

IDENTIFICATION

Product code: ENKAZ VERSION 1.3
Product name: CRD HARDCORE INSTRUCTION and SIZING TESTS
Product date: 23-MAY-1983
Maintainer: BASE SYSTEMS DIAGNOSTIC ENGINEERING

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital or its affiliated companies.

Copyright (c) 1982 by Digital Equipment Corporation. All Rights Reserved.

The following are trademarks of Digital Equipment Corporation.

DEC
DECUS
UNIBUS

DECsystem-10
MASSBUS
VAX

DECSYSTEM-20
PDP
VMS

digital

Table of Contents

1.0	ABSTRACT	3
2.0	MAINTENANCE HISTORY	3

1 PROGRAM ABSTRACT

This Level 4 diagnostic is run during VAX-11/730/725 Customer Runnable Diagnostics AUTO and MENU mode. Its primary functions are:

- 1) Providing a small subset of the VAX Macro-Hardware Instruction Set prior to booting the Diagnostic Supervisor.
- 2) Sizing base VAX-11/730/725 disk system load devices, (via IDC, UDA50 or LESI controllers), using an algorithm which is defined in ENSAB.DOC for AUTO and MENU CRD. Sizing is done to pass the correct parameters through VMB.EXE to boot the Diagnostic Supervisor on the correct load device.
- 3) Verification of LESI and UDA50 CRD supported load disk options by testing if the VAX-11/730/725 CRD supported load device (RA80, RA81, RA60, or RC25) is spinning and reading "CRDPACK" label for the RA60 and RC25.

This diagnostic was designed to be run on the VAX-11/730/725 specifically and it is not intended to be APT compatible. Refer to ENSAB.DOC the CRD AUTO and MENU mode documentation for further information on this diagnostic and complete documentation on VAX-11/730/725 CRD AUTO test.

2 MAINTENANCE HISTORY

DATE	VERSION	DESCRIPTION
23-MAY-1983	1.0	Initial release.
25-JUL-1983	1.1	Added LCN VAX-11/725 syntax.
16-JAN-1984	1.2	Added code for detection of IDC and detection of drive 1 not present in which we boot from drive 0 European system configuration.
02-FEB-1984	1.2	Added code for printing on the VR100 terminal in simulated VT52 mode. LCN RC25 system configuration.
14-Jun-1984	1.3	Code for booting from drive 0 when no RL02 is present was fixed. The 730/725 processor register definitions were added. The RC25 based systems can now go straight into menu mode.

22-ENKAZ-1.3 Map
ENKAZ

E 1

14-JUN-1984 16:08

Fiche 1 Frame E1
VAX-11 Linker V3A-18

Sequence 4 Page 1

+-----+
! Object Module Synopsis !
+-----+

Module Name	Ident	Bytes	File	Creation Date	Creator
ENKAZ01	01-30	5080	[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.OBJ;5	14-JUN-1984 16:01	VAX-11 Macro V03-01
ENKAZ02	01-30	2265	[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.OBJ;12	14-JUN-1984 16:04	VAX-11 Macro V03-01

ZZ-ENKAZ-1.3 Map
DRBO:[BALL.ENKAZ.ENKAZ_1]ENKAZ.EXE;13

F 1

14-JUN-1984 16:08

Fiche 1 Frame F1
VAX-11 Linker V3A-18

Sequence 5
Page

2

-----+
! Image Section Synopsis !
-----+

<u>Cluster</u>	<u>Type</u>	<u>Pages</u>	<u>Base Addr</u>	<u>Disk VBN</u>	<u>PFC</u>	<u>Protection and Paging</u>	<u>Global Sec. Name</u>	<u>Match</u>	<u>Majorid</u>	<u>Minorid</u>
DEFAULT_CLUSTER	0	15	00000000		1	0 READ ONLY				

Key for special characters above:

-----+
! R - Relocatable !
! P - Protected !
-----+

+-----+
 ! Program Section Synopsis !
 +-----+

Psect Name	Module Name	Base	End	Length	Align	Attributes
A_CONIO	ENKAZ01	00000000 00000000	0000125F 0000125F	00001260 () 00001260 ()	4704.) PAGE 9 4704.) PAGE 9	PIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
A_HARDCORE	ENKAZ01	00001260 00001260	000013D7 000013D7	00000178 () 00000178 ()	376.) BYTE 0 376.) BYTE 0	PIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
SIZING	ENKAZ02	00001400 00001400	00001CD8 00001CD8	000008D9 () 000008D9 ()	2265.) PAGE 9 2265.) PAGE 9	NOPI,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC

-----+
 ! Symbol Cross Reference !
 -----+

Symbol	Value	Defined By	Referenced By ...
BAD_DISK	00001688-R	ENKAZ02	ENKAZ01
BOOT\$READPROMPT	00000E00-R	ENKAZ01	
C_D_RDY	00000081	ENKAZ02	
DRIVE0	000017B5-R	ENKAZ02	
DSSGB_UDASO_INIT	00001480-R	ENKAZ02	ENKAZ01
DS1	00000100	ENKAZ02	
ERROR_ROUTINE	00000F29-R	ENKAZ01	ENKAZ02
HMBLK_ERR	00001687-R	ENKAZ02	ENKAZ01
IDC_CSR	00F26200	ENKAZ01	ENKAZ02
IDC_HLD	0000168A-R	ENKAZ02	
IDC_PARAM_0	00001784-R	ENKAZ02	
IDC_PARAM_1	0000175C-R	ENKAZ02	
IDC_TST	0000168E-R	ENKAZ02	
INIT_DRVO	00001847-R	ENKAZ02	ENKAZ01
INTERRUPT_STACK	00000A00-R	ENKAZ01	
KERNEL_STACK	00000800-R	ENKAZ01	
LES1_UDA_PARAM	00001736-R	ENKAZ02	ENKAZ01
MCHK_IDC	000011F4-R	ENKAZ01	ENKAZ02
MCHK_LES_UDA	00001208-R	ENKAZ01	ENKAZ02
NOCNTRL_FND	00000207-R	ENKAZ01	ENKAZ02
NODRIVE_O_1	00001686-R	ENKAZ02	ENKAZ01
NO_LOAD	00000F6A-R	ENKAZ01	
RA80_TEST	00000FE0-R	ENKAZ01	ENKAZ02
RA8081_TEST	0000109E-R	ENKAZ01	ENKAZ02
RA81 PRES	00001684-R	ENKAZ02	ENKAZ01
RC25 PRES	00001685-R	ENKAZ02	
RC25 TEST	00001121-R	ENKAZ01	ENKAZ02
RPB_BUFF	00000C00-R	ENKAZ01	ENKAZ02
SCB	00000600-R	ENKAZ01	
SCB0	00000600-R	ENKAZ01	
SCB20	00000620-R	ENKAZ01	
SCB24	00000624-R	ENKAZ01	
SCB4	00000604-R	ENKAZ01	ENKAZ02
SCB60	00000660-R	ENKAZ01	
SCB64	00000664-R	ENKAZ01	
SCB8	00000608-R	ENKAZ01	
SPIN_DWN	00001689-R	ENKAZ02	ENKAZ01
TEST_6	000016E3-R	ENKAZ02	ENKAZ01
TEST_7	000017AC-R	ENKAZ02	
TEST_8	00001C00-R	ENKAZ02	ENKAZ01
TOP_OF_ISTACK	00000BFC-R	ENKAZ01	
TOP_OF_KSTACK	000009FC-R	ENKAZ01	
UDA_LES1	0000171F-R	ENKAZ02	ENKAZ01
UDA_LES1_CSR	40FFF468	ENKAZ01	ENKAZ02
UDA_RESPONSE	00001650-R	ENKAZ02	
UD_DRIVER1	00001692-R	ENKAZ02	ENKAZ01
UD_INIT	00001400-R	ENKAZ02	
UNEX_E_1	000011CB-R	ENKAZ01	ENKAZ02
WHAT_DRIVE	00001680-R	ENKAZ02	ENKAZ01

ZZ-ENKAZ-1.3 Map
DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ.EXE;13

I 1
14-JUN-1984 16:08

Fiche 1 Frame 11
VAX-11 Linker V3A-18

Sequence 8 Page 5

<u>Symbol</u>	<u>Value</u>	<u>Defined By</u>	<u>Referenced By ...</u>
WHAT_MODE	00000203-R	ENKAZ01	ENKAZ02

 ! Symbols By Value !

Value	Symbols...
00000081	C_D_RDY
00000100	DS1
00000203	R-WHAT_MODE
00000207	R-NOCNTRLR_FND
00000600	R-SCB R-SCB0
00000604	R-SCB4
00000608	R-SCB8
00000620	R-SCB20
00000624	R-SCB24
00000660	R-SCB60
00000664	R-SCB64
00000800	R-KERNEL_STACK
000009FC	R-TOP_OF_KSTACK
00000AC0	R-INTERROPT_STACK
00000BFC	R-TOP_OF_ISTACK
00000C00	R-RPB_BUFF
00000E00	R-BOOT\$READPROMPT
00000F29	R-ERROR_ROUTINE
00000F6A	R-NO_LOAD
00000FE0	R-RA80_TEST
0000109E	R-RA80B1_TEST
00001121	R-RC25_TEST
000011C8	R-UNEX_E_I
000011F4	R-MCHK_IDC
00001208	R-MCHK_LES_UDA
00001400	R-UD_INIT
000014B0	R-DS\$GB_UDA50_INIT
00001650	R-UDA_RESPONSE
00001680	R-WHAT_DRIVE
00001684	R-RA81-PRES
00001685	R-RC25-PRES
00001686	R-NODRIVE_0_1
00001687	R-HMBLK_ERR
00001688	R-BAD_DISK
00001689	R-SPIN_DWN
0000168A	R-IDC_HLD
0000168E	R-IDC-TST
00001692	R-UD_DRIVER1
000016E3	R-TEST_6
0000171F	R-UDA_LESI
00001736	R-LEST_UDA_PARAM
0000175C	R-IDC_PARAM_1
00001784	R-IDC_PARAM_0
000017AC	R-TEST_7
000017B5	R-DRIVE0
00001877	R-INIT_DRVO
00001C00	R-TEST_8
00F26200	IDC_CSR
40FFF468	UDA_LESI_CSR

Value

Symbols...

