FIELD MAINTENANCE PRINT SET

TABLE OF CONTENTS

FIELD MAINTENANCE PRINT SET REV11	B-TC-REV11-Ø-1
LSI11 REF BOOT CABLE CONN.	D-CS-M94ØØ-Ø-1
LSI11 REFRESH/BOOT/TERM-REV11	A-PL-REV11-Ø-Ø
LSI11 ROM LIST/BOOT NO. 1	B-CS-REV11-Ø-3
REV11 SHIPPING LIST	A-PL-REV11-Ø-2

UNIT VARIATIONS COVERED BY THIS PRINT SET
REV11-A
REV11-C
REV11-J

REVII VOL. I

Field Maintenance Print Set

Digital Equipment Corporation

PRINT SET ORDER NO.
MPØØØ73

2441
5 juin 76

REVISIONS	REV.	CHG. NO.	DATE

USED ON OPTION/MODEL	
LSI11	
SHEET I OF 1	

DRN. D. HEALY	DATE 1/20/76	TITLE: FIELD MAINTENANCE PRINT SET REV11 - VOLUME NO. 1
CHK'D D. HEALY	DATE 1/20/76	
PROJ. ENG.	DATE 3/12/76	
FIELD SERV.	DATE 3/10/76	

digital

SIZE	CODE	NUMBER	REV.
B	TC	REV11-Ø-1	
DIST.			

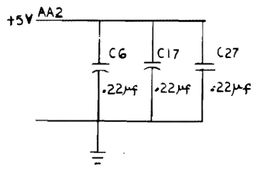
EN-01124-19-N975-(327)

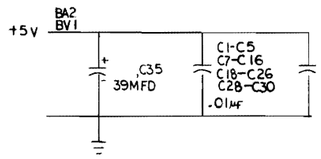
DRB 124

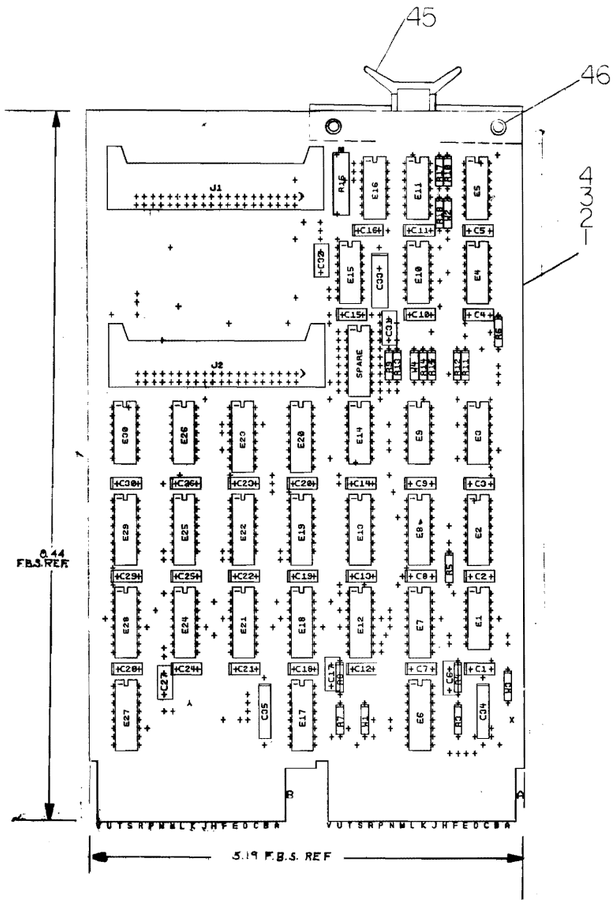
NOTES:

1. BD INIT L AND BD INIT H ARE SIGNALS GENERATED BY THIS MODULE THAT ARE USED ON THIS MODULE.
2. A STANDARD ROM STARTING LOCATION FOR THE FOLLOWING BOOTSTRAPS HAVE BEEN ESTABLISHED. ALL FUTURE REV II-X's THAT INCLUOR ANY OF THESE BOOTSTRAPS SHOULD USE THE SAME STARTING LOCATIONS. IF A NEW BOOTSTRAP IS ESTABLISHED, PLEASE ECO THIS NOTE TO INCLUDE THE DEVICE AND STARTING LOCATION.

BOOTSTRAP	STARTING LOCATION
.RXØI	165242
.ABSOLUTE LOADER	165406
.RKØ5	165650
.CPU DIAGNOSTIC (MEMORY ADDIFYING)	173302
.CPU DIAGNOSTIC (NON-MEMORY MODIFYING)	173000
.MEMORY DIAGNOSTIC	173626
.ODT	







IC 5624	8	16
IC 8136	8	16
IC 74174	8	16
IC 8556	8	16
IC 8641	8	16
IC TYPE	GND	+5V

GND AND 5V ARE USUALLY PIN 7 AND 14 RESPECTIVELY EXCEPTIONS ARE STATED ABOVE

IC PIN LOCATIONS

REF	REF	REF	REF	REF	REF	REF	REF				
REF	REF	REF	REF	REF	REF	REF	REF			ASSY/DRILLING HOLE LAYOUT	D-AH-M9400-0-5	2
REF	REF	REF	REF	REF	REF	REF	REF			MODULE ECO HISTORY	B-MH-M9400-0-6	3
1	1	1	1	1	1	1	1			ETCHED CIRCUIT BOARD	5011903	4
27	27	27			27		27	C1 THRU C5,C7 THRU C16,C18 THRU C26,C28,C29,C30		CAPACITOR, .01uf, 50V	1001610-01	5
3		3	3			3	3	C6,C17,C27		CAPACITOR, .22uf, 50V	1010274	6
1	1	1			1		1	C31		CAPACITOR, 680pf, 100V	1000026	7
1	1	1			1		1	C32		CAPACITOR, 22pf, 100V	1005820	8
1	1	1			1		1	C33		CAPACITOR, .01uf, 100V	1005784	9
2	2	2	2	2	2	2	2	C34,C35		CAPACITOR, 39uf, 10V, 10%	1000078	10
2			2	2				J1,J2		CONNECTOR	1209941-02	11
~~4~~	~~4~~	~~4~~			~~4~~		~~4~~	~~XE15,XE22,XE25,XE29~~		~~I.C. SOCKET~~	~~1211813~~	~~12~~
1	1	1			1		1	R9		RESISTOR, 75 OHM, 1/4W, 5%	1302379	13
~~1~~	~~1~~	~~1~~			~~1~~		~~1~~	~~R8~~		~~RESISTOR, 330 OHM, 1/4W, 5%~~	~~1300283~~	~~14~~
~~2~~	~~2~~	~~2~~			~~2~~		~~2~~	~~R11,R13~~		~~RESISTOR, 330 OHM, 1/4W, 5%~~	~~1300295~~	~~15~~
4	4	4			4		4	R4,R8,R11,R13		RESISTOR, 330 OHM, 1/4W, 5%	1300295	16
~~1~~	~~1~~	~~1~~			~~1~~		~~1~~	~~R7~~		~~RESISTOR, 680 OHM, 1/4W, 5%~~	~~1301424~~	~~17~~
~~2~~	~~2~~	~~2~~			~~2~~		~~2~~	~~R12,R14~~		~~RESISTOR, 680 OHM, 1/4W, 5%~~	~~1301424~~	~~18~~
4	4	4			4		4	R3 R7, R12, R14		RESISTOR, 680 OHM, 1/4W, 5%	1301424	19
4	4	4			4		4	R5,R6,R10,R15		RESISTOR, 1K, 1/4W, 5%	1300365	20
1	1	1			1		1	R16		RESISTOR, 50K, 3/4 W, 20%	1309143-13	21
1	1	1			1		1	R17		RESISTOR, 4.7K, 1/4W, 5%	1300447	22
1	1	1			1		1	R18		RESISTOR, 5.6K, 1/4W, 5%	1301874	23
1	1	1			1		1	E16		I.C. DEC 7404	1909686	24
1	1	1			1		1	E9		I.C. DEC 7408	1910155	25
4	4	4			4		4	E3,E5,E10,E26		I.C. DEC 7474	1905547	26
1	1	1			1		1	E1		I.C. DEC 8881	1909705	27
1	1	1			1		1	E14		I.C. DEC 7402	1909004	28
1	1	1			1		1	E11		I.C. DEC 7427	1910878	29
2	2	2			2		2	E4,E30		I.C. DEC 7400	1905575	30
6	6	6			6		6	E7,E12,E18,E21,E24,E28		I.C. DEC 8641	1911579	31
1	1	1			1		1	E15		I.C. DEC 4024	1911038	32
2	2	2			2		2	E20,E23		I.C. DEC DM 8556	1912951	33
1	1	1			1		1	E2		I.C. DEC8136	1912395	34
2	2	2			2		2	E8,E13		I.C. DEC 74174	1910652	35
1					1		1	E19		I.C. DEC IM5624	23974A9-00	36
1					1		1	E22		I.C. DEC IM5624	23975A9-00	37
1					1		1	E25		I.C. DEC IM5624	23976A9-00	38
1					1		1	E29		I.C. DEC IM5624	23977A9-00	39
		3				3	3	E6,E17,E27		RESISTOR PAK	1311003-01	40
3			3					E6,E17,E27		RESISTOR PAK	1311003-02	41
			1	1		1		W1		INSULATED JUMPER	9009185	42
2	2	2			2		2	W2,W4		INSULATED JUMPER	9009185	43
1	1	1	1	1	1	1	1	W3		INSULATED JUMPER	9009185	44
1	2	2	1	1	2	2	2			HANDLE, FLIP CHIP MAGENTA	9008337-06	45
4	4	4	4	4	4	4	4			EYELET	9006732	46
1	1	1	1	1	1	1	1			SPLIT LUG	9006735	47
1			1	1						SPACER	9009781	49
M9400-YJ	M9400-YH	M9400-YF	M9400-YE	M9400-YD	M9400-YC	M9400-YB	M9400-YA					

FIRST USED ON OPTION MODEL

QTY	REF. DESIGNATION	DESCRIPTION	PART NO.	ITEM NO.

PARTS LIST

ETCH BOARD REV. G

REV.	CHANGE NO.	CHK	DATE	NAME
F	M9400-00005		10/22/76	M.E. LEWANDOWSKI
E	M9400-00004		10/19-74	M.E. LEWANDOWSKI
D	M9400-00003		9-JUNE-76	M.E. LEWANDOWSKI
C	M9400-00002		4-29-76	M. LEWANDOWSKI
B	M9400-00001		4-12-76	M. LEWANDOWSKI

REVISIONS

DEC NO.	EIA NO.	DEC NO.	EIA NO.

