8510 DATA PROCESSOR
IBM 3740 COMPATIBLE
DISKETTE SYSTEM

ABSTRACT

This document is proprietary to TERAK Corporation. It is provided for internal use or furnished under license as a part of the QX.SYS Software Source Kit. This document aquaints Terak 8510 Data Processor users with the diskette system. The user is assumed to be familiar with both DEC LSI-11 system architecture and IBM 3740 soft sector format definitions.

TERAK PUB NO 500021

First Release Aug 76
Revised May 77

1.0 SCOPE

The TERAK 8510 diskette data storage system utilizes an IBM compatible diskette. The diskette is a cartridge that consists of a flexible magnetic disk enclosed in a jacket. The disk is free to rotate within the jacket.

The media used is IBM compatible. Data written on the diskette system is capable of being read by an IBM 3740 system. In addition, data written on a 3740 system can be read on the 8510 diskette system. The 8510 diskette system also provides a means for automatic bootstrap of operating systems or user software from a diskette.

2 0 APPLICABLE DOCUMENTS

2.1 IBM Diskette
OEM Information GA 21-9190-1
File No. - Genl. 19

2.2 IBM 3740 Manual

2.3 RT-11/85 System Release Notes

2.4 RT-11/85 Software Support Manual

3.0 GENERAL DESCRIPTION

3.1 Physical Description

The 8510 diskette system is designed for applications requiring diskette interchangeability with an IBM 3740 system. The recording characteristics are controlled to the IBM format.

The diskette system consists of two main parts:

Diskette drive
Drive controller

The drive provides the following mechancial characteristics required to achieve IBM 3740 interchangeability:

Tracks per inch: 48
Total number of tracks: 77
Inside track radius: 2.029"
Disk rotational speed: 360 RPM

Rotational direction
(Recording side): CCW

Read/Write track width: 0.012"

The drive is designed to employ an IBM Diskette, or equivalent, as described in the referenced document (see Paragraph 2.0).

The drive controller provides the electronics necessary to interface the diskette drive to the LSI-11 data bus (Q bus) plus record and recover information under the IBM format.

3 2 Recording Parameters

The 8510 diskette drive is designed to the following recording characteristics:

Data Transfer Rate:	250 K Bits/Sec
Recording Density:	6,621 Bits/Radian
	(3248 BPI Inside Track)
Maximum Bits/Track:	26624 (Formatted)
Maximum Bits/Disk:	2,050,048 (Formatted)

The recording mode is frequency modulation (double frequency recording). The read and write circuitry is designed to record and recover data interchangeably with an IBM 33FD as applied in the 3740 system. The interface provides write current control to meet these IBM requirements.

3.3 Formatting

The diskette system is designed to employ IBM diskettes, or equivalent (see Paragraph 2.0) that are pre-initialized (formatted) to provide:

Sectors per track:	26
Index Tracks:	1
Data Tracks:	73
Spare Tracks:	2
Reserved Track:	
Data Bytes per sector:	128
Records per diskette:	1898

This organisation may be altered, as required, to meet special applications.

3.4 Write Protect Option

If a disk drive is installed with the Write Protect option, the write electronics are disabled only if a hole is provided on the jacket of a diskette. When the diskette is inserted in the 8510, the hole is positioned in a LED/photo transistor server

assembly. If the hole is provided on the jacket, the light passes through it and activates the photo transistor circuit which disables the write electronics. If the hole is not provided, the photo transistor circuit is not activated and the write operation can be performed on the diskette.
An interface status register bit indicates whether or not the diskette is write protected.

4.0 PERFORMANCE CHARACTERISTICS

4.1 Rotational Speed

360 RPM
Speed Variation: ± 2.5%

Rotational direction: counter clockwise (CCW) viewed from recording side
Average latency: 83.3 milliseconds

4.2 Access Times

Track-to-track: 6 msec (step time) (min)
Head settling time: 24 msec (at last step) (max)
Head load time. 50 msec (from head down CMD) (max)

4.3 Error Rate

The error rate, exclusive of external sources such as disk defects, and other disk contamination shall be a maximum of:

1 recoverable error per 10^9 bits
1 non-recoverable error per 10^{12} bits

An unrecoverable error is defined as an error that persists after the error recovery procedure is performed. The error recovery procedure consists of 10 retries to read the record in error.

5.0 COMPATIBILITY

The diskette system is compatible with systems that meet the intent and requirements of the IBM 3740.

6.0 LSI-11 INTERFACE PROTOCOL

6.1 General

The diskette drive controller provides the electronics necessary to interface the Diskette Drive to the LSI-11 data bus plus record and recover information under the IBM format. In addition, the controller contains 64 words (16 bits) of Read Only Memory for initial system bootstrap (see section 7). Up to four disk drive units can be supported. Due to the daisy chain structure of the drive interconnects only one drive may be addressed at a time. Processor software