Key for special characters above:

*	- Undefined
U	- Universal
R	- Relocatable
X	- External

+-----+
! Image Synopsis !
+-----+

virtual memory allocated: 00000000 00001DFF 00001E00 (7680. bytes, 15. pages)
Stack size: 0. pages
Image binary virtual block limits: 1. 15. (15. blocks)
Image name and identification: ENKAZ 01-30
Number of files: 2.
Number of modules: 2.
Number of program sections: 5.
Number of global symbols: 50.
Number of cross references: 76.
Number of image sections: 1.
Image type: SYSTEM.
Map format: FULL WITH CROSS REFERENCE in file DRBO:[BALL.ENKAZ.ENKAZ_1]ENKAZ.MAP,13
Estimated map length: 26. blocks

+-----+
! Link Run Statistics !
+-----+

Performance Indicators	Page Faults	CPU Time	Elapsed Time
Command processing	102	00:00:00.29	00:00:01.87
Pass 1:	35	00:00:00.17	00:00:00.70
Allocation/Relocation:	10	00:00:00.14	00:00:01.17
Pass 2:	13	00:00:00.22	00:00:01.43
Map data after object module synopsis:	19	00:00:00.40	00:00:03.10
Symbol table output:	0	00:00:00.11	00:00:01.07
Total run values:	179	00:00:01.33	00:00:09.34

Using a working set limited to 500 pages and 33 pages of data storage (excluding image)

Total number object records read (both passes): 90
of which 0 were in libraries and 4 were DEBUG data records containing 145 bytes

Number of modules extracted explicitly = 0
with 0 extracted to resolve undefined symbols

0 library searches were for symbols not in the library searched

A total of 5 global symbol table records was written

LINK/SYMBOL=ENKAZ/NODEBUG/MAP=ENKAZ/FULL/CROSS/SYSTEM=%X0/EXE=ENKAZ ENKAZ01A,ENKAZ02A

Table of contents

(4)	71	DECLARATIONS
(4)	335	BOO\$READPROMPT - Prompt and read input string
(4)	435	PRCMT1 - Routine to prompt console for input
(4)	456	BOO\$TYPE ASCII - Type a counted ASCII string
(4)	497	PRINT ROUTINE
(4)	517	ERROR ROUTINE
(4)	602	RA60 PRINT ROUTINE
(4)	657	RA80 AND RA81 PRINT ROUTINE
(4)	700	RC25 PRINT ROUTINE
(4)	747	EXCEPTION HANDLER ROUTINES
(6)	803	INITIALIZATION ROUTINE
(7)	860	TEST 1: AOBLEQ, AOBLSSTest
(8)	908	TEST 2: SOBGEQ, SOBGTRTest
(9)	954	TEST 3: BBS, BBCTest
(10)	982	TEST 4: BBSS, BBCTest
(11)	1020	TEST 5: BLBS, BLBCTest
(12)	1051	Diagnostic Subroutines

ZZ-ENKAZ-1.3
ENKAZ01
01-30

CRD HARDWARE INSTRUCTION TESTS
CRD HARDWARE INSTRUCTION TESTS

N 1
14-JUN-1984

Fiche 1 Frame N1

Sequence 13

14-JUN-198 16:01:36 VAX-11 Macro V03-01 Page 1
14-JUN-198 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(1)

```
0000 1 .TITLE ENKAZ01 CRD HARDWARE INSTRUCTION TESTS
0000 2 .IDENT /01-30/
0000 3
0000 4
0000 5 : Copyright (c) 1982
0000 6 : Digital Equipment Corporation, Maynard, Massachusetts 01754
0000 7
0000 8 : This software is furnished under a license for use only on a single computer
0000 9 : system and may be copied only with the inclusion of the above copyright
0000 10 : notice. This software, or any other copies thereof, may not be provided or
0000 11 : otherwise made available to any other person except for use on such system
0000 12 : and to one who agrees to these license terms. Title to and ownership of the
0000 13 : software shall at all times remain in DEC.
0000 14
0000 15 : The information in this software is subject to change without notice and
0000 16 : should not be construed as a commitment by Digital Equipment Corporation.
0000 17
0000 18 : DEC assumes no responsibility for the use or reliability of its software on
0000 19 : equipment which is not supplied by DEC.
0000 20
0000 21
0000 22 : **
0000 23 : Facility:
0000 24
0000 25 : Vax Diagnostic Library
0000 26
0000 27 : Abstract:
0000 28
0000 29 : This diagnostic provides level 4 coverage while executing Customer
0000 30 : Runnable Diagnostics. Its primary functions are:
0000 31
0000 32 : 1) Providing a small subset of the VAX Macro-Hardware Instruction
0000 33 : Set prior to booting the Diagnostic Supervisor.
0000 34
0000 35 : 2) Sizing base VAX-11/730 disk system load devices via IDC, UDA50 or
0000 36 : LESI using an algorithm which is defined in ENSAB.DOC for AUTO and
0000 37 : MENU CRD. Sizing is done to pass the correct parameters through
0000 38 : VMB.EXE to boot the Diagnostic Supervisor on the correct load
0000 39 : device. This is done utilizing PUBTDIVER module.
0000 40
0000 41 : 3) Verification of KLESI and UDA50 CRD supported load disk options by
0000 42 : testing if the VAX-11/730 CRD supported load device (RA80, RA81,
0000 43 : RA60, or RC25) is spinning and reading "CRDPACK" label for the RA60
0000 44 : and RC25.
0000 45
0000 46 : Environment:
0000 47
0000 48 : Stand-alone
0000 49
0000 50 : Author:
0000 51
0000 52 : Peter Green 23-May-83 Version 1.0
0000 53
0000 54 : Peter Green 5-JUL-83 Version 1.1
0000 55 : Added LCN syntax.
0000 56
0000 57
```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

CRD HARDCORE INSTRUCTION TESTS
CRD HARDCORE INSTRUCTION TESTS

B 2
14-JUN-1984

Fiche 1 Frame B2

Sequence 14

14-JUN-1984 16:01:36
14-JUN-1984 15:53:29

VAX-11 Macro V03-01
DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(1)

Page 2

0000 58 ;--

ZZ-ENKAZ-1.3
ENKAZ01
01-30

CRD HARDWARE INSTRUCTION TESTS
CRD HARDWARE INSTRUCTION TESTS

C 2
14-JUN-1984

Fiche 1 Frame C2

Sequence 15

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 3
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(3)

	0000	60		
	0000	61		
	0000	62	:	
	0000	63	:	Revision
	0000	64	:	
00000001	0000	65	:	PRIM_REV
00000003	0000	66	:	SEC_REV
	0000	67	:	
	0000	68	:	
	0000	69	:	

= 1
= 3

ZZ-ENKAZ-1.3
ENKAZ01
01-30

DECLARATIONS

CRD HARDWARE INSTRUCTION TESTS
DECLARATIONS

D 2
14-JUN-1984

Fiche 1 Frame D2

Sequence 16

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 4
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1JENKAZ01A.(4)

```
0000 71 .SBTTL DECLARATIONS
0000 72 .PSECT A_CONIO,PAGE,PIC ;
0000 73
0000 74 .LIST MEB
0000 75
0000 76 :
0000 77 : Macro's
0000 78 :
0000 79
0000 80
0000 81 .MACRO PASS_MESSAGE
0000 82 .ASCIC / PASSED/<13><10><13><10>
0000 83 .ENDM PASS_MESSAGE
0000 84
0000 85 .MACRO FAIL_MESSAGE
0000 86 .ASCIC / *FAILED*/<13><10><13><10><13><10>
0000 87 .ENDM FAIL_MESSAGE
0000 88
0000 89 .MACRO ABORT_MESSAGE
0000 90 .ASCIC /TEST ABORTED - /
0000 91 .ENDM ABORT_MESSAGE
0000 92
0000 93 .MACRO CPU_MESSAGE
0000 94 .ASCIC /CPU BAD - /
0000 95 .ENDM CPU_MESSAGE
0000 96
0000 97 .MACRO FIELD_MESSAGE
0000 98 .ASCIC /CALL FIELD SERVICE/
0000 99 .ENDM FIELD_MESSAGE
0000 100
0000 101 .MACRO NOCNTLR_MESSAGE
0000 102 .ASCIC /NO SUPPORTED DISK CONTROLLER FOUND/<13><10>
0000 103 .ENDM NOCNTLR_MESSAGE
0000 104
0000 105 .MACRO NODRIVE01_MESSAGE
0000 106 .ASCIC /UNSUPPORTED DISK CONFIGURATION FOUND - DRIVE 0 AND 1/<13><10>
0000 107 .ENDM NODRIVE01_MESSAGE
0000 108
0000 109 .MACRO LOADMCK_MESSAGE
0000 110 .ASCIC /UNEXPECTED MACHINE CHECK WHILE VERIFYING LOAD PATH/<13><10>
0000 111 .ENDM LOADMCK_MESSAGE
0000 112
0000 113 .MACRO DISKERR_MESSAGE
0000 114 .ASCIC /DRIVE BAD - /
0000 115 .ENDM DISKERR_MESSAGE
0000 116
0000 117 .MACRO END_MESSAGE
0000 118 .ASCIC <13><10>?**** END OF VAX-11/730,725 ?
0000 119 .ENDM END_MESSAGE
0000 120
0000 121 .MACRO TEND_MESSAGE
0000 122 .ASCIC ?TEST ****?<13><10><13><10>
0000 123 .ENDM TEND_MESSAGE
0000 124
0000 125 .MACRO RET_MESSAGE
0000 126 .ASCIC ?RETURNING TO VAX-11/730,725 CONSOLE. WAIT FOR '>>>' PROMPT?-
0000 127 <13><10>
```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

DECLARATIONS

CRD HARDWARE INSTRUCTION TESTS
DECLARATIONS

E 2
14-JUN-1984

Fiche 1 Frame E2

Sequence 17

14-JUN-1984 16:01:36

VAX-11 Macro V03-01

Page 5

14-JUN-1984 15:53:29

DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(4)

```
0000 128 .ENDM RET_MESSAGE
0000 129
0000 130 .MACRO PROPER_MESSAGE
0000 131 .ASCIC <13><10><13><10>/** PROPER SETUP REQUIRED **/
0000 132 .ENDM PROPER_MESSAGE
0000 133
0000 134 .MACRO NSPN6025_MESSAGE
0000 135 .ASCIC <13><10>/** CHECK FOR DIAGNOSTIC PACK CRDPACK **/-
0000 136 <13><10>/** INSERTED AND SPINNING IN /
0000 137 .ENDM NSPN6025_MESSAGE
0000 138
0000 139 .MACRO DRVERR_MESSAGE
0000 140 .ASCIC /** IF SETUP IS CORRECT, SUSPECT DRIVE **/<13><10>-
0000 141 /** PROBLEM - /
0000 142 .ENDM DRVERR_MESSAGE
0000 143
0000 144 .MACRO SPNEND_MESSAGE
0000 145 .ASCIC / **/<13><10>
0000 146 .ENDM SPNEND_MESSAGE
0000 147
0000 148 .MACRO NSPN8081_MESSAGE
0000 149 .ASCIC <13><10>/** CHECK THAT DUA0: IS SPINNING **/<13><10>
0000 150 .ENDM NSPN8081_MESSAGE
0000 151
0000 152 .MACRO AUTO_MESSAGE
0000 153 .ASCIC /AUTO /
0000 154 .ENDM AUTO_MESSAGE
0000 155
0000 156 .MACRO MENU_MESSAGE
0000 157 .ASCIC /MENU /
0000 158 .ENDM MENU_MESSAGE
0000 159
0000 160 .MACRO TEST_MESSAGE
0000 161 .ASCIC /TESTING STARTED - RUN TIME = 0:30 MINUTES/
0000 162 .ENDM TEST_MESSAGE
0000 163
0000 164 .MACRO PART_MESSAGE
0000 165 .ASCIC /PART 1 /
0000 166 .ENDM PART_MESSAGE
0000 167
0000 168 .MACRO RC25_MESSAGE
0000 169 .ASCIC /RC25 /
0000 170 .ENDM RC25_MESSAGE
0000 171
0000 172 .MACRO RA60_MESSAGE
0000 173 .ASCIC /RA60 /
0000 174 .ENDM RA60_MESSAGE
0000 175
0000 176 .MACRO RA80_MESSAGE
0000 177 .ASCIC /RA80 /
0000 178 .ENDM RA80_MESSAGE
0000 179
0000 180 .MACRO RA81_MESSAGE
0000 181 .ASCIC /RA81 /
0000 182 .ENDM RA81_MESSAGE
0000 183
0000 184 .MACRO DAA0_MESSAGE
```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