SEMICONDUCTOR CONVERSION CHART

		DATE
DRN.	Gregory Wilder	
CHK'D.		DATE
ENG.		DATE 12-18-75
PROJ. ENG.		DATE 12-18-75
PROD.		DATE 12-18-75

NEXT HIGHER ASSY

SCALE

SHEET 1 OF 5

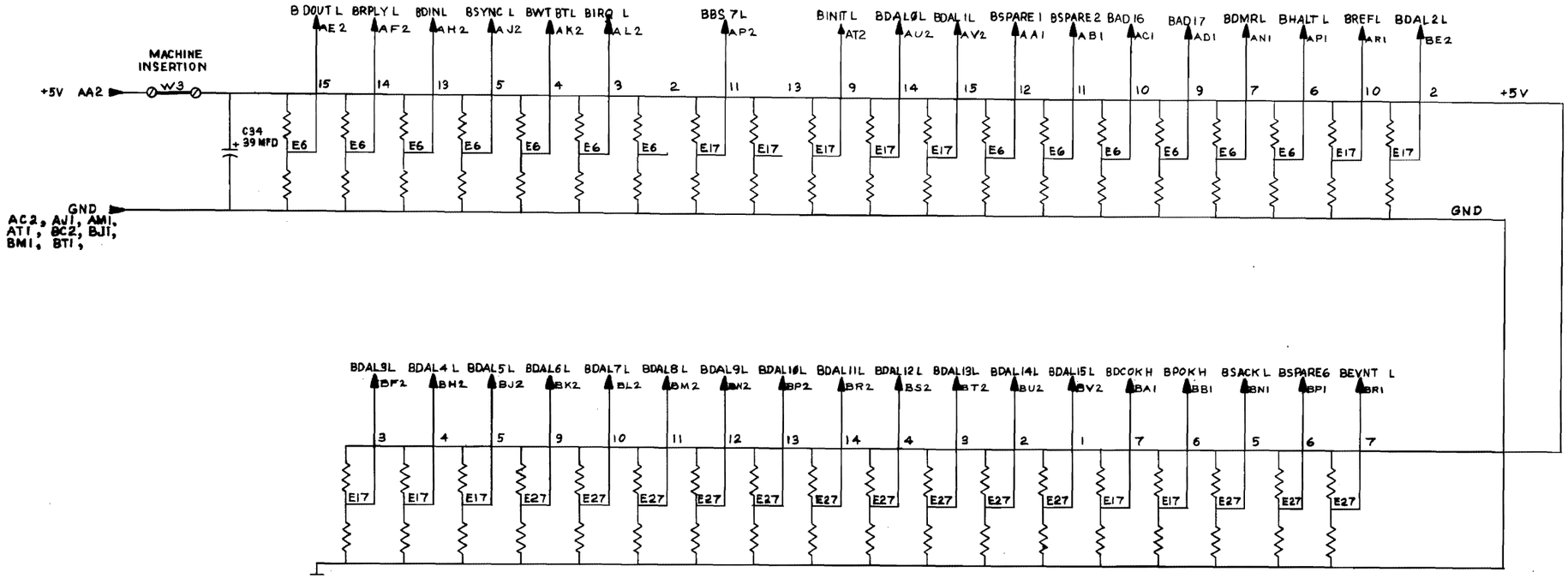
digital

TITLE: LSI REF. BOOT CABLE CONN.

SIZE	CODE	NUMBER	REV.
D	CS	M9400-0-1	F

DIST.

DEC FORM NO. DRD 135.8



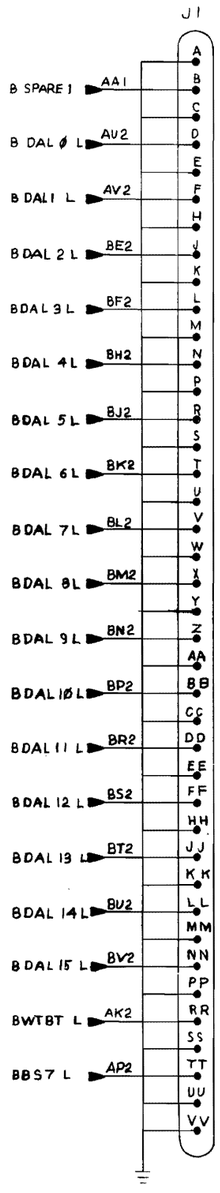
NOTE:
GROUND IS ON PIN 8
+5V IS ON PIN 16 OF EACH RESISTOR PACK.

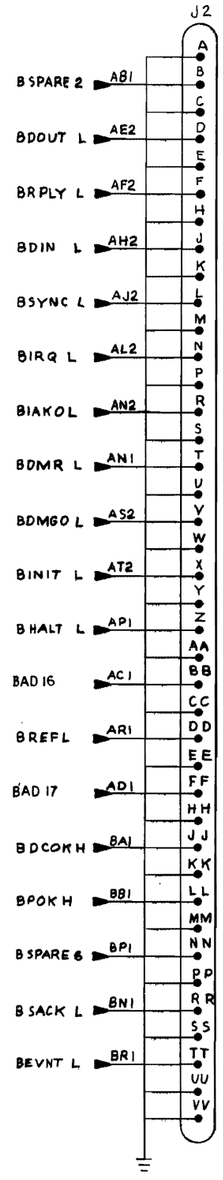
REVISIONS		
CHK	CHANGE NO.	REV.

TITLE	SIZE	CODE	NUMBER	REV.
LSI REF BOOT CABLE CONN	D	CS	M9400-0-1	F
SCALE — / —	SHEET 2 OF 5	DIST.		

DEC FORM NO. DRD 138

8 7 6 5 4 3 2 1





REVISIONS		
CHK	CHANGE NO.	REV.

DEC FORM NO. DRD 138

TITLE	SIZE	CODE	NUMBER	REV.
LSI REF BOOT CABLE CONN	D	CS	M94ØØ-Ø-1	E
SCALE —/—/—	SHEET 3 OF 5		DWG.	

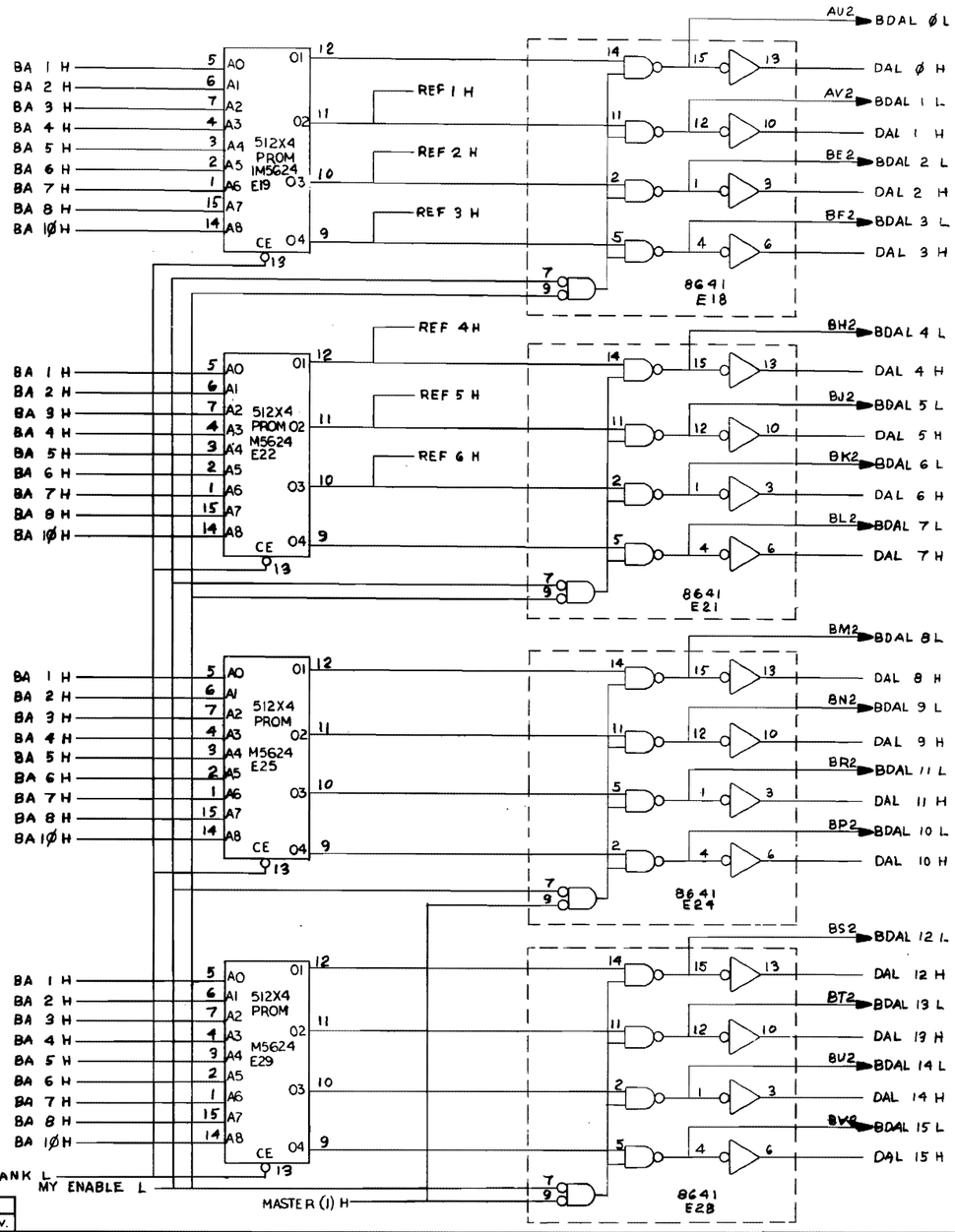
DCSM9400-0-1 REV F

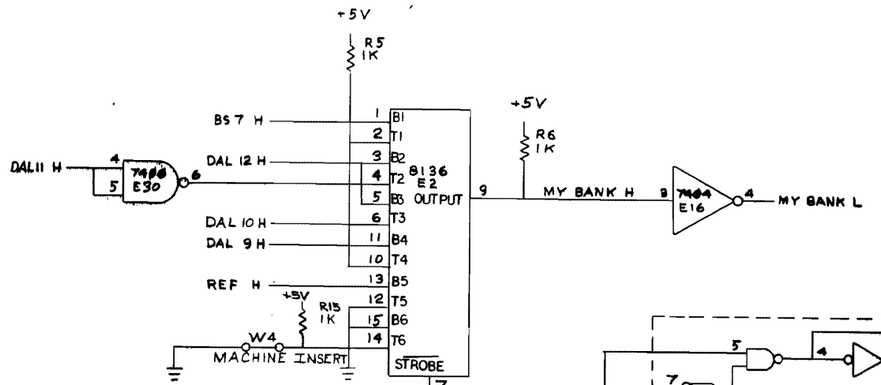
+5V
MACHINE INSERTION
R10 1K
W2
GRAB BUS L
7427 E11
REQ H
+3V
7474 E9
8881 E1
10 AN
BDMR L
R PENDING (1) H
BD INIT H
7402 E14
SACK (1) H
+3V
7474 E3
DMGR H
SACK (Ø) H
R CLR L
8641 E12
BN1
B SACK L
7404 E16
7404 E16
R9 75Ω
C31 680 PF
DMGR H
8881 E1
AS2
BDMGO L
7498 E9
SACK (Ø) H
DMG DLYD H
+3V
SACK (1) H
DMG DLYD H
7400 E30
7474 E26
MASTER (1) H
MASTER (Ø) H
R PLY H
SYNC H
7427 E11
BD INIT H
7402 E14
BCLR L
R PENDING (1) H
7427 E11
DONE (Ø) H
+5V
C32 22 PF
1024 E15
MSPAREA MSPAREA
AK1
AL1
SCLK H
+5V
R16 50KΩ
R17 4.7K
R18 5.6K
C33 .Ø1 uF
1024 E15
MSPAREB MSPAREB
BK1
BL1
+5V
+3V
7474 E26
B (Ø) H
DONE (Ø) H
MASTER SYNC (1) H
MASTER (111) H
8881 E1
AR1
BREF L
8641 E12
AJ2
BSYNC L
SYNC H
BD INIT H
A (Ø) H
7402 E14
RPLY (Ø) H
MASTER SYNC (1) H
B (Ø) H
7400 E4
7408 E4
7474 E10
A (1) H
SCLK H
A (Ø) H
MASTER (1) H
B (1) H
RPLY (1) H
7400 E4
A (1) H
7474 E10
B (1) H
SCLK H
B (Ø) H
BD INIT L
+3V
MASTER (1) H
7474 E5
MASTER SYNC (1) H
SCLK H
BD INIT L
+3V
RPLY H
SCLK H
7474 E5
R PLY (1) H
R PLY (Ø) H
PRE SYNC H
A (Ø) H
B (Ø) H
7400 E4
7408 E9
BD INIT L
MASTER SYNC (1) H
8881 E1
AJ2
B SYNC L
MASTER (1) H
7402 E14
MY ENABLE L
GRAB BUS H
7404 E16
GRAB BUS L
MY BANK H
DIN H
7408 E9
A (1) H
B (1) H
7408 E9
8641 E7
+5V
R8 330Ω
AR2
BDMG I L
R7 680Ω
DMGR H
BAI
BDCOK (H)
BD INIT H
7404 E16
BD INIT L
AF2
BRPLY L
RPLY H
AH2
BDIN L
DIN H
+5V
+5V
R13 330Ω
R11 330Ω
+3V
R14 680Ω
R12 680Ω
MASTER (Ø) H
DM8556 E23
I/O D
I/O C
I/O B
I/O A
CLK
LOAD
CE
RESET
REF 6 H
REF 5 H
BD INIT H
MASTER (Ø) H
DM8556 E20
MAX COUNT
I/O D
I/O C
I/O B
I/O A
CLK
LOAD
CE
RESET
REF 4 H
REF 3 H
REF 2 H
REF 1 H
BD INIT H
MASTER (1) H
RPLY (1) H
7400 E30
+3V

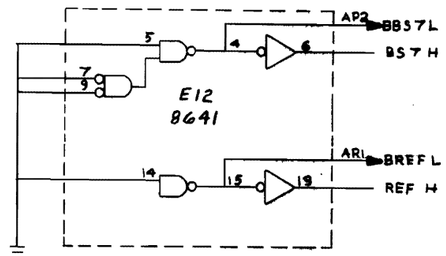
REVISIONS		
CHK	CHANGE NO.	REV.

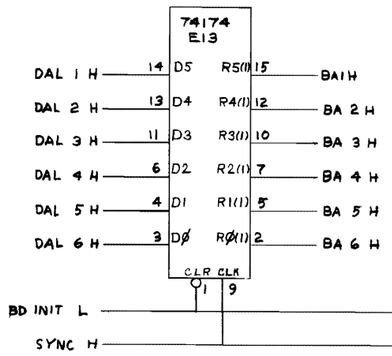
TITLE	SIZE	CODE	NUMBER	REV.
LSI REF BOOT CABLE CONN	D	CS	M9400-Ø-1	F
SCALE —	SHEET 4 OF 5		DIST.	

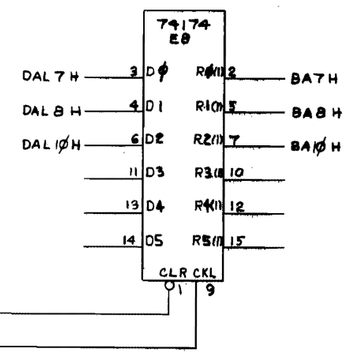
DEC FORM NO. DRD 138











REVISIONS		
CHK	CHANGE NO.	REV.

TITLE	SIZE	CODE	NUMBER	REV.
LSI REF BOOT CABLE CONN	D	CS	M9400-0-1	F
SCALE	SHEET 5 OF 5	DIST.		

DEC FORM NO. DRD 138

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS

PARTS LIST

MADE BY D. HEALY	CHECKED D. HEALY	SECTION 1
DATE 1/20/76	DATE 1/20/76	
ENG Tom Hitowski	PROD Dan DeRane	ISSUED SECT. 1
DATE 3/11/76	DATE 3-15-76	

ITEM NO.	DWG NO./PART NO.	DESCRIPTION	QUANTITY/VARIATION REV11-A	REV11-C	REV11-H	REV11-J
	D-CS-M9400-YA-1	BUS TERMINATOR/REFRESH & BOOT #1	1	-	-	-
	D-CS-M9400-YC-1	REFRESH & BOOT #1	-	1	-	-
	D-CS-M9400-YH-1	REFRESH & BOOT #2	-	-	1	-
	D-CS-M9400-YJ-1	REV11-A CABLE CONNECTORS	-	-	-	1

TITLE LSI-11 REFRESH/BOOT/TERM-REV11	ASSY NO. NONE	SIZE A	CODE PL	NUMBER REV11-0-0	REV. / ECO NO.
	SHEET 1 OF 1	DIST.			

DEC FORM DEC 16-(325)-1031-N870
DRA 110

M94ØØ TABLE OF CONTENTS

3	DEFINITIONS
47	ROM AREA 173ØØØ - 173777
377	MEMORY DIAGNOSTICS
489	ROM AREA 165ØØØ - 165777
497	KEYBOARD DISPATCH ROUTINE
541	GETCHR - GET & ECHO CHARACTER
596	GETNUM - GET OCTAL NUMBER INTO RO
623	OD - HALT FOR MICRO-ODT FUNCTIONS
664	RXV11 - FLOPPY DISK BOOTSTRAP
722	ABSLDR - ABSOLUTE BINARY LOADER -- VØØ7A
876	RK11 - DISK DRIVE BOOTSTRAP

FIRST USED ON OPTION MODEL	QTY.	DESCRIPTION	PART NO.	ITEM NO.
LSI11		PARTS LIST		

DRN. D. Neary	DATE 3 MAR 76	digital EQUIPMENT CORPORATION MAYNARD, MASSACHUSETTS	
CHK'D. D. Neary	DATE 4 MAR 76		
ENG. Tom Hotowski	DATE 3/11/76	TITLE LSI11 ROM LIST/BOOT NO. 1	
PROJ. ENG. M E Lawandowski	DATE 3/12/76		
PROD.	DATE 3-15-76		
NEXT HIGHER ASSEMBLY B-DD-REV11-Ø		SIZE K / CODE CS / NUMBER REV11-Ø-3	REV.
SCALE			
SHEET 1 OF 26		DIST.	

REVISIONS | REV. | CHANGE NO. | CHK

DEC FORM NO. DRB 109

SH #2

```
                                .SBTTL  DEFINITIONS
000000                  L.CKSM  =       %0
000000                  R0      =       %0
000001                  L.ADR   =       %1
000001                  R1      =       %1
000002                  L.BC    =       %2
000002                  R2      =       %2
000003                  L.BYT   =       %3
000003                  R3      =       %3
000004                  R4      =       %4
000005                  L.PTR   =       %5
000005                  R5      =       %5
000006                  SP      =       %6
000007                  PC      =       %7

177560                  TKS     =       177560  ;KEYBOARD STATUS REGISTER
177562                  TKB     =       177562  ;KEYBOARD BUFFER REGISTER
177564                  TPS     =       177564  ;CONSOLE PRINTER STATUS REGISTER
177566                  TPB     =       177566  ;CONSOLE PRINTER BUFFER REGISTER

000015                  CR      =       15      ;CARRIAGE RETURN
000012                  LF      =       12      ;LINE FEED

177170                  RXCS    =       177170  ;FLOPPY DISK COMMAND STATUS REGISTER

177550                  PCCS    =       177550  ;PC11 COMMAND STATUS REGISTER

177404                  RKCS    =       177404  ;WORD COUNT REGISTER FOR RK11


002000                  BIT10   =       002000  ;BIT 10 MASK

100000                  BIT15   =       100000  ;BIT 15 MASK

020040                  BIT5,13 =       020040  ;BIT 5 & 13 MASK
```

SH#3


```
                                        .ENABLE ABS

                                        .IIF NDF GONOGO, GONOGO = 173000

                                        .SBTTL  ROM AREA 173000 - 173777
                                        ;*************************************************
                                        ;*THIS AREA OF ROM CONTAINS THE GO-NOGO CPU
                                        ;*AND MEMORY TESTS
                                        ;*************************************************

            173000                      .       =       GONOGO
                                        ;*************************************************
                                        ;*THE GO-NOGO CPU INSTRUCTION TEST1 THRU TEST6
                                        ;*DO NOT MODIFY MEMORY
                                        ;*************************************************


                                        ;*************************************************
                                        ;*TEST1:
                                        ;*CHECK ALL SINGLE OPERAND INSTRUCTIONS
                                        ;*WITH DESTINATION MODE ZERO
                                        ;*************************************************
173000  012706  177777          TEST1:  MOV     #177777,SP      ;INITIALIZE SP TO CAUSE A DOUBLE
                                                                ; BUS ERROR ON A TRAP
173004  012700  000340                  MOV     #340,R0         ;HIGHEST PRIORITY
173010  106400                          MTPS    R0              ;DISABLE INTERRUPTS
                                                                ;SHOWN ARE THE CONTENTS OF THE REGISTER AND
                                                                ;CONDITION CODES THAT APPEAR AFTER THE
                                                                ;EXECUTION OF THE INSTRUCTION
                                                                ;  REG      NZVC
                                                                ; -----    ------
173012  005001                          CLR     R1              ;000000     0100
173014  005101                          COM     R1              ;177777     1001
173016  005401                          NEG     R1              ;000001     0001
173020  006001                          ROR     R1              ;100000     1000
173022  005501                          ADC     R1              ;100001     1000
173024  000261                          SEC                     ;100001     1001
173026  006101                          ROL     R1              ;000003     0011
173030  006201                          ASR     R1              ;000001     0011
173032  005601                          SBC     R1              ;000000     0100
173034  006301                          ASL     R1              ;000000     0100
173036  005201                          INC     R1              ;000001     0000
173040  005301                          DEC     R1              ;000000     0100
173042  001377                          BNE     .
```


SH#4

```
                                        ;*************************************************
                                        ;*TEST2:
                                        ;*CHECK ALL BYTE SINGLE OPERAND INSTRUCTIONS
                                        ;*WITH DESTINATION MODE ZERO
                                        ;*************************************************
173044  112702  000377          TEST2:  MOVB    #000377,R2      ;177777     1000
173050  105002                          CLRB    R2              ;177400     0100
173052  105102                          COMB    R2              ;177777     1001
173054  105602                          SBCB    R2              ;177776     1000
173056  105202                          INCB    R2              ;177777     1000
173060  105402                          NEGB    R2              ;177401     0001
173062  105302                          DECB    R2              ;177400     0101
173064  000302                          SWAB    R2              ;000377     1000
173066  106302                          ASLB    R2              ;000376     1001
173070  106002                          RORB    R2              ;000377     1010
173072  106202                          ASRB    R2              ;000377     1001
173074  106102                          ROLB    R2              ;000377     1001
173076  105502                          ADCB    R2              ;000000     0101
173100  105702                          TSTB    R2              ;000000     0100
173102  001377                          BNE     .
```

SH#5

```
                                        ;*****************************************************
                                        ;*TEST3:
                                        ;*CHECK ALL SOURCE MODES FOR DOUBLE OPERAND
                                        ;*WORD INSTRUCTIONS WITH DESTINATION MODE ZERO
                                        ;*THE SOURCE MODE BEING EXECUTED IS SHOWN AS
                                        ;*(SMX) IN THE COMMENTS
                                        ;*****************************************************

173104  012702  173264          TEST3:  MOV     #T3DATA,R2      ;MOVE DATA ADDRESS TO R2
173110  011201                          MOV     (R2),R1         ;MOVE TO (T3DATA) TO R1, (SM1)
173112  022201                          CMP     (R2)+,R1        ;CHECK DATA, (SM2)
173114  001377                          BNE     .               ;LOOP IF INCORRECT DATA
173116  063201                          ADD     @(R2)+,R1       ;ADD T3DATA TO R1, (SM3)
173120  165201                          SUB     @-(R2),R1       ;SUBTRACT T3ATA FROM R1, (SM5)
173122  044201                          BIC     -(R2),R1        ;CLEAR R1 BY NAND WITH ITSELF, (SM4)
173124  056201  000004                  BIS     4(R2),R1        ;INSERT ALL 1'S IN R1, (SM6)
173130  037201  000006                  BIT     @6(R2),R1       ;TEST ANY BITS SET, (SM7)
173134  001777                          BEQ     .               ;LOOP IF NOT
173136  074101                          XOR     R1,R1           ;CLEAR R1 BY XOR WITH SELF
173140  001377                          BNE     .               ;LOOP IF NOT ZERO


                                        ;*****************************************************
                                        ;*TEST4:
                                        ;*CHECK JUMP INSTRUCTION DESTINATION MODES - 1,3,6
                                        ;*THE DESTINATION MODE BEING EXECUTED IS SHOWN AS
                                        ;*(DMX) IN THE COMMENTS
                                        ;*****************************************************

173142  012701  173152          TEST4:  MOV     #JMP1,R1        ;LOAD ADDRESS FOR JUMP
173146  000111                          JMP     (R1)            ;JUMP TO JMP1, (DM1)
173150  000777                          BR      .               ;LOOP IF JUMP FAILS
173152  022121                  JMP1:   CMP     (R1)+,(R1)+     ;ADJUST POINTER FOR JMP3 ADDRESS
173154  000131                          JMP     @(R1)+          ;GET ADDR OF ADDR FOR JUMP TO JMP3, (DM3)
173156  173160                          .WORD   JMP3            ;ADDRESS MODE 3 JUMP
173160  000161  000006          JMP3:   JMP     6(R1)           ;JUMP AHEAD BY 4 TO JMP6, (DM6)
173164  000777                          BR      .               ;LOOP IF JUMP FAILED
```


SH #6

```
                                        ;*****************************************************
                                        ;*TEST5:
                                        ;*CHECK "TSTB" & "TST", DESTINATION MODES - 1,2,4,6
                                        ;*THE DESTINATION MODE BEING EXECUTED IS SHOWN AS
                                        ;*(DMX) FOR WORD OR (DMBX) FOR BYTE INSTRUCTIONS
                                        ;*****************************************************

173166  012701  173274          TEST5:  MOV     #T5DATA,R1      ;LOAD TEST DATA ADDRESS
173172  005767  000076                  TST     T5DATA          ;SHOULD BE 100000, (DM6)
173176  100377                          BPL     .               ;LOOP IF NOT MINUS
173200  105767  000070                  TSTB    T5DATA          ;SHOULD BE ZERO, (DMB6)
173204  001377                          BNE     .               ;LOOP IF NOT ZERO
173206  105721                          TSTB    (R1)+           ;TEST LOW ORDER BYTE = 0, (DMB2)
173210  001377                          BNE     .               ;LOOP IF NOT ZERO
173212  105711                          TSTB    (R1)            ;TEST HI ORDER BYTE = 200, (DMB1)
173214  100377                          BPL     .               ;LOOP IF NOT MINUS
173216  105741                          TSTB    -(R1)           ;TEST LOW ORDER BYTE = 0, (DMB4)
173220  001377                          BNE     .               ;LOOP IF NOT ZERO
173222  005721                          TST     (R1)+           ;TEST (T5DATA) = 100000, (DM2)
173224  100377                          BPL     .               ;LOOP IF NOT MINUS
173226  005711                          TST     (R1)            ;TEST (T5DATA+2) = 0, (DM1)
173230  001377                          BNE     .               ;LOOP IF NOT ZERO
173232  005741                          TST     -(R1)           ;TEST (T5DATA) = 100000, (DM4)
173234  100377                          BPL     .               ;LOOP IF NOT MINUS


                                        ;*****************************************************
                                        ;*TEST6:
                                        ;*CHECK BYTE DOUBLE OPERAND INSTRUCTIONS WITH
                                        ;*DESTINATION MODE ZERO
                                        ;*****************************************************

173236  012701  173300          TEST6:  MOV     #T6DATA,R1      ;LOAD DATA ADDRESS
173242  112102                          MOVB    (R1)+,R2        ;LOAD 377 IN R2
173244  141102                          BICB    (R1),R2         ;COMPLEMENT BITS 0,3,5,7 IN R2
173246  131102                          BITB    (R1),R2         ;TEST BITS 0,3,5,7 = 0
173250  001377                          BNE     .               ;LOOP IF SET
173252  151102                          BISB    (R1),R2         ;SET BITS 0,3,5,7
173254  124102                          CMPB    -(R1),R2        ;TEST R2 = 377
173256  001377                          BNE     .               ;LOOP IF NOT EQUAL

173260  000167  171522                  JMP     KBDS            ;GO TO KEYBOARD ROUTINE

173264  173264          T3DATA: .WORD   T3DATA          ;WARNING - THIS DATA IS POSITION
173266  173264                  .WORD   T3DATA          ;   DEPENDANT, DO NOT INSERT DATA
173270  177777          T3DAT4: .WORD   177777          ;  WITHIN THE LIST
173272  173270                  .WORD   T3DAT4
173274  100000          T5DATA: .WORD   100000
173276  000000                  .WORD   0
173300  125377          T6DATA: .WORD   125377
```

SH#7

```
                        ;*****************************************************
                        ;*THE FOLLOWING CPU INSTRUCTION TESTS MODIFY MEMORY
                        ;*LOCATIONS 500 & 502 AS NOTED IN THE COMMENT FIELDS
                        ;*****************************************************

                        ;*****************************************************
                        ;*TEST7:
                        ;*CHECK WORD INSTRUCTIONS WITH DESTINATION
                        ;*MODES NOT ZERO
                        ;*
                        ;*CHECK "MOV" INSTRUCTION WITH DESTINATION MODES =1,2,3,4
                        ;*
                        ;*ALL DESTINATION MODES ARE CHECKED & A LIST OF INSTRUCTIONS
                        ;*AND DESTINATION MODES [IN PARENTHESIS] USED TO PERFORM
                        ;*THIS CHECK IS GIVEN HERE FOR REFERENCE
                        ;*    CLR(1,6),COM(1),INC(7),DEC(3,5)
                        ;*    NEG(4),ASR(1),ASL(5),ROR(3)
                        ;*    ROL(6),SUB(1),BIT(6),BIC(2),BIS(3)
                        ;*****************************************************

                                                                ;                       CONTENTS AFTER EXECUTION
                                                                ;                            500
                                                                ;                           ------
173302  012702  173436  TEST7:  MOV     #T7DATA,R2      ;LOAD TEST DATA ADDRESS
173306  016203  000002          MOV     2(R2),R3        ;LOAD 500 IN R3
173312  012213                  MOV     (R2)+,(R3)      ;WRITE -1 IN LOCATION 500       177777
173314  024223                  CMP     -(R2),(R3)+     ;CHECK DATA
173316  001046                  BNE     T7ERR           ;ERROR, IF NOT EQUAL
                                                        ;                            502
                                                        ;                           ------
173320  005063  177776          CLR     -2(R3)          ;CLEAR 500                      000000
173324  012232                  MOV     (R2)+,@(R2)+    ;LOAD -1 IN LOCATION 500         177777
173326  062243                  ADD     (R2)+,-(R3)     ;ADD 1 TO LOCATION 500           000000
173330  164213                  SUB     -(R2),(R3)      ;SUBTRACT 1 FROM LOCATION 500    177777
173332  005272  177776          INC     @-2(R2)         ;INCREMENT LOCATION 500          000000
173336  005352                  DEC     @-(R2)          ;DECREMENT LOCATION 500          177777
173340  044223                  BIC     -(R2),(R3)+     ;ZERO 500 BY COMPLEMENTING ALL BITS  000000
173342  001034                  BNE     T7ERR           ;ERROR, IF NOT ZERO
                                                        ;                            502
                                                        ;                           ------
173344  012223                  MOV     (R2)+,(R3)+     ;LOAD -1 INTO LOCATION 502       177777
173346  005443                  NEG     -(R3)           ;NEGATE LOCATION 502             000001
173350  031362  000002          BIT     (R3),2(R2)      ;TEST FOR BIT 0 ON
173354  001427                  BEQ     T7ERR           ;ERROR, IF NOT SET

173356  006213                  ASR     (R3)            ;SHIFT RIGHT LOCATION 502        000000
173360  116213  000002          MOVB    2(R2),(R3)      ;LOAD +1 IN LOCATION 502         000001
173364  005623                  SBC     (R3)+           ;SUBTRACT CARRY(=1) FROM LOCATION 502  000000
173366  006163  177776          ROL     -2(R3)          ;ROTATE LEFT LOCATION 502        000000
173372  001020                  BNE     T7ERR           ;ERROR, IF NOT ZERO
```


SH#8

```
                                                        ;                            500
                                                        ;                           ------
173374  056232  000002          BIS     2(R2),@(R2)+    ;SET BIT 0 IN LOCATION 500       000001
173400  006352                  ASL     @-(R2)          ;SHIFT LEFT LOCATION 500         000002
173402  006032                  ROR     @(R2)+          ;ROTATE RIGHT LOCATION 500       000001
173404  000261                  SEC                     ;SET CARRY
173406  005552                  ADC     @-(R2)          ;ADD CARRY TO LOCATION 500       000002
173410  005332                  DEC     @(R2)+          ;DECREMENT LOCATION 500          000001
173412  021252                  CMP     (R2),@-(R2)     ;COMPARE LOCATION 500 TO +1
173414  001007                  BNE     T7ERR           ;ERROR, IF NOT EQUAL
                                                        ;                            502
                                                        ;                           ------
173416  016243  000002          MOV     2(R2),-(R3)     ;LOAD +1 IN LOCATION 502         000001
173422  005013                  CLR     (R3)            ;CLEAR LOCATION 502              000000
173424  005113                  COM     (R3)            ;COMPLEMENT LOCATION 502         177777
173426  021362  177776          CMP     (R3),-2(R2)     ;COMPARE LOCATION 502 TO -1
173432  001404                  BEQ     TEST8           ;ERROR, IF NOT EQUAL
173434  000000          T7ERR:  HALT                    ;HALT

                                                        ;WARNING -
                                                        ;THE ORDER OF THIS DATA IS POSITION DEPENDANT
173436  177777          T7DATA: .WORD   177777          ;-1
173440  000500                  .WORD   000500          ;DEFERRED ADDRESS
173442  000001                  .WORD   000001          ;+1
```

SH#9

```
                                        ;*****************************************************
                                        ;*TEST8:
                                        ;*CHECK ALL BYTE DESTINATION MODES WHILE EXECUTING
                                        ;*ALL BYTE INSTRUCTIONS FOR DESTINATION MODES NOT ZERO
                                        ;*DESTINATION MODES FOR EACH BYTE INSTRUCTION IS
                                        ;*SHOWN AS (DMBX) IN THE COMMENT FIELDS
                                        ;*****************************************************

                                                                ;CONTENTS AFTER INSTRUCTION EXECUTION
                                                                ;  500          NZVC
                                                                ; -----        ------
173444  011203                  TEST8:  MOV     (R2),R3         ;LOAD 500 INTO R3
173446  016213  177776                  MOV     -2(R2),(R3)     ;177777
173452  105013                          CLRB    (R3)            ;177400 (DMB1)  0100
173454  105123                          COMB    (R3)+           ;177777 (DMB2)  0000
173456  105632                          SBCB    @(R2)+          ;177776 (DMB3)  1001
173460  105243                          INCB    -(R3)           ;177777 (DMB4)  1000
173462  105452                          NEGB    @-(R2)          ;177401 (DMB5)  0001
173464  105323                          DECB    (R3)+           ;177400 (DMB2)  0101
173466  000363  177777                  SWAB    -1(R3)          ;000377 (DMB6)  1000
173472  106332                          ASLB    @(R2)+          ;000376 (DMB3)  1010
173474  106072  177776                  RORB    @-2(R2)         ;000377 (DMB7)  1010
173500  106243                          ASRB    -(R3)           ;000377 (DMB4)  1001
173502  106113                          ROLB    (R3)            ;000377 (DMB1)  1001
173504  105523                          ADCB    (R3)+           ;000000 (DMB2)  0101
173506  121343                          CMPB    (R3),-(R3)      ;000000 (DMB4)  0101
173510  001007                          BNE     T8ERR           ;BR, IF LOCATION IS NOT ZERO
                                        ;
173512  156252  177774                  BISB    -4(R2),@-(R2)   ;000377 (DMB5)  1001
173516  146223  177776                  BICB    -2(R2),(R3)+    ;000000 (DMB2)  0101
173522  136243  177776                  BITB    -2(R2),-(R3)    ;000000 (DMB4)  0101
173526  001401                          BEQ     TEST9           ;BR, IF LOC, 500 IS ZERO
173530  000000                  T8ERR:  HALT                    ;ERROR OCURRED IN TEST8
```


SH#10

```
                                        ;*****************************************************
                                        ;*TEST9:
                                        ;*CHECK "JSR" INSTRUCTION WITH DESTINATION MODE = 7
                                        ;*USE R1 AND THEN PC AS THE LINKAGE REGISTER
                                        ;*****************************************************

173532  012702  173620          TEST9:  MOV     #TJSR,R2        ;INITIALIZE R2 WITH DATA ADDRESS
173536  010306                          MOV     R3,SP           ;SET SP = 500
173540  005726                          TST     (SP)+           ;INCREMENT SP TO 502
173542  011203                          MOV     (R2),R3         ;SET R1 = 177777
173544  004372  000002                  JSR     R3,@2(R2)       ;PERFORM JSR TO JSR7, (DM7)
173550  000000                          HALT                    ;JSR DID NOT WORK
173552  004772  000002                  JSR     PC,@2(R2)       ;PERFORM JSR TO JSR7P
173556  000000                  T9ERR:  HALT                    ;JSR FAILED
173560  022704  173302                  CMP     #TEST7,R4       ;WAS IT A DIAG COMMAND
173564  001020                          BNE     MEM             ;BR TO MEM TEST IF BOOT COMMAND
173566  000167  171214          KBJMP:  JMP     KBD$            ;RESTART KEYBOARD DISPATCH ROUTINE
173572  005723                  JSR7:   TST     (R3)+           ;INCRTEMENT RETURN PAST HALT AND
173574  001370                          BNE     T9ERR           ;BR IF A HALT WAS NOT IN THE RETURN LOCATION
                                        ;
173576  021622                          CMP     (SP),(R2)+      ;TEST FOR AN ALL 1'S WORD ON THE STACK
173600  001366                          BNE     T9ERR           ;ERROR IF NOT
173602  000203                          RTS     R3              ;RETURN
173604  000000                          HALT                    ;"RTS" RETURN FAILED
173606  005736                  JSR7P:  TST     @(SP)+          ;TEST THAT THE RETURN LOCATION IS A HALT
173610  001362                          BNE     T9ERR           ;ERROR IF NOT ZERO
173612  062746  000002                  ADD     #2,-(SP)        ;INCREMENT RETURN ADDRESS PAST HALT
173616  000207                          RTS     PC              ;RETURN
                                        ;
                                                                ;WARNING -
173620  177777          TJSR:   .WORD   177777          ;THE ORDER OF THIS DATA IS POSITION DEPENDANT
173622  173572                  .WORD   JSR7
173624  173606          TJSRP:  .WORD   JSR7P
```

SH #11

```
                                        .SBTTL  MEMORY DIAGNOSTICS
                                        ;**************************************************
                                        ;*MEMORY TEST:
                                        ;*THIS TEST CHECKS THE UPPER 4K OF MEMORY AND THEN
                                        ;*CHECKS THE REMAINDER OF THE MEMORY UP TO 28K
                                        ;**************************************************

                                        ;*CHECK UPPER 4K OF MEMORY

173626  005006                  MEM:    CLR     SP              ;STARTING ADDRESS FOR TEST
173630  012705  017776                  MOV     #17776,R5       ;UPPER BOUNDARY FOR 4K
173634  005003                          CLR     R3              ;SET DATA = 0
173636  010701                          MOV     PC,R1           ;SUBROUTINE CALL
173640  000424                          BR      MEMDO           ;GO WRITE MEMORY, (UPPER 4K)

                                        ;*NOW SIZE THE REMAINDER OF MEMORY

173642  012705  160000          MEMSIZ: MOV     #160000,R5      ;SET MAX ADDRESS
173646  012702  000006                  MOV     #6,R2           ;SET POINTER FOR TIMEOUT
173652  012712  000340                  MOV     #340,(R2)       ;KEEP PRIORITY AT 7
173656  010742                          MOV     PC,-(R2)        ;STORE PC FOR TIMEOUT TRAP
173660  012706  000502                  MOV     #502,SP         ;RESET STACK POINTER
173664  005045                          CLR     -(R5)           ;CHECK FOR HIGHEST WRITABLE MEMORY

                                        ;*RESTORE TRAPS CATCHER

173666  010701                          MOV     PC,R1           ;SUBROUTINE CALL
173670  000167  171254                  JMP     TRAPS           ;SET UP TRAP HALTS

                                        ;*CHECK REMAINDER OF MEMORY

173674  010701                          MOV     PC,R1           ;SUBROUTINE CALL
173676  000405                          BR      MEMDO           ;GO WRITE REMAINDER OF MEMORY
173700  022704  173626                  CMP     #MEM,R4         ;WAS IT A DIAG COMMAND
173704  001730                          BEQ     KBJMP           ;BR TO KEYBOARD ROUTINE IF YES
173706  000167  171332                  JMP     BTSTP           ;GO DO BOOT
```


SH #12

```
                                        ;**************************************************
                                        ;*SUBROUTINE TO WRITE, READ & VERIFY MEMORY
                                        ;*
                                        ;*CALLING SEQUENCE
                                        ;*      SET R3=0
                                        ;*      SET SP TO LOW ADDRESS FOR VERIFICATION
                                        ;*      SET R5 TO HIGH ADDRESS FOR VERIFICATION
                                        ;*      MOV PC,R1
                                        ;*      BR MEMDO
                                        ;*
                                        ;*SUBROUTINE ONLY RETURNS IF NO ERROR IS DETECTED
                                        ;*THE CONTENTS OF R2 ARE LOST.
                                        ;*THE TEST CONSISTS OF A MEMORY ADDRESS & DATA STORAGE TEST
                                        ;**************************************************

                                        ;**************************************************
                                        ;*DUAL MEMORY ADDRESS TEST:
                                        ;*THIS TEST WRITES ALL MEMORY WITH ITS ADDRESS AND
                                        ;*THEN READS AND VERIFIES ALL MEMORY
                                        ;**************************************************

173712  010602          MEMDO:  MOV     SP,R2           ;COPY STARTING ADDRESS
173714  010222          ADDRT:  MOV     R2,(R2)+        ;STORE ADDRESS AT ADDRESS
173716  020205                  CMP     R2,R5           ;FINISHED WITH ADDRESSES?
173720  101775                  BLOS    ADDRT           ;NO CONTINUE
173722  024202          CHECK:  CMP     -(R2),R2        ;YES, CHECK DATA
173724  001402                  BEQ     ADCONT          ;BR IF NO COMPARISION ERROR
173726  010203                  MOV     R2,R3           ;STORE EXPECTED DATA IN R3
173730  000000          MEMERA: HALT                    ;MEMORY ADDRESS ERROR, EXPECTED DATA IS IN R3
                                                        ;   BAD DATA IS POINTED TO BY ADDRESS IN R2
                                                        ;   TYPE "P" TO CONTINUE TEST
173732  020206          ADCONT: CMP     R2,SP           ;FINISHED CHECK?
173734  001372                  BNE     CHECK           ;BR IF NOT FINISHED
```

SH # 13

```
                                        ;*************************************************
                                        ;*DATA STORAGE TEST:
                                        ;*THE STEPS OF THIS TEST ARE:
                                        ;*    1. FILL MEMORY WITH ZEROES
                                        ;*    2. WALK AN ALL 1'S WORD THRU MEMORY & VERIFY EACH LOCATION
                                        ;*    3. FILL MEMORY WITH ONES
                                        ;*    4. WALK AN ALL 0'S WORD THRU MEMORY & VERIFY EACH LOCATION
                                        ;*BY PERFORMING THESE STEPS ALL BIT POSITIONS ARE CHECK FOR 0/1 STORAGE
                                        ;*AND THE SENSE AMPS ARE STRESSED IN SEMICONDUCTOR MEMORIES
                                        ;*************************************************

173736  010322          MEMT:   MOV     R3,(R2)+        ;MOVE BACKGROUND DATA TO MEMORY
173740  020205                  CMP     R2,R5           ;DONE?
173742  101775                  BLOS    MEMT            ;BR, IF MEMORY NOT FILLED
173744  005103                  COM     R3              ;COMP. TEST DATA
173746  010342          WALK:   MOV     R3,-(R2)        ;LOAD TEST DATA IN MEMORY LOCATION @(R2 - 2)
173750  020312                  CMP     R3,(R2)         ;CHECK FOR CORRECT DATA
173752  001401                  BEQ     DCONT           ;BR, IF DATA GOOD
173754  000000          MEMERD: HALT                    ;HALT, BAD DATA. EXPECTED DATA IS IN R3
                                                        ;   BAD DATA IS POINTED TO BY ADDRESS IN R2
                                                        ;   TYPE "P" TO CONTINUE TEST
173756  005103          DCONT:  COM     R3              ;COMP. TEST DATA TO GET PREVIOUS BACKGROUND DATA
173760  010312                  MOV     R3,(R2)         ;RESTORE BACKGROUND DATA FOR PRESENT MEMORY LOCATION
173762  005103                  COM     R3              ;RESTORE TEST DATA FOR THIS PASS
173764  020206                  CMP     R2,SP           ;DONE?
173766  001367                  BNE     WALK            ;BR, IF TEST IS NOT COMPLETED FOR ENTIRE MEMORY
173770  005703                  TST     R3              ;DONE WALKING 1'S & 0'S ? (0'S ARE DONE LAST)
173772  001361                  BNE     MEMT            ;BR, IF WALKING 0'S HAS NOT BEEN DONE
173774  000161  000002          JMP     2(R1)           ;RETURN TO CALLER
```


SH # 14

```
                                        .SBTTL  ROM AREA 165000 - 165777
                                        .IIF NDF $BOOTS, $BOOTS=165000
                                        ;*************************************************
                                        ;*THIS AREA OF ROM CONTAINS THE KEYBOARD DISPATCH
                                        ;*ROUTINE AND THE LOADER PROGRAMS
                                        ;*************************************************
165000                          .       =       $BOOTS

                                        .SBTTL  KEYBOARD DISPATCH ROUTINE
                                        ;*************************************************
                                        ;*KEYBOARD DISPATCH ROUTINE:
                                        ;*THIS ROUTINE OUTPUTS THE PROMPT CHARACTERS AND
                                        ;*INTERPRETS AND EXECUTES THE COMMAND STRINGS
                                        ;*************************************************

165000  012700  165734          KBDQ:   MOV     #MSGQ,R0        ;OUTPUT -> <?><LF>
165004  000402                  KBDS:   BR      PRINT           ;BRANCH TO INDICATE A BAD COMMAND
165006  012700  165736                  MOV     #MSG,R0         ;OUTPUT -> <CR><LF><'S><0>

165012  112002          PRINT:  MOVB    (R0)+,R2        ;PRINT A CHARACTER
165014  010703                  MOV     PC,R3           ;COPY PC FOR RETURN
165016  000440                  BR      PUTCHR          ;PRINT CHARACTER
165020  001374                  BNE     PRINT           ;BR IF MORE TO PRINT

165022  013701  177562          MOV     @#TKB,R1        ;CLEAR KEYBOARD RBUF
165026  010703          READ:   MOV     PC,R3           ;COPY PC FOR RETURN
165030  000425                  BR      GETCHR          ;GET AND ECHO CHAR
165032  000302                  SWAB    R2              ;ROTATE CHAR
165034  001774                  BEQ     READ            ;BR IF 2ND CHAR NOT READ

165036  042702  020040          BIC     #BIT5,13,R2     ;CLEAR BITS 5 & 13 OF COMMAND
                                                        ; (ALWAYS MAKES COMMAND UPPER CASE ASCII)
165042  012004          SEARCH: MOV     (R0)+,R4        ;COPY -> FUNCTION ADDRESS
165044  001755                  BEQ     KBDQ            ;IF END OF LIST, BR & PRINT<?><LF>
                                                        ;   TO INDICATE A BAD COMMAND
165046  020220                  CMP     R2,(R0)+        ;SEARCH FOR MATCHING COMMAND
165050  001374                  BNE     SEARCH          ;BR IF NO MATCH MADE
165052  005710                  TST     (R0)            ;IF R0 IS ZERO, THEN THE PRESENT COMMAND
165054  001410                  BEQ     DOIT            ;   IS "OD" & TRAPS ARE NOT SET UP
165056  010701                  MOV     PC,R1           ;COPY PC FOR RETURN
165060  000445                  BR      GETNUM          ;GET OPTIONAL VALUE FOR CMD
                                                        ;INITIALIZE TRAPS
165062  010701                  MOV     PC,R1           ;SUBROUTINE CALL
165064  000167  000060          JMP     TRAPS           ;GO SET TRAPS HALTS
165070  032704  002000          BIT     #BIT10,R4       ;TEST BIT 10 TO DISTINGUISH A BOOT
                                                        ;ADDRESS FROM A DIAGNOSTIC ADDRESS
165074  001401                  BEQ     BTCOMM          ;BR IF BOOT
165076  000114          DOIT:   JMP     (R4)            ;PERFORM DIAGNOSTIC OR "OD" COMMAND
165100  000167  006176  BTCOMM: JMP     TEST7           ;DO DIAGNOSTIC (CPU & MEMORY) TESTS BEFORE BOOT
```

SH#15

```
                                        .SBTTL  GETCHR  - GET & ECHO CHARACTER
                                        ;*************************************************
                                        ;*SUBROUTINE TO READ & ECHO A CHARACTER
                                        ;*THE CHARACTER IS RETURNED IN R2
                                        ;*THE CALLING SEQUENCE IS:
                                        ;*      MOV     PC,R3
                                        ;*      BR      GETCHR
                                        ;*TO JUST OUTPUT A CHARACTER ENTER AT "PUTCHR"
                                        ;*************************************************

165104  105737  177560        GETCHR: TSTB    @#TKS           ;CHAR READY
165110  100375                        BPL     GETCHR          ;BR IF NOT
165112  105002                        CLRB    R2              ;CLEAR FOR TRANSFER
165114  153702  177562                BISB    @#TKB,R2        ;TRANSFER CHAR
165120  105737  177564        PUTCHR: TSTB    @#TPS           ;PRINTER READY
165124  100375                        BPL     PUTCHR          ;BR IF NOT
165126  110237  177566                MOVB    R2,@#TPB        ;PRINT CHAR
165132  022323                        CMP     (R3)+,(R3)+     ;INDEX POINTER TO BR + 4
165134  142702  000200                BICB    #200,R2         ;CLEAR PARITY BIT
165140  000163  177776                JMP     -2(R3)          ;RETURN TO BR + 2


                                        ;*************************************************
                                        ;*RETURN INSTRUCTIONS FOR SUBROUTINES ENTERED BY
                                        ;*      MOV     PC,R1
                                        ;*      BR      XXXXX
                                        ;*************************************************

165144  005721                X1RTN:  TST     (R1)+           ;SET POINTER TO RTN TO BRANCH+2
165146  000111                        JMP     (R1)            ;RETURN


                                        ;*************************************************
                                        ;*SUBROUTINE TO SET UP TRAP & DEVICE INTERRUPT HALTS
                                        ;*
                                        ;* CALLING SEQUENCE
                                        ;*      MOV  PC,R1
                                        ;*      JMP  TRAPS
                                        ;*
                                        ;*THE CONTENTS OF R2 IS LOST, R2 RETURNS WITH ZERO
                                        ;*************************************************

165150  012706  000500        TRAPS:  MOV     #500,SP         ;INITIALIZE STACK POINTER
165154  012702  000400                MOV     #400,R2         ;POINT R2 TO THE TOP OF THE VECTOR AREA
165160  005042                TCONT:  CLR     -(R2)           ;PUT HALT IN LOCATION AFTER TRAP VECTOR
165162  010242                        MOV     R2,-(R2)        ;POINT TRAP VECTOR TO HALT
165164  005702                        TST     R2              ;IF R2=0 TRAPS ARE ALL DONE
165166  001374                        BNE     TCONT           ;BR IF NOT DONE
165170  000161  000004                JMP     4(R1)           ;RETURN TO CALLER
```


SH#16

```
                                        .SBTTL  GETNUM  - GET OCTAL NUMBER INTO R0
                                        ;*************************************************
                                        ;*THIS SUBROUTINE READS A CHARACTER, VERIFIES THAT IT
                                        ;*IS A VALID OCTAL NUMBER, AND STORES IT IN R0
                                        ;*THE CALLING SEQUENCE IS:
                                        ;*      MOV     PC,R1
                                        ;*      BR      GETNUM
                                        ;*THE CONTENTS OF R2 & R3 ARE LOST
                                        ;*************************************************

165174  005000                GETNUM: CLR     R0              ;INITIALIZE RESULT
165176  005002                2$:     CLR     R2              ;CLEAR FOR BYTE XFER
165200  010703                        MOV     PC,R3           ;COPY PC FOR RETURN
165202  000740                        BR      GETCHR          ;GET & ECHO CHAR
165204  120227  000015                CMPB    R2,#CR          ;TERMINATED VIA CR?
165210  001755                        BEQ     X1RTN           ;RETURN IF SO
165212  162702  000070                SUB     #'8,R2          ;CHECK FOR OCTIT
165216  062702  000010                ADD     #'8-'0,R2       ;       "
165222  103266                        BCC     KBDQ            ;BR IF NOT
165224  006300                        ASL     R0              ;MAKE ROOM FOR IT IN RESULT
165226  006300                        ASL     R0              ;       "
165230  006300                        ASL     R0              ;       "
165232  050200                        BIS     R2,R0           ;PUT IT IN RESULT
165234  000760                        BR      2$


                                        .SBTTL  OD      - HALT FOR MICRO-ODT FUNCTIONS
                                        ;*************************************************
                                        ;*HALT FOR ODT OPERATIONS
                                        ;*HIT "P" TO RETURN TO KEYBOARD DISPATCH ROUTINE
                                        ;*PROVIDED PC("R7") IS UNALTERED
                                        ;*************************************************

165236  000000                OD:     HALT                    ;LET THE MICRO-CODE TAKE OVER
165240  000662                        BR      KBDS            ;RETURN TO KEYBOARD DISPLAY ROUTINE
```

SH#17

```
                                        ;*************************************************
                                        ;*STARTING LOCATION FOR DOING BOOT WITHOUT DIAGNOSTICS
                                        ;*  BEFORE STARTING, LOAD:
                                        ;*      R4 = ADDRESS OF BOOT ROUTINE
                                        ;*      R5 = HIGHEST AVAILABLE MEMORY LOCATION
                                        ;*              (R5-ONLY REQUIRED FOR ABSOLUTE LOADER)
                                        ;*BOOT DEFAULTS TO UNIT ZERO AT THIS ENTRY POINT
                                        ;*************************************************

165242  005000              BOOT:   CLR     R0              ;DEFAULT TO UNIT ZERO

                                        ;*************************************************
                                        ;*NORMAL BOOT ENTRY AFTER PERFORMING DIAGNOSTICS
                                        ;*************************************************

165244  012703  000004      BTSTP:  MOV     #4,R3           ;SET TIMEOUT VECTOR TO RETURN TO
165250  012713  165006              MOV     #KBDS,(R3)      ;  KEYBOARD DISPATCH ROUTINE
165254  012401                      MOV     (R4)+,R1        ;COPY COMMAND STATUS REG &
                                                            ;POINT R4 TO BEGINNING OF BOOT ROUTINE
165256  010506                      MOV     R5,SP           ;SET TO TOP OF MEMORY
                                                            ;  [THE MEMORY TEST LEAVES R5 POINTING
                                                            ;    TO THE TOP OF MEMORY]
165260  000005                      RESET
165262  000114                      JMP     (R4)            ;PERFORM BOOT
```


SH#18

```
                                        .SBTTL  RXV11   - FLOPPY DISK BOOTSTRAP
                                        ;*************************************************
                                        ;*THIS ROUTINE PERFORMS A BOOT FROM THE RX01 FLOPPY DRIVE
                                        ;*UPON ENTERING THIS ROUTINE :
                                        ;*      R0 = DEVICE NUMBER
                                        ;*      R1 = COMMAND/STATUS REGISTER ADDRESS
                                        ;*
                                        ;*ENTRY IS MADE AT (RX11 + 2)
                                        ;*THE CONTENTS OF R2,R3,R4 ARE LOST
                                        ;*************************************************

                                        .ENABLE LSB
165264  177170              RX11:   .WORD   RXCS            ;RX11 CONTROL/STATUS REGISTER
165266  012704                      MOV     (PC)+,R4        ;COPY COMMAND/STATUS BITS
165270      267                      .BYTE  200!40!20!6!1    ;TR, DONE, UNIT 1, READ, GO
165271      247                      .BYTE  200!40!00!6!1    ;TR, DONE, UNIT 0, READ, GO
165272  131700                      BITB    (PC),R0         ;WHICH UNIT?
                                                            ;LOW BYTE OF NEXT MUST BE 00 000 001
165274  001001                      BNE     12$             ;BR IF UNIT 1, R4 IS OK
165276  000304                      SWAB    R4              ;SELECT UNIT 0, MOVE HIGH BYTE TO LOW
165300  005002              12$:    CLR     R2              ;CLEAR TIMING REGISTER
165302  105711              121$:   TSTB    (R1)            ;'TR' OR 'DONE' SET?
165304  001005                      BNE     13$             ;BR IF DONE
165306  005202                      INC     R2              ;INCREMENT TIMING COUNTER
165310  005702                      TST     R2              ;THIS IS TO INCREASE THE TIME OF LOOP
165312  001373                      BNE     121$            ;BR IF TIME HAS NOT EXPIRED
165314  000000                      HALT                    ;HALT, "DONE" NOT SET --> TIMEOUT
165316  000552                      BR      KBJMP2          ;RETURN TO KEYBOARD DISPATCH ROUTINE("$")
165320  012703  000003      13$:    MOV     #3,R3           ;SET READ SEQUENCING BITS
165324  000261                      SEC                     ;SET FOR READ SEQUENCE
165326  110411              14$:    MOVB    R4,(R1)         ;COMMAND: "READ" THEN "EMPTY BUFFER"
165330  141704                      BICB    (PC),R4         ;CHANGE "READ(6)" TO "EMPTY BUFFER(2)"
                                                            ;LOW BYTE OF NEXT MUST BE 0X 000 100
165332  031104              16$:    BIT     (R1),R4         ;'ERR', 'TR', OR 'DONE' SET?
165334  001776                      BEQ     16$
165336  100414                      BMI     20$             ;BR IF BOOT ERROR
165340  103405                      BCS     18$             ;BR IF READ SEQUENCE
165342  131104                      BITB    (R1),R4         ;'TR' OR 'DONE' SET?
165344  100011                      BPL     20$             ;DONE, GO TEST FOR 240 AT 0
165346  116123  000002              MOVB    2(R1),(R3)+     ;STORE DATA
165352  000767                      BR      16$             ;GO WAIT FOR 'TR' OR 'DONE'
165354  006203              18$:    ASR     R3              ;READ COMMAND SEQUENCE:
                                                            ; 1ST - R3=1, CARRY =1
                                                            ; 2ND - R3=0, CARRY=1
                                                            ; 3RD - R3=0, CARRY=0
165356  103363                      BCC     14$             ;3RD TIME LOAD EMPTY BUFFER COMMAND
165360  112761  000001  000002      MOVB    #1,2(R1)        ; 1ST TIME LOAD SECTOR # 1
                                                            ; 2ND TIME LOAD TRACK # 1
165366  000761                      BR      16$             ;CONTINUE READ SEQUENCE
165370  005721              20$:    TST     (R1)+           ;ANY ERRORS?
165372  100522                      BMI     BTERR           ;BR IF ERROR
165374  122737  000240  000000      CMPB    #240,@#0        ;240 MUST BE AT ZERO
165402  001116                      BNE     BTERR           ;BR IF ERROR
165404  005007                      CLR     PC              ;START SECONDARY BOOT
```

SH#19

```
                                        .DSABLE LSB
```


SH#20

```
                                        .SBTTL  ABSLDR  - ABSOLUTE BINARY LOADER -- V007A
                                        ;*****************************************************
                                        ;*
                                        ;*   INPUT FORMAT --
                                        ;*      FRAME    -1   001
                                        ;*               -2   000
                                        ;*               -3   BYTE COUNT - LOWER ORDER
                                        ;*               -4   BYTE COUNT - HIGHER ORDER
                                        ;*               -5   LOAD ADDRESS - LOWER ORDER
                                        ;*               -6   LOAD ADDRESS - HIGHER ORDER
                                        ;*                         ,       DATA
                                        ;*                         ,          PLACED
                                        ;*                         ,             HERE
                                        ;*              CKSM - LAST FRAME CONTAINS THE CHECKSUM
                                        ;*
                                        ;* IF THE BYTE COUNT IS EQUAL TO 6, THE LOAD ADDRESS
                                        ;*  SPECIFIED WILL BE CONSIDERED TO BE THE DESIRED JUMP
                                        ;*   ADDRESS, IF THIS ADDRESS IS ODD, THE LOADER WILL HALT,
                                        ;*
                                        ;*
                                        ;*  IF THE BYTE COUNT IS > 6, DATA WILL BE LOADED INTO MEMORY,
                                        ;*
                                        ;*  PROGRAMMING CONSIDERATIONS AND CAUTIONS = THE TOP THREE WORDS
                                        ;*  OF MEMORY ARE USED FOR THE LOADER SP STACK
                                        ;*
                                        ;*ALL GENERAL REGISTERS ARE USED
                                        ;*****************************************************

165406  177560          AR:     .WORD   TKS             ;DL11 COMMAND STATUS REGISTER
165410  000000                  HALT                    ;ENTER SWR SETTINGS INTO R4 USING MICRO-ODT
                                                        ;   THEN TYPE "P" TO PROCEED
165412  000402                  BR      DEVNUM          ;BR TO CHECK FOR OPTIONAL CSR
165414  177560          AL:     .WORD   TKS             ;DL11 COMMAND STATUS REGISTER
165416  005004                  CLR     R4              ;PSEUDO SWITCH REGISTER CLEARED
165420  005700          DEVNUM: TST     R0              ;CHECK FOR OPTIONAL CSR
165422  001401                  BEQ     ABSL            ;IF NONE, GO TO ABSOLUTE LOADER WITH R1=CSR=177560
165424  010001                  MOV     R0,R1           ;LOAD OPTIONAL CSR INTO R1

                                        ;*MEMORY SIZE = THE "BTSTP" ROUTINE HAS ALREADY  POINTED SP TO
                                        ;*THE HIGHEST WRITABLE MEMORY ADDRESS

165426  010116          ABSL:   MOV     R1,(SP)         ;-> READER CSR

165430  011646          L.LOAD: MOV     (SP),-(SP)      ;MAKE 2 COPIES FOR L.READ
```

SH#21

```
768  165432  012705  165542          L.LD1:  MOV     #L.READ,L.PTR   ;=> READ ROUTINE
769  165436  005001                  CLR     L.ADR           ;CLEAR THE ROAD ADDRESS
770  165440  010446                  L.LD1B: MOV     R4,-(SP)                ;PICK UP THE CONTENT OF
771                                                          ;THE SOFTWARE SWITCH REGISTER
772  165442  006016                  ROR     @SP             ;CHECK RELOCATION FACTOR
773  165444  103402                  BCS     L.LD1C          ;JUMP IF SOME RELOCATION NEEDED
774  165446  005016                  CLR     @SP             ;USE ADDRESS SPECIFIED ON THE TAPE
775  165450  000403                  BR      L.LD2           ;GO DO LOAD
776  165452  006316                  L.LD1C: ASL     @SP             ;CHECK FOR NON-ZERO
777  165454  001001                  BNE     L.LD2           ;JUMP IF LOAD ADDRESS SPECIFIED
778  165456  010116                  MOV     L.ADR,@SP       ;OTHERWISE CONTINUE LOADING FROM LAST LOAD
779
780                                  ;* LOOK FOR THE BEGINNING OF A BLOCK
781
782  165460  005000                  L.LD2:  CLR     L.CKSM          ;INITIALIZE CHECKSUM
783  165462  004715                  JSR     PC,@L.PTR       ;READ A FRAME
784  165464  105303                  DECB    L.BYT           ;CHECK FOR +1 (START OF A BLOCK)
785  165466  001374                  BNE     L.LD2           ;LOOP UNTIL +1 IS FOUND
786  165470  004715                  JSR     PC,@L.PTR       ;READ ANOTHER FRAME
787
788                                  ;*
789                                  ;* INPUT AND SAVE BYTE COUNT, IF BYTE COUNT IS EQUAL TO 6
790                                  ;*  GO TO PROCEED JUMP
791                                  ;*
792
793  165472  004767  000074          JSR     PC,L.GWRD       ;GET FULL BYTE COUNT
794  165476  010402                  MOV     R4,L.BC         ;
795  165500  062702  177774          ADD     #-4,L.BC        ;SUBTRACT 4 TO MAKE BYTE COUNT CORRECT
796  165504  022702  000002          CMP     #2,L.BC         ;WAS BYTE COUNT EQUAL TO 6?
797  165510  001436                  BEQ     L.JMP           ;JUMP IF NO DATA (E.G. - JUMP BLOCK)
798  165512  004767  000054          JSR     PC,L.GWRD       ;GET LOAD ADDRESS
799  165516  061604                  ADD     @SP,R4          ;GENERATE ACTUAL ADDRESS
800  165520  010401                  MOV     R4,L.ADR                ;AND PUT IT INTO THE PROPER CELL
801
802                                  ;*
803                                  ;* READ IN REMAINDER OF DATA
804                                  ;*  IF THE LOADER HALTS AT L.BAD, A CHECKSUM ERROR
805                                  ;*  HAS OCCURED,  R3 WILL CONTAIN THE EXPECTED CHECKSUM,
806                                  ;*  AND R0 WILL CONTAIN THE DEVIATION FROM THE EXPECTED
807                                  ;*  CHECKSUM,
808                                  ;*
809  165522  004715          L.LD3:  JSR     PC,@L.PTR       ;READ A FRAME
810  165524  002004                  BGE     L.LD4           ;BRANCH IF MORE DATA REMAINS
811  165526  105700                  TSTB    L.CKSM          ;IF CHECKSUM IS
812  165530  001753                  BEQ     L.LD2           ;CORRECT, THEN CONTINUE
813  165532  000000          L.BAD:  HALT                    ;CHECKSUM ERROR
814  165534  000751                  BR      L.LD2           ;PRESS CONTINUE TO IGNORE CHECKSUM
815  165536  110321          L.LD4:  MOVB    L.BYT,(L.ADR)+  ;STORE 8 BITS AT A TIME
816  165540  000770                  BR      L.LD3           ;     THE RE-LOOP
```


SH#22

```
818
819                                  ;*
820                                  ;* INPUT A FRAME, DECREMENT BYTE COUNT, AND ACCUMULATE CHECKSUM
821                                  ;*
822  165542  016603  000006  L.READ: MOV     6(SP),L.BYT     ;DEVICE ADDRESS TO L.BYT
823  165546  105213                  INCB    @L.BYT          ;SELECT READER
824  165550  105713          L.R1:   TSTB    @L.BYT          ;DONE?
825  165552  100376                  BPL     L.R1            ;NO
826  165554  116303  000002          MOVB    2(L.BYT),L.BYT  ;GET CHARACTER
827  165560  042703  177400          BIC     #177400,L.BYT   ;CLEAR GARBAGE BITS
828  165564  060300                  ADD     L.BYT,L.CKSM    ;ADD TO CHECKSUM
829  165566  005302                  DEC     L.BC            ;DECREMENT BYTE COUNT BY ONE
830  165570  000207                  RTS     PC
831
832                                  ;*
833                                  ;* ASSEMBLE ONE FULL WORD OF DATA
834                                  ;*
835  165572  004715          L.GWRD: JSR     PC,@L.PTR       ;GET ONE CHARACTER
836  165574  010304                  MOV     L.BYT,R4        ;SAVE R3 IN TEMPORARY
837  165576  004715                  JSR     PC,@L.PTR       ;GET ANOTHER FRAME
838  165600  000303                  SWAB    L.BYT           ;PLACE ANOTHER FRAME
839  165602  050304                  BIS     L.BYT,R4        ;ASSEMBLE BOTH FRAMES INTO A COMPLETE WORD
840  165604  000207                  RTS     PC              ;RETURN
841
842                                  ;*
843                                  ;* CHECK CORRECTNESS OF JUMP ADDRESS
844                                  ;* HALT IF ADDRESS IS ODD, JUMP TO PROGRAM IF ADDRESS IS EVEN
845                                  ;*
846  165606  004767  177760  L.JMP:  JSR     PC,L.GWRD       ;GET POSSIBLE TRANSFER ADDRESS
847  165612  004715                  JSR     PC,@L.PTR       ;GET CHECKSUM
848  165614  105700                  TSTB    L.CKSM          ;IF INCORRECT
849  165616  001345                  BNE     L.BAD           ; GO TO CHECKSUM HALT ADDRESS
850  165620  006204                  ASR     R4              ;GET LOW ORDER BIT
851  165622  103002                  BCC     L.JMP1          ;SKIP IF ADDRESS IS EVEN
852  165624  000000                  HALT                    ;OTHERWISE HALT
853  165626  000704                  BR      L.LD1B          ;RETURN TO START OF LOADING LOOP
854  165630  006304          L.JMP1: ASL     R4              ;RESTORE REGISTER
855  165632  061604                  ADD     @SP,R4
856  165634  000114                  JMP     @R4             ;JUMP TO USER
857
858
859                                  ;*****************************************************
860                                  ;*BOOT ERROR:
861                                  ;*A HALT HERE INDICATES A BOOT ERROR HAS OCCURRED
862                                  ;*
863                                  ;*HIT "P" TO RETURN TO THE KEYBOARD ROUTINE,
864                                  ;*  PROVIDED THE PC IS UNALTERED
865                                  ;*****************************************************
866
867  165636  005741          BTERR1: TST     -(R1)           ;POINT R1 AT ERROR REGISTER FOR RK11
868  165640  011102          BTERR:  MOV     (R1),R2         ;MOVE CONTENTS OF ERROR REG TO R2
869  165642  000000                  HALT
870  165644  000167  177136  KBJMP2: JMP     KBDS            ;RETURN TO KEYBOARD DISPLAY ROUTINE
871
```

SH#23

```
                                        .ENABLE LSB

                                .SBTTL  RK11    - DISK DRIVE BOOTSTRAP
                                ;*********************************************
                                ;*THIS ROUTINE PERFORMS A BOOT FROM THE RK05 DISK DRIVE
                                ;*UPON ENTERING THE ROUTINE:
                                ;*      R0 = DEVICE NUMBER
                                ;*      R1 = COMMAND/STATUS REGISTER ADDRESS
                                ;*
                                ;*ENTRY IS MADE AT (RK11 + 2)
                                ;*THE CONTENTS OF R3 IS LOST
                                ;*********************************************

165650  177404          RK11:   .WORD   RKCS            ;=> CONTROL/STATUS REG.
165652  052700  000010          BIS     #10,R0          ;SHIFT UNIT # TO BITS 15 - 13
165656  006300          2$:     ASL     R0              ;       "
165660  103376                  BCC     2$              ;       "
165662  010061  000006          MOV     R0,6(R1)        ;MOVE IN TO RKDA REG.
165666  005000                  CLR     R0              ;CLEAR UNIT #
165670  062700  000005  DSKRD:  ADD     #5,R0           ;LOAD DISK READ COMM
165674  000400                  BR      OTHER           ;UNNECESSARY INSTRUCTION
```


SH#24

```
165676  012761  177000  000002  OTHER:  MOV     #-256.*2,2(R1)  ;LOAD WORD COUNT OR BYTE COUNT
165704  010011                  MOV     R0,(R1)         ;LOAD READ FUNCTION & GO
165706  005003                  CLR     R3              ;CLEAR REG FOR TIMING INTERFACE TIMEOUT
165710  105711          TDONE:  TSTB    (R1)            ;TEST FOR DONE
165712  100405                  BMI     DONE            ;BR IF DONE
165714  005203                  INC     R3              ;INCREMENT COUNTER
165716  005703                  TST     R3              ;THIS IS TO INCREASE TIME OF LOOP
165720  001373                  BNE     TDONE           ;TEST FOR TIMEOUT
165722  000000                  HALT                    ;HALT, "DONE" NOT SET --> TIMEOUT
165724  000747                  BR      KBJMP2          ;RETURN TO KEYBOARD DISPATCH ROUTINE("$")
165726  005711          DONE:   TST     (R1)            ;ERRORS ?
165730  100742                  BMI     BTERR1          ;BR IF ERROR
165732  005007                  CLR     PC              ;START SECONDARY BOOT

                                .DSABLE LSB
```

SH#25

```
                                        ;*************************************************
                                        ;*OUTPUT PROMPT STRINGS
                                        ;*************************************************

165734  005077              MSGQ:   .ASCII  <'?><LF>        ;ERRONEOUS COMMAND INDICATOR
165736  005015  000044      MSG:    .ASCII  <CR><LF><'S><0> ;READY PROMPT OUTPUT


                                        ;*************************************************
                                        ;*COMMAND DEFINITION TABLE
                                        ;*************************************************

165742  173626              CTABLE: MEM
165744  046530                      .ASCII  /XM/            ;COMMAND TO DO JUST MEMORY TEST
165746  165406                      AR
165750  051101                      .ASCII  /AR/            ;COMMAND TO DO CPU & MEMORY TESTS AND
                                                            ;THE ABSOLUTE LOADER WITH A RELOCATABLE PROGRAM
165752  165414                      AL
165754  046101                      .ASCII  /AL/            ;SAME COMMAND AS "AR",EXCEPT NO RELOCATION DESIRED
165756  165264                      RX11
165760  054104                      .ASCII  /DX/            ;SAME AS "AR",EXCEPT THE FLOPPY BOOT IS
                                                            ;EXECUTED IN PLACE OF THE ABSOLUTE LOADER
165762  165650                      RK11
165764  045504                      .ASCII  /DK/            ;SAME AS "AR", EXCEPT THE DISK DRIVE BOOT
                                                            ;IS EXECUTED IN PLACE OF THE ABSOLUTE LOADER
165766  173302                      TEST7
165770  041530                      .ASCII  /XC/            ;COMMAND TO DO JUST THE CPU TEST
165772  165236                      OD
165774  042117                      .ASCII  /OD/            ;COMMAND TO HALT FOR ODT OPERATION
                                                            ;WARNING -
                                                            ;THE "OD" COMMAND MUST BE THE LAST COMMAND IN THE LIST
165776  000000                      0                       ;INDICATION OF END OF TABLE

        000001                      .END
```


SH#26

```
ABSL    165426          ADCONT  173732          ADORT   173714          AL      165414
AR      165406          BIT10 = 002000          BIT15 = 100000          BIT5,1= 020040
BOOT    165242          BTCOMM  165100          BTERR   165640          BTERR1  165636
BTSTP   165244          CHECK   173722          CR    = 000015          CTABLE  165742
DCONT   173756          DEVNUM  165420          DOIT    165076          DONE    165726
DSKRD   165670          GETCHR  165104          GETNUM  165174          GONOGO= 173000
JMP1    173152          JMP3    173160          JSR7    173572          JSR7P   173606
KBDQ    165000          KBDS    165006          KBJMP   173566          KBJMP2  165644
LF    = 000012          L.ADR =%000001          L.BAD   165532          L.BC  =%000002
L.BYT =%000003          L.CKSM=%000000          L.GWRD  165572          L.JMP   165606
L.JMP1  165630          L.LD1   165432          L.LD1B  165440          L.LD1C  165452
L.LD2   165460          L.LD3   165522          L.LD4   165536          L.LOAD  165430
L.PTR =%000005          L.READ  165542          L.R1    165550          MEM     173626
MEMDO   173712          MEMERA  173730          MEMERD  173754          MEMSIZ  173642
MEMT    173736          MSG     165736          MSGQ    165734          OD      165236
OTHER   165676          PC    =%000007          PCCS  = 177550          PRINT   165012
PUTCHR  165120          READ    165026          RKCS  = 177404          RK11    165650
RXCS  = 177170          RX11    165264          R0    =%000000          R1    =%000001
R2    =%000002          R3    =%000003          R4    =%000004          R5    =%000005
SEARCH  165042          SP    =%000006          TCONT   165160          TDONE   165710
TEST1   173000          TEST2   173044          TEST3   173104          TEST4   173142
TEST5   173166          TEST6   173236          TEST7   173302          TEST8   173444
TEST9   173532          TJSR    173620          TJSRP   173624          TKB   = 177562
TKS   = 177560          TPB   = 177566          TPS   = 177564          TRAPS   165150
T3DATA  173264          T3DAT4  173270          T5DATA  173274          T6DATA  173300
T7DATA  173436          T7ERR   173434          T8ERR   173530          T9ERR   173556
WALK    173746          X1RTN   165144          $BOOTS= 165000          .     = 166000

 ERRORS DETECTED:  0

 *M9400,M9400=RM9400/DS:ERFZ
 RUN-TIME:  3  6  0 SECONDS
 CORE USED:  4K
```

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS

PARTS LIST

MADE BY D. HEALY	CHECKED D. HEALY	SECTION
DATE 26 FEB 76	DATE 26 FEB 76	1
ENG Tom Witowski	PROD D. DeCarlo	ISSUED SECT.
DATE 3/11/76	DATE 3-15-76	1

ITEM NO.	DWG NO./PART NO.	DESCRIPTION	QUANTITY/VARIATION REV11-A	REV11-C	REV11-J
1	MPØØØ73	REV11 VOLUME NO. 1 PRINT SET	1	1	1
2	DEC-11-DZM9A-D	REV11 DIAGNOSTIC LISTING	*	*	*
3	DEC-11-DZM9A-PB	REV11 DIAGNOSTIC PAPER TAPE	*	*	*

*NOTE: THESE ITEMS ARE ONLY TO BE SHIPPED WITH FIRST SHIP TO CUSTOMER.

TITLE	ASSY NO.	SIZE	CODE	NUMBER	REV. ECO NO.
REV11 SHIPPING LIST	NONE	A	PL	REV11-Ø-2	
	SHEET 1 OF 1	DIST			

DEC FORM DRA 110 DEC 16-(325)-1031-N870