DISKETTE CONTROLLER CONTROL + STATUS REG

QXCS = $I7700_8$ QXDB = $I77002_8$ VECTOR = 250_8

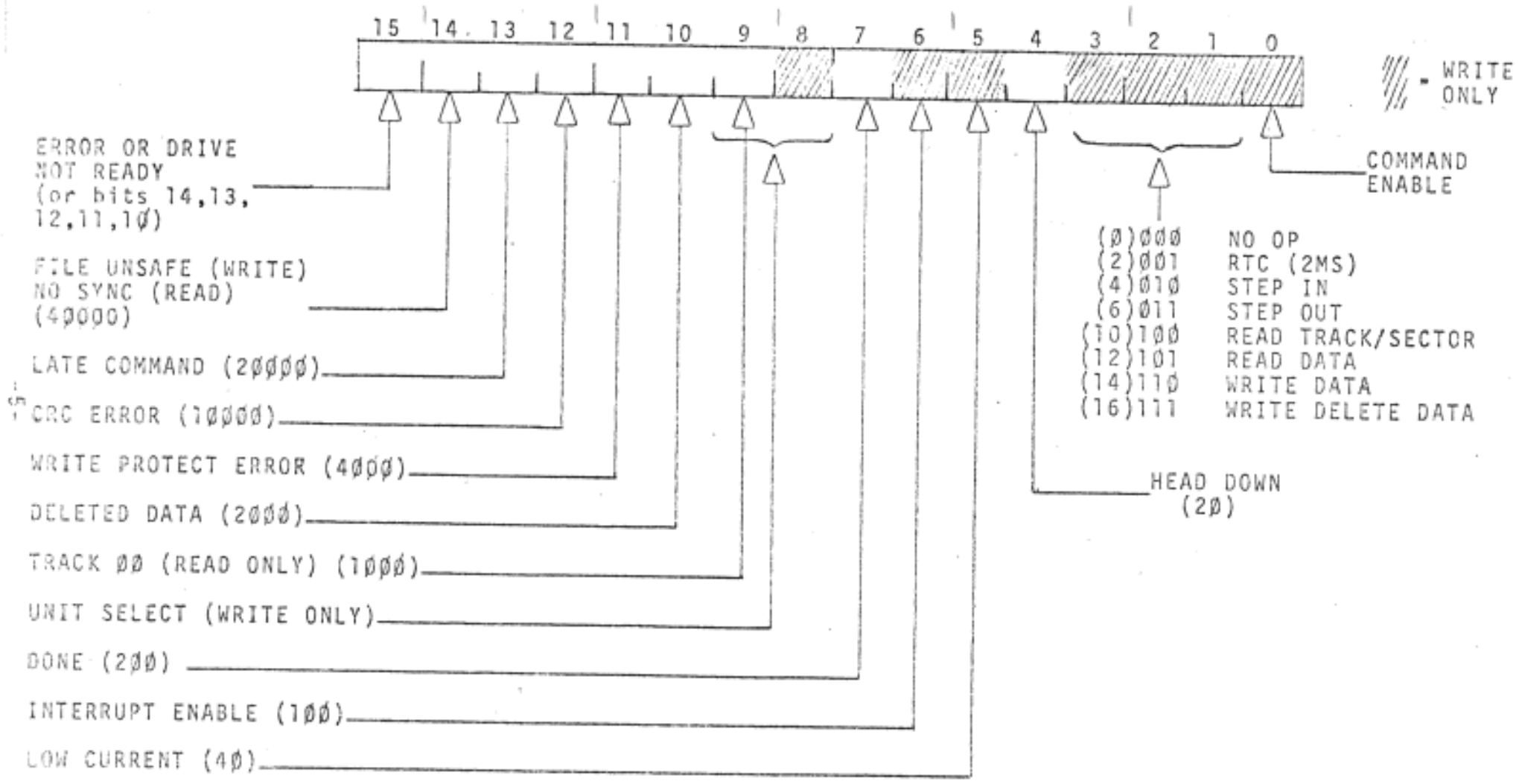

FIGURE 1

support is required for the following functions:

- Head initialisation to Track 00, head positioning on each of the four possible drives (stepping head in and out) and head step settling timing.
- Head down and settling.
- Low write current selection when recording inside of track 4310
- Location of the desired sector on a track.
- Loading and unloading one sector (128 bytes) of information via a data buffer.

The interface consists of two device registers addressable on the processor data bus. The QX Control and Status Register (QXCS) and the QX Data Buffer (QXDB). QXDB is a 64 word (128 byte) buffer. Data is transferred to and from the disk by the QXDB. The processor writes data to be recorded into the buffer and reads recovered data from the buffer. Data is read or written as 64 consecutive 16 bit words.
Each reference to QXDB increments to the next word. In general the buffer is initialised to the first word whenever QXCS is referenced. During the location of a sector on a track QXDB is also used to transfer track address and sector address to the LSI-ll. The following are the definitions of QXCS and QXDB and rules governing their use.

6.2 QX Control and Status Register (QXCS) - 177000_8 (see fig. 1)

6.2.1 Control

- While an operation is in progress, the controller will not respond to any other operations except QXCS read until "Done". This includes references to the resident bootstrap Read Only Memory. All attempts will result in a Bus Fault trap through location 4.
- If altering any bit, all bits must be re-established, even if no change is desired.
- QXCS is word oriented. Byte addressing will be treated as word addressing.

Bits 9 & 8 - Unit Select
Binary coded drive select lines

BIT		DRIVE
9	8	SELECTED
0	0	QX0
0	1	QX1
1	0	QX2
1	1	QX3

Bit 6 - Interrupt Enable

When set will enable an interrupt to occur through location 250_8 upon completion of an operation or when an error occurs.

Bit 5 - Low Current

Must be reset when writing on tracks $0\text{-}43_{10}$ ($0 - 53_8$) and set for tracks $44\text{-}76_{10}$ ($54 - 114_8$).
-ignored during read operations.

Bit 4 - Head Down

a) When set indiates the head is in contact with the diskette. If not previously loaded, allow 5Oms for the head to load and settle before attempting read or write operations. During stepping operations, it is recommended the head be loaded approximately 4 tracks from the final destination. This is computed from:

50ms load settling - 30ms seek settling
6ms track to track access = approx 4 tracks

b) The head will unload under processor control if-the bit is reset. The head will unload automatically if: (1) the drive door is opened, or (2) within 500.0 to 666.8ms (3 to 4 disk revolutions) following the completion of an operation if a new operation is not started. Any reference to QXCS will reset the time out. Note: the drive door/head interlock is performed by the disk drive and will not be reflected in the QXCS register.

c) Head control is a function of "Unit Select". Selection of a new drive will unload the head on the deselected drive.

Bits 3, 2, and 1 - Function Command

Binary coded command lines. Completion of a command or occurance of an error sets "Done" (bit 7) and/or interrupts through location 250_8. The operation is performed once per "Command Enable.' (bit 0). The codes are:

0) No Op: No disk drive operation. Command completes within 400ms.

1) Real Time Clock: No disk drive operation. Command completes within 1.8 to 2.2 ms (2.0 nominal) regardless of drive ready status (Error Condition bit 15).

2/3) Step In/Step Out: Moves the head one track position toward/away from the center of the disk. If re-executing, wait 5ms for track to track access. Allow an additional 24ms for track settling when the desired track is reached or if changing direction (6ms + 24ms = 30ms total on the last step). Verify track number before attempting to read or write. The real time clock function command is provided to support these timing requirements.