DECLARATIONS

CRD HARDWARE INSTRUCTION TESTS
DECLARATIONS

F 2
14-JUN-1984

Fiche 1 Frame F2

Sequence 18

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 6
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(4)

```
0000 185 .ASCIC /DAA0 /
0000 186 .ENDM DAA0_MESSAGE
0000 187
0000 188 .MACRO DJA0_MESSAGE
0000 189 .ASCIC /DJA0 /
0000 190 .ENDM DJA0_MESSAGE
0000 191
0000 192 .MACRO DJA1_MESSAGE
0000 193 .ASCIC /DJA1 /
0000 194 .ENDM DJA1_MESSAGE
0000 195
0000 196 .MACRO DUA0_MESSAGE
0000 197 .ASCIC /DUA0 /
0000 198 .ENDM DUA0_MESSAGE
0000 199
0000 200 :
0000 201 : JMP to the starting address.
0000 202 :
0000 203 : = ^X200
101D 31 0200 204 BRW INIT1
0203 205
0203 206
0203 207 :
0203 208 : Equated symbols:
0203 209 :
0203 210
00000012 0203 211 IPL = 18
00000004 0203 212 MCKADD = 4
00000022 0203 213 TXCS = 34
00000023 0203 214 TXDB = 35
0000000D 0203 215 CR = 13
0000000A 0203 216 LF = 10
00F26200 0203 217 IDC_CSR = ^XF26200
40FFF468 0203 218 UDA_LESI_CSR = ^X40FFF468
00000084 0203 219 BOOTC_COMBUFSZ = 132
00000000 0203 220 WHAT_MODE:: .LONG 0
00000000 0207 221 NOCNTLR_FND:: .LONG 0
00000000 020B 222 LOAD_ERROR: .LONG 0
00000000 020F 223 CPU_BAD: .LONG 0
00000000 0213 224 PASS_COUNT: .LONG 0
00000000 0217 225 RA60_MESS_ST: .LONG 0
00000000 021B 226 RA80B1_MESS_ST: .LONG 0
00000000 021F 227 RC25_MESS_ST: .LONG 0
0223 228 PRINTPASS: PASS_MESSAGE
0D 44 45 53 53 41 50 20 20 20 20 0C' 0223 .ASCIC / "PASSED/<13><10><13><10>
0A 0D 0A 022F
0E C223
20 44 4C 45 49 46 20 4C 4C 41 43 00' 0232 229 PRINTFIELD: FIELD_MESSAGE
45 43 49 56 52 45 53 0232 .ASCIC /CALL FIELD SERVICE/
12 0232
2A 44 45 4C 47 41 46 2A 20 20 20 00' 0243 230 PRINTFAIL: FAIL_MESSAGE
0A 0D 0A 0D 0A 0D 0245 .ASCIC / "FAILED"/<13><10><13><10><13><10>
11 0251
45 54 52 4F 42 41 20 54 53 45 54 00' 0257 231 PRINTABORT: ABORT_MESSAGE
0257 .ASCIC /TEST ABORTED - /
```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

DECLARATIONS

CRD HARDWARE INSTRUCTION TESTS
DECLARATIONS

G 2
14-JUN-1984

Fiche 1 Frame G2

Sequence 19

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 7
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(4)

```
20 2D 20 44 0263
OF 0257
20 44 4E 45 20 2A 2A 2A 2A 0A 0D 00' 0267
33 37 2F 31 31 2D 58 41 56 20 46 4F 0267
20 35 32 37 2C 30 0273
1D 027F
0A 0D 2A 2A 2A 2A 20 54 53 45 54 00' 0267
OA 0D 0285
OD 0285
0291
0285
54 20 47 4E 49 4E 52 55 54 45 52 00' 0293
30 33 37 2F 31 31 2D 58 41 56 20 4F 0293
45 4C 4F 53 4E 4F 43 20 35 32 37 2C 029F
27 20 52 4F 46 20 54 49 41 57 20 2C 02AB
OD 54 50 4D 4F 52 50 20 27 3E 3E 3E 02B7
OA 02C3
3C 02CF
0293
55 54 45 53 20 46 49 20 20 2A 2A 00' 02D0
54 43 45 52 52 4F 43 20 53 49 20 50 02D0
52 44 20 54 43 45 50 53 55 53 20 2C 02DC
4F 52 50 20 20 2A 2A 20 20 45 56 49 02E8
20 20 20 20 20 2A 2A 0A 02F4
20 20 20 4D 45 4C 42 0308
3E 02D0
2D 20 44 41 42 20 45 56 49 52 44 00' 030F
20 030F
OC 031B
030F
20 44 45 54 43 45 50 58 45 4E 55 00' 031C
43 45 48 43 20 45 4E 49 48 43 41 4D 031C
49 52 45 56 20 45 4C 49 48 57 20 4B 0328
50 20 44 41 4F 4C 20 47 4E 49 59 46 0334
OA 0D 48 54 41 0340
34 034C
031C
45 54 52 4F 50 50 55 53 20 4F 4E 00' 0351
52 54 4E 4F 43 20 4B 53 49 44 20 44 0351
OD 44 4E 55 4F 46 20 52 45 4C 4C 4F 035D
OA 0369
24 0375
0351
44 45 54 52 4F 50 50 55 53 4E 55 00' 0376
47 49 46 4E 4F 43 20 4B 53 49 44 20 0376
4E 55 4F 46 20 4E 4F 49 54 41 52 55 0382
20 30 20 45 56 49 52 44 20 2D 20 44 038E
OA 0D 31 20 44 4E 41 039A
36 03A6
0376
20 20 20 20 20 2A 2A 0A 0D 0A 0D 00' 03AD
54 45 53 20 52 45 50 4F 52 50 20 20 03AD
09 44 45 52 46 55 51 45 52 20 50 55 03B9
2A 2A 20 20 20 20 20 20 20 20 20 03C5
2D 03D1
03AD
```

```
232 PRINTEND: END MESSAGE
.ASCIC <13><10>?**** END OF VAX-11/730,725 ?

233 PRINTTEND: TEND MESSAGE
.ASCIC ?TEST ****?<13><10><13><10>

234 PRINTRET: RET MESSAGE
.ASCIC ?RETURNING TO VAX-11/730,725 CONSOLE, WAIT FOR '>>>'  
PROMPT?--

235 PRINTDRVERR: DRVERR MESSAGE
.ASCIC /** IF SETUP IS CORRECT, SUSPECT DRIVE **/<13><10>--

/** PROBLEM - /

236 PRINTDISKERR: DISKERR MESSAGE
.ASCIC /DRIVE BAD - /

237 PRINTLOADMCK: LOADMCK MESSAGE
.ASCIC /UNEXPECTED MACHINE CHECK WHILE VERIFYING LOAD PATH/<13><10>

238 PRINTNOCNTRL: NOCNTRL MESSAGE
.ASCIC /NO SUPPORTED DISK CONTROLLER FOUND/<13><10>

239 PRINTNODRIVE01: NODRIVE01 MESSAGE
.ASCIC /UNSUPPORTED DISK CONFIGURATION FOUND - DRIVE 0 AND 1/<13><10>

240 PRINTPROPER: PROPER MESSAGE
.ASCIC <13><10><13><10>/** PROPER SETUP REQUIRED **/
```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

DECLARATIONS

CRD HARDCOKE INSTRUCTION TESTS
DECLARATIONS

H 2
14-JUN-1984

Fiche 1 Frame H2

Sequence 20

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 8
14-JUN-1984 15:53:29 DRBO:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(4)

4B 43 45 48 43 20 20 2A 2A 0A 0D 00' 03DB
53 4F 4E 47 41 49 44 20 52 4F 46 20 03E7
44 52 43 20 4B 43 41 50 20 43 49 54 03F3
2A 2A 09 4B 43 41 50 03FF
45 53 4E 49 20 20 20 20 2A 2A 0A 0D 0406
49 50 53 20 44 4E 41 20 44 45 54 52 0412
20 4E 49 20 47 4E 49 4E 4E 041E
4B 03DB
0427
48 43 20 20 20 20 20 2A 2A 0A 0D 00' 0427
41 55 44 20 54 41 48 54 20 4B 43 45 0433
49 4E 4E 49 50 53 20 53 49 20 3A 30 043F
0A 0D 2A 2A 20 20 20 20 20 47 4E 044B
2E 0427
0A 0D 2A 2A 20 20 20 20 00' 0456
08 0456
20 4F 54 55 41 00' 045F
05 045F
20 55 4E 45 4D 00' 0465
05 0465
20 20 20 31 20 54 52 41 50 00' 046B
09 046B
41 54 53 20 47 4E 49 54 53 45 54 00' 0475
54 20 4E 55 52 20 2D 20 44 45 54 52 0481
4D 20 30 33 3A 30 20 3D 20 45 4D 49 048D
53 45 54 55 4E 49 0499
29 0475
20 20 20 44 41 42 20 55 50 43 00' 049F
0A 049F
20 35 32 43 52 00' 04AA
05 04AA
20 30 41 41 44 00' 0480
05 0480
20 30 41 55 44 00' 0486
05 0486
20 30 41 4A 44 00' 04BC
05 04BC
20 31 41 4A 44 00' 04C2
05 04C2
20 30 36 41 52 00' 04C8
05 04C8
20 30 38 41 52 00' 04CE
05 04CE

241 PRINTNSPN6025: NSPN6025 MESSAGE
.ASCIC <13><10>7** CHECK FOR DIAGNOSTIC PACK CRDPACK **/-

<13><10>/** INSERTED AND SPINNING IN /

242 PRINTNSPN8081: NSPN8081 MESSAGE
.ASCIC <13><10>7** CHECK THAT DUA0: IS SPINNING **/<13><10>

243 PRINTSPNEND: SPNEND MESSAGE
.ASCIC / **/<13><10>

244 PRINTAUTO: AUTO MESSAGE
.ASCIC /AUTO /

245 PRINTMENU: MENU MESSAGE
.ASCIC /MEND /

246 PRINTPART: PART MESSAGE
.ASCIC /PART 1 /

247 PRINTTEST: TEST MESSAGE
.ASCIC /TESTING STARTED - RUN TIME = 0:30 MINUTES/

248 PRINTCPU: CPU MESSAGE
.ASCIC /CPU BAD - /

249 PRINTRC25: RC25 MESSAGE
.ASCIC /RC25 /

250 PRINTDAA0: DAA0 MESSAGE
.ASCIC /DAA0 /

251 PRINTDUA0: DUA0 MESSAGE
.ASCIC /DUA0 /

252 PRINTDJA0: DJA0 MESSAGE
.ASCIC /DJA0 /

253 PRINTDJA1: DJA1 MESSAGE
.ASCIC /DJAT /

254 PRINTR60: RA60 MESSAGE
.ASCIC /RA60 /

255 PRINTR80: RA80 MESSAGE
.ASCIC /RA80 /

ZZ-ENKAZ-1.3
ENKAZ01
01-30

DECLARATIONS

CRD HARDWARE INSTRUCTION TESTS
DECLARATIONS

1 2
14-JUN-1984

Fiche 1 Frame 12

Sequence 21

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 9
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(4)

```
20 31 38 41 52 00' 04D4 256 PRINTR81: RA81 MESSAGE
05 04D4 .ASCIC /RA8T /
04DA 257
0000055E 04DA 258 BOO$GT_COMBUF: .BLKB 132
000005E2 055E 259 BOO$GL_FREEMEM: .BLKB 132
05E2 260
05E2 261 .ALIGN PAGE ; Align SFB on page boundary
0600 262
0600 263
0600 264 SCB:: ; SYSTEM CONTROL BLOCK
00000000 0600 265 SCB0:: .LONG 0 ; (RESERVED)
00000000 0604 266 SCB4:: .LONG 0 ; MACHINE CHECK
00000620 0608 267 SCB8:: .BLKL 6
00000000 0620 268 SCB20:: .LONG 0 ; ACCESS CONTROL VIOLATION
00000660 0624 269 SCB24:: .BLKL 15
00000000 0660 270 SCB60:: .LONG 0 ; WRITE NON-EXISTENT MEMORY
00000800 0664 271 SCB64:: .BLKL 103
0800 272
0800 273
0800 274 .ALIGN PAGE ; Align INTERRUPT STACK
08C0 275 ; on page boundary
000009FC 0800 276 KERNEL_STACK:: .BLKL 127 ; Allocate a stack area
00000000 09FC 277 TOP_OF_KSTACK:: .LONG 0 ; Set top of stack pointer
0A00 278
0A00 279 .ALIGN PAGE ; Align INTERRUPT STACK
0A00 280 ; on page boundary
00000BFC 0A00 281 INTERRUPT_STACK:: .BLKL 127 ; Allocate a stack area
00000000 0BFC 282 TOP_OF_ISTACK:: .LONG 0 ; Set top of stack pointer
0C00 283
0C00 284 .ALIGN PAGE ; Align RPB on page boundary
0C00 285
00000E00 0C00 286 RPB_BUFF:: .BLKL 128 ; Set top of stack pointer
0E00 287
0E00 288 ;++
0E00 289
0E00 290 ; Facility: System Bootstrapping
0E00 291
0E00 292 ; Abstract: CONIO provides basic console read,readprompt and write facilities.
0E00 293
0E00 294 ; Environment: Mode=kernel
0E00 295
0E00 296 ; Author: RICHARD I. HUSTVEDT, Creation date: 27-APR-1978
0E00 297
0E00 298 ; Modified by: PETER GREEN 23-MAY-1983
0E00 299
0E00 300 ; MODIFIED TO PROMPT FOR ERROR PREPARATION MESSAGES.
0E00 301
0E00 302 ; Darryl L. Ball 14-June-1984
0E00 303
0E00 304 ; Added 730/725 Processor register definitions. Needed for compilation
0E00 305 ; under Version 4.X of VAX/VMS.
0E00 306
0E00 307
0E00 308 ;--
0E00 309
0E00 310 ;
```

```
OE00 311 ; Include files:
OE00 312 ;
OE00 313 $PRDEF ; Define processor registers
OE00 .SAVE LOCAL_BLOCK
OE00 .PSECT $ABSS,ABS
0000OE00 .RESTORE
OE0C 314 $PR730DEF ; Define 730/725 processor registers
OE00 .SAVE LOCAL_BLOCK
OE00 .PSECT $ABSS,ABS
0000OE00 .RESTORE
OE00 315 $SSDEF ; Define status code values
OE00 .SAVE LOCAL_BLOCK
OE00 .PSECT $ABSS,ABS
0000OE00 .RESTORE
OE00 316
OE00 317
OE0C 318 : MACROS:
OE00 319 :
OE00 320 :
OE00 321 :
OE00 322 : Equated Symbols:
OE00 323 :
OE00 324 :
0000000D OE00 325 CR = 13 ; Character code for carriage return
0000000A OE00 326 LF = 10 ; Character code for line feed
00000015 OE00 327 CONTROL U = 21 ; Character code for control-U
0000007F OE00 328 RUBOUT = 127 ; Character code for rubout
00000000 OE00 329 V_RUB = 0 ; Rubout sequence in progress
OE00 330
OE00 331 :
OE00 332 : Own Storage:
OE00 333 :
```

```

OE00 335 .SBTTL BOO$READPROMPT - Prompt and read input string
OE00 336
OE00 337 ++
OE00 338 Functional Description:
OE00 339 BOO$READPROMPT outputs the specified ASCIIZ prompt string on the
OE00 340 console terminal then checks the count of characters to be read.
OE00 341 If zero it exits, otherwise it reads the console terminal until
OE00 342 either a carriage return is encountered or the character count
OE00 343 is satisfied. The specified buffer is filled with an ASCII
OE00 344 string containing the characters read but not including the
OE00 345 terminating carriage return.
OE00 346
OE00 347 Calling Sequence:
OE00 348 CALLG ARGLIST,BOO$READPROMPT
OE00 349
OE00 350 Input Parameters:
OE00 351 PROMPT(AP) - Address of ASCIIZ prompt string
00000004 OE00 352 PROMPT = 4
OE00 353
OE00 354 SIZE(AP) - Maximum length of input string
00000008 OE00 355 SIZE = 8
OE00 356 Note: if size is zero, then nothing is read
OE00 357 and only the prompt string is written.
OE00 358
OE00 359 BUF(AP) - Address of input buffer
0000000C OE00 360 BUF = 12
OE00 361
OE00 362 Output Parameters:
OE00 363 R0 - Completion status code (always SS$_NORMAL)
OE00 364
OE00 365 Buffer located by BUF(AP) will be filled with the string
OE00 366 read as an ASCII string.
OE00 367
OE00 368 --
OE00 369 .LIST MEB
OE00 370 BOO$READPROMPT::
OE00 371 .WORD ^M<R2,R4,R7,R8,R9>
58 04 AC 0394 OE00 372 5$: MOVL PROMPT(AP),R8
0089 30 OE00 373 BSBW OUTZSTR
52 08 AC 00 OE00 374 MOVL SIZE(AP),R2
6B 13 OE00 375 BEQL 70$
54 D4 OE00 376 CLRL R4
59 0C AC 00 OE00 377 MOVL BUF(AP),R9
89 94 OE00 378 CLRB (R9)+
57 69 9E OE00 379 MOVAB (R9),R7
07 52 F5 OE00 380 SOBGTR R2,10$
50 0D 9A OE00 381 MOVZBL #CR,R0
65 10 OE00 382 BSBB OUTCHAR
49 11 OE00 383 BRB 60$
50 20 DB OE00 384 10$: MFPR #PR$ RXCS,R0
F9 50 07 E1 OE00 385 BBC #7,R0,10$
58 21 DB OE00 386 MFPR #PR$ RXDB,R8
8F 8A OE00 387 BICB #^X80,R8
7F 8F 58 91 OE00 388 CMPB R8,#RUBOUT
1F 12 OE00 389 BNEQ 40$
02 54 00 E2 OE00 390 BBSS #V RUB,R4,20$
40 10 OE00 391 BSBB OUTBSLSH

```

```

: Show macro expansions
:
: Print to prompt string
: And send it to the console
: Maximum number of characters to read
: Done if nothing to read
: Clear flags
: Set address of input buffer
: Initialize string count
: Remember start of buffer
: Decrement and test character count
: Set to send CR/LF
: Send CR
: End of read
: Get console status
: Wait for ready
: Get console character
: Strip parity
: Check for rubout
: No
: Set start of rubout sequence
: Output back slash

```

58	79	9A	OE3E	392	20\$:	MOVZBL	-(R9),R8	:	Get rubbed out character
	08	12	OE41	393		BNEQ	30\$	:	Skip increment
	59	D6	OE43	394		INCL	R9	:	Point at start of buffer
52	08	AC	OE45	395		MOVL	SIZE(AP),R2	:	Restore size limit
	D9	11	OE49	396		BRB	10\$	:	And get another character
	37	10	OE4B	397	30\$:	BSBB	OUTR8	:	Output rubbed out character
	52	D7	OE4D	398		DECL	R2	:	One less character
	D3	11	OE4F	399		BRB	10\$	:	And get another
02	54	00	OE51	400		BBCC	#V RUB,R4,40\$	:	Terminate rubout sequence
	27	10	OE55	401		BSBB	OUTBSLSH	:	Output backslash
	58	15	OE57	402	40\$:	CMPB	#CONTROL_U,R8	:	Check for line cancel (control-U)
	A6	13	OE5A	403		BEQL	5\$	:	Branch if so
03	58	06	OE5C	404		BBC	#6,R8,50\$	:	Branch if not alpha
	58	20	OE60	405		BICB	#32,R8	:	Set to upper case
			OE63	406	50\$:			:	
	1F	10	OE63	407		BSBB	OUTR8	:	Echo character
89	58	90	OE65	408		MOVB	R8,(R9)+	:	Buffer new character
	0D	91	OE68	409		CMPB	#CR,R8	:	Is character a carriage return?
	B7	12	OE6B	410		BNEQ	10\$	:	No,
	50	0A	OE6D	411	60\$:	MOVZBL	#LF,R0	:	Yes send line feed also
	15	10	OE70	412		BSBB	OUTCHAR	:	Output character in R0
	59	57	OE72	413		SUBL	R7,R9	:	Compute character count
	59	D7	OE75	414		DECL	R9	:	Remove carriage return
	77	59	OE77	415		MOVB	R9, -(R7)	:	and form counted string
	50	01	OE7A	416	70\$:	MOVZWL	#SS\$_NORMAL,R0	:	Return normal completion status
		04	OE7D	417		RET		:	
			OE7E	418	OUTBSLSH:			:	Output back slash
50	5C	8F	OE7E	419		MOVZBL	#^AX\X,R0	:	Set character code
	03	11	OE82	420		BRB	OUTCHAR	:	And output it
	50	58	OE84	421	OUTR8:	MOVZBL	R8,R0	:	Get character to output
			OE87	422	OUTCHAR:			:	Output character in R0
	51	22	OE87	423		MFPR	#PR\$ TXCS,R1	:	Get console transmitter status
F9	51	07	OE8A	424		BBC	#7,RT,OUTCHAR	:	Wait for ready
	23	5D	OE8E	425		MTPR	R0,#PR\$ TXDB	:	Send character to console
		05	OE91	426	OUTEX:	RSB		:	Return
			OE92	427	OUTZSTR:			:	Output ASCII string
	50	88	OE92	428		MOVZBL	(R8)+,R0	:	Get next character to output
	FA	13	OE95	429		BEQL	OUTEX	:	Branch if at end of string
	EE	10	OE97	430		BSBB	OUTCHAR	:	Send character
	F7	11	OE99	431		BRB	OUTZSTR	:	Continue for all characters
			OE9B	432				:	
			OE9B	433				:	

22-ENKAZ-1.3
ENKAZ01
01-30

PROMPT1 - Routine to prompt console for
CRD HARDWARE INSTRUCTION TESTS
PROMPT1 - Routine to prompt console for

M 2
14-JUN-1984

Fiche 1 Frame M2

Sequence 25

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 13
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(4)

```

                                0E9B 435      .SBTTL PROMPT1 - Routine to prompt console for input
                                0E9B 436      ;
                                0E9B 437      ; Calling Sequence:
                                0E9B 438      BSBW PROMPT1
                                0E9B 439      .ASCIZ /prompt string/
                                0E9B 440      ;
                                0E9B 441      ; Output Parameters:
                                0E9B 442      COMBUF - Filled with ASCII string representing reply.
                                0E9B 443      Reply does not include terminator.
                                0E9B 444      ;
                                0E9B 445      ;
                                0E9B 446      PROMPT1:
                                0E9B 447      MOVL (SP),R0 ; Get prompt string pointer
                                0E9E 448      PUSHAB W^BOO$GT_COMBUF ; Set buffer address
                                0EA2 449      MOVZBL #BOO$C_COMBUFSZ,-(SP) ; and size for read.
                                0EA6 450      PUSHL R0 ; Set address of prompt string
                                0EA8 451      CALLS #3,W^BOO$READPROMPT ; Prompt and read response
                                0EAD 452      LOCC #0,#256,a(SP) ; Find end of string
                                0EB4 453      JMP 1(R1) ; and return
                                0EB7 454
                                0EB7 455
                                0EB7 456      .SBTTL BOO$TYPE_ASCII - Type a counted ASCII string
                                0EB7 457      ;++
                                0EB7 458      ; Functional Description:
                                0EB7 459      ; This routine accepts a string descriptor and types the message
                                0EB7 460      ; on the console terminal
                                0EB7 461      ;
                                0EB7 462      ; Calling Sequence:
                                0EB7 463      BSBW BOO$TYPE_ASCII
                                0EB7 464      ;
                                0EB7 465      ; Inputs:
                                0EB7 466      R1 = Address of ASCII string
                                0EB7 467      ;
                                0EB7 468      ; Outputs:
                                0EB7 469      R0,R1 altered
                                0EB7 470      All other registers preserved
                                0EB7 471      ;--
                                0EB7 472
                                0EB7 473      BOO$TYPE_ASCII:
                                0EB7 474      MOVZBL (R1)+,R0 ; R0 = Size, R1 = Address of string
                                0EBA 475      PUSHB #^M<R2,R3,R4,R5> ; Save registers
                                0EBC 476      MOVL W^BOO$GL_FREEMEM,R3 ; Address of scratch storage
                                0EC1 477      CLRQ -(SP) ; No read for PROMPTREAD call
                                0EC3 478      PUSHL R3 ; Address of ASCII string
                                0EC5 479      MOVCL R0,(R1),(R3) ; Move the string to scratch storage
                                0EC9 480      CLRB (R3)+ ; and make it ASCII
                                0ECB 481      CALLS #3,W^BOO$READPROMPT ; Type the string
                                0ED0 482      POPR #^M<R2,R3,R4,R5> ; Recover saved registers
                                0ED2 483      RSB
                                0ED3 484      ;
                                0ED3 485      ; THIS ROUTINE PRINTS THE MESSAGE SHOWN AND WAITS FOR USER
                                0ED3 486      ; INTERVENTION ,<CR>, FOR PREPARATION ERRORS.
                                0ED3 487      ;
                                0ED3 488      PROMPT_MESS:
                                0ED3 489      BSBW PROMPT1 ; NO, PROMPT TO SPIN
                                0ED5 490      .ASCIZ /** THEN TYPE <CR> TO CONTINUE **/
                                0EF1

```

50 6E D0 0E9B 447 MOVL (SP),R0 ; Get prompt string pointer
F638 CF 9F 0E9E 448 PUSHAB W^BOO\$GT_COMBUF ; Set buffer address
7E 84 8F 9A 0EA2 449 MOVZBL #BOO\$C_COMBUFSZ,-(SP) ; and size for read.
50 DD 0EA6 450 PUSHL R0 ; Set address of prompt string
FF53 CF 03 FB 0EA8 451 CALLS #3,W^BOO\$READPROMPT ; Prompt and read response
CO BE 0100 8F 00 3A 0EAD 452 LOCC #0,#256,a(SP) ; Find end of string
01 A1 17 0EB4 453 JMP 1(R1) ; and return

50 81 9A 0EB7 474 MOVZBL (R1)+,R0 ; R0 = Size, R1 = Address of string
3C BB 0EBA 475 PUSHB #^M<R2,R3,R4,R5> ; Save registers
53 F69E CF D0 0EBC 476 MOVL W^BOO\$GL_FREEMEM,R3 ; Address of scratch storage
7E 7C 0EC1 477 CLRQ -(SP) ; No read for PROMPTREAD call
53 DD 0EC3 478 PUSHL R3 ; Address of ASCII string
63 61 50 28 0EC5 479 MOVCL R0,(R1),(R3) ; Move the string to scratch storage
83 94 0EC9 480 CLRB (R3)+ ; and make it ASCII
FF30 CF 05 FB 0ECB 481 CALLS #3,W^BOO\$READPROMPT ; Type the string
3C 8A 0ED0 482 POPR #^M<R2,R3,R4,R5> ; Recover saved registers
05 0ED2 483 RSB

C6 10 0ED3 488 PROMPT_MESS:
4F 54 20 3E 52 43 3C 20 45 50 59 54 0ED3 489 BSBW PROMPT1 ; NO, PROMPT TO SPIN
0ED5 490 .ASCIZ /** THEN TYPE <CR> TO CONTINUE **/
0EF1

22-ENKAZ-1.3 PRINT ROUTINE
ENKAZ01
01-30

CRD HARDWARE INSTRUCTION TESTS
PRINT ROUTINE

B 3
14-JUN-1984

Fiche 1 Frame B3

Sequence 27

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 15
14-JUN-1984 15:53:29 CRBO:CBALL.ENKAZ.ENKAZ_1JENKAZ01A.(4)

```

OFOE 497 .SBTTL PRINT ROUTINE
OFOE 498
OFOE 499 :++
OFOE 500 : FUNCTIONAL DESCRIPTION:
OFOE 501 :
OFOE 502 : THIS ROUTINE IS USED TO TRANSMIT AN ASCII MESSAGE TO THE CONSOLE
OFOE 503 : TERMINAL.
OFOE 504 :--
OFOE 505 PRINT_ROUTINE:
51 01 D0 OFOE 506 MOVL #1,R1 ; Initialize index
53 64 9A OF11 507 MOVZBL (R4),R3 ; Get number of chars to print
52 52 D4 OF14 508 CLRL R2 ; Clear character stuffer
52 64 9A OF16 509 10$: MOVZBL (R4) [R1],R2 ; Get a character
F9 50 22 DB OF1A 510 20$: MFPR #TXCS,R0 ; Get tx status
23 07 E1 OF1D 511 BBC #7,R0,20$ ; Transmitter ready?
EE 51 52 DA OF21 512 MTPR R2,#TXDB ; Print a character
53 53 F3 OF24 513 AOBLEQ R3,R1,10$ ; Print the whole message
05 05 OF28 514 RSB ; EXIT
OF29 515
```

```

OF29 517 .SBTTL ERROR ROUTINE
OF29 518 :++
OF29 519 : FUNCTIONAL DESCRIPTION:
OF29 520 :
OF29 521 : THIS ROUTINE IS USED TO DISPLAY AN ERROR MESSAGE IN CASE OF CPU OR
OF29 522 : LOAD PATH FAILURE OR WHEN WARNING MESSAGE WHEN NO CRD SUPPORTED
OF29 523 : DISK CONTROLLER IS FOUND. IT ALSO ABORTS AFTER ALL MESSAGES ARE PRINTED.
OF29 524 :
OF29 525 : R4 = ADDRESS OF ASCII MESSAGE
OF29 526 :--
OF29 527
OF29 528 ERROR_ROUTINE::
12 F2D6 CF E8 OF29 529 BLBS WHAT_MODE,10$ : DETERMINE WHAT MODE
OF29 530 : AUTO OR MENU
54 F313 CF DE OF2E 531 MOVAL PRINTFAIL,R4 : FAILED
D8 AF 16 OF33 532 JSB PRINT_ROUTINE : PRINT MESSAGE
54 F525 CF DE OF36 533 MOVAL PRINTAUTO,R4 : AUTO
D0 AF 16 OF3B 534 JSB PRINT_ROUTINE : PRINT MESSAGE
OF3E 535 BRB 20$
54 F521 CF DE OF40 536 10$: MOVAL PRINTMENU,R4 : MENU
C6 AF 16 OF45 537 JSB PRINT_ROUTINE : PRINT MESSAGE
OF48 538
54 F30B CF DE OF48 539 20$: MOVAL PRINTABORT,R4 : FAILED
BE AF 16 OF4D 540 JSB PRINT_ROUTINE : PRINT MESSAGE
OF50 541
15 F2B3 CF E8 OF50 542 BLBS NOCNTLR_FND,NO_LOAD : IS THERE A CRD SUPPORTED
OF55 543 : CONTROLLER?
18 00000000'EF E8 OF55 544 BLBS NODRIVE_0_1,DRIVE_NFND : IS THERE A CRD SUPPORTED
OF5C 545 : DISK DRIVE 0 OR 1?
1D F2AB CF E8 OF5C 546 BLBS LOAD_ERROR,LOAD_BAD : IS THERE A CRD SUPPORTED
OF61 547
20 00000000'EF E8 OF61 548 BLBS BAD_DISK,DISK_ERR : BAD STATUS FROM ONLINE COMMAND
OF68 549
29 11 OF68 550 BRB CPU_ERR : CPU ERROR
OF6A 551 NO_LOAD::
54 F3E3 CF DE OF6A 552 MOVAL PRINTNOCNTLR,R4 : NO CRD SUPPORTED CONTROLLER FOUND
9C AF 16 OF6F 553 JSB PRINT_ROUTINE : PRINT MESSAGE
28 11 OF72 554 BRB ERR_END
OF74 555
54 F3FE CF DE OF74 556 DRIVE_NFND:
OF74 557 MOVAL PRINTNODRIVE01,R4 : NO CRD SUPPORTED DISK DRIVE
OF79 558 : 0 OR 1 FOUND
92 AF 16 OF79 559 JSB PRINT_ROUTINE : PRINT MESSAGE
1E 11 OF7C 560 BRB ERR_END
OF7E 561
54 F39A CF DE OF7E 562 LOAD_BAD:
OF7E 563 MOVAL PRINTLOADMCK,R4 : UNEXPECTED MACHINE CHECK WHILE
OF83 564 : VERIFYING LOAD PATH
88 AF 16 OF83 565 JSB PRINT_ROUTINE : PRINT MESSAGE
14 11 OF86 566 BRB ERR_END
OF88 567
54 F383 CF DE OF88 568 DISK_ERR:
OF88 569 MOVAL PRINTDISKERR,R4 : UNEXPECTED MACHINE CHECK WHILE
FF7D CF 16 OF8D 570 JSB PRINT_ROUTINE : VERIFYING LOAD PATH
09 11 OF91 571 BRB ERR_END : PRINT MESSAGE
OF93 572 CPU_ERR:
54 F508 CF DE OF93 573 MOVAL PRINTCPU,R4 : CPU BAD

```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

ERROR ROUTINE

CRD HARDWARE INSTRUCTION TESTS
ERROR ROUTINE

D 3
14-JUN-1984

Fiche 1 Frame D3

Sequence 29

14-JUN-1984 16:01:36 VAX-11 Macro V03-01
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(4)

Page 17

	FF72 CF	16	OF98	574	JSB	PRINT_ROUTINE	; PRINT MESSAGE
			OF9C	575			
			OF9C	576	ERR_END:		
54	F292 CF	DE	OF9C	577	MOVAL	PRINTFIELD,R4	; CALL FIELD SERVICE
	FF69 CF	16	OFA1	578	JSB	PRINT_ROUTINE	; PRINT MESSAGE
			OFA5	579			
54	F2BE CF	DE	OFA5	580	MOVAL	PRINTEND,R4	; END OF VAX-11/730
	FF60 CF	16	OFAA	581	JSB	PRINT_ROUTINE	; PRINT MESSAGE
			OFAE	582			
0B	F251 CF	E8	OFAE	583	BLBS	WHAT_MODE,30\$	; DETERMINE WHAT MODE
			OFB3	584			; AUTO OR MENU
54	F4A8 CF	DE	OFB3	585	MOVAL	PRINTAUTO,R4	; AUTO
	FF52 CF	16	OFB8	586	JSB	PRINT_ROUTINE	; PRINT MESSAGE
		09	OFBC	587	BRB	40\$	
			OFBE	588			
54	F4A3 CF	DE	OFBE	589	MOVAL	PRINTMENU,R4	; MENU
	FF47 CF	16	OF C3	590	JSB	PRINT_ROUTINE	; PRINT MESSAGE
			OF C7	591			
54	F2BA CF	DE	OF C7	592	MOVAL	PRINTTEND,R4	; TEST **
	FF3E CF	16	OFCC	593	JSB	PRINT_ROUTINE	; PRINT MESSAGE
			OFD0	594			
54	F2BF CF	DE	OFD0	595	MOVAL	PRINTRET,R4	; RETURNING TO CONSOLE PROMPT
	FF35 CF	16	OFD5	596	JSB	PRINT_ROUTINE	; PRINT MESSAGE
			OFD9	597			
00000174'EF		16	OFD9	598	JSB	CLEAN_UP	; Go to Clean up routine
		00	OFDF	599	HALT		; Halt the processor
			OFEO	600			

```

OFE0 602 .SBTTL RA60 PRINT ROUTINE
OFE0 603 :++
OFE0 604 : FUNCTIONAL DESCRIPTION:
OFE0 605 :
OFE0 606 : This routine is used to display a message when RA60 CRD supported
OFE0 607 : disk controller is found. It also CHECKS WHAT DRIVE AND IF THE DRIVE IS
OFE0 608 : SPINNING and CORRECT PACK LABEL.
OFE0 609 :--
OFE0 610 :
OFE0 611 RA60_TEST::
48 F21F CF E8 OFF0 612 BLBS WHAT_MODE,30$
43 F22E CF E8 OFF5 613 BLBS RA60_MESS_ST,30$
54 F235 CF DE OFEA 614 MOVAL PRINTPASS,R4 : PRINT PASSED TO CPU PART 3
FF1B CF 16 OFEF 615 JSB PRINT_ROUTINE : PRINT MESSAGE
F220 CF D6 OFF3 616 INCL RA60_MESS_ST
54 F4CD CF DE OFF7 617 MOVAL PRINTRA60,R4 : PRINT MESSAGE
FF0E CF 16 OFFC 618 JSB PRINT_ROUTINE : RA60
54 F467 CF DE 1000 619 MOVAL PRINTPART,R4 : PRINT MESSAGE
FF05 CF 16 1005 620 JSB PRINT_ROUTINE : PART 1
OB 00000000'EF E9 1009 621 BLBC WHAT_DRIVE,10$ : WHAT DRIVE?
54 F4AE CF DE 1010 622 MOVAL PRINTDJA1,R4 : PRINT MESSAGE
FEF5 CF 16 1015 623 JSB PRINT_ROUTINE : DJA1
09 11 1019 624 BRB 20$
54 F49D CF DE 101B 625 10$: MOVAL PRINTDJA0,R4 : PRINT MESSAGE
FEEA CF 16 1020 626 JSB PRINT_ROUTINE : DJA0
54 F44D CF DE 1024 627 20$: MOVAL PRINTTEST,R4 : PRINT MESSAGE
FEE1 CF 16 1029 628 JSB PRINT_ROUTINE : TESTING STARTED AND TIME
04 00000000'EF E9 102D 629 30$: BLBC BAD_DISK,40$
FEF1 CF 16 1034 630 JSB ERROR_ROUTINE
03 00000000'EF E8 1038 631 40$: BLBS SPIN_DWN,50$ : IS IT SPINNING
EFBE' 31 103F 632 BRW TEST_8 : YES READ PACK LABEL
54 F367 CF DE 1042 633 50$: MOVAL PRINTPROPER,R4 : PRINT MESSAGE
FEC3 CF 16 1047 634 JSB PRINT_ROUTINE : PROPER SETUP REQUIRED
54 F38C CF DE 104B 635 MOVAL PRINTNSPN6025,R4 : PRINT MESSAGE
FEBA CF 16 1050 636 JSB PRINT_ROUTINE : CHECK CRDPACK AND SPINNING
1054 637
OB 00000000'EF E9 1054 638 BLBC WHAT_DRIVE,60$
105B 639
54 F463 CF DE 105B 640 MOVAL PRINTDJA1,R4 : PRINT MESSAGE
FEAA CF 16 1060 641 JSB PRINT_ROUTINE : DJA1
09 11 1064 642 BRB 70$
54 F452 CF DE 1066 643 60$: MOVAL PRINTDJA0,R4 : PRINT MESSAGE
FE9F CF 16 106B 644 JSB PRINT_ROUTINE : DJA0
54 F3E3 CF DE 106F 645 70$: MOVAL PRINTSPNEND,R4 : PRINT MESSAGE
FE96 CF 16 1074 646 JSB PRINT_ROUTINE : **
1B 00000000'EF E9 1078 647 BLBC HMBLK_ERR,80$ : HOME BLOCK ERROR?
54 F24D CF DE 107F 648 MOVAL PRINTDRVERR,R4 : PRINT MESSAGE
FE86 CF 16 1084 649 JSB PRINT_ROUTINE : IF SETUP CORRECT SUSPECT DRIVE ERR
54 F1A6 CF DE 1088 650 MOVAL PRINTFIELD,R4 : PRINT MESSAGE
FE7D CF 16 108D 651 JSB PRINT_ROUTINE : CALL FIELD SERVICE
54 F3C1 CF DE 1091 652 MOVAL PRINTSPNEND,R4 : PRINT MESSAGE
FE74 CF 16 1096 653 JSB PRINT_ROUTINE : **
FE35 CF 16 109A 654 80$: JSB PROMPT_MESS : PROMPT FOR SETUP
109E 655

```

```

109E 657 .SBTTL RA80 AND RA81 PRINT ROUTINE
109E 658 :++
109E 659 : FUNCTIONAL DESCRIPTION:
109E 660 :
109E 661 : This routine is used to display a message when AN RA80 OR RA81 CRD
109E 662 : supported disk configuration is found, (in drive 0). It also CHECKS for
109E 663 : DRIVE 0 AND IF THE DRIVE IS SPINNING.
109E 664 :
109E 665 :--
109E 666 RA8081_TEST::
03 00000000'EF E9 109E 667 BLBC WHAT_DRIVE,10$ ; IF DRIVE 1
EF58' 31 10A5 668 BRW INIT_DRIVE ; CHECK DRIVE 0
48 F157 CF E8 10A8 669 10$: BLBS WHAT_MODE,40$
43 F16A CF E8 10AD 670 BLBS RA8081_MESS_ST,40$
54 F16D CF DE 10B2 671 MOVAL PRINTPASS,R4 ; PRINT PASSED
FE53 CF 16 10B7 672 JSB PRINT_ROUTINE ; PRINT MESSAGE
F15C CF D6 10BB 673 INCL RA8081_MESS_ST
0B 00000000'EF E9 10BF 674 BLBC RA81_PRES,20$ ; RA81?
10C6 675 ; IF 0 THEN RA80
54 F40A CF DE 10C6 676 MOVAL PRINTR81,R4 ; PRINT MESSAGE
FE3F CF 16 10CB 677 JSB PRINT_ROUTINE ; RA81
09 11 10CF 678 BRB 30$ ; CONTINUE TO PRINT MESSAGE
10D1 679 ; TEST STARTED
54 F3F9 CF DE 10D1 680 20$: MOVAL PRINTR80,R4 ; PRINT MESSAGE
FE34 CF 16 10D6 681 JSB PRINT_ROUTINE ; RA80
54 F38D CF DE 10DA 682 30$: MOVAL PRINTPART,R4 ; PRINT MESSAGE
FE2B CF 16 10DF 683 JSB PRINT_ROUTINE ; PART 1
54 F3CF CF DE 10E3 684 MOVAL PRINTDUA0,R4 ; PRINT MESSAGE
FE22 CF 16 10E8 685 JSB PRINT_ROUTINE ; DUA0
54 F385 CF DE 10EC 686 MOVAL PRINTTEST,R4 ; PRINT MESSAGE
FE19 CF 16 10F1 687 JSB PRINT_ROUTINE ; TESTING STARTED AND TIME
04 00000000'EF E9 10F5 688 40$: BLBC BAD_DISK,50$ ;
FE29 CF 16 10FC 689 JSB EPROR_ROUTINE ;
00000000'EF 95 1100 690 50$: TSTB SPIN_DWN ; IS IT SPINNING
03 12 1106 691 BNEQ 60$ ;
EEF5' 31 1108 692 BRW LES1_UDA_PARAM ; YES BOOT SUPERVISOR PORTION
54 F29E CF DE 110B 693 60$: MOVAL PRINTPROPER,R4 ; PRINT MESSAGE
FDFA CF 16 1110 694 JSB PRINT_ROUTINE ; PROPER SETUP REQUIRED
54 F30F CF DE 1114 695 MOVAL PRINTNSPN8081,R4 ; PRINT MESSAGE
FDF1 CF 16 1119 696 JSB PRINT_ROUTINE ; CHECK THAT DUA0 IS SPINNING
FDB2 CF 16 111D 697 JSB PROMPT_MESS ; PROMPT FOR SETUP
1121 698

```

```

1121 700 .SBTTL RC25 PRINT ROUTINE
1121 701 : **
1121 702 : FUNCTIONAL DESCRIPTION:
1121 703 :
1121 704 : This routine is used to display a message when RC25 CRD supported
1121 705 : disk OPTION is found. It also CHECKS for DRIVE 0. IF THE DRIVE IS
1121 706 : SPINNING and for correct pack label.
1121 707 : --
1121 708 RC25_TEST::
03 00000000'EF E9 1121 709 BLBC WHAT_DRIVE,10$ : IF DRIVE 1
      EED5' 31 1128 710 BRW INIT_DRV0 : CHECK DRIVE 0
      36 FOD4 CF E8 1128 711 10$: BLBS WHAT_MODE,20$
      31 FOEB CF E8 1130 712 BLBS RC25_MESS_ST,20$
54 FOEA CF DE 1135 713 MOVAL PRINTPASS,R4 : PRINT PASSED
      FDD0 CF 16 113A 714 JSB PRINT_ROUTINE : PRINT MESSAGE
      FODD CF D6 113E 715 INCL RC25_MESS_ST
54 F364 CF DE 1142 716 MOVAL PRINTRC25,R4 : PRINT MESSAGE
      FDC3 CF 16 1147 717 JSB PRINT_ROUTINE : RC25
54 F31C CF DE 114B 718 MOVAL PRINTPART,R4 : PRINT MESSAGE
      FDBA CF 16 1150 719 JSB PRINT_ROUTINE : PART 1
54 F358 CF DE 1154 720 MOVAL PRINTDAA0,R4 : PRINT MESSAGE
      FDB1 CF 16 1159 721 JSB PRINT_ROUTINE : DUA0
54 F314 CF DE 115D 722 MOVAL PRINTTEST,R4 : PRINT MESSAGE
      FDA8 CF 16 1162 723 JSB PRINT_ROUTINE : TESTING STARTED AND TIME
04 00000000'EF E9 1166 724 20$: BLBC BAD_DISK,30$ : DRIVE ERROR?
      FDB8 CF 16 116D 725 JSB ERROR_ROUTINE
03 00000000'EF E8 1171 726 30$: BLBS SPIN_DWN,40$ : IS IT SPINNING
      EE85' 31 1178 727 BRW TEST_8 : YES READ PACK LABEL
54 F22E CF DE 117B 728 40$: MOVAL PRINTPROPER,R4 : PRINT MESSAGE
      FDBA CF 16 1180 729 JSB PRINT_ROUTINE : PROPER SETUP REQUIRED
54 F253 CF DE 1184 730 MOVAL PRINTSPN6025,R4 : PRINT MESSAGE
      FDB1 CF 16 1189 731 JSB PRINT_ROUTINE : CHECK CRDPACK AND SPINNING
54 F31F CF DE 118D 732 MOVAL PRINTDAA0,R4 : PRINT MESSAGE
      FD78 CF 16 1192 733 JSB PRINT_ROUTINE : DUA0
54 F28C CF DE 1196 734 MOVAL PRINTSPNEND,R4 : PRINT MESSAGE
      FD6F CF 16 119B 735 JSB PRINT_ROUTINE : **
1B 00000000'EF E9 119F 736 BLBC HMBLK_ERR,50$ : HOME BLOCK ERROR?
54 F126 CF DE 11A6 737 MOVAL PRINTDRVERR,R4 : PRINT MESSAGE
      FDSF CF 16 11AB 738 JSB PRINT_ROUTINE : IF SETUP CORRECT SUSPECT DRIVE ERR
54 F07F CF DE 11AF 739 MOVAL PRINTFIELD,R4 : PRINT MESSAGE
      FDS6 CF 16 11B4 740 JSB PRINT_ROUTINE : CALL FIELD SERVICE
54 F29A CF DE 11B8 741 MOVAL PRINTSPNEND,R4 : PRINT MESSAGE
      FD4D CF 16 11BD 742 JSB PRINT_ROUTINE : **
      FDOF CF 16 11C1 743 50$: JSB PROMPT_MESS : PROMPT FOR SETUP
      11C5 744

```

```

11C5 746
11C5 747      .SBTTL  EXCEPTION HANDLER ROUTINES
11C5 748
11C5 749      .ALIGN LONG
11C8 750      ;++
11C8 751      ; FUNCTIONAL DESCRIPTION:
11C8 752      ;
11C8 753      ; This routine is used to handle all unexpected exceptions or
11C8 754      ; urgent interrupts (over priority ^X18). The results of an
11C8 755      ; unexpected exception or urgent interrupt is printing an error
11C8 756      ; message pointing to the CPU and aborting CRD Auto or Menu mode.
11C8 757      ;--
11C8 758      UNEX_E_I::
11C8 759      MTPR      #-1, #PR$ MCSR      ; CLEAR REGISTER
11CF 760      ADDL2    (SP)+, SP-        ; ADJUST SP
F03A CF 0064 8F 81 11D2 761      CMPW      #100, PASS_COUNT      ; TST IF IN HARDWARE
11D9 762      BGEQ     10$              ; OR LOAD PATH SECTION
F02F CF 01 11DB 763      MOVL      #1, CPU_BAD
6E FD45 CF 9E 11E0 764      MOVAB     ERROR_ROUTINE, (SP)      ; PRINT ERROR
11E5 765      BRB      20$              ; THEN ABORT
F01F CF 01 11E7 766      MOVL      #1, LOAD_ERROR
6E FD39 CF 9E 11EC 767 10$: MOVAB     ERROR_ROUTINE, (SP)
02 11F1 768 20$: REI
11F2 769
11F2 770
11F2 771      .ALIGN LONG
11F4 772      ;++
11F4 773      ; This routine is used to handle expected MACHINE CHECKS
11F4 774      ; when an IDC is not round. The results of this expected
11F4 775      ; MACHINE CHECK is to continue and look for a LESI or
11F4 776      ; UDA50 controller on the UNIBUS.
11F4 777      ;--
11F4 778      MCHK_IDC::
11F4 779      MTPR      #-1, #PR$ MCSR      ; CLEAR REGISTER
11FB 780      ADDL2    (SP)+, SP-        ; ADJUST SP
6E 00000000 EF DE 11FE 781      MOVAL     UDA_LESI, (SP)      ; CONTINUE TO
1205 782      ; SIZE
02 1205 783      REI
1206 784
1206 785      .ALIGN LONG
1208 786      ;++
1208 787      ; This routine is used to handle expected MACHINE CHECKS
1208 788      ; when a LESI or UDA50 is not found. The results of this expected
1208 789      ; MACHINE CHECK is to print a warning message due to the lack of
1208 790      ; a supported disk load device and then abort.
1208 791      ;--
1208 792      MCHK_LES_UDA::
1208 793      MTPR      #-1, #PR$ MCSR      ; CLEAR REGISTER
120F 794      ADDL2    (SP)+, SP-        ; ADJUST SP
EFFF CF 01 1212 795      MOVL      #1, NOCNTLR_FND
6E FDOE CF 9E 1217 796      MOVAB     ERROR_ROUTINE, (SP)      ; PRINT WARNING
121C 797      ; THEN ABORT
02 121C 798      REI
121D 799
121D 800      .ALIGN LONG      ; A4

```

```

1220 802
1220 803
1220 804
1220 805
1220 806
1220 807
1220 808
1220 809
1220 810
1220 811
1220 812
1220 813
1220 814
1220 815
1220 816
1220 817
1220 818
1220 819
1220 820
1220 821
1220 822
1220 823
1220 824
1220 825
1220 826
1220 827
1223 828
1223 829
1228 830
1228 831
1228 832
1220 833
1230 834
1230 835
1235 836
1238 837
1238 838
123D 839
123D 840
1240 841
1240 842
1243 843
1247 844
1247 845
124A 846
1252 847
1252 848
1255 849
1255 850
1257 851
1257 852
125D 853
125E 854
1260 855

      38 00 DA
EFDB CF 55 DO
      5E F9D0 CF DE
      04 5E DA
      5E F7C8 CF DE
      00 5E DA
      51 F3C4 CF DE
      11 51 DA
      53 01 DO
      50 82 AF DE
      81 60 DE
F5 53 0000007F 8F F3
      12 18 DA
      00 DD
      0000125E 'EF DF
      A0' 11

      .SBTTL  INITIALIZATION ROUTINE
      ++
      FUNCTIONAL DESCRIPTION:

      This routine initializes the stack pointer, turns off
      memory management and the stack pointers are invokled.
      This routine also sets up the SCB buffer to refer to an unexpected
      exception and urgent interrupt routine, moves to the
      KERNEL stack and sets the IPL level to ^X18.

      Assumptions:

      1) Init was performed in the Console mode prior to
      running this program.

      Therefore :

      MODE = KERNEL MODE
      IPL = 31
      STACK = INTERRUPT STACK

      --

      INIT1:  MTPR    #0,#PR$_MAPEN      ; Turn off memory management
              MOVL    R5,WHAT_MODE      ; AUTO OR MENU/ R5 PASSED FROM
              ; CRAB00.CMD AND CRMB00.CMD
              MOVAL    TOP_OF_ISTACK,SP  ; SP Points to the top of stack
              MTPR     SP,#PR$_ISP      ; Load the ISP
              MOVAL    TOP_OF_KSTACK,SP  ; SP Points to the top of stack
              MTPR     SP,#PR$_KSP      ; Load the KSP with the top of stack
              MOVAL    SCB,R1           ; SET UP SYSTEM CONTROL
              ; BUFFER BASE ADDRESS
              MTPR     R1,#PR$_SCBB     ; Load BASE System Control Block
              ; address
              MOVL     #1,R3           ; Initialize counter
              MOVAL    UNEX_E_1,R0      ; Load UNEXPECTED EXCEPTION or
              ; URGENT INTERRUPT address.
              MOVAL    (R0),(R1)+       ; Load the SCB with trap catchers
              AOBLEQ    #127,R3,10$     ; Continue until the SCB is full
              MTPR     #^X18,#PR$_IPL  ; Lower IPL to ^X18 (Urgent Interrupts)
              PUSHL    #0               ; Push zero on PSL to change to
              ; KERNEL STACK
              PUSHAL    20$             ; Push next Address on Stack
              REI                     ; Pop the Stack
              BRB      TEST_1          ; Start test

```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

INITIALIZATION ROUTINE

CRD HARDWARE INSTRUCTION TESTS
INITIALIZATION ROUTINE

J 3
14-JUN-1984

Fiche 1 Frame J3

Sequence 35

14-JUN-1984 16:01:36 VAX-11 Macro V03-01
14-JUN-1984 15:53:29 DRBO:[BALL.ENKAZ.ENKAZ_1]ENKA_01A.(7)

Pz Je 23

```
00000000 858 .PSECT A_HARDWARE, BYTE, PIC
0000 859
0000 860 .SBTTL TEST 1: AOBLEQ, AOBLS test
0000 861 ;++
0000 862 ; Test description:
0000 863 ;
0000 864 ; In this test, the AOBLEQ and AOBLS instructions are functionally
0000 865 ; verified.
0000 866 ;
0000 867 ; Assumptions:
0000 868 ;
0000 869 ;
0000 870 ;--
0000 871 START:
0000 872
0000 873 TEST_1:
0000 874 ;++
0000 875 ; Subtest 1: AOBLEQ
0000 876 ;--
0000 877 T1_S1:
EA 50 FFFFFFFF 8F D0 0000 878 MOVL #-1, R0 ; Initialize counter
51 51 50 D4 0007 879 CLRL R1 ; Initialize index
52 51 50 D6 0009 880 10$: INCL R0 ; Bump counter
00000F29'EF 06 13 000F 881 XORL3 R0, R1, R2 ; Counter = index?
0000007F 8F F3 0011 882 BEQL 20$ ; Branch if so
00000080 8F D0 001F 883 JMP ERROR_ROUTINE ; Else, report the error
50 50 50 CD 0026 884 20$: AOBLEQ #127, R1, 10$ ; Do the AOBLEQ
00000F29'EF 06 13 002A 885 MOVL #128, R0 ; Expect index = 128
00000080 8F D0 0026 886 XORL3 R0, R1, R2 ; Is index = 128?
00000F29'EF 06 13 002C 887 BEQL 30$ ; Branch if so
00000080 8F D0 002C 888 JMP ERROR_ROUTINE ; Else, report the error
0032 889 30$: ; End of subtest
0032 890 ;++
0032 891 ; Subtest 2: AOBLS
0032 892 ;--
0032 893 T1_S2:
EA 50 FFFFFFFF 8F D0 0032 894 MOVL #-1, R0 ; Initialize counter
51 51 50 D4 0039 895 CLRL R1 ; Initialize index
52 51 50 D6 003B 896 10$: INCL R0 ; Bump counter
00000F29'EF 06 13 0041 897 XORL3 R0, R1, R2 ; Counter = index?
00000080 8F F2 0043 898 BEQL 20$ ; Branch if so
00000080 8F D0 0049 899 JMP ERROR_ROUTINE ; Else, report the error
50 50 50 CD 0058 900 20$: AOBLS #128, R1, 10$ ; Do the AOBLS
00000F29'EF 06 13 005C 901 MOVL #128, R0 ; Expect index = 128
00000080 8F D0 0058 902 XORL3 R0, R1, R2 ; Is index = 128?
00000F29'EF 06 13 005E 903 BEQL 30$ ; Branch if so
00000080 8F D0 005E 904 JMP ERROR_ROUTINE ; Else, report the error
0064 905 30$: ; End of subtest
0064 906 ;***** End of test *****
```

```

0064 908 .SBYTL TEST 2: SOBGEQ, SOBGTR Test
0064 909 :++
0064 910 : Test description:
0064 911 :
0064 912 : In this test the instructions SOBGEQ and SOBGTR are functionally
0064 913 : verified.
0064 914 :
0064 915 : Assumptions:
0064 916 :
0064 917 : Test 1
0064 918 :--
0064 919 TEST_2:
0064 920 :++
0064 921 : Subtest 1: SOBGEQ
0064 922 :--
0064 923 t2_s1:
50 00000128 8F D0 0064 924 MOVL #^X128,R0 : Initialize reference counter
51 00000127 8F D0 0068 925 MOVL #^X127,R1 : Initialize index operand
52 51 50 D7 0072 926 10$: DECL R0 : Decrement reference counter
00000F29'EF 06 13 0074 927 XORL3 R0,R1,R2 : Is R0 = R1?
EF 51 F4 0078 928 BEQL 20$ : Branch if so
52 51 50 D7 007A 929 JMP ERROR_ROUTINE : Else, report the error
00000F29'EF 06 13 0080 930 20$: SOBGEQ R1,10$ : Should branch till R1 = -1
EF 51 F4 0083 931 DECL R0 : Decrement reference counter
52 51 50 D7 0085 932 XORL3 R0,R1,R2 : Is R0 = R1?
00000F29'EF 06 13 0089 933 BEQL 30$ : Exit if so
EF 51 F4 008B 934 JMP ERROR_ROUTINE : Else, report the error
0091 935 30$: : End of subtest
0091 936 :++
0091 937 : Subtest 2: SOBGTR
0091 938 :--
0091 939 t2_s2:
50 00000128 8F D0 0091 940 MOVL #^X128,R0 : Initialize counter
51 00000127 8F D0 0098 941 MOVL #^X127,R1 : Initialize index operand
52 51 50 D7 009F 942 10$: DECL R0 : Decrement counter
00000F29'EF 06 13 00A1 943 XORL3 R0,R1,R2 : Is R1 = R0?
EF 51 F4 00A5 944 BEQL 20$ : Branch if so
52 51 50 D7 00A7 945 JMP ERROR_ROUTINE : Else, report the error
00000F29'EF 06 13 00AD 946 20$: SOBGTR R1,10$ : Should branch until R1=0
EF 51 F4 00B0 947 DECL R0 : Expect R1 = 0
52 51 50 D7 00B2 948 XORL3 R0,R1,R2 : Is R1 = 0?
00000F29'EF 06 13 00B6 949 BEQL 30$ : Exit if zero
EF 51 F4 00B8 950 JMP ERROR_ROUTINE : Else, report the error
00BE 951 30$: : End of subtest
00BE 952 ;***** End of test *****

```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

TEST 3: BBS, BBC Test

CRD HARDWARE INSTRUCTION TESTS
TEST 3: BBS, BBC Test

L 3
14-JUN-1984

Fiche 1 Frame L3

Sequence 37

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 25
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A.(9)

```
00BE 954 .SBTTL TEST 3: BBS, BBC Test
00BE 955 ;++
00BE 956 ; Test description:
00BE 957 ;
00BE 958 ; In this test, the instructions BBS and BBC are functionally verified.
00BE 959 ;
00BE 960 ; Assumptions:
00BE 961 ;
00BE 962 ; Test 1-Test 2
00BE 963 ;--
00BE 964 TEST_3:
00BE 965 ;++
00BE 966 ; Subtest 1: BBS,BBC
00BE 967 ;--
00BE 968 T3_S1:
52 50 D4 00BE 969 CLRL R0 ; Initialize index
51 51 D4 00C0 970 CLRL R1 ; Initialize operand
52 FFFF FFFF 8F D0 00C2 971 MOVL #-1,R2 ; Initialize operand
04 51 50 E0 00C9 972 10$: BBS R0,R1,20$ ; Should not branch
06 51 50 E1 00C0 973 BBC R0,R1,30$ ; Should branch
00000F29'EF 17 0CD1 974 20$: JMP ERROR_ROUTINE ; Else, report the error
04 52 50 E1 00D7 975 30$: BBC R0,R2,40$ ; Should not branch
06 52 50 E0 00D8 976 BBS R0,R2,50$ ; Should branch
00000F29'EF 17 00DF 977 40$: JMP ERROR_ROUTINE ; Else, report the error
E0 50 1F F3 00E5 978 50$: AOBLEQ #31,R0,10$ ; Do for all 32 bit positions
00E9 979 ; End of subtest
00E9 980 ;***** End of test *****
```

```

00E9 982 .SBTTL TEST 4: BBSS, BBCS Test
00E9 983 ;++
00E9 984 ; Test description:
00E9 985 ;
00E9 986 ; In this test, the instructions BBSS and BBCS are functionally
00E9 987 ; verified.
00E9 988 ;
00E9 989 ; Assumptions:
00E9 990 ;
00E9 991 ; Test 1-Test 3
00E9 992 ;--
00E9 993 TEST_4:
00E9 994 ;++
00E9 995 ; Subtest 1: BBSS
00E9 996 ;--
00E9 997 T4_S1:
00E9 998 ; CLRL R3 ; Initialize position operand
00E9 999 ; CLRL R4 ; Initialize test operand
04 54 53 D4 00ED 1000 10$: BBSS R3,R4,20$ ; Should not branch
06 54 53 E2 00F1 1001 BBSS R3,R4,30$ ; Should branch
00000F29'EF 17 00F5 1002 20$: JMP ERROR_ROUTINE ; Else, report the error
EE 53 1F F3 00FB 1003 30$: AOBLEQ #31,R3,10$ ; Do until all 32 bits are set
00FF 1004 ; End of subtest
00FF 1005 ;++
00FF 1006 ; Subtest 2: BBCS
00FF 1007 ;--
00FF 1008 T4_S2:
00FF 1009 ; CLRL R3 ; Initialize position operand
00FF 1010 ; CLRL R4 ; Initialize test operand
06 54 53 D4 0101 1010 BBSS R3,R4,20$ ; Should branch
00000F29'EF 17 0107 1011 10$: BBSS R3,R4,30$ ; Should not branch
03 54 53 E3 0103 1012 20$: BBSS R3,R4,30$ ; Branch if okay
00000F29'EF 17 0114 1013 30$: JMP ERROR_ROUTINE ; Else, report the error
E5 53 1F F3 011A 1016 40$: AOBLEQ #31,R3,10$ ; Do until all 32 are set
011E 1017 ; End of subtest
011E 1018 ;***** End of test *****

```

```

011E 1020 .SBTTL TEST 5: BLBS, BLBC Test
011E 1021 ;++
011E 1022 ; Test description:
011E 1023 ;
011E 1024 ; In this test the instructions BLBS and BLBC are functionally verified.
011E 1025 ;
011E 1026 ; Assumptions:
011E 1027 ;
011E 1028 ; Test 1-Test 5
011E 1029 ;--
011E 1030 TEST_5:
011E 1031 ;++
011E 1032 ; Subtest 1: BLBS, BLBC
011E 1033 ;--
011E 1034 T5_S1:
50 D4 011E 1035 CLRL R0 ; Initialize test operand
51 D4 0120 1036 CLRL R1 ; Initialize test operand
52 01 00 0122 1037 MOVL #1,R2 ; Initialize test operand
53 FFFFFFFE 8F 00 0125 1038 MOVL #-2,R3 ; Initialize test operand
03 50 E8 012C 1039 BLBS R0,10$ ; Should not branch
06 50 E9 012F 1040 BLBC R0,20$ ; Should branch
00000F29'EF 17 0132 1041 10$: JMP ERROR_ROUTINE ; Else, report the error
03 52 E9 0138 1042 20$: BLBC R2,30$ ; Should not branch
06 52 E8 013B 1043 BLBS R2,40$ ; Should branch
00000F29'EF 17 013E 1044 30$: JMP ERROR_ROUTINE ; Else, report the error
C3 53 E8 0144 1045 40$: BLBS R3,50$ ; Should not branch
06 53 E9 0147 1046 BLBC R3,60$ ; Should branch
00000F29'EF 17 014A 1047 50$: JMP ERROR_ROUTINE ; Else, report the error
0150 1048 60$: ; End of subtest
0150 1049 ***** End of test *****

```

ZZ-ENKAZ-1.3
ENKAZ01
01-30

Diagnostic Subroutines

CRD HARDWARE INSTRUCTION TESTS
Diagnostic Subroutines

B 4
14-JUN-1984

Fiche 1 Frame B4

Sequence 40

14-JUN-1984 16:01:36

VAX-11 Macro V03-01

Page 28

14-JUN-1984 15:53:29

DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A(12)

```

                                0150 1051 .SBTTL Diagnostic Subroutines
                                0150 1052 :++
                                0150 1053 : Endpass
                                0150 1054 :
                                0150 1055 : Here is the endpass routine.
                                0150 1056 :
                                0150 1057 :--
                                0150 1058 END_PASS:
SO 00000213'EF 00000174'EF 16 0150 1059 JSB CLEAN_UP ; CLEANUP THE CPU
00000213'EF 00000213'EF D6 0156 1060 INCL PASS_COUNT ; Bump the pass count once
00000064 8F 7B 015C 1061 EDIV #100,PASS_COUNT,R0,R1 ; GET NUMBER OF 100 PASSES
                                0168
                                0169 1062 10$: TSTL R1 ; DONE 100 PASSES?
                                016B 1063 BNEQ 20$ ; BRANCH IF NO
                                016D 1064 BRW TEST_6
                                0170 1065 20$: JMP START
                                0174 1066
                                0174 1067 :++
                                0174 1068 : Cleanup
                                0174 1069 :--
                                0174 1070 CLEAN_UP:
                                12 1F DA 0174 1071 MTPR #31,#IPL ; SET IPL TO 31
                                05 0177 1072 RSB ; RETURN
                                0178 1073
                                0178 1074 .END
```

ZZ-ENKAZ-1.3 Symbol table
ENKAZ01
Symbol table

CRD HARDWARE INSTRUCTION TESTS

C 4
14-JUN-1984

Fiche 1 Frame C4

Sequence 41

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 29
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A(12)

BAD_DISK	*****	X	01
BOO\$C_COMBUFSZ	= 00000084		
BOO\$GL_FREEMEM	0000055E	R	01
BOO\$GI_COMBUF	000004DA	R	01
BOO\$READPROMPT	00000E00	RG	01
BOO\$TYPE_ASCIC	00000EB7	R	01
BUF	= 0000000C		
CLEAN_UP	00000174	R	03
CONTROL_U	= 00000015		
CPU_BAD	0000020F	R	01
CPU_ERR	00000F93	R	01
CR	= 0000000D		
DISK_ERR	00000F88	R	01
DRIVE_NFND	00000F74	R	01
DSS\$GB_UDASO_INIT	*****	X	01
END_PASS	00000150	R	03
ERROR_ROUTINE	00000F29	RG	01
ERR_END	00000F9C	R	01
HMBLK_ERR	*****	X	01
IDC_CSR	= 00F26200	G	
INIT1	00001220	R	01
INIT_DRVO	*****	X	01
INTERRUPT_STACK	00000A00	RG	01
IPL	= 00000012		
KERNEL_STACK	00000800	RG	01
LES1_UDA_PARAM	*****	X	01
LF	= 0000000A		
LOAD_BAD	00000F7E	R	01
LOAD_ERROR	0000020B	R	01
MCHK_IDC	000011F4	RG	01
MCHK_LES_UDA	00001208	RG	01
MCKADD	= 00000004		
NOCNTLR_FND	00000207	RG	01
NODRIVE_O_1	*****	X	01
NO_LOAD	00000F6A	RG	01
OUTBSLSH	00000E7E	R	01
OUTCHAR	00000E87	R	01
OUTEX	00000E91	R	01
OUTR8	00000E84	R	01
OUTZSTR	00000E92	R	01
PASS_COUNT	00000213	R	01
PR\$TLP	= 00000012		
PR\$ISP	= 00000004		
PR\$KSP	= 00000000		
PR\$MAPEN	= 00000038		
PR\$MCE\$R	= 00000026		
PR\$RXCS	= 00000020		
PR\$RXDB	= 00000021		
PR\$SCBB	= 00000011		
PR\$TXCS	= 00000022		
PR\$TXDB	= 00000023		
PRM_REV	= 00000001		
PRINTABORT	00000257	R	01
PRINTAUTO	0000045F	R	01
PRINTCPU	0000049F	R	01
PRINTDAAO	00000480	R	01
PRINTDISKERR	0000030F	R	01

PRINTDJA0	
PRINTDJA1	
PRINTDRVERR	
PRINTDUAO	
PRINTEND	
PRINTFAIL	
PRINTFIELD	
PRINTLOADMCK	
PRINTMENU	
PRINTNOCNTLR	
PRINTