure. Specific hardware modules are then described including jumper and cabling instructions. Finally, a SYSGEN procedure is described along with trouble shooting ideas.

(CHAPTER 2 INSTALLATION continued)

2.1 HARDWARE CONFIGURATIONS

The following table is intended to give a cross reference of available support hardware different CPU's. These have been tested and are known to work under the PDOS operating system.

	CPU		
	/101MA	/102	SBC95/1
MEMORY	------	------	------
ER3232	X		
TM990/201-206	X		
TM990/203A	X	X	
MATRIX 7911/RPC			X
STORAGE			
TM990/303A	X	X	
ER3314	X	X	
TM990/210	X	X	
ER3300	X		
GW3 FDC/1			X
MICRO/SYS SASI			X
SPECIAL			
TM990/302 EPROM BURNER	X		
TM990/306 SPEECH	X	X	
TM990/307 I/O	X	X	
ER3318 GRAPHICS	X	X	

Typical system configurations:

1-TM990/101MA	9900 CPU, 64K byte SRAM
2-ER3232	4 ports, floppy storage
1-TM990/303A	4 cards, 3 amps
1-TM990/101MA	9900 CPU, 64K byte DRAM
1-TM990/203A	2 ports, floppy storage
1-TM990/303A	3 cards, 4 amps
1-TM990/101MA	9900 CPU, 128K byte SRAM
4-ER3232	6 ports, floppy & winchester
1-ER3314	6 cards, 10 amps
1-TM990/102	9900 CPU, 128K byte DRAM
1-TM990/307	5 ports, floppy storage
1-TM990/303A	3 cards, 4 amps
1-SBC 95/1	9995 CPU, 64K byte SRAM
2-Matrix RAM	2 ports, floppy & winchester
1-GW3 FDC/1	5 cards, 2 amps
1-M/S SASI	

2.2 GETTING STARTED

Although there are many different PDOS system configurations, only one is shown here. Changes from one configuration to another are mainly limited to different controllers and CPU switch settings. (The TM990/102 does not have CPU switches and hence the boot menu prompts for installed devices.)

Shown here is a system consisting of a TM990/101MA CPU card, a TM990/303A floppy controller, and any combination of RAM cards such that memory is contiguous from >0000 to a minimum of >8000 and a maximum of >E000. The boot workspace is at >7000. Terminals should be configured for 1 start bit, 7 bit character, even parity, and 2 stop bits.

You must follow each step carefully for a correct system boot. Do not proceed to the next step without successfully completing the previous step.

Typical system configuration:

TM990/101MA	9900 CPU
(32K bytes RAM)	RAM from >1000 to >8000+
TM990/303A	Floppy controller
2-SA800	Floppy drives
TM990/510	Card cage

9902 initialized for 11 bits:
- 1 start bit
- 7 bit character
- 1 even parity
- 2 stop bits

2.2.1 SYSTEM BOOT

STEP 1 VERIFY BOOT PROGRAM.

**Do not insert PDOS disk until Step 4.

- [ ] Verify correct voltages on empty card cage.
- [ ] Install PDOS boot EPROMs on CPU card.
- [ ] Verify all CPU jumpers.
- [ ] Turn OFF (open) all switches on CPU card.
- [ ] Install CPU and RAM only in card cage.
- [ ] Connect terminal to main port.
- [ ] Power up system.
- [ ] Ground RESTART.B momentarily (pin 93 on card cage) and hit <carriage return>.

**Do not proceed until PDOS menu comes up. Error messages 'DTR LOW' and 'CHECKSUM ERROR' should be resolved before continuing.

```
*PDOS BOOT R2.4
0-99=BOOT
100=MEMORY TEST
101=IAC
102=BOOT
103=MAKE BOOT
104=AUX
?_
```

(2.2.1 SYSTEM BOOT continued)

STEP 2 VERIFY SYSTEM MEMORY.

[ ] Run memory test by entering '100<carriage return>'.

**Do not proceed until memory verifies. Each successful pass of memory is indicated by a period. Memory errors are reported as the memory address followed by the exclusive 'OR' of the read and write data.

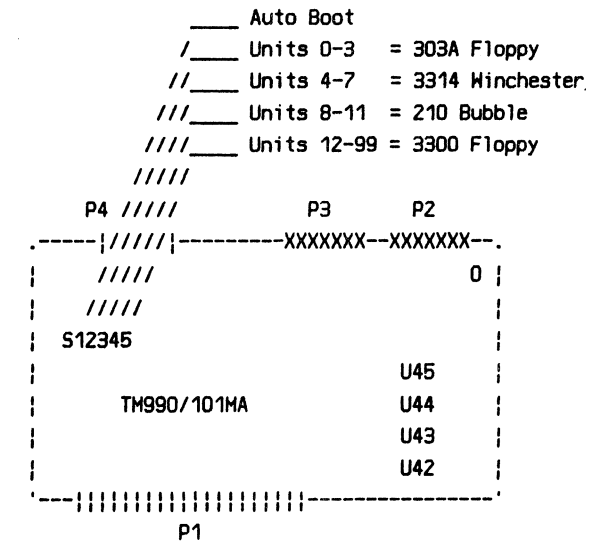
```
?100.....

?101
0 10
0000:  2FDC 2306 2F7C 02FA 2F7C 02FA 2FDC ...
0010:  2F7C 0644 2F7C 0644 2F7C 0644 2F7C ...
10
0010: 2F7C
0012: 0644^C
```

STEP 3 VERIFY STATIC DISK CONTROLLER OPERATION.

[ ] Turn off power.
[ ] Enable CPU controller switch (switch #2 on).
[ ] Install one mass storage controller (e.g. 303A).
[ ] Connect drives to controller verifying correct cabling.
[ ] Power up system and drives.
[ ] Ground RESTART.B and hit <carriage return>.
[ ] Verify correct controller self test (if any).

**Do not insert PDOS disk at this time!
**Do not proceed until PDOS menu again comes up.



STEP 4 VERIFY DISK READ OPERATION BY BOOTING SYSTEM.

[ ] Verify that PDOS boot disk is write protected.
[ ] Power up system and drives. NEVER power up or down the system with a diskette in drive and door closed!
[ ] Ground RESTART.B and hit <carriage return>.
[ ] Insert PDOS boot disk in drive and close door.
[ ] Enter disk number followed by <CR>.
[ ] Verify drive selects and R/W head loads.
[ ] Wait for 'BOOTED!' reply.
[ ] Hit <carriage return>.
[ ] Verify PDOS banner and that the LED on the CPU card is blinking on and off at 1 second intervals.

```
*PDOS BOOT R2.4
0-99=BOOT
100=MEMORY TEST
101=IAC
102=BOOT
103=MAKE BOOT
104=AUX
?0
BOOTED!
PDOS/101 R2.4
ERII, COPYRIGHT 1982
DATE=MN,DY,YR _
```

(2.2.1 SYSTEM BOOT continued)

**If the system seems to hang and returns error 102 (>0066) after an extended period, then:

1. Verify controller jumpers.
2. Verify by ohming out drive cabling.
3. Verify correct drive termination.
4. Verify BUS grant signals on card cage.

**If any other errors are reported, a disk alignment problem would be indicated. Verify with drive distributor that disk drives are correctly aligned.

**If LED is blinking but the PDOS prompt does not appear, verify that DTR (pin 20) signal at CPU main port is high. (This condition is indicated by the message 'DTR LOW' before the boot menu.)

STEP 5 VERIFY CORRECT PDOS BOOT.

[ ] Enter DATE and TIME. Spaces or commas may be used as delimiters. Seconds are optional. All line entries end with <carriage return>.

[ ] Enter 'CS'. A period should be the only response.

[ ] Enter 'LT'. Verify the 'BM' is >6000 and 'EM' is the end of the system memory.

[ ] Enter 'LS'. The disk directory should list to console.

**If the system returns an error 80 to the 'CS' command, then the boot was not correctly read. The boot itself may have been altered. Verify correct memory configuration again and then consult your distributor.

```
PDOS/101 R2.4
ERII, COPYRIGHT 1982
DATE=MN,DY,YR 7 14 82
TIME=HR,MN,SC 10 1
.CS
.LT
TASK  PAGE  TIME    TB    WS    PC ...

*0/0   0     3    >6020 >619A >0828...
.LS
DISK=PDOS R2.4
LEV  NAME:EXT      TYPE      SIZE  ...

 1   ALOAD         SY         8/8  ...
 1   ASM           SY        52/52 ...
 1   BACKUP        SY         6/6  ...
 1   BFIX          SY        11/11 ...
...
```

You have now correctly booted the PDOS system. Only steps 4 and 5 need be repeated to boot a system after a system has been checked out. If this is an initial system boot, continue on with steps 6 through 9 to backup your system disk and configure PDOS for your specific terminal.

(2.2.1 SYSTEM BOOT continued)

STEP 6 FORMAT A NEW DISKETTE.

[ ] Verify boot disk is in drive 0.
[ ] Place new disk in drive 1.
[ ] Enter 'FRMTxxx', where 'xxx' is device type.
[ ] When format header displays, open the door on drive 0.
[ ] Proceed with formatting disk in drive 1.
Enter 'LOGICAL DISK UNIT=' as 1.
Enter 'TRACKS=' as <carriage return>.
Enter 'FORMAT LOGICAL DEVICE 1?' with 'Y'.

**If formatting is not successful, verify hardware configuration and then repeat STEP 6 with another new disk. (Always use good quality, double density diskettes.)

```
.FRMT303
TM990/303A STANDARD FLOPPY FORMAT R2.4
LOGICAL DISK DEVICE=1
SIDES=1
TRACK=0,76
FORMAT LOGICAL DEVICE 1, SECTORS 0 TO 76?Y
.....................................
SUCCESS!!
```

STEP 7 BACKUP SYSTEM DISK.

[ ] Close door on disk drive 0.
[ ] Enter 'BACKUP'.
[ ] Backup disk in disk drive 0 (source disk) to disk in disk drive 1 (destination disk).
Verify the correct number of sectors.
Enter <carriage return> for 'DISK NAME'.

```
.BACKUP
DISK BACKUP R2.4
SOURCE DISK=0
DESTINATION DISK=1
NUMBER OF SECTORS=1976
READY?Y
DUPLICATE 'PDOS 2.4'?Y
FINISHED SECTOR 100
....
SUCCESS!  BACKUP DISK NAME=
RENAMED 'PDOS 2.4'
```

STEP 8 CONFIGURE PDOS SYSTEM.

[ ] Enter 'BFIX'.
[ ] Select system terminal type.
[ ] Answer all 'ADJUST' questions with <CR>'s.
[ ] Answer 'DISK #=' with 1 to fix a new boot on disk device 1.
[ ] Remove boot disk from drive 0 and store in a safe, dry, cool place.

**If your terminal type is not listed, then refer to section 2.4.3 CUSTOMIZING PDOS for instructions on how to define your own control code sequences.

```
.BFIX
BOOT FIX R2.4
**CAUTION: EXECUTE ONLY AFTER NEW BOOT!
TERMINALS:
    A=ADDS REGENT 25
    D=DECSCOPE (VT52 or VT100)
    H=HAZELTINE 1520
    I=INTERTUBE II
    L=LEAR SEIGLER
    S=SOROC
    U=USER DEFINED
TYPE=S
ADJUST TIMER EVENTS (Y OR N)?
ADJUST SYSTEM CLOCK (Y OR N)?
ADJUST CONSOLE CRU BASE ADDRESSES (Y OR N)?
CHANGE SYSTEM INTERRUPT MASK (Y OR N)?
AUTO-START UPON BOOT (Y OR N)?
ADJUST AUTO-START FILE NAME (Y OR N)?
ADJUST INITIAL MEMORY LIMIT (Y OR N)?
ADJUST PDOS PROMPT (Y OR N)?
DISK #=1
BOOT SECTOR=1846
CONTINUE?Y
BOOT SUCCESSFULLY WRITTEN
```

STEP 9 [ ] Place new backed up boot disk in drive 0.
[ ] Reboot system (steps 4 and 5).

(2.2.1 SYSTEM BOOT continued)

FLOPPY DISK MAINTENANCE

Safety precautions should always be followed with respect to the treatment of diskettes.

1.	Never power up or down a system with a diskette in the drive and the door closed. Only after power has been turned on and a RESET or RESTART activated, should the drive door be closed.	Power up procedure
2.	Protect diskettes by always storing them in protective jackets when not in use.	Store diskettes in jackets
3.	Do not touch the media with your fingers. Do not bend or fold the diskette.	Do not touch media
4.	Store your diskettes in a cool, dry place. Preferably, the temperature and humidity should be constant and the same as where they are used. Never expose the diskettes to the sun or extreme temperatures.	Environment
5.	Carefully load and unload the diskettes from the drives. Avoid bending.	Loading and unloading
6.	Do not expose the diskettes to strong magnetic fields. (e.g. terminal transformers, power supplies, magnets, etc.)	Magnetic fields

2.2.2 BOOT EPROMS

The PDOS boot EPROMs contain read and write logical sector routines and a system boot program. Entry addresses are at address >F800 and include controller initialization and motor off routines. Other functions include memory inspect/change and test programs.

The read and write sector routines are the link between PDOS and a secondary storage device. Reference to a 256 byte sector is by disk number (R0), logical sector number (R1), and buffer address (R2). Errors are returned in register R0. They are device dependent and range from 100 to 32768.

Memory addresses >F000 through >FFFF are reserved for EPROM routines. The boot EPROMs reside at memory addresses >F800 through >FFFF. The entry points are located at address >F800 and are defined as follows:

>F800 READ LOGICAL SECTOR. XRSE and XRSZ primitives pass R0, R1, and R2 to this routine. (See 5.2.11 READ SECTOR.)

>F804 WRITE LOGICAL SECTOR. XWSE passes R0, R1, and R2 to this routine. (See 5.2.23 WRITE SECTOR.)

>F808 INITIALIZE LOGICAL SECTOR. XISE passes R0, R1, and R2 to this routine. Initialize sector is equivalent to write sector except that no PDOS ID check is made on the header sector. (See 5.2.7 INIT SECTOR.)

>F80C INITIALIZE CONTROLLER. This routine is called once via a 'BL' instruction before PDOS system initialization. Device dependent initialization procedures are handled here.

>F810 MOTOR OFF ROUTINE. This routine is called once every second via a 'BL' instruction and is for controller devices, which need constant attention. Such is the case with 5" mini-floppies, which require the motor to be turned off after a period of inactivity.

```
Read and write logical sector
System boot
Memory inspect and change
Memory test

R0=disk #
R1=logical sector
R2=buffer address
```

```
        AORG >F800
BOOTV   B @XRSEZ        ;READ SECTOR
        B @XWSEZ        ;WRITE SECTOR
        B @XISEZ        ;INITIALIZE SECTOR
        B @XDITC        ;INIT CONTROLLER
        RT              ;MOTOR OFF (IGNORED)
```

(2.2.2 BOOT EPROMS continued)

The bootstrap program begins executing via the LOAD vector at memory address >FFFC. If auto-boot has been selected (generally through a CPU switch), then the program proceeds to boot the system from the lowest installed disk device. Otherwise, the program waits for a character on the main CPU port.

The first character entered is timed and used to set the baud rate of the main console port. This is referred to as auto-bauding a port. A carriage return auto-bauds all ports correctly.

After the port has been bauded, the boot program sums memory from >F800 to >FFFA. If this sum is nonzero, then a 'CHECKSUM ERROR' message is reported indicating that something has happened to the EPROMs and could be a source of problems.

Next, you may be queried as to which storage devices are installed. This occurs only when there are no configuration switches available, such as with a TM990/102 or SBC95/1 system. A single 'Y' character selects the device. Anything else ignores the device.

Finally, the PDOS boot menu is displayed. You may now select from various functions including:

1. System boot from disk numbers 0 through 99. Boot sector constants within the EPROMs select the correct sector of the boot. Auto-boot selects the lowest numbered storage device.

2. System memory test. A pass is made through memory writing random data and then a second pass verifies memory content.

3. Memory inspect and change. System memory is examined, altered, or copied. Both a hex and an ASCII dump is provided.

4. BOOT and MAKE BOOT routines. A system boot from any logical sector is done by the boot routine. The MAKE BOOT routine writes memory from >0000 to >6000 to any disk # and logical sector.

```
9902 initialized for 11 bits:
        1 start bit
        7 bit character
        1 even parity
        2 stop bits


<LOAD vector>
<carriage return>
SELECT TM990/303? Y     {Optional}
SELECT ER3314?          {Optional}
*PDOS BOOT R2.4
0-99=BOOT
100=MEMORY TEST
101=IAC
102=BOOT
103=MAKE BOOT
104=AUX
?
```

```
System boot

Memory test

Memory inspect and change

Boot and make boot
```

(2.2.2 BOOT EPROMS continued)

5. AUX programs. Other routines are placed in the lower EPROMs (>F000 through >F7FF) and called via an entry at >F000 with the AUX selection.

Auxiliary programs

BOOT MEMORY TEST

Option 100 of the boot EPROMs selects a memory test routine. An optional second decimal parameter is used to select a memory test range other than from >0000 to >7000. The routine first passes through memory, writing random data. A second pass is then made to verify the data. For each successful memory pass, a period is output to the console. If an error occurs, the address is printed along with the exclusive OR of the data read and the correct data.

```
?100,57312      Tests >0000->E000
?100,53216      Tests >0000->D000
?100,49120      Tests >0000->C000
?100,45024      Tests >0000->B000
?100,40928      Tests >0000->A000
?100,36832      Tests >0000->9000
?100,32736      Tests >0000->8000
```

AUTO-BOOT

The PDOS boot EPROMs have the facility to automatically boot PDOS into RAM and set the auto-start flag at memory address >0070. On a TM990/101MA system, this option is selected by switch #1 on the CPU card. Other systems require external switches or hard coded auto-boot.

Switch 1 ON = Auto boot

AUTO-START

If the auto-start flag (byte >0070) is nonzero, then PDOS automatically executes the file named 'SY$STRT' on the system disk. Care must be taken that a baud port command of some type is executed under control of the 'SY$STRT' file, since the system console port is not auto-bauded.

The file type of 'SY$STRT' indicates how the file is to be executed. Normally, it would be a procedure file with the first command being a BAUD PORT (BP) for the console port. Other commands could include configuring other user tasks and the starting of a turn-key application program. The auto-start file name is changed by the BFIX utility.

```
.SA SY$STRT,AC
.SF SY$STRT
BP 1,19200
BP 2,9600
SY 1
LV 10
MENU
RC
._
```

(2.2.2 BOOT EPROMS continued)

SYSTEM DISK

Location >0071 is loaded with the boot disk number after the system is booted and just before a 'BLWP @>0000' is executed. PDOS loads the default system disk number from this location. Thus, the system comes up using the same disk from which it was booted.

>0070 = Auto-run flag
>0071 = Initial default disk #

2.3 PDOS SUPPORTED HARDWARE

2.3.1 CPU

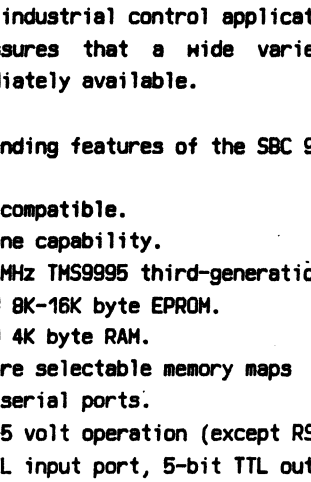
2.3.1.1 SBC 95/1 CPU

The SBC 95/1 is a single-board computer which offers 16-bit minicomputer performance at a very low cost, in a format suitable for most industrial control applications. STD Bus compatibility assures that a wide variety of support products are immediately available.

Some of the outstanding features of the SBC 95/1 include:

- STD Bus compatible.
- Standalone capability.
- Uses 12 MHz TMS9995 third-generation microprocessor.
- On-board 8K-16K byte EPROM.
- On-board 4K byte RAM.
- 2 software selectable memory maps
- 2 Async serial ports.
- Single +5 volt operation (except RS232).
- 8-bit TTL input port, 5-bit TTL output port.

TMS9995 STD BUS CPU



JUMPERS:
A=1,2,3
B=4,5,6
C=7,8,9
D=10,11,12
E=13,14,15
F=16,17,18
G=19,20,21
H=22,23,24
I=25,26,27
J=28,29,30
K=31,32
L=33,34
M=35,36
N=37,38,39

The SBC 95/1 PDOS is shipped with a TMS2532 EPROM containing the read/write sector routines and boot utilities. A 74S472 PROM is also included which contains the PDOS memory map. Since the RESTART vector (>FFFC) is in CPU RAM, the boot EPROM must be initially in low memory and contain the RESTART vector (>0000). Once initialization has occurred, the EPROM is mapped high to address >F000 through >FFFF for PDOS operation.

1-TMS2532 EPROM
1-74S472 PROM

The boot EPROM is installed in the 28-pin socket U9 on the CPU card. The 74S472 PROM is installed in the 20-pin socket U8. The 'auto first wait state' jumper N (E35-E36) should be removed.

It is necessary to have one momentary closed switch to ground connected to PBRESET- (pin 48) on the STD bus. This is used to reset the 9995 and initiate the boot ROM program. An optional momentary closed switch to ground is connected to NMIRQ- (pin 46). This switch restarts PDOS without having to reboot the system. The boot ROM places the vector @ >0000 in the NMI location >FFFC.

RESET switch
RESTART switch

The correct jumper settings for PDOS to operate with the SBC 95/1 CPU are:

E1-E2, E4-E5	U9 = 2532 EPROM
E8-E9, E10-E11, E13-E14	U10 = 4016 RAM
E17-E18,E19-E20, E22-E23	U11 = 4016 RAM
E26-E27	PORT A RS232
E29-E30	PORT B RS232
E31-E32	PORT B -CTS TO GND
E33-E34	PORT B -DTR FROM P20

2.3.1.2 TM990/101MA CPU MODULE

TM990/100MA CPU module

The Texas Instruments TM990/101MA is a self-contained microcomputer on a single printed-circuit board. Its features include a central processing unit (CPU) with hardware multiply and divide, programmable serial and parallel I/O lines, external interrupts, three programmable interval timers, and onboard RAM and EPROM.

PDOS system

The TM990/101MA PDOS is shipped with 2 2708 EPROMs containing the read/write sector routines and boot utilities. PDOS requires the TM990/101MA onboard EPROM to be addressed high and the onboard RAM to be addressed low. The two 2708 boot EPROMs are inserted into sockets U43 and U45 as indicated on the EPROMs. The following jumpers must be in place on the CPU card for a RAM system:

Function	Stake Pins Used	
HI/LO memory map	E15-E16	RAM low
Interrupt 4 source	E2-E3	INT4 on
Interrupt 5 source	E5-E6	INT5 on
EPROM enable	E13-E14	EPROM enabled
Slow EPROM	E8-E53	Fast EPROM
2708/2716 map	E10-E11	2708
	E27-E28,E29-E30	"
	E32-E33,E34-E35	"

DTR high

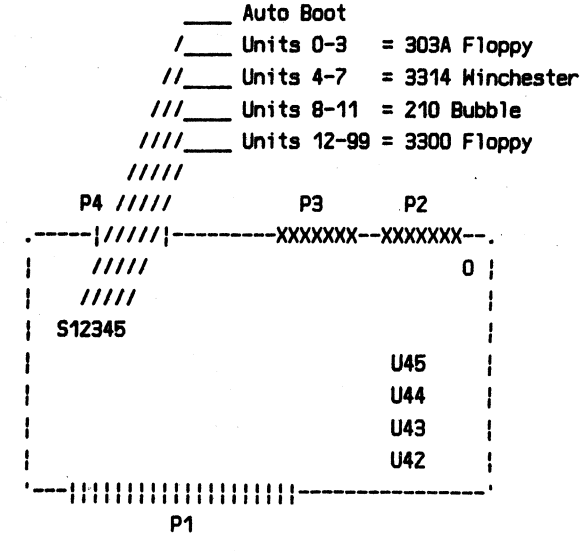
The Data Terminal Ready (DTR) signal must be present on all console ports. This is hardwired by presenting pin 5 to pin 20 on the port connector.

Power on RESTART.B capacitor

The CPU RESTART.B capacitor should be installed on the CPU card to filter (effectively debounce) the -RESTART.B signal. A 39 microfarad capacitor at C23 is the suggested value for manual actuation by a SPST pushbutton to ground. (See Section 6.7.3 of TM990/101MA manual.)

The TM990/101MA CPU ID DIP-switch is used by the READ/WRITE SECTOR EPROMS to configure the system according to the hardware storage devices attached. These are presently defined as follows:

S1 = ON	AUTO BOOT and AUTO EXECUTE OF 'SY$STRT'
S2 = ON	TM990/303A Controller (Disks 0-3)
S3 = ON	ER3314 Winchester Controller (Disks 4-7)
S4 = ON	TM990/210 6 Bubble (Disks 8-11)
S5 = ON	ER3300 Floppy Controller (Disks 12-127)



2.3.1.3 TM990/102 CPU MODULE

The TM990/102 CPU module is a single board computer which features the TMS9900 CPU, 128K bytes of dynamic RAM, a programmable serial port, two programmable interval timers, and up to 16k bytes of EPROM. The extended address lines XA0 through XA3 are also supported such that up to a megabyte of memory can be added to the system. PDOS supports up to 512K bytes.

The TM990/102 PDOS is shipped with two 2516 EPROMs containing the read/write sector routines and boot utilities. The two EPROMs are installed in sockets U10 and U12. The 102 jumpers are configured as follows for the 2516's:

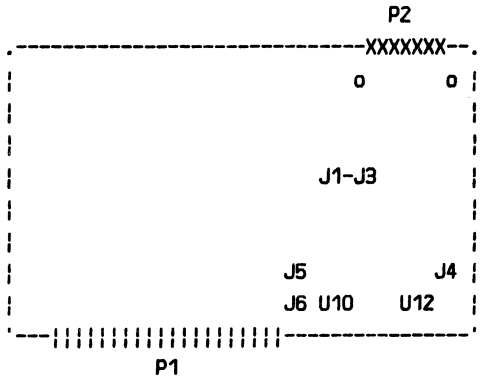
J1	OUT = EPROM ON
J3	IN = 2516
J4	1-2 = 2516
J5	2-3 = 2516
J6	2-3 = 2516

Since the TM990/102 CPU card does not have any DIP switches, the PDOS boot EPROMs prompt for each mass storage device after auto-bauding but before the boot menu. A single 'Y' character selects the device for use by PDOS. Selected devices are stored at memory location >2FFE and tested before each sector access. An error 100 results from access to a non-selected device.

Auto-boot is available only by programming new EPROMs. Two methods of auto-boot are available for the 102. First, if the ABFLG of the boot EPROMs is set, then the system always auto-boots. Second, if the SWFLG is set, then the DIP switches on a TM990/307 card are sampled for both auto-boot and device selection.

PDOS 102 defines eight 9902 console ports. Port #1 is the system console port and is on the TM990/102 CPU card. The other 9902 ports are dependent upon additional hardware. A TM990/303B has an RS232 port which is addressed at location >0180 and used for port #2, and other ports obtained from TM990/307 I/O extender cards.

TM990/102 CPU module:




```
SELECT TM990/303? Y
SELECT ER3314?
*PDOS BOOT R2.4
0-99=BOOT
100=MEMORY TEST
101=IAC
102=BOOT
103=MAKE BOOT
104=AUX
?_
```

```
PORT #1 >0080   P2 TM990/102

PORT #2 >0180   PORT D TM990/307
PORT #3 >0500   PORT A TM990/307
PORT #4 >0580   PORT B TM990/307
PORT #5 >0600   PORT C TM990/307

PORT #6 >0680   PORT A TM990/307
PORT #7 >0700   PORT B TM990/307
PORT #8 >0780   PORT C TM990/307
DRIVER  >0800   PORT D TM990/307
```

(2.3.1.3 TM990/102 CPU continued)

One or two TM990/307 cards may be used in conjunction with the TM990/102 CPU. Ports #2, #3, #4, and #5 would be on the first TM990/307 board and ports #6, #7, and #8 on the optional second 307 board. The remaining port of the second TM990/307 card would be used for drivers, such as a modified $TTA.

If a TM990/303B supplies an RS232 port at base >0180, then port D of the first TM990/307 board should be changed to >0680. An optional second TM990/307 would then have port A changed to >0880 and used for a driver.

If TM990/307 cards are used in conjunction with the TM990/102 card, then the 307 switches should be set as follows:

1) Both TM990/307 cards are set for interrupt level 8. For the 1st TM990/307, switch packs S1 through S4 are set with switch 1 ON and switches 2-8 OFF. (See figure 2-2 of 307 manual.) For the 2nd TM990/307, switch packs S1, S2, and S3 have switch 1 ON and 2-8 OFF. Switch pack S4 has switches 1-8 OFF.

2) The first TM990/307 board is set with CRU base addresses >0500, >0580, >0600, and >0180.

3) The second TM990/307 board, if used, is set with CRU base addresses of >0680, >0700, >0780, and >0800 (not used).

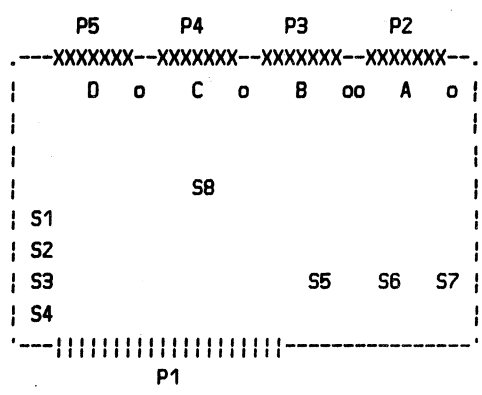
Optional PDOS 102 boot EPROMs which use DIP switches for auto-boot and device selection (SWFLG EQU 1) look to the first TM990/307 switch pack S8. These switches are defined as follows:

SW8 OFF = AUTO BOOT
SW7 OFF = TM990/303A (disks 0-3)
SW6 OFF = ER3314 INSTALLED (disks 4-7)
SW5 OFF = TM990/210 BUBBLE (disks 8-11)

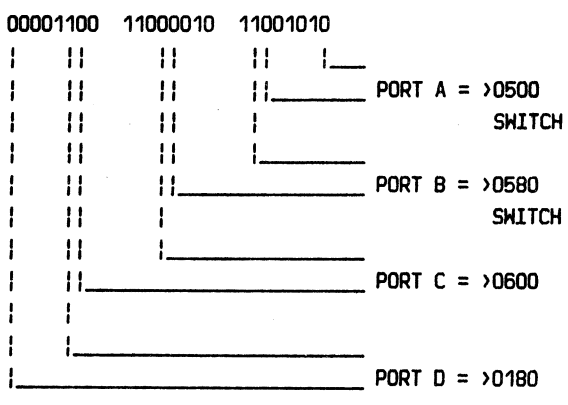
Note #1: The 'sense' of these switches is opposite that found on the TM990/101M card. An 'OFF' (OPEN) indicates the feature is selected.

Note #2: In order for the TM990/102 to use the configuration switches on the first TM990/307 card, the CRU base addresses for ports A and B MUST be at >0500 and >0580 respectively. The system will not work if this is not correct.

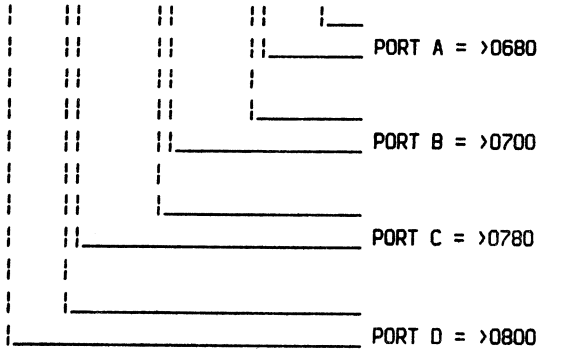
TM990/307 RS232 EXPANSION CARD:



TM990/307 card #1 switch packs:



TM990/307 card #2 switch packs:



2.3.2 MEMORY

2.3.2.1 ER3232 32K STATIC RAM

The ER3232 is a 32K byte static RAM module. Each module is CRU page selectable and has an optional RS232C I/O port. Only one CRU instruction is required to switch from one memory plane to another. Memory mappings provide for single and multiple board PDOS memory configurations.

Switches S3 and S4 select four different memory maps. One is for a 20K byte system page and the other three are for full 32K byte RAM systems. These switches also select the CRU bases for the paging latch and the RS232C port. The following are the map switch definitions:

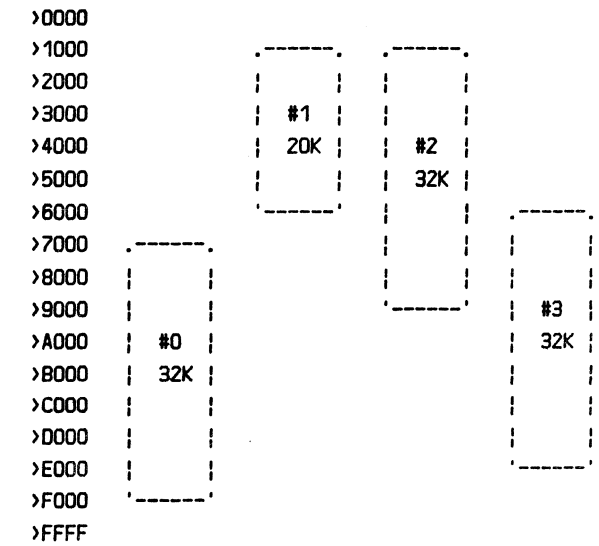
SELECT	S3,S4	MEMORY ADDR	PAGE CRU	RS232 CRU
#0	ON,ON	>7000->EFFF	>980	>A00->BC0
#1	ON,OFF	>1000->5FFF	>D80	>E00->FC0
#2	OFF,ON	>1000->8FFF	>D80	>E00->FC0
#3	OFF,OFF	>6000->DFFF	>980	>A00->BC0

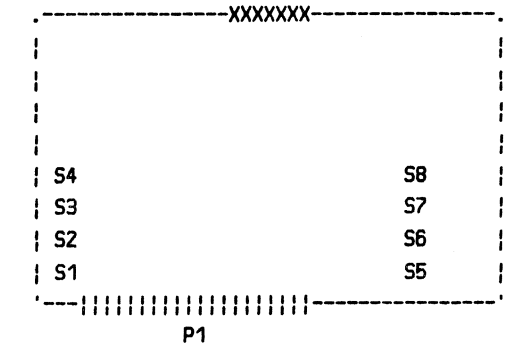
Switches S5, S6, and S7 specify the page number. Up to eight memory pages may reside at the same memory address. The page CRU base is used to select one memory plane while deselecting all others. This greatly facilitates memory swapping for multiple users. These switches also determine the CRU base for the optional RS232C port thus allowing new users to be added to a PDOS system with their own console terminal. The following are the page switch definitions:

		RS232C CRU BASE	
PAGE	S5,S6,S7	S3=S4	S3<>S4
#0	ON,ON,ON	>A00	>E00
#1	ON,ON,OFF	>A40	>E40
#2	ON,OFF,ON	>A80	>E80
#3	ON,OFF,OFF	>AC0	>EC0
#4	OFF,ON,ON	>B00	>F00
#5	OFF,ON,OFF	>B40	>F40
#6	OFF,OFF,ON	>B80	>F80
#7	OFF,OFF,OFF	>BC0	>FC0

Finally, switch S8 selects one memory wait state if slow static RAM is used. Normally, RAMs with access times of 450 nanoseconds or less do not require a memory wait state. One memory wait state is selected by setting S8 OFF (open). S8 ON (closed) disables any wait states.

```
LI R12,>980     ;POINT TO PAGE SELECT
LI R0,>0300     ;GET PAGE 3 SELECT
LDCR R0,3       ;SELECT PAGE 3
```





S8 OFF = one memory wait state

2.3.2.2 MATRIX 7911/RPC 32K STATIC RAM

The MATRIX 7911/RPC is a 32K byte static RAM card that conforms to all STD BUS standards and is used in conjunction with the SBC 95/1 CPU card.

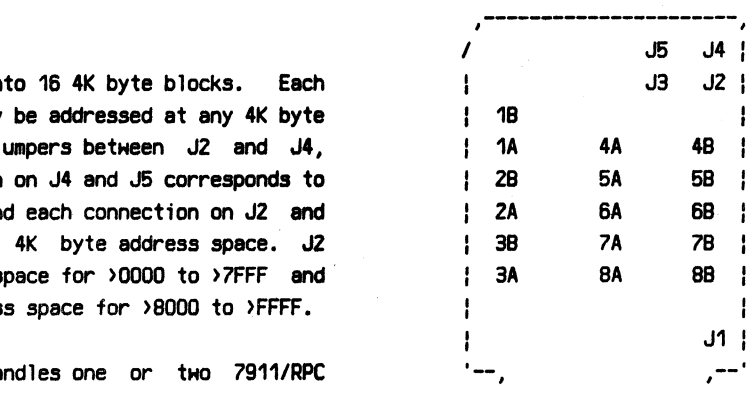
The address lines are decoded into 16 4K byte blocks. Each 4K byte block of memory may be addressed at any 4K byte boundary by means of wire wrap jumpers between J2 and J4, and J3 and J5. Each connection on J4 and J5 corresponds to its numbered block of memory, and each connection on J2 and J3 corresponds to a decoded 4K byte address space. J2 decodes 8 4K blocks of address space for >0000 to >7FFF and J3 decodes 8 4K blocks of address space for >8000 to >FFFF.

A SBC 95/1 - PDOS environment handles one or two 7911/RPC cards. The first card is addressed from >0000 to >7FFF and the second from >8000 to >DFFF. This is done as follows:

1st card:	J2/0 - J4/1	>0000->0FFF
	J2/1 - J4/2	>1000->1FFF
	J2/2 - J4/3	>2000->2FFF
	J2/3 - J4/4	>3000->3FFF
	J2/4 - J4/5	>4000->4FFF
	J2/5 - J4/6	>5000->5FFF
	J2/6 - J4/7	>6000->6FFF
	J2/7 - J4/8	>7000->7FFF
2nd card:	J3/8 - J5/1	>8000->8FFF
	J3/9 - J5/2	>9000->9FFF
	J3/10 - J5/3	>A000->AFFF
	J3/11 - J5/4	>B000->BFFF
	J3/12 - J5/5	>C000->CFFF
	J3/13 - J5/6	>D000->DFFF

The 7911/RPC memory card is used in "primary" map mode. This is enabled by installing a jumper between pins 1 and 2 of J1.

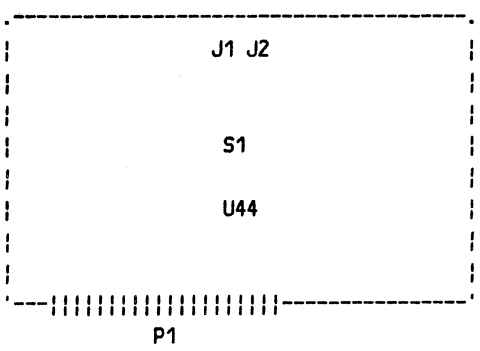
MATRIX 32K STATIC RAM



2.3.2.3 TM990/201 STATIC RAM

TM990/201 and TM990/206 RAM cards can be used to provide contiguous RAM memory for a PDOS system. However, they do require a modified decode PROM in socket U44 such that contiguous memory exists from >0000 to at least >8000.

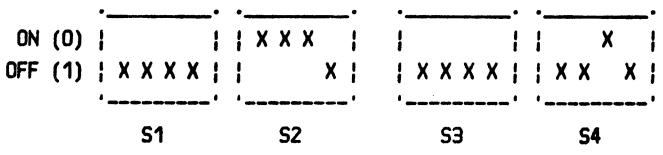
TM990/201 EPROM/RAM CARD



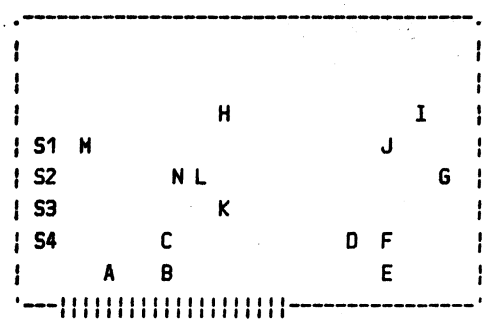
2.3.2.4 TM990/203A DYNAMIC RAM

The TM990/203 line of memory expansion boards are designed to expand system memory from 64K to 256K bytes. The TM990/203 module works only with the TM990/101MA series while the TM990/203A provides expanded memory for the TM990/102 and TM990/103 CPU modules.

A =	E4-E5	Reset parity interrupt
	E8-E9 OUT	Disable parity interrupt
B =	E13-E14	Select RAM size (-2)
C =	E16-E18	Select RAM size
	E20-E22	Select RAM size
D =	E23-E24	Memory wait state
E =	E27-E28	Memory wait state
F =	E31-E34	Number of refresh cycles
G =	E45-E65	Cycle-steal refresh
H =	E47-E48	Size of memory device
I =	E73-E74	Early/late MEMCYC-
J =	N/I	Select application
K =	E82-E83	TMS4116 device used
L =	E84-E85	Select memory speed
M =	E90-E91	Select 15-bit address (E89-E90 for 102)
	E93-E94	Select 15-bit address (E92-E93 for 102)
N =	E36-E39	Select 15-bit address (E36-E37 for 102)



TM990/203A DYNAMIC RAM



2.3.3 SECONDARY STORAGE

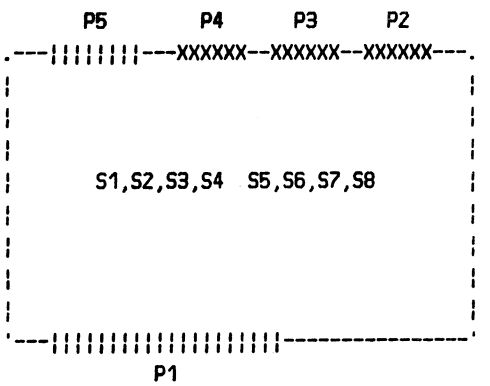
2.3.3.1 ER3300 FLOPPY CONTROLLER

The ER3300 floppy controller module supports large and small floppy disk drives. The board also includes three RS232C I/O ports with full modem capability. The board is referenced by PDOS as disk devices 12 through 15.

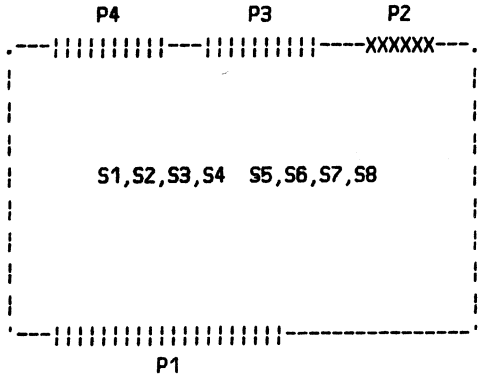
The floppy edge connector pins are defined as follows:

P5-2	-MOTOR ON #1
P5-4	-MOTOR ON #2
P5-6	-HEAD SEL
P5-8	-IP
P5-10	-SELECT #0
P5-12	-SELECT #1
P5-14	-SELECT #2
P5-16	-MOTOR ON #0
P5-18	-DIRECTION
P5-20	-STEP
P5-22	-WRITE DATA
P5-24	-WRITE CLOCK
P5-26	-TRACK 00
P5-28	-WRITE PROTECT
P5-30	RAW DATA
P5-32	-MOTOR ON #3
P5-34	-SELECT #3
P5-36	-HEAD LOAD
P5-38	READY
P5-40	-WRITE FAULT
P5-1 thru 39	GND

ER3300:



ER3300A:



TMS9901 CRU addresses are as follows:

>0400->043F
>0440->047F

1793 floppy controller register memory map addresses are as follows:

>E001 = Status Register
>E003 = Read Track
>E005 = Read Sector
>E007 = Read Data
>E009 = Command Register
>E00B = Write Track
>E00D = Write Sector
>E00F = Write Data

2.3.3.2 ER3314 WINCHESTER INTERFACE

Winchester hard disk

The ER3314 interface is a multipurpose hardware interface board for integration of a Winchester hard disk into the PDOS system. The I/O port provides an interface to Shugart Associates System Interface (SASI) SA1400 series intelligent disk controllers. Also available on the board are two fully functional IEEE 488-1978 controller / talker / listener ports. The main bus connector of the board is compatible with the TM990 bus specification in all respects.

2 IEEE 488-1978 interfaces

Programmed I/O or DMA

The SASI interface may be operated in either the programmed I/O mode or in the direct memory access (DMA) mode. The IEEE 488-1978 ports (also referred to as GPIB ports) may operate as either polled or interrupt driven devices. The GPIB ports have no DMA capability.

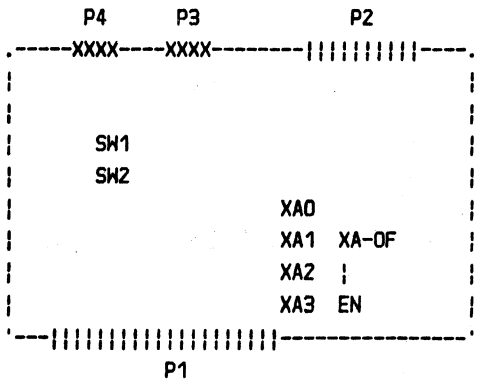
Standard GPIB connectors

The interface card occupies one slot of the processor chassis. Connection to the CPU is via the TM990 standard bus. The GPIB ports are terminated in standard GPIB connectors. The SA1400 cable should be terminated in a 50-pin SCOTCHFLEX 3425-5000 or equivalent connector to mate with the interface card connector (SCOTCHFLEX 3433-1002).

Byte oriented transfers

Data transfers to and from the GPIB ports are byte oriented. The SASI interface is also byte oriented when transfers are done under program control. Data transfers are word oriented when the SASI interface is operating in DMA mode. Byte oriented transfers to any of the three ports are done through the most significant byte (D0-D7) of the TM990 bus.

ER3314:



Switch packs SW1 and SW2 control address and operation of GBIP ports P4 and P3 respectively. Switch positions are defined as follows:

1-5 = Port address (0-31) (ON=0, OFF=1)
6 ON = Tristate outputs
6 OFF = Open collector outputs

The extended address drive capability is enabled with a jumper from XA to EN and disabled with a jumper from XA to OF. XA0 through XA3 determine the extended address states.

(2.3.3.2 ER3314 WINCHESTER INTERFACE continued)

The Winchester device service routine (BT3314:SR) is shipped with logical device numbers 0 and 1 accessing the Winchester, and device numbers 2 and 3 accessing floppies. The boot EPROMs translate these to disk numbers 4 through 7 respectively. The ER3314 interface is selected by switch #3 of the CPU card.

Disk #4 and #5 = SA1004 Winchester

Disk #6 = SA800 single sided floppy

Disk #7 = SA850 double sided floppy

S3 on = ER3314 Winchester system

Disk #4 selects the first half of a SA1004 10 megabyte Winchester and disk #5 the second half. Disk #6 is initialized for a SA800 single sided floppy drive or equivalent. Disk #7 is initialized for a SA850 double sided floppy drive.

If you have a SA1002 instead of a SA1004, then disk #4 accesses the complete disk. Depending on the floppy type (single or double sided), the floppy drive should be connected as logical unit 2 or 3. After the PDOS disks have been transferred to the Winchester, the drives are configured as desired.

ER3314 error numbers

The error numbers returned from the ER3314 device service routine (BT3314:SR) range from 200 to 299. They are defined as follows:

201 = No Index Signal Detected
202 = No Seek Complete Detected
203 = Write Fault
204 = Drive Not Ready
205 = Drive Not Selected
206 = No Track 000 Detected

216 = ID Field ECC or CRC error
217 = Data Field ECC or CRC error
218 = ID Address Mark Not Found
219 = Data Address Mark Not Found
220 = Record Not Found
221 = Seek Error
222 = DMA Timeout Error
223 = Write Protected
224 = Correctable Data Field ECC error
225 = Bad Sector Found
226 = Format Error

232 = Invalid Command received from Host
233 = Illegal Logical Sector Address

(2.3.2.2 ER3314 WINCHESTER INTERFACE continued)

ER3314 memory mapped addresses are defined as follows: Memory mapped addresses

ADDRESS	READ	WRITE
>E020	DATA FROM SA1400	DATA TO SA1400
>E022	STATUS FROM SA1400	------
>E040	PORT 1 INTRPT STATUS 0	PORT 1 INTRPT MASK 0
>E042	PORT 1 INTRPT STATUS 1	PORT 1 INTRPT MASK 1
>E044	PORT 1 ADDRESS STATUS	------
>E046	PORT 1 BUS STATUS	PORT 1 AUX. COMMAND
>E048	PORT 1 ADDRESS SWITCH 1	PORT 1 ADDRESS REG.
>E04A	------	PORT 1 SERIAL POLL
>E04C	PORT 1 CMND PASS THROUGH	PORT 1 PARALLEL POLL
>E04E	PORT 1 DATA IN	PORT 1 DATA OUT
>E050	PORT 1 ADDRESS SWITCH REG	------
>E060	PORT 2 INTRPT STATUS 0	PORT 2 INTRPT MASK 0
>E062	PORT 2 INTRPT STATUS 1	PORT 2 INTRPT MASK 1
>E064	PORT 2 ADDRESS STATUS	------
>E066	PORT 2 BUS STATUS	PORT 2 AUX. COMMAND
>E068	PORT 2 ADDRESS SWITCH 1	PORT 2 ADDRESS REG.
>E06A	------	PORT 2 SERIAL POLL
>E06C	PORT 2 CMND PASS THROUGH	PORT 2 PARALLEL POLL
>E06E	PORT 2 DATA IN	PORT 2 DATA OUT
>E070	PORT 2 ADDRESS SWITCH REG	------

ER3314 CRU addresses are defined as follows: CRU addresses

ADDRESS	CRUIN	CRUOUT
>0480	SA1400 BUSY	SA1400 RESET
>0482	SA1400 MESSAGE	SA1400 SELECT
>0484	SA1400 COMMAND/DATA	DMA MODE
>0486	SA1400 REQUEST	------
>0488	SA1400 INPUT/OUTPUT	XA3
>048A	SA1400 STATUS VALID	XA2
>048C	SA1400 MSG COMPLETE	XA1
>048E	------	XA0
>04C0->04FF	TMS 9911	TMS 9911

2.3.3.3 GW3 FDC/1 FLOPPY CONTROLLER

The FDC/1 floppy disk controller board is an advanced mass storage controller for STD Bus computer systems. The FDC/1 is tailored for operation with GW3's SBC 95/1 processor board using the high performance TMS 9995. Important freatures include:

- 8" or 5" drive support, software selectable
- Single or double sided, single or double density
- Phase-lock-loop data separator
- High speed DMA data transfers

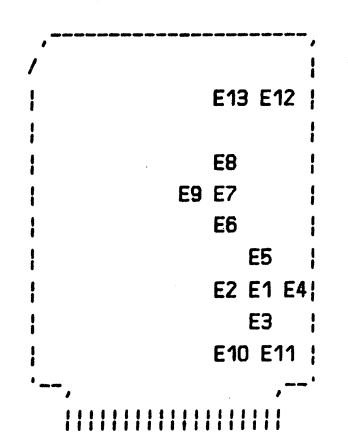
Either an 8" or 5" floppy drive is directly connected to the P2 connector of the FDC/1 without need for pinout adaptors or jumper changes on the board. Four 8" or three 5" drives may be daisy-chained from a single controller board. (8" and 5" drives may be intermixed on the same cable!) All controler commands and status are I/O mapped into an 8 byte block.

Although the drive size, side, and density are software selectable, a single jumper allows the drive size to be forced to 8" or 5" or software select. The position of the jumper can be read by software allowing OEM applications to automatically adapt to the final configuration.

Only two jumpers are required for operation of the FDC/1 board with the PDOS BTFDC1:SR DSR, namely:

E1-E2 Select >EFC0 base address
E6-E8 Select 8" drive

GW3 FDC/1 FLOPPY CONTROLLER



2.3.3.4 TM990/210 BUBBLE MEMORY

The TM990/210 Bubble Memory Module provides non-volatile bubble memory storage for the PDOS system. Access time is from 860 microseconds minimum to 7.3 milliseconds maximum per sector. PDOS configures a 6 bubble module as a single disk, complete with directory and file storage.

Switch 4 on the CPU card is used to indicate that TM990/210 card(s) are in the system beginning at memory address >E100. Up to 4 boards are supported (disk numbers 8 through 11) and are addressed in 32 byte increments. A TM990/210-3 has 6 MMU's (Memory Module Unit) with 69k bytes of storage.

The bubble is accessed in single page mode. This allows interrupts and other system functions to proceed while the bubble is being accessed. One PDOS sector (256 bytes) consists of 15 pages. Each bubble contains 42 sectors for a total of 42 x 6 = 252 sectors. The bubble can be configured for a boot module, leaving 156 sectors for program storage.

Note: Use jumper P3A to P3B to avoid bubble reference problems cause by IORST.

Units 8 thru 11 = Bubble

S4 ON = TM990/210-3 bubble modules

Unit 8 = @>E100
Unit 9 = @>E120
Unit 10 = @>E140
Unit 11 = @>E160

Single page mode

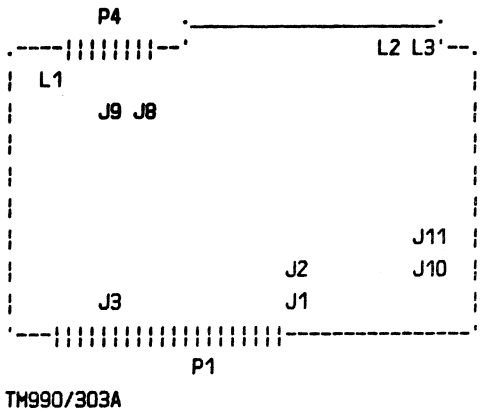
15 pages/sector
42 sectors/bubble
6 bubbles/card = 252 sectors

Boot = 0-155 File storage
156-251 Boot

Address >E100 => XADDR S0 = ON
S1 = ON
S2 = ON
S3 = ON
XAEN = OFF
ADDR SEL S0 = OFF
S1 = OFF
S2 = OFF
S3 = ON
S4 = ON
S5 = ON
S6 = ON
S7 = OFF
S8 = ON
S9 = ON
S10 = ON

2.3.3.5 TM990/303A FLOPPY CONTROLLER

The TM990/303A is a Texas Instruments Inc. floppy disk controller module. This board provides a controlling interface between the microcomputer and an 8 inch floppy disk drive. The PDOS device service routine (BT303:SR) initializes the TM990/303A for 8 inch, IBM, single or double sided, double density format. The TM990/303A board is selected by switch #2 on the TM990/101MA CPU card and is addressed by PDOS as disk numbers 0 through 3. The TM990/102 CPU selects the TM990/303A by answering the device prompt with a 'Y'.



TM990/303A

To install a TM990/303A:

STEP 1 - JUMPER SETTINGS

The TM990/303A should be jumpered as follows for a standard drive system:

J1	= E1-E3	No boot load
J2	= E5-E6	" " "
J3	= INT2	Interrupt level 2
J8	= IN	Disk format
J9	= IN	Disk size
J10	= STD	Standard 8" disk
J11	= E16-E17	" " "

Suggested Shugart SA800 disk drive jumper settings are listed in TABLE 2-2 of the TM990/303A manual. These must be set correctly for proper system initialization.

SHUGART SA800 jumper settings

Jumper Name	Intermediate Board	Terminated Board
A	IN	IN
B	IN	IN
C (P18)	IN	IN
D	OUT	OUT
DC	IN	IN
DDS	OUT	OUT
DS	IN	IN
DS1	Drive 0 select.....	
DS2	Drive 1 select.....	
DS3	Drive 2 select.....	
DS4	Drive 3 select.....	
HL	OUT	OUT
T1	OUT	IN
T2	IN	IN
T3	OUT	IN
T4	OUT	IN
T5	OUT	IN
T6	OUT	IN
X	OUT	OUT
Y	OUT	OUT
Z	IN	IN
800	IN	IN
801	OUT	OUT

STEP 2 - CARD CAGE

When using a TM990/510 or equivalent, the etch on the backplane between lines 95 and 96 must be open (cut) where the TM990/303A board is installed. Generally, this slot is the closest to the bus termination.

STEP 3 - ATTACH DRIVE and TEST

When system installation has been checked and verified, power can be applied. With power applied, the controller executes a self-test. LEDs L2 and L3 go on. When the test is complete, these go off and L1 remains on, indicating no error.

(2.3.3.4 TM990/303A FLOPPY CONTROLLER continued)

There are two types of errors reported by the system boot EPROMs. The first type comes directly from the 303A controller. These errors are reported as a single 16-bit 2's complement decimal integer by PDOS and a 4 digit hex number by the boot EPROMs. The error number is made up of the combined primary and secondary status words of the 303A command list. Since bit 1 is the error indicator, the absolute value of all 303A errors are always be greater than 16383. (Write protect error is intercepted and returned as error 103.)

TM990/303A disk controller errors are defined as follows:

Bit	Value	Description
0	>8000	OL = Off Line
1	>4000	ER = Error occurred
5	>0400	SI = Seek Incomplete
6	>0200	ST = Self Test Error
7	>0100	BC = Bad Command
9	>0040	DE = Data Error
11	>0010	ID = Disk ID Error
12	>0008	OV = Overrun Error
14	>0002	SE = Data Field Search Error
15	>0001	UE = Unit Error

The second type of error is generated by the PDOS Device Service Routine. These errors include device time-out, device not installed, boundary errors, and errors resulting from incompatible data or media.

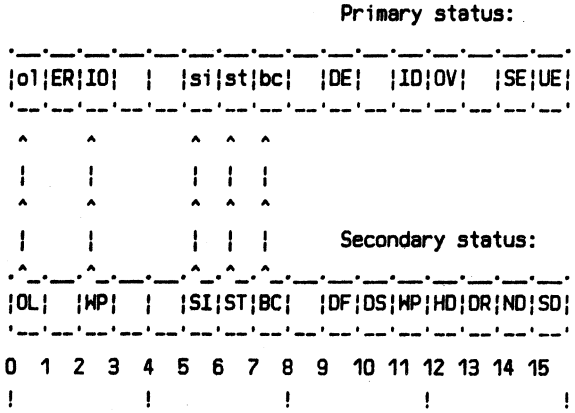
These errors are defined as follows:

68 = Attempt to write non-PDOS header.
100 = Device not installed.
101 = Sector too large.
102 = Device timeout.
103 = Write protect error.
109 = Attempt to read/write across 4k boundary.
110 = # of disk sides does not match data.

Track 0 is reserved for future use. For a standard disk, tracks 1 through 76 are available for data storage. This equals 1976 sectors or 1976 x 256 = 505,856 bytes per disk. (1,011,712 bytes for double sided disk.)

A boot disk restricts disk storage to tracks 1 through 71 or 1846 sectors (472,576 or 945,152 bytes/disk). Tracks 72 through 76 are reserved for the PDOS boot.

303A errors



Device Service Routine errors

Standard disk: 0 = Reserved
1-76 = PDOS file storage

Boot disk: 0 = Reserved
1-71 = PDOS file storage
72-76 = PDOS boot

2.3.4 SPECIAL PURPOSE

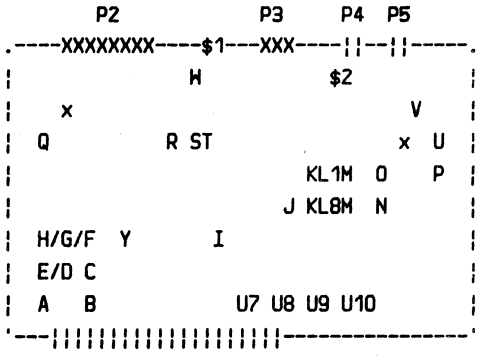
2.3.4.1 ER3318 GRAPHICS

The ER3318 is a multi-purpose graphics board featuring two TMS9918 graphic processors, AM9511 arithmetic processor, ADC 0808 8 channel, 8 bit analog to digital converter, battery backed up clock, TMS5220 speech processor, 75469NA sound chip, and RAM/EPROM support.

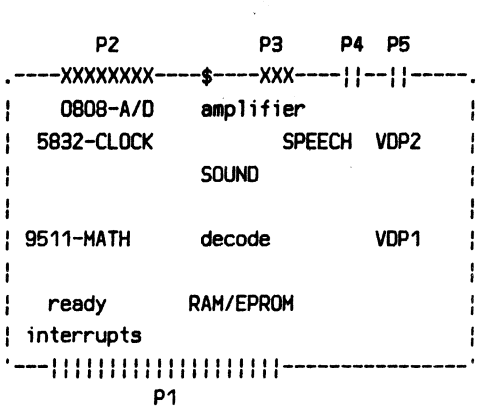
Jumper configurations are defined as follows:

A	IN = INT8 ENABLE	*OUT = INT8 DISABLE (SPEECH)
B	IN = INT9 ENABLE	*OUT = INT9 DISABLE (9511)
C	*IN = READY.B ENABLE	OUT = READY.B DISABLE
D	1 = XA3 LOW	*2 = XA3 HIGH
E	*1 = XA2 LOW	2 = XA2 HIGH
F	1 = EXT ADR ENABLE	*2 = EXT ADR DISABLE (101MA)
G	*1 = XA1 LOW	2 = XA1 HIGH
H	*1 = XA0 LOW	2 = XA0 HIGH
I	1 = 2532 4K EPROM	*2 = 2516/4016 2K EPROM/RAM
J	1 = 2532 4K EPROM	*2 = 2516/4016 2K EPROM/RAM
L1	K =	M =
L2	K =	M =
L3	K =	M =
L4	K =	M =
L5	K =	M =
L6	K =	M =
L7	K =	M =
L8	K =	M =
N	*1 = 4016 RAM	2 = 2516/2532 EPROM
O	1 = 2532 4K EPROM	*2 = 2516/4016 2K EPROM/RAM
P	*1 = 4116 16K VDP1 RAM	2 = 4027 4K VDP1 RAM
Q	1 = 2 mhz 9511 CLOCK	*2 = 3 mhz 9511 CLOCK
R	1 = >x1xx ADDR BIAS	*2 = >x3xx ADDR BIAS
S	1 = >6xxx DECODE BASE	*2 = >Exxx DECODE BASE
T	1 = >6xxx DECODE BASE	*2 = >Exxx DECODE BASE
U	*1 = 4116 16K VDP2 RAM	2 = 4027 4K VDP2 RAM
V	*1 = VDP2 OVERLAY VDP1	2 = VDP2 OUT P5
W	*1 = SPEECH AND SOUND	2 = SPEECH ONLY
Y	*IN = NO WAIT STATE	OUT = WAIT STATE
$1	SOUND VOLUMN CONTROL	
$2	SPEECH FREQUENCY	

ER3318 MULTI-PURPOSE GRAPHICS BOARD:



ER3318 FUNCTIONAL LAYOUT:



(2.3.4.1 ER3318 GRAPHICS continued)

P2 Pin #1 = +5 VOLT
2 = +5 VOLT
3 = +5 VOLT
4 = GND
5 = GND
6 = GND
7 = GND
8 = GND
9 = GND
10 = GND
11 = GND
12 = GND
13 = GND
Pin #14 = +5 VOLT
15 = GND
16 = GND
17 = CHANNEL 7 INPUT
18 = CHANNEL 6 INPUT
19 = CHANNEL 5 INPUT
20 = CHANNEL 4 INPUT
21 = CHANNEL 3 INPUT
22 = CHANNEL 2 INPUT
23 = CHANNEL 1 INPUT
24 = CHANNEL 0 INPUT
25 = GND

P3 Pin #1 = GND
2 = GND
3 = GND
4 = GND
5 = GND
6 = VIDEO OUT AUX
7 = VIDEO OUT MAIN
8 = EXTERNAL AUDIO AMP OUTPUT
9 = 8 OHM SPEAKER OUTPUT

P4/P5 Pin #1 = GND
2 = Composite Video
3 = N/C
4 = +12 volts

2.3.4.2 TM990/302 EPROM PROGRAMMER

The TM990/302 is used for EPROM burning only in a PDOS system. The onboard RAM cannot be used and must be removed from the board. If problems still occur, the DBIN, WE, and READY signals should also be cut. The BURN302 utility is demonstrated below:

```
.BURN302
BURN302 R2.3e
*NOTE: ALL NUMBERS ARE HEX.
*
HIGHEST PC=0000
BUFFER LIMITS ARE 0000 TO 7072
 0,<file>,<adr>           LOAD BINARY FILE
 1,<file>{,<adr>}         LOAD OBJECT FILE
 2,<adr1>,<adr2>,<byte>   LOAD EPROM DATA
 A                        VERIFY BLANK EPROM
 B{,<adr>}                SET BUFFER BASE
 C,<adr1>,<adr2>,{<adr3>} COMPUTE CHECKSUM
 E                        EXIT TO PDOS
 I{,<adr>}                SET EPROM INDEX
 M                        MODIFY BUFFER MEMORY
   <adr1>                 INSPECT
   <adr1>,<adr2>          DISPLAY MEMORY
   <adr1>,<adr2>,<adr3>   COPY MEMORY
 O,<adr1>,<adr2>,<file>   OUTPUT OBJECT TO FILE
 P,<adr1>,<adr2>,<byte>   PROGRAM EPROM
 S{,<step>}               SET STEP
 T{,<eprom>}              SPECIFY EPROM TYPE
 V,<adr1>,<adr2>,<byte>   VERIFY EPROM WITH MEMORY
*BF000
HIGHEST PC=0000
BUFFER LIMITS ARE F000 TO FFFF
*1,BOOT
ENTRY ADDRESS=0000
*PF000,FFFF,L
................................................................
VERIFYED!
*P,F000,FFFF,R
................................................................
VERIFYED!
*E
.
```

2.3.4.3 TM990/307 I/O EXTENDER

One or two TM990/307 cards may be used in conjunction with the TM990/101MA or TM990/102 CPU. A TM990/101MA CPU defines ports 3-6 on the first TM990/307 and ports 7 and 8 on the second. A TM990/102 would be the same if a TM990/303B with aux port is included. Otherwise, ports 2-5 would be on the first TM990/307 and ports 6-8 on the second. Any ports left over could be used for drivers, such as a modified $TTA.

TM990/307 switches are set as follows:

1) Both TM990/307 cards are set for interrupt level 8. For the 1st TM990/307, switch packs S1 through S4 are set with switch 1 ON and switches 2-8 OFF. (See figure 2-2 of 307 manual.) For the 2nd TM990/307, switch packs S1, S2, and S3 have switch 1 ON and 2-8 OFF. Switch pack S4 has switches 1-8 OFF.

2) The first TM990/307 board is set with CRU base addresses >0500, >0580, >0600, and >0180.

3) The second TM990/307 board, if used, is set with CRU base addresses of >0680, >0700, >0780, and >0800 (not used).

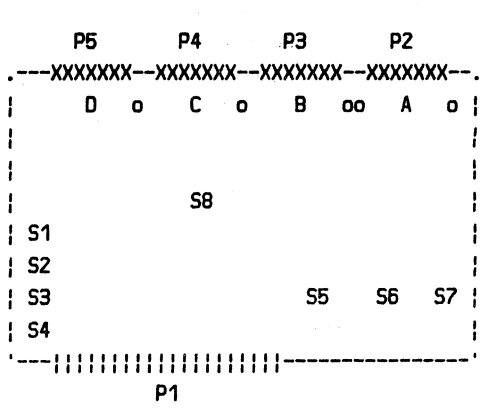
Optional PDOS 102 boot EPROMs which use DIP switches for auto-boot and device selection (SWFLG EQU 1) look to the first TM990/307 switch pack S8. These switches are defined as follows:

SW8 OFF = AUTO BOOT
SW7 OFF = TM990/303A (disks 0-3)
SW6 OFF = ER3314 INSTALLED (disks 4-7)
SW5 OFF = TM990/210 BUBBLE (disks 8-11)

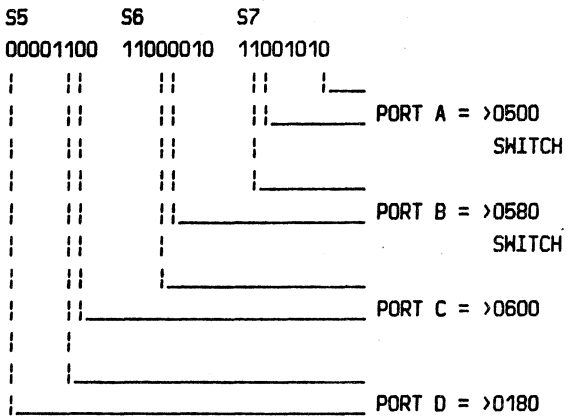
Note #1: The 'sense' of these switches is opposite that found on the TM990/101M card. An 'OFF' (OPEN) indicates the feature is selected.

Note #2: In order for the TM990/102 to use the configuration switches on the first TM990/307 card, the CRU base addresses for ports A and B MUST be at >0500 and >0580 respectively. The system will not work if this is not correct.

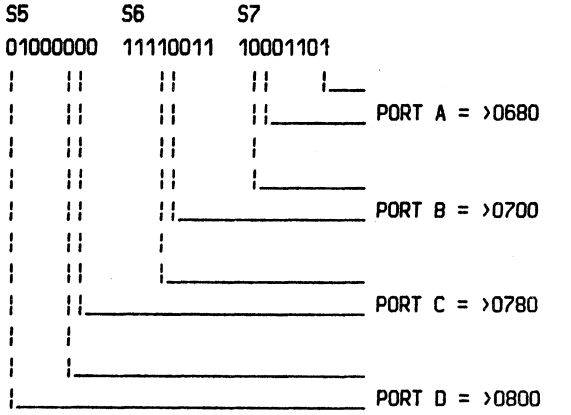
TM990/307 RS232 EXPANSION CARD:



TM990/307 card #1 switch packs:



TM990/307 card #2 switch packs:



2.3.5 CARD CAGE

For DMA logic cards to work properly, the etch between lines 95 and 96 of the backplane on the card cage must be open. This is done by either cutting the etch or by removing a stake-pin jumper at the slot on TM990/5X0A card cages. Jumpers marked J5, J6, J7, and J8 correspond to jumpers to set up slots 1 to 4 respectively. These jumpers are located inside the chassis on the backplane adjacent to their respective slots.

The cutting of the trace or removal of the stake-pin jumper allows the DMA logic to work correctly by letting the GRANTIN-GRANTOUT lines control DMA priority.

2.4 SYSGEN

Due to the device independent nature of PDOS, very little system customizing is required. The main changes from system to system involve the terminal type and secondary storage devices. The latter is completely a function of the boot EPROMs and is covered in Chapter 8.

Storage media preparation is covered in this section, as well as procedures for customizing PDOS and backing up diskettes.

Customize 1) terminal type
2) secondary storage devices

2.4.1 DISK FORMATTING

Before PDOS can use a disk or diskette, the disk must have various control and timing marks written on it (formatted), as well as PDOS directory and sector data (initialized). Formatting a disk is device dependent while initializing is common to all secondary storage devices.

A disk is formatted by one of the 'FRMTxxx' utilities. Since formatting is physical device dependent, the correct format program must be used. The device specification is always a logical unit number with respect to the physical device. This may be the same as the PDOS disk number, but not necessarily.

Formatting a disk unit completely destroys any existing data. Care should be exercised when formatting new disks so that good data is not inadvertently destroyed. Some suggestions include:

1) Enable write protect whenever possible on good diskettes.

2) Open drive doors or spin down drives of those not intended for formatting if unsure.

3) Always maintain current backups of all programs and data.

Devices such as bubble memories or RAM storage devices do not require formatting.

```
.FRMT303
TM990/303A STANDARD FLOPPY FORMAT R2.4
DEVICE LOGICAL UNIT=1
SIDES=1
TRACK=0,76
FORMAT LOGICAL UNIT 1, SECTORS 0 TO 76?Y
.................................
SUCCESS!!
.

.FRMTH
FORMAT WINCHESTER R2.4
DEVICE LOGICAL UNIT=2
TRACK=COMPLETE DISK
INTERLEAVE=3
FORMAT LOGICAL UNIT 2?Y
*** CAUTION: DO NOT ACCESS DISK WHILE FORMATTING ***
PLEASE WAIT.....SUCCESS!!
.
```

2.4.2 DISK INITIALIZING

Before PDOS can work with a new disk, it must be initialized so that directory and sector information is available for the file management module. The INIT utility initializes a formatted disk.

Six parameters are required to properly initialize a disk. First, enter the disk device number. Since INIT uses the system read/write routines, the disk device number corresponds to a physical device through the boot EPROMs. Thus, this number may be different from the FRMTxxx logical unit.

Next, enter the number of diskette sides and density. These parameters are required in the header sector for some devices. For others, such as Winchester units, these parameters are meaningless. The boot EPROMs are the only routines that use these parameters.

The number of possible directory entries on the new disk unit is next. For best usage of disk storage, this number should be a multiple of 8. Once the number of files has been set, it can not be expanded until the disk is again initialized.

The number of PDOS sectors is the fifth prompt. This number is device dependent but the INIT menu helps you in remembering some of the common device sizes. The number of sectors can be less than the total number available on a device if you desire to reserve storage for other purposes (such as storing a system boot on the disk).

Finally, you enter the disk name. This name can be up to 16 characters in length and appears at the beginning of a directory, assembly, or catalog listing. Other than identification, the name has no significance to PDOS.

Since the INIT utility also destroys all data on a disk unit (up to the number of sectors specified), you must verify all parameters before proceeding.

The INIT utility initializes a formatted disk by creating the header and bit map sectors from prompt information and bad sector information gathered by writing to all disk sectors. If a bad sector is encountered during the initialize process, it is deallocated (removed) from the bit map and an error message is reported. Only the header and bit map sectors are required to be good sectors.

```
.INIT
PDOS DISK INITIALIZATION R2.4
DISK #=6
SIDES=1
DENSITY=0
MAXIMUM DIRECTORY SIZE=128
TOTAL NUMBER OF SECTORS=1846
DISK NAME=BACKUP #004
INIT:   DISK #6
        SINGLE SIDED
        DOUBLE DENSITY
        128 FILE DIRECTORY
        1846 PDOS SECTORS
OK?Y
PLEASE WAIT.....
SUCCESS!!
.
```

2.4.3 CUSTOMIZING PDOS

The BFIX utility customizes a PDOS system for specific applications. To ensure a valid system, you should execute BFIX only after a system boot. Creating new tasks or executing BASIC may alter the PDOS initialization routines making reinitialization impossible.

The following areas of PDOS are customized with the BFIX utility:

1. PDOS and BASIC terminal control codes.
2. Event timers.
3. System clock rate.
4. Console CRU addresses.
5. System 9901 interrupt mask.
6. Auto-start flag.
7. Auto-start file name.
8. Upper memory limit.
9. PDOS prompt characters.

1. PDOS and BASIC terminal control codes.

PDOS requires a console terminal to be able to clear the screen and position the cursor. PDOS BASIC supports other optional commands such as cursor movement, clear screen modes, and screen protection.

BFIX prompts for a terminal type from a set of predefined terminal definitions. If a code other than 'U' or <carriage return> is selected, then both PDOS and BASIC are adjusted accordingly. Otherwise, a series of prompts are output showing current control codes and asking for any changes.

A command is changed by entering ASCII literals (hexadecimal number within angle brackets), characters, or predefined control codes (i.e. <BS>, <LF>, <FF>, <CR>, and <ESC>). Inputs are checked for both syntax and allowable codes for any particular command.

PDOS sends from one to four characters to clear the screen. (e.g. one or two characters; an escape followed by one or two characters; or an escape, character, escape, and a final character.)

A one or two character sequence precedes the X and Y screen coordinates when positioning the cursor. The X and Y order can be reversed as well as biased by a blank.

A PDOS BASIC terminal command is a single character or a character preceded by an escape character.

```
.BFIX
BOOT FIX R2.4
**CAUTION: EXECUTE ONLY AFTER NEW BOOT!
TERMINALS:
    A=ADDS REGENT 25
    D=DECSCOPE (VT52 or VT100)
    H=HAZELTINE 1520
    I=INTERTUBE II
    L=LEAR SEIGLER
    S=SOROC
    U=USER DEFINED
TYPE=U
CLEAR SCREEN CHARACTERS.....<ESC>*   NEW=
POSITION CURSOR LEAD CHARACTERS..<ESC>=   NEW=^L
         BIAS ROW/COLUMN BY >20?Y
       REVERSE ROW/COLUMN ORDER?N
BASIC SCREEN CONTROL TABLE:
   (U) CURSOR UP.............<0B>     NEW=
   (D) CURSOR DOWN...........<LF>     NEW=
   (R) CURSOR RIGHT..........<FF>     NEW=
   (L) CURSOR LEFT...........<BS>     NEW=
   (B) BEGINNING OF LINE.....<CR>     NEW=
   (H) HOME CURSOR...........<1E>     NEW=
   (S) CLEAR TO END OF SCREEN....<ESC>Y   NEW=
   (E) CLEAR TO END OF LINE......<ESC>T   NEW=
   (W) RESET WRITE PROTECT.......<ESC>G   NEW=
   (P) SET WRITE PROTECT.....<ESC>F   NEW=
   (() START WRITE PROTECT.....<ESC>I   NEW=
   ()) END WRITE PROTECT.....<ESC>H   NEW=
   (Z) CLEAR UNPROTECTED.....<ESC>K   NEW=
   (N) SKIP TO NEXT FIELD.....<09>    NEW=
```

(2.4.3 CUSTOMIZING PDOS continued)

2. Event timers.

Events 112 through 115 are system timer events. PDOS sets these events every 1/5, 1, 10, and 20 seconds respectively. Event 112 is defined in clock tics while events 113 through 115 are defined as second counts. The event limits are stored in memory locations >00A8 through >00AE. Entries are in decimal and a <CR> leaves the old count.

```
ADJUST TIMER EVENTS (Y OR N)?Y
EVENT 112 = 25 TICS.  NEW TIC COUNT=
EVENT 113 = 1 SECONDS.  NEW SECOND COUNT=
EVENT 114 = 10 SECONDS.  NEW SECOND COUNT=
EVENT 115 = 20 SECONDS.  NEW SECOND COUNT=200
```

3. System clock rate.

The system clock rate is normally at 125 tics per second or 8 milli-seconds per tic. This can be adjusted to a faster rate of 2.6666 ms or slower rates of 13.333 ms or 40 ms. Only options 1 through 4 are allowed.

```
ADJUST SYSTEM CLOCK (Y OR N)?Y
CURRENT TICS/SECOND=125
    1=375 or 2.666 ms
    2=125 or 8 ms
    3=75 or 13.333 ms
    4=25 or 40 ms
NEW TICS/SECOND OPTION (1-4)=4
```

4. Console CRU base addresses.

Console port CRU base addresses can be adjusted to suit a particular system configuration. These CRU bases are entered in hexadecimal and are used by the PDOS input and output routines.

```
ADJUST CONSOLE CRU BASE ADDRESSES (Y OR N)?Y
PORT #1 = >0080.  NEW HEX BASE=
PORT #2 = >0180.  NEW HEX BASE=
PORT #3 = >0E00.  NEW HEX BASE=500
PORT #4 = >0A00.  NEW HEX BASE=580
PORT #5 = >0A40.  NEW HEX BASE=600
PORT #6 = >0A80.  NEW HEX BASE=680
PORT #7 = >0AC0.  NEW HEX BASE=700
PORT #8 = >0B00.  NEW HEX BASE=780
```

5. System 9901 interrupt mask.

All desired interrupt levels must be enabled at the system 9901 before the CPU recognizes them. PDOS supports user interrupts for levels 1, 2, and 9 through 15. Level 3 is reserved for the system clock. Levels 4 through 8 are dedicated to user 9902 terminal I/O. Uninitialized 9902's generate spurious interrupts and must be masked out if not included in the console ports. Levels 3 and 4 are always be enabled.

Note: PDOS/9995 does not use a 9901 to disable interrupts and hence BFIX ignores this prompt.

```
CHANGE SYSTEM INTERRUPT MASK (Y OR N)?Y
CURRENT ENABLED LEVELS=3,4,5,6
      ENTER NEW LEVELS=5,6,8
```

6. Auto-start flag.

A boot can be written with the auto-start byte at location >0070 set such that PDOS by-passes auto-bauding and proceeds directly to processing the auto-start file name.

```
AUTO-START UPON BOOT (Y OR N)?Y
```

(2.4.3 CUSTOMIZING PDOS continued)

7. Auto-start file name.

The auto-start file name is used in conjunction with a turn-key system configuration. If the auto-start byte is nonzero (at memory address >0070), then PDOS automatically passes the auto-start file name to the PDOS monitor for execution. The file name can be up to 7 characters in length.

```
ADJUST AUTO-START FILE NAME (Y OR N)?Y
CURRENT FILE NAME=SY$STRT
    NEW FILE NAME=AUTO-GO
```

8. Upper memory limit.

When PDOS initializes itself, memory is sized by writing and reading zeros until memory does not verify or a upper limit is reached. This upper limit can be adjusted down to >8000.

```
ADJUST INITIAL MEMORY LIMIT (Y OR N)?Y
CURRENT MEMORY LIMIT=E000
    NEW MEMORY LIMIT=C000
```

9. PDOS prompt characters.

The PDOS monitor prompts with a <line feed> <carriage return> followed by one or two characters. The latter two characters can be changed to any other desired prompt.

```
ADJUST PDOS PROMPT (Y OR N)?Y
  CURRENT PROMPT CHARACTERS.....<BEL>.   NEW=
```

Finally, the new customized PDOS is written to disk. All adjustments are made directly to memory and hence a copy of memory for >0000 to >6000 comprises the new boot. Since some of the system constants are generated during PDOS initialization, the system should be rebooted to reflect the new parameters.

```
DISK #=1
BOOT SECTOR=1846
CONTINUE?Y
```

2.4.4 DISK BACKUP

Disk backup is very important to maintaining current and reliable data and procedures. Regular backups should be made to prevent loss of data through human error and mechanical or electrical failures.

Two standard methods of disk backup are available with your PDOS system. First, a complete, sector by sector copy of a disk is made with the BACKUP utility. Second, a selective file backup is made with the TRANS utility.

Example:

```
.TRANS @:@;0,6
    $LPT;0          BN          1
    $TTA;0          BN          1
    $TTO;0          BN          1
    $TTS;0          BN          1
    HLPTX;0         TX          34
    SY$STRT;0       AC          1
.TRANS A@:@;5,6
    ADV:DAT;5       BN      C   206
    ADVENT;5        SY          68
    AMAZING;5       EX          25
.
```

```
.BACKUP
DISK BACKUP R2.4
SOURCE DISK #=0
DESTINATION DISK #=1
NUMBER OF SECTORS=1976
READY?Y
DUPLICATE 'PDOS 2.4 #1'?Y
FINISHED SECTOR 121
FINISHED SECTOR 242
FINISHED SECTOR 363
FINISHED SECTOR 484
FINISHED SECTOR 605
FINISHED SECTOR 726
FINISHED SECTOR 847
FINISHED SECTOR 968
FINISHED SECTOR 1089
FINISHED SECTOR 1210
FINISHED SECTOR 1331
FINISHED SECTOR 1452
FINISHED SECTOR 1573
FINISHED SECTOR 1694
FINISHED SECTOR 1815
FINISHED SECTOR 1936
FINISHED SECTOR 1975
SUCCESS!  BACKUP DISK NAME=
RENAMED 'PDOS 2.4 #1'
.
```

2.5 SYSTEM ADDRESSING

2.5.1 CRU ADDRESS ASSIGNMENTS

CRU ADDR	ASSIGNMENT
>0000->003E	990/101MA LED
>0040->007E	990/101MA DIP SWITCH
>0080->00BE	990/101MA MAIN 9902
>00C0->00FE	_____
>0100->013E	990/101MA 9901 I/O
>0140->017E	_____
>0180->01BE	990/101MA AUX 9902 / 303B / 307D
>01C0->01FE	990/101MA MULTI-DROP
>0200->023E	990/303A FLOPPY CONTROLLER
>0240->027E	_____
>0280->02BE	_____
>02C0->02FE	_____
>0300->033E	SPEECH MODULE (TM990/303B AUX 9902)
>0340->037E	_____
>0380->03BE	_____
>03C0->03FE	_____
>0400->043E	ER3300 FLOPPY CONTROLLER 9901
>0440->047E	ER3300 FLOPPY CONTROLLER 9911
>0480->04BE	ER3314 WINCHESTER DISK CONTROLLER
>04C0->04FE	ER3314 WINCHESTER 9911
>0500->053E	307 #1 PORT A
>0540->057E	_____
>0580->05BE	307 #1 PORT B
>05C0->05FE	_____
>0600->063E	ER3300 PORT #1 307 #1 PORT C
>0640->067E	" " "
>0680->06BE	ER3300 PORT #2 307 #1 PORT D
>06C0->06FE	" " "
>0700->073E	ER3300 PORT #3 307 #2 PORT A
>0740->077E	" " "
>0780->07BE	307 #2 PORT B
>07C0->07FE	_____
>0800->083E	307 #2 PORT C
>0840->087E	_____
>0880->08BE	307 #2 PORT D
>08C0->08FE	_____

Onboard	101MA	CPU	Decode
v	v	v	v
v	v	v	v
v	v	v	v
v	v	v	v
v	v	v	v
v	v	v	v

Onboard	101MA	CPU	Decode

(2.5.1 CRU ADDRESS ASSIGNMENTS continued)

>0900->093E	_____
>0940->097E	_____
>0980->09BE	ER3232 PAGE SELECT S3=S4
>09C0->09FE	_____
>0A00->0A3E	ER3232 PORT 0 S3=S4 S5,S6,S7 = 000
>0A40->0A7E	ER3232 PORT 1 S3=S4 S5,S6,S7 = 001
>0A80->0ABE	ER3232 PORT 2 S3=S4 S5,S6,S7 = 010
>0AC0->0AFE	ER3232 PORT 3 S3=S4 S5,S6,S7 = 011
>0B00->0B3E	ER3232 PORT 4 S3=S4 S5,S6,S7 = 100
>0B40->0B7E	ER3232 PORT 5 S3=S4 S5,S6,S7 = 101
>0B80->0BBE	ER3232 PORT 6 S3=S4 S5,S6,S7 = 110
>0BC0->0BFE	ER3232 PORT 7 S3=S4 S5,S6,S7 = 111
>0C00->0C3E	_____
>0C40->0C7E	_____
>0C80->0CBE	_____
>0CC0->0CFE	_____
>0D00->0D3E	_____
>0D40->0D7E	_____
>0D80->0DBE	ER3232 PAGE SELECT S3<>S4
>0DC0->0DFE	_____
>0E00->0E3E	ER3232 PORT 8 S3<>S4 S5,S6,S7 = 000
>0E40->0E7E	ER3232 PORT 9 S3<>S4 S5,S6,S7 = 001
>0E80->0EBE	ER3232 PORT 10 S3<>S4 S5,S6,S7 = 010
>0EC0->0EFE	ER3232 PORT 11 S3<>S4 S5,S6,S7 = 011
>0E00->0E3E	ER3232 PORT 12 S3<>S4 S5,S6,S7 = 100
>0E40->0E7E	ER3232 PORT 13 S3<>S4 S5,S6,S7 = 101
>0E80->0EBE	ER3232 PORT 14 S3<>S4 S5,S6,S7 = 110
>0EC0->0EFE	ER3232 PORT 15 S3<>S4 S5,S6,S7 = 111
>0F00->0F3E	_____
>0F40->0F7E	_____
>0F80->0FBE	_____
>0FC0->0FFE	_____

(2.5.1 CRU ADDRESS ASSIGNMENTS continued)

>1000->103E	_____
>1040->107E	_____
>1080->108E	_____
>10C0->10FE	_____
>1100->113E	_____
>1140->117E	_____
>1180->118E	_____
>11C0->11FE	_____
>1200->123E	_____
>1240->127E	_____
>1280->128E	_____
>12C0->12FE	_____
>1300->133E	_____
>1340->137E	_____
>1380->138E	_____
>13C0->13FE	_____
>1400->143E	_____
>1440->147E	_____
>1480->148E	_____
>14C0->14FE	_____
>1500->153E	_____
>1540->157E	_____
>1580->158E	_____
>15C0->15FE	_____
>1600->163E	_____
>1640->167E	_____
>1680->168E	_____
>16C0->16FE	_____
>1700->173E	TM990/302
>1740->177E	_____
>1780->178E	_____
>17C0->17FE	_____

2.5.2 MEMORY MAPPED I/O ASSIGNMENTS

4K byte memory mapped space

A 4K byte address space is reserved by PDOS for memory mapped I/O devices. These addresses range from >E000 to >EFFF and are decoded by user hardware.

Generally, device memory map addresses are allocated in 32 byte increments. Defined addresses are as follows:

Address	Assignment
>E000->E01F	3300 Floppy controller
>E020->E03F	3314 Winchester controller
>E040->E05F	3314 GPIB port #1
>E060->E07F	3314 GPIB port #2
>E080->E09F	_____
>E0A0->E0BF	_____
>E0C0->E0DF	_____
>E0E0->E0FF	_____
>E100->E11F	Bubble card #1
>E120->E13F	Bubble card #2
>E140->E15F	Bubble card #3
>E160->E17F	Bubble card #4
>E180->E19F	_____
>E1A0->E1BF	_____
>E1C0->E1DF	_____
>E1E0->E1FF	_____
>E200->E21F	_____
>E220->E23F	_____
>E240->E25F	_____
>E260->E27F	_____
>E280->E29F	_____
>E2A0->E2BF	_____
>E2C0->E2DF	_____
>E2E0->E2FF	_____
>E300->E31F	3318 WRITE 0808 A/D CONVERTER
>E320->E33F	3318 READ 0808 A/D CONVERTER
>E340->E35F	3318 9918A VDP1 GRAPHICS
>E360->E37F	3318 9918A VDP2 GRAPHICS
>E380->E39F	3318 5220 SPEECH
>E3A0->E3BF	3318 SN76489N SOUND
>E3C0->E3DF	3318 5832 CLOCK
>E3E0->E3FF	3318 9511 FLOATING POINT

(2.5.2 MEMORY MAPPED I/O ASSIGNMENTS continued)

>E400->E41F _____
>E420->E43F _____
>E440->E45F _____
>E460->E47F _____

>E480->E49F _____
>E4A0->E4BF _____
>E4C0->E4DF _____
>E4E0->E4FF _____

>E500->E51F _____
>E520->E53F _____
>E540->E55F _____
>E560->E57F _____

>E580->E59F _____
>E5A0->E5BF _____
>E5C0->E5DF _____
>E5E0->E5FF _____

>E600->E61F _____
>E620->E63F _____
>E640->E65F _____
>E660->E67F _____

>E680->E69F _____
>E6A0->E6BF _____
>E6C0->E6DF _____
>E6E0->E6FF _____

>E700->E71F _____
>E720->E73F _____
>E740->E75F _____
>E760->E77F _____

>E780->E79F _____
>E7A0->E7BF _____
>E7C0->E7DF _____
>E7E0->E7FF _____

2.5.3 PDOS MEMORY CONSTANTS

The following is a list of key PDOS memory variables and constants. Locations >0080 through >0100 would be bias by >0040 for PDOS 102 to allow for the mapping registers.

Name	Address	Definition
ASFLG	>0070	Auto start flag
SYSDK	>0071	Default system disk
XPID	>0080	PDOS ID
CKSUM	>0084	PDOS checksum
CR9902	>0086	9902 control register constant
DISKZ	>0088	Hard swap BLWP vector
TIMC	>008C	9901 clock constant
STPS	>008E	Tics/second
CLSC	>0090	Clear screen control characters
PSSC	>0092	Position cursor control characters
INTE	>0094	9901 interrupt enable mask
INTBP	>0096	CRU port table pointer
CRUTB	>0098	9902 CRU base table
EVTC1	>00A8	System event 112 timer constant
EVTC2	>00AA	System event 113 timer constant
EVTC3	>00AC	System event 114 timer constant
EVTC4	>00AE	System event 115 timer constant
STRTFL	>00B0	Auto start file name
PMES	>00BA	PDOS prompt characters
MAIL	>2200	BASIC MAIL array address
ECTAB	>2240	BASIC CALL table
TMBF	>2270	Task message buffers (8/50)
TMD	>2400	Task message destination table
TMS	>2408	Task message source table
XCHB	>2410	File slot buffers
XFSL	>2810	PDOS file slots
SHTAB	>2C10	Task List
XCHI	>2D52	File slot buffer assignment queue
PORTS	>2D5A	Character input buffers
PATB	>2E5A	Input port allocation table
INTAB	>2E64	Port CRU base table
MAP0	>2F00	Page 0 memory allocation bit map
MAP1	>2F08	Page 1 memory allocation bit map
MAP2	>2F10	Page 2 memory allocation bit map
MAP3	>2F18	Page 3 memory allocation bit map
MAP4	>2F20	Page 4 memory allocation bit map
MAP5	>2F28	Page 5 memory allocation bit map
MAP6	>2F30	Page 6 memory allocation bit map
MAP7	>2F38	Page 7 memory allocation bit map

(2.5.3 PDOS MEMORY CONSTANTS continued)

EVTAB	>2F40	Event table
EVTM	>2F50	System events 112-115 timers
F8	>2F58	Port 8 bit input flags
HRS	>2F60	Hours
MIN	>2F62	Minutes
SEC	>2F64	Seconds
MON	>2F66	Month
DAY	>2F68	Day
YRS	>2F6A	Year
OUTAB	>2F6C	Interrupt output Port pointers
INTWS	>2F7C	Port interrupt workspace
BRK	>2F7C	Port break flags
TICS	>2F86	2 word clock counter
FPAC	>2F9C	Floating point accumulator
FPEAD	>2FA2	FP error return address
D303C	>2FBC	303 command table
DSFLG	>2FD0	303 double sided flags
DDFLG	>2FD4	1793 double density flags
MOFLG	>2FD8	Motor ON counters
CLKWS	>2FDC	Clock interrupt workspace
CMPN	>2FE4	Current memory page number (byte)
TSKN	>2FE5	Current task # (byte)
L2LOCK	>2FE6	Level 2 lock
L3LOCK	>2FE8	Level 3 lock
SWLOCK	>2FEA	Swap lock
TIME	>2FEC	Current task time slice
SBPTR	>2FEE	Current task status block pointer
SWTP	>2FF0	Swap table pointer
PMFLG	>2FFC	Permanent memory flag
XRSE	>F800	Read sector entry point
XWSE	>F804	Write sector entry point
XISE	>F808	Init sector entry point
XDIT	>F80C	Initialize disk controller entry
XDOF	>F810	Drive motor off entry

2.5.4 INTERRUPT AND XOP VECTOR DEFINITIONS

TMS9900 interrupt vectors are defined as follows:

```
>0000   Level 0 = RESET vector
>0004   Level 1 = Reserved
>0008   Level 2 = Reserved
>000C   Level 3 = System clock
>0010   Level 4 = CPU main 9902 port
>0014   Level 5 = CPU aux 9902 port
>0018   Level 6 = ER3232 9902 port
>001C   Level 7 = 9902 port
>0020   Level 8 = TM990/307 9902 ports / ER3318 speech
>0024   Level 9 = ER3318 AM9511 arithmetic processor
>0028   Level 10 = User defined
>002C   Level 11 = User defined
>0030   Level 12 = User defined
>0034   Level 13 = User defined
>0038   Level 14 = User defined
>003C   Level 15 = User defined
```

TMS9900 XOP vectors are defined as follows:

```
>0040    XOP 0 = Load FPAC
>0044    XOP 1 = Store FPAC
>0048    XOP 2 = Add to FPAC
>004C    XOP 3 = Subtract from FPAC
>0050    XOP 4 = Multiply FPAC
>0054    XOP 5 = Divide into FPAC
>0058    XOP 6 = Scale FPAC
>005C    XOP 7 = Miscellaneous
>0060    XOP 8 = BASIC evaluate and fix
>0064    XOP 9 = BASIC floating point output
>0068   XOP 10 = BASIC integer output
>006C   XOP 11 = BASIC error call
>0070   XOP 12 = XBUG break point
>0074   XOP 13 = PDOS file primitives
>0078   XOP 14 = PDOS monitor calls
>007C   XOP 15 = PDOS support routines
```

2.6 TROUBLE SHOOTING

If the start up procedure shown in section 2.2 GETTING STARTED is followed carefully, then, any problems encountered in bringing up a new system will be easily isolated. Each step introduces a new parameter. These are discussed as follows:

STEP 1 VERIFY BOOT PROGRAM.

1. All voltages should be verified at the card cage before any boards are plugged in.

2. If the boot menu fails to come up:

[ ] 2a. Verify CPU jumpers.
[ ] 2b. Verify correct EPROM sockets.
[ ] 2c. Verify correct EPROM insertion.
[ ] 2e. Verify RAM at address >7000.
[ ] 2f. Verify terminal cable pins 1,2,3,7.
[ ] 2g. Verify momentary RESTART.B switch.

3. If boot menu comes up but with messages:

CHECKSUM ERROR

[ ] 3a. Verify correct EPROM insertion.
[ ] 3b. Verify EPROM speeds (wait state?)
[ ] 3c. Contact dealer for replacement.

DTR LOW

[ ] 3d. Correct pin 20 to + voltage.
[ ] 3e. PDOS will not operate until corrected.

STEP 2 VERIFY SYSTEM MEMORY

4. The memory test does not catch all types of memory errors but, in most cases, should indicate memory problems. Obviously, the PDOS system will not operate correctly with faulty memory.

[ ] 4a. Verify memory boards correctly seated.
[ ] 4b. Verify RAM on CPU card.
[ ] 4c. Do not user TM990/302 RAM.
[ ] 4d. Verify correct wait states for memory.

(2.6 TROUBLE SHOOTING continued)

STEP 3 VERIFY STATIC DISK CONTROLLER OPERATION

5. The disk controllers may be at fault even before trying to boot PDOS. Do not use the PDOS disk for debugging controller problems! The PDOS boot program does not interact with the controllers until after the the menu is displayed.

[ ] 5a. Verify correct cabling (ohm out).
[ ] 5b. Verify all controller jumpers.
[ ] 5c. Verify all drive jumpers. This includes termination, parity, etc.
[ ] 5d. Verify BUS grant signals on card cage.

STEP 4 VERIFY DISK READ OPERATION BY BOOTING SYSTEM

6. Here is where most problems are encountered. This is the last hurdle. If the system seems to hang and returns error 102 (>0066) after an extended period, then:

[ ] 6a. Verify controller jumpers.
[ ] 6b. Verify correct drive cabling.
[ ] 6c. Verify drive termination.
[ ] 6d. Verify BUS grant signals on card cage.
[ ] 6e. Verify system memory. (DMA possible?)
[ ] 6f. Verify extended address lines.

7. Does the head load? This is indicated by the drive LED coming on.

[ ] 7a. Is the drive turned ON?
[ ] 7b. What error does the controller report in its LED's?
[ ] 7c. Is the drive spinning? (Belt off?)
[ ] 7d. Does the system have proper grounding?
[ ] 7e. Does the system have the same ground reference? (Ground loop?)

8. If the controller reports an error to the boot EPROMs, then the appropriate manual should be consulted. All disk errors such as ID not found and CRC errors may result from drive incompatibility.

[ ] 8a. Verify drive alignment with CE disk.
[ ] 8b. Verify drive speed with strobe.
[ ] 8c. Verify PDOS media not damaged.
[ ] 8d. Try both PDOS disks.

(2.6 TROUBLE SHOOTING continued)

9. If the boot EPROMs indicate the system has been booted but no menu appears and the CPU LED does not blink, then:

[ ] 9a. Verify system memory.
[ ] 9b. Try other PDOS disk.
[ ] 9c. Strike key during boot. When menu returns, use IAC to examine memory.
[ ] 9d. Contact distributor for new media.

10. If the boot EPROMs indicate the system has been booted, the CPU LED is blinking at one second intervals, but no menu, then:

[ ] 10a. Verify DTR signal (pin 20) at port.
[ ] 10b. Verify -DTR signal at pin 7 of 9902.
[ ] 10c. Verify terminal cabling.