NOTE: When stepping after a write command, the step will not be performed until after the track trim erase has been turned off (approximately 600us after "Done"). This time should be accounted for in head settling when stepping after write.

4) Read Track/Sector ID: Detect any address mark, then load the track and sector identity into QXDB at location 1770028. If no address mark is found in 3 to 4 revolutions of the diskette, "No Sync" (Error Condition bit 14) will be set and the command terminated.

5) Read Data: Detect any data mark, then load the sector data into-the data buffer. If the command is attempted at any time except within 300us of completion of "Read Track/ Sector", "Late Command" (Error Condition bit 13) will be set and the command terminated. Detection of a Delete Data Mark will set "Delete Data" (Error Condition bit 10) and the command will continue to completion. If no Data Mark or Delete Data Mark is found within lms, "No Sync" (Error Condition bit 14) will be set and the command terminated.

6) Write Data:- Sector data previously loaded into the data buffer, is written on the disk. If the command is attempted at any time except within 300us of completion of "Read Track/Sector". "Late Command" (Error Condition bit 13) will be set and the command terminated.

7) Write Delete Data: Same as "Write Data" except a Delete Data Mark will be written preceding the data instead of a Data Mark.

Bit 0 - Command Enable Setting this bit causes the command indicated by the Function code bits 3:1 to be executed on the selected drive. Once enabled, the controller will not respond to any operation except QXCS read until "Done". Attempts to write QXCS, reference QXDB, or the Boostrap Read Only Memory will result in a Bus Fault trap through location 4.

6.2.2 Error Conditions

- Any error condition (bits 15:10), once set, will remain set until QXCS is written. When QXCS is written these bits are cleared.

- Occurrence of any error (except "Delete Data") will set "Done" and interrupt through location 2508 if Interrupt Enable" (QXCS 6) is set.

Bit 15 - Drive Not Ready/Lost Ready

a) Indicates selected drive is not loaded and ready to use or selected disk drive was disturbed during a command. If awaiting a status change, QXCS must be periodically written until this bit is no longer set.

b) Also set when any bit from 14 thru 10 is set.

Bit 14 - File Unsafe/No Sync

a) File Unsale: During write operations indicates the occurrence of one of the following conditions which may jeopardise data integrity.

(1) Write gate and no write data.
(2) Write gate and no write current.
(3) Write gate and no write enable (option).
(4) Write gate and step.
(5) Write gate and head not loaded.
(6) Write gate and no erase gate.

b) No Sync: During read operations, valid IBM format was not detected within 3 to 4 revolutions of the disk.

Bit 13 - Late Command

Read or write data command did not arrive within 300us after completion of the "Read Track/Sector ID" command.

Bit 12 - CRC Error - Information read from the diskette did not generate the same check characters as were generated when data was written on the diskette.

Bit 11 - Write Protect Error Diskette file protect is enabled (option).

Bit 10 - Deleted Data

Delete data mark was detected during sector read command.

6.2.3 Status

Bit 7 – Done

Indicates controller is ready to receive a new command. When clear, the controller will not respond to any operation except QXCS read. Attempts to write QXCS, to read or write QXDB, or to read the bootstrap ROM, will result in a Bus Fault trap through location 4.

Bit 9 - Track 00

Indicates the selected drive's read/write head is positioned over track 00.

6.3 QX Data Buffer (QXDB) - 177002_8 (see figure 2)

6.3.1 Track/Sector I D

- Track and Sector is available in QXDB following a "Read Track/Sector" command.

- Track and Sector is only available during the 300us period following completion of a "Read Track/Sector" command (the period during which a "Read Data" or "Write Data"/"Write Delete Data" must be issued to avoid a "Late Command" error condition). During this period QXDB is not available for transfers to/from the 64 word buffer. During this period the buffer is not initialised to the first word with a QXCS reference and referencing QXDB will not disturb the 64 word data buffer (see 6.3.2).

Bits 15:8 Sector Address
Indicates the sector passing beneath the read/write head ($1\text{-}26_{10}$ $1\text{-}32_8$).

Bits 7:0 Track Address
Indicates the track over which the read/write head is positioned ($0\text{-}76_{10}$; $0\text{-}114_8$).

6.3.2 Read/Write Data

- Data is loaded/unloaded, to/from the data buffer thru QXDB as 64 consecutive 16 bit words at any time except during the 300us following a "Read Track/Sector" command or during any command.

- Each reference to QXDB increments to the next word in the buffer. During buffer reading, however, the controller reads one word ahead in anticipation of the next sequential read request. Therefore, alternation of consecutive reads and writes will result in every third word being written.
  QXDB is word oriented. Byte addressing will be treated as word addressing.

- The buffer is initialised to the first word whenever QXCS is referenced, except when "Track/Sector ID" is available (see 6.3.1).

- The 128 bytes of the sector are mapped into the 64, 16 bit words of the buffer as follows:

  Counting the 128 consecutive bytes of a sector from 1, the odd bytes are mapped info the lower consecutive bytes (bits 7:0), the even bytes are mapped into the upper consecutive bytes (bits 15:8). The effect of this mapping is to place the even (low order) byte of words transferred to/from memory, from/to the disk, as the physically leading byte on the disk.

DISKETTE CONTROLLER DATA BUFFER (QXDB) 177002_8

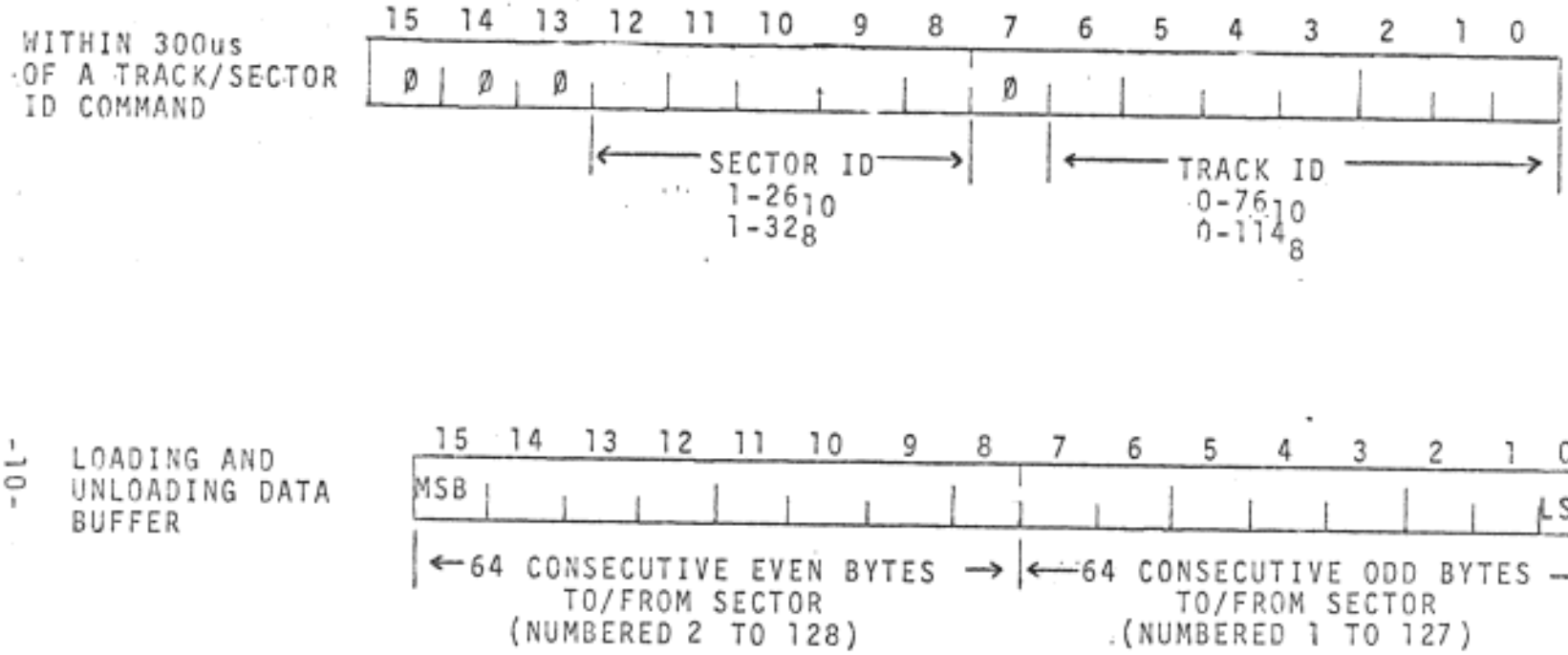

FIGURE 2

Bits 15:8 - Upper Read/Write Data Bytes. 64 consecutive even bytes to/from sector counting sector bytes from 2 to 128_{10}.

Bits 7:0 - Lower Read/Write Data Bytes 64 consecutive odd bytes to/from sector counting sector bytes from 1 to 127_{10}.

6.4 Interrupt Priority

When the controller is enabled for interrupts (QXCS 6), an interrupt request will be issued upon completion or termination of a command. Interrupt service is prioritised by passing the processor service grant from device to device along the data bus. The controller is the electrically closest device and therefore has the highest device prioity. To enable interrupt request servicing, the LSI-11 Processor Status Word priority level must be set to less than 4. That is, Processor Status bit 7 must be 0.

6.5 Power Up

Initial position of the read/write head with respect to data tracks is indeterminate immediately after application of power. In order to assure proper positioning of the read/write head prior to any read/write operation, a Step Out operation should be performed until Track 00 (QXCS 9) is set, or a single step (to align the head) followed by a "head Track/Sector" command.

6.6 Error Recovery

To guard against degradation from imperfection in the media, no more than 4 attempts to write a record should be made when read after write errors are encountered. In the event a record cannot be successfully written with 4 attempts, it is recommended that the sector or track be fabled defective and an alternate sector or track be used. If more than 2 defective tracks are encountered, it is recommended the diskette be replaced.

In the event of a read error up to 10 attempts should be made to recover with re-reads. If after 10 attempts the data was not recovered, step the head one track away in the same direction and then re-position to recover the data.

Unloading the head when not transferring data will increase the data reliability and extend diskette life. The controller will lift the head automatically if a new command is not received within 3 to 4 revolutions of the diskette.

7.0 BOOTSTRAP SYSTEM*

The diskette controller supports a read only memory for system bootstrapping and serialisation. At absolute location 173000 and extending thru location 173176 is the read only memory containing the 8510 bootstrap, (location 173000 thru 173174) and the system serial number (location 173176). A restriction on the use of this memory is that any reference to locations 173000 thru 173176 while

*NOTE - all numbers in this section are octal unless otherwise noted

a controller command is in progress or data is stored in the controller data buffer, will result in a bus fault, or loss of data. For this reason, any reference to the controller read only memory, after the operating system has been established, must be done carefully to preclude simultaneous disk I/O activity. In particular, location 173176 of the read only memory is reserved for the system serial number. It will normally contain a binary number unique to each 8510 processor, and may be used as required by the system software, provided the above precaution is observed.
The 8510 bootstrap is initiated when:

- Power is initially applied to the 8510
- A power fail restart is triggered by a line voltage fluctuation
- The 8510 power switch is pressed to the upward position and released
- Software transfers control to location 173000.

Any of the above will start the processor at absolute location 173000. A listing of the code in the read only memory is presented in Figure 3. Since the execution of instructions from the read only memory and controller I/O activity are incompatible, the bootstrap first moves itself into low memory, at locations 10000 thru 10174, and then enters this code. This bootstrap routine will then attempt to load sector 1 of track 1 from the diskette in unit 0 into memory locations 0 thru 176. If the drive is not ready when the bootstrap is started, the boot will simply wait for a diskette to be inserted, and the drive door closed before proceeding. If I/O errors are encountered during this read activity, retrys will be attempted up to a finite limit. Each retry will jog the disk head. If excessive I/O errors are encountered, the bootstrap routine will hang the processor, and no error message will be printed on the console. Once sector 1 has been successfully loaded, the contents of location 0 (corresponding to the first word of sector 1) is verified against a fixed validation code (010001). If the correct validation code is not found, the bootstrap routine will hang the processor and no error message will print on the console. If the correct validation code is found, control will be transferred from the bootstrap routine to location 0.

At this stage of the boostrap, any user software previously loaded into sector 1, track 1 of the disk will be executed. The first instruction (at location 0) must be a "MOV R0, R1", which is the validation code.

The bootstrap routine in locations 10000 thru 10174 has been designed such that it may also be used as a read only driver by more advanced stages of the bootstrap. The following table defines the entry and exit conditions for the use of this routine. Note that all processor registers are used, including the stack pointer (SP). Control is initially transferred to location 0 with all interrupts locked out. It is essential that interrupts and any

other activity requiring the stack be locked out until the bootstrap routine is no longer required (and the SP is correctly loaded). Note also that the bootstrap routine re-checks the validation code in location 0 each time it is called. The correct code must be left in location 0 until the bootstrap routine is no longer required. Last, note that locations 10000 thru 10174 must not be modified (e.g., by loading the advanced bootstrap) until the bootstrap routine is no longer required.

TABLE 1
ENTRY AND EXIT CONDITIONS FOR BOOTSTRAP ROUTINE

Entry:	Location 10110 (absolute)
Exit:	Location 0 (absolute)
Call:	JMP @#10110
With:	R5 = Sector to be loaded (1 thru 31 octal) R1 - Buffer address
Return:	CLR PC
With:	R0 -> first word above end of buffer R1 preserved R5 preserved
Notes:	All registers are used (including SP) and must be unmodified by user code with the exception of R0, R1, R5. Location 0 must retain a 010001 code which is a "MOV R0, R1" instruction. Thus, R1 will be updated to ascending sequential buffers unless the user code modifies R1. Sectors may be loaded in any order, but will be read only from track 1 of unit 0.
Errors:	Re-trys of any read request are automatic. If excessive I/O errors are encountered, the bootstrap routine will hang the processor. Errors are cumulative from the start of the bootstrap.

Since most operating system bootstraps require more code than can be loaded into one sector, additional code is typically loaded into memory to continue the bootstrap. An example of such code is displayed in Figure 4. This bootstrap will load memory locations 200 thru 1776 with the contents of sectors 3, 5 7, 21, 23, 25 and 27. These correspond to logical blocks 0 and 2, which are the standard locations for the system bootstrap on disks operated under the RT-11 operating system. The space in sector 1 (locations 0 thru 176) which is unused by the primitive bootstrap may be used

for vector loading, subroutines or data for the advanced bootstrap. It should be noted that when the advanced bootstrap is entered, any I/O thru the disk controller to other than track 1, or using logical block addressing (requiring a mapping algorithm), must be performed by a read-only handler loaded along with the advanced bootstrap; the bootstrap routine cannot be used for such purposes. For additional information on the RT-11/85 bootstrap, see the RT-11/85 System Release Notes and related software support bulletins.

Another type of bootstrap which is required by most systems is a "dummy bootstrap which will inform the operator, by printing a message on the console, that the diskette which was mounted into unit 0 for bootstrap does not contain a valid system bootstrap. This bootstrap is typically carried on data diskettes, or diskettes carrying files other than those required to complete the advanced bootstrap. An example of such code is displayed in Figure 5. This bootstrap is written into sector 1 only; additional code is not required. After the message is printed on the console terminal, the code will hang the processors preventing operator access to micro-ODT. This precludes any potential for damaging data on a diskette by abuse of peripheral controller registry, especially, the diskette controller.

Some of the detail in these bootstraps is provided as a suggestion of the manner in which a user of the 8510 processor may bootstrap foreign software successfully. Details, such as the vector skipping, downward-going sector list, and filler characters need not be used in all circumstances.

APPENDIX

1. SERIALIZATION OF HARDWARE

As documented in section 7.0, absolute address 173176 in the BOOT ROM is reserved for serialisation of 8510 processors. Two other locations are also available for user installation identification. These are the low order bytes of the WORDS at absolute locations 173064 and 173104.

The current contents of these locations are:

173064 / 2310
173104 / 23734

but can be modified to

173064 / 2400
173104 / 24000

to meet requirements for additional serialisation, (0 bits are un-programmed) at a very slight cost in the primary bootstrap's speed. The low order bytes of each of these locations can then be factory programmed to 0 thru 377. Included with the standard serialisation word (173176), this provides a total of 32 bits available for serialisation. The same rules apply to reading the contents of these locations as apply to reading the standard serialisation word.

```
        .TITLE  QXBROM
                ;
                ;PROPRIETARY TO TERAK CORPORATION
                ;
                ;THIS SOFTWARE IS PROVIDED UNDER A LICENSE FOR USE ON A
                ;SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
                ;INCLUSION OF THIS NOTICE. THIS SOFTWARE OR ANY COPIES
                ;THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE
                ;IN ANY WAY TO ANY PERSON EXCEPT FOR USE ON SUCH SYSTEM
                ;AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO
                ;AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES
                ;REMAIN IN TERAK CORPORATION.
                ;
                ;TERAK ASSUMES NO RESPONSIBILITY FOR THE RELIABILITY OR
                ;CONSEQUENCE OF USE OF ITS SOFTWARE ON EQUIPMENT WHICH
                ;IS NOT SUPPLIED BY TERAK. THE INFORMATION IN THIS
                ;SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD
                ;NOT BE CONSTRUED AS A COMMITMENT BY TERAK CORPORATION.
                ;
```

```
                        ;
                        ;QX CONTROLLER BOOTSTRAP ROM
                        ;
                        ;
                        ;THIS CODE WILL RESIDE IN LOCATIONS 173720 THRU 173176,
                        ;OR LOCATIONS 173220 THRU 173276, DEPENDING
                        ;UPON A STRAP OPTION ON THE DISK CONTROLLER.
                        ;WHEN HIGHER ADDRESS IS SELECTED FOR BOOT ROM, THE
                        ;CONTROLLER CONTROL & STATUS REGISTER ADDRESS
                        ;AND DATA REGISTER ADDRESS CHANGE ACCORDINGLY:
                        ;
                        ;       BOOT PROM       CNTRL & STAT    VECTOR
                        ;
                        ;LO:    173020-         177170          250
                        ;HI:    173220-         177220          254
                        ;
                        ;BOOT STARTS AT 173220 (AUTO) OR 173200 (VIA ODT)
                        ;AND IMMEDIATELY MOVES ITSELF TO LOW MEMORY
                        ;(THIS IS NECESSARY DUE TO INTERLOCK OF QX CONTROLLER
                        ;AND BOOT ROM). THE CONTROL AND STATUS REGISTER ADDRESS
                        ;WILL BE CALCULATED BASED UPON THE LOCATION OF THE BOOT CODE,
                        ;SUCH THAT A BOOT STARTED AT EITHER ADDRESS WILL
                        ;PROCEED USING THE CORRESPONDING CONTROLLER. THIS IS
                        ;PROVIDED PRIMARILY FOR MANUFACTURING TEST.
                        ;
                        ;THE MOVED BOOT ROUTINE WILL THEN WAIT FOR UNIT 0 TO
                        ;BECOME READY (DISK MOUNTED) AND THEN SEEK AND LOAD
                        ;SECTOR 1, ON TRACK 1, ON UNIT 0 INTO LOCATIONS 0 THRU 176
                        ;(777).  IF LOCN 0 = 240731, THE SECOND STAGE
                        ;OF BOOT IS STARTED AT LOCN 0.
                        ;
                        ;AFTER THE SECOND STAGE IS ENTERED, THE BOOTSTRAP ROUTINE
                        ;IN LOCATIONS 12270 THRU 12176 MAY BE USED AS A READ-ONLY
                        ;DRIVER TO LOAD FURTHER CODE FROM SECTORS ON TRACK 1.
                        ;
                        ;
                        ;D. KOZIMER      12-APRIL-76
```

```
                        .ASECT
                        ;
                        ;CONTROLLER DEPENDENT EQUATES
                        ;
  172870        CXBTLO  =       172222
  177222        CXCSLO  =       177222
  012000        BOOTLO  =       12000   ;LOC'N FOR MOVED BOOT
  000021        XRTS    =       21      ;READ DWN/READ TRKSEC/UNIT #
  000023        XREAD   =       23      ;READ DWN/READ/UNIT #
  000025        XSEEKN  =       25      ;READ DOWN/STEP IN/UNIT #
  000027        XSEEKO  =       27      ;READ UP/STEP OUT/UNIT #
  001000        TRK00   =       1000    ;TRK 0 SENSE SWITCH BIT
                        ;
                        .GLOBL BOOT79           ;ADDRESS TO "CALL" BOOT ROUTINE
                        ;
  015072        DISPL = BOOTLO - BOOT   ;NET DISPLACEMENT BETWEEN ROM & RAM
  012112        BOOT7G = BOOT7 + DISPL  ;USED TO CALCULATE ENTRANCE
                        ;
                        .MCALL .REGDEF
  000000        .REGDEF
                        ;
                        ;CAUTION...ASSEMBLY REQ'S CXBTLO = 172000
                        ;
```

```
                        173200          . = QVBTLO                      ;START OF BOOT PROM
173200 105417 BOOT:     MTPS (PC)                       ;A 12704 TO PS...NO INTRPTS
173202 012704           MOV #BOOTY + DISPL, R4  ;R4=>BOOTY IN MOVED CODE
       214212
173206 010702           MOV PC, R2                      ;R4 = BOOTY
173210 012224 BOOTX:    MOV (R2)+, (R4)+        ;MOVE A WORD
173212 105704           TSTB R4                         ;100000 => NEG BYTE
173214 100375           BPL BOOTY                       ;NEG BIT->64 WORDS
173216 000137           JMP @#BOOTM + DISP:     ;JMP TO BOOTM IN MOVED CODE
       212422
                        ;
                        ;FOLLOWING CODE IS ACTUALLY EXECUTED AS IF
                        ;A  . = BOOTLO WERE INSERTED HERE
                        ;
                                                        ;R4 = 10220 ON ENTRY
173222 062702 BOOTM:    ADD #QIBLO-BOOT-220,R2  ;R2=173200 OR 173402 ON ENTRY
       000020
                                                        ;NOW R2 = CSR REG (PIC)
173226 111705           MOVB (PC), R5           ;R5 = SECTOR TO BE FETCHED
173230 005201           CLR R1                          ;R1 = BUFFER START ADRS
173232 005412 BOOTR:    CLR (R2)                        ;NULL CMD TO CLEAR ERRORS
173234 005712           TST (R2)                        ;R4 BYTE = 200 = -128 DEC
173236 100775           BLT BOOTR                       ;WAIT ON READY BIT
173240 105274           INCB R4                         ;R4 BYTE COUNTS # RETRIES
173242 002377 HALT:     BUS HALT                        ;IF NOT NED, HANG UP
173244 032712 BOOT1:    BIT #CRERR, (R2)        ;TRACK 0 ???
       021023
173250 001010           BNE BOOT4                       ;YES..GO STEP IN
173252 012712           MOV #XSEEK0, (R2)       ;HEAD UP, STEP OUT
       022067
173256 105712 BOOT2:    TSTB (R2)                       ;WAIT FOR DONE BIT
173260 100376           BPL BOOT2
173262 012703           MOV #2312, R3
       002312
173266 077301 BOOT3:    SOB R3, BOOT3           ;WAIT 8 MILLISECS
173270 002726           BR BOOT2                        ;GO CHECK TRACK 0 SWITCH
173272 012712 BOOT4:    MOV #XSEEKN, (R2)       ;STEP IN, HEAD DOWN
       022025
173276 105712 BOOT5:    TSTB (R2)                       ;DONE??
173300 100376           BPL BOOT5
173302 012703           MOV #22724, R3          ;WAIT 50 MILLISECS
       022724
173306 077301 BOOT6:    SOB R3, BOOT6
                        ;
                        ;BOOTY IS ENTRANCE FOR STAGE II
                        ;ON ENTRY....   R0      DESTROYED
                        ;               R1      PTS 2 FIRST BUFFER WORD
                        ;               R2      CONTROLLER STATUS..DO NOT DESTROY
                        ;               R3      DESTROYED
                        ;               R4      RETRY COUNT..DO NOT DESTROY
                        ;               R5      SECTOR NMBR TO FETCH
                        ;               SP      DESTROYED
                        ;RETURN IS TO LOCN 2
```

```
173110 010203 BOOT7:  MOV R2, R3          ;SET UP DATA PNTR
173112 012712         MOV #RXTS, (R2)     ;READ TRKSEC, READ DOWN
       000031
173116 105712 BOOT8:  TSTB (R2)           ;WAIT ON DONE BIT
173120 100376         BPL BOOT8
173122 005723         TST (R3)+           ;ERRORS??
173124 100742         BMI BOOT7           ;RETRY IF SO (=> STEP OUT & IN)
173126 122327         CMPB (R3)+, #1      ;STILL TRACK 1 ??
       000001
173132 001337         BNE BOOT7           ;RETRY IF NOT
173134 120513 BOOT9:  CMPB R5, (R3)       ;CORRECT SECTOR ??
 73136 001364         BNE BOOT7           ;NO..KEEP LOOKING
 73140 012712         MOV #RREAD, (R2)    ;READ THEM BITS
       000003
 73144 005303         DEC R3              ;ADJ TO WORD BOUNDARY
 73146 111706         MOVB (PC), SP       ;PUT A 100 INTO SP
 73150 010100         MOV R1, R0          ;PUT BUF ADR INTO
 73152 105712 BOOTW:  TSTB (R2)           ;WORKING REGISTER
 73154 100376         BPL BOOTW           ;DONE??
 73156 005712         TST (R2)
 73160 100724         BMI BOOT7           ;ERRORS...RETRY
 73162 011320 BOOTA:  MOV (R3), (R0)+     ;SPREAD THEM BITS
 73164 077602         SOB SP, BOOTA       ;MOVE 64 WORDS
 73166 021027         CMP (SP), (R0)+     ;CHECK FIRST WORD
 73170 010001         MOV R0, R1          ;VALIDATION CODE
 73172 001777 ALTO:   BNE ALTO            ;IF NOT, HANG UP
 73174 005007         CLR PC              ;JMP TO LOCATION 0
                      ;
       000001         .END
```

```
CXBROM  RT-11 MACRO VM02-12   19-MAY-77 08:40:54 PAGE 5+
SYMBOL TABLE

ADDR     173172          BOOT     173ØØØ          BOOTA    173162
BOOTB    173134          BOOTLO=  Ø1ØØØØ          BOOTM    173Ø22
BOOTX    173Ø32          BOOTY    173Ø12          BOOT1    173Ø44
BOOT2    173Ø56          BOOT3    173Ø66          BOOT4    173Ø72
BOOT5    173Ø76          BOOT6    1731Ø6          BOOT7    17311Ø
BOOT7G=  Ø1Ø11Ø G        BOOT8    173116          BOOT9    173152
DISPL  = Ø16ØØØ          HALT     173Ø42          PC      =%ØØØØØ7
CXBTLO=  173ØØØ          CXDSLO=  177ØØØ          R0      =%ØØØØØØ
R1      =%ØØØØØ1         R2      =%ØØØØØ2         R3      =%ØØØØØ3
R4      =%ØØØØØ4         R5      =%ØØØØØ5         SP      =%ØØØØØ6
TRKØØ  = ØØ1ØØØ          XREAD  = ØØØØ33          XRTS   = ØØØØ21
XSIIIN=  ØØØØØ5          XSEEKO=  ØØØØØ7
. ABS.   173176     ØØØ
         ØØØØØØ     ØØ1
ERRORS DETECTED:  Ø
FREE CORE: 154Ø2. WORDS

,CXBROM=CXBROM
```

```
                .TITLE  STANDARD SYSTEM BOOT
                        ;
                        ;PROPRIETARY TO TERAK CORPORATION
                        ;
                        ;THIS SOFTWARE IS PROVIDED UNDER A LICENSE FOR USE ON A
                        ;SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
                        ;INCLUSION OF THIS NOTICE. THIS SOFTWARE OR ANY COPIES
                        ;THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE
                        ;IN ANY WAY TO ANY PERSON EXCEPT FOR USE ON SUCH SYSTEM
                        ;AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO
                        ;AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES
                        ;REMAIN IN TERAK CORPORATION.
                        ;
                        ;TERAK ASSUMES NO RESPONSIBILITY FOR THE RELIABILITY OR
                        ;CONSEQUENCE OF USE OF ITS SOFTWARE ON EQUIPMENT WHICH
                        ;IS NOT SUPPLIED BY TERAK. THE INFORMATION IN THIS
                        ;SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD
                        ;NOT BE CONSTRUED AS A COMMITMENT BY TERAK CORPORATION.
                        ;
```

```
                        ;       8510 DATA PROCESSOR       SAMPLE "SYSTEM" BOOTSTRAP
                        ;
                        ;       THE FOLLOWING CODE WILL RESIDE ON SECTOR 1, TRACK 1 OF
                        ;       A SYSTEM DISK. THE REMAINING SPACE IN SECTOR 1 CAN BE
                        ;       USED TO LOAD VECTORS, AND LOAD OTHER CODE,ETC.
                        ;       THE LIST OF SECTORS IS IN DESCENDING ORDER TO SAVE SPACE.
                        ;       THE SECTOR LIST SHOWN CORRESPONDS TO THE RT-11 LOCATION
                        ;       FOR THE COMPLETE BOOTSTRAP, BUT MAY BE MODIFIED TO INCLUDE
                        ;       ANY OR ALL OF THE SECTORS ON TRACK 1 (0 THRU 31 OCTAL).
                        ;       THE SECTORS WILL BE LOADED INTO ASCENDING MEMORY SPACE FROM
                        ;       LOCATION 200 UP.
                        ;
                                .MCALL  .REGDEF
000000                          .REGDEF
                        ;
        000000                  .ASECT
                        ;
                        ;
        010112                  BOOTRTN = 10112         ;PERMANENT ADDRESS OF BOOT ROUTINE
        000200                  BOOT = 200              ;ADDRESS OF NEXT STAGE OF BOOT
                                ;DEFINED HERE ONLY FOR SAMPLE CODE SANS ERROR MESSAGE
                                ;
        000000                  . = 0
000000  010001                  MOV R0, R1              ;VALIDATION CODE, AND
                                                        ;UPDATE BUFFER POINTER
                                                        ; // SKIP OVER VECTORS
000002  000415                  BR BOOT1                ;//  FROM LOC'N 4 THRU 34
        000036                  . = 36                  ;\\  THIS ALLOWS LOADING OF VECTORS
                                                        ; \\ CONCURRENT WITH LOADING OF BOOTSTRAP

000036  113705  BOOT1:  MOVB @(PC)+, R5                 ;NEXT SECTOR INTO R5 (ALSO TRAP PSW)
000040  000177  POINTR: .WORD SECLST                    ;LIST OF BYTES
000042  001405          BEQ BOOT                        ;NON-ZERO SECTOR => CONTINUE LOADING
                        ;0 SECTOR =>    LOAD COMPLETE, BR TO NEXT STAGE OF THE BOOTSTRAP
000044  005367          DEC POINTR                      ;ADVANCE POINTER
        177770
000050  000137          JMP @#BOOTRTN   ;"CALL" BOOT ROUTINE ... IT RETURNS TO LOC'N 0
        010112
                        ;
                        ;THE ABOVE CODE FALLS SHORT OF LOC'N 54 (RT-11 SYSPTR)
                        ;
                        ;
        000170          . = 170         ;ADJUST THIS LOC'N TO REFLECT LENGTH OF LIST
000170     000          .BYTE 0                         ;LIST TERMINATOR
000171     027          .BYTE 27                        ;\\
000172     025          .BYTE 25                        ; \\     LOGICAL BLOCK 0
000173     023          .BYTE 23                        ; //     UNDER RT-11
000174     021          .BYTE 21                        ;//
000175     007          .BYTE 7                         ;\\      REMAINING THREE SECTORS
000176     005          .BYTE 5                         ; >>     OF LOGICAL BLOCK 0
000177     003  SECLST: .BYTE 3                         ;//      UNDER RT-11
                        ;LAST BYTE IS AT LOCATION 177 TO MAXIMIZE
                        ;PROGRAM SPACE AVAILABLE IN SECTOR 1
                        ;
                        ;
        000001'         .END
```

```
STANDARD SYSTEM BOOT     RT-11 MACRO V5.02-12   19-MAY-77 09:41:18 PAGE 2-
SYMBOL TABLE

BOOT   = 000000          BOOTRT= 012110          BOOT1  000035
PC     =000007           POINTR  000047          R0     =000000
R1     =000001           R2     =000002          R3     =000003
R4     =000004           R5     =000005          SYSLST  000177
SP     =000006
. ABS.  000377    000
        000002    001
ERRORS DETECTED: 0
FREE CORE: 15888. WORDS

.BOOT=BOOT
```

```
                .TITLE  STANDARD DUMMY BOOT
                ;
                ;PROPRIETARY TO TERAK CORPORATION
                ;
                ;THIS SOFTWARE IS PROVIDED UNDER A LICENSE FOR USE ON A
                ;SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE
                ;INCLUSION OF THIS NOTICE. THIS SOFTWARE OR ANY COPIES
                ;THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE
                ;IN ANY WAY TO ANY PERSON EXCEPT FOR USE ON SUCH SYSTEM
                ;AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO
                ;AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES
                ;REMAIN IN TERAK CORPORATION.
                ;
                ;TERAK ASSUMES NO RESPONSIBILITY FOR THE RELIABILITY OR
                ;CONSEQUENCE OF USE OF ITS SOFTWARE ON EQUIPMENT WHICH
                ;IS NOT SUPPLIED BY TERAK. THE INFORMATION IN THIS
                ;SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD
                ;NOT BE CONSTRUED AS A COMMITMENT BY TERAK CORPORATION.
                ;
```

```
                      ;                  9510 DATA PROCESSOR     SAMPLE "DUMMY" BOOT
                      ;
                      ;           THIS BOOT STRAP, WHEN IN PLACE ON SECTOR 1, TRACK 0 OF A
                      ;           FLOPPY DISK, WILL CAUSE THE ERROR MESSAGE TO BE PRINTED
                      ;           ONTO THE CONSOLE TERMINAL IF BOOTSTRAP IS ATTEMPTED
                      ;           FROM THIS DISK. THIS IS TYPICALLY USED TO PROTECT DATA DISKS.
                      ;
                                  .MCALL  .REGDEF
 000000                           .REGDEF
                      ;
                      ;
        000000                    .ASECT
                      ;
                      ;
        177564                    TPS = 177564                  ;CONSOLE
        177566                    TPB = 177566                  ;ADDRESS
                      ;
                      ;
        000000                    . = 0
 000000 010001                    MOV R0, R1                    ;VALIDATION CODE
 000002 000403                    BR DUMMY                      ;SKIP OVER BUS FAULT VECTOR
                      ;
                      ;
 000004 000030                    .WORD HANG                    ;SET UP BUS FAULT VECTOR
 000006 000340                    .WORD 340                     ;IN CASE THERE'S NO CONSOLE
                      ;
                      ;
 000010 012700        DUMMY:      MOV #MSG, R0
        000032
 000014 105737        1$:         TSTB @#TPS                    ;CONSOLE READY ???
        177564
 000020 100375                    BPL 1$                        ;NO ... LOOP
 000022 112037        2$:         MOVB (R0)+, @#TPB             ;PRINT A BYTE
        177566
 000026 100372                    BPL 1$                        ;LOOP TO END OF MESSAGE
 000030 000777        HANG:       BR HANG               ;KEEP OPERATOR AWAY FROM MICRO-ODT
                      ;
                      ;
                                  .NLIST BEX
 000032    000        MSG:        .BYTE 0,0,15,12,0,0,0,0,0         ;CR-LF AND FILLERS
 000043    077                    .ASCII /??-NO BOOT ON VOLUME/<15><12><200>
        000001'                   .END
```

SYMBOL TABLE

```
DUMMY   000210    R4%?    000232    MSG     000232
P0    =%000007    R0    =%007770    R1    =%000771
R2    =%177002    R3    =%000003    R4    =%000004
R5    =%777005    SP    =%000706    TP3   = 177564
TPS   = 177564
. ABS.  007372      000
        000000      001
ERRCPS DETECTED: 0
FREE CORE: 15494. WORDS

,DUMMY=DUMMY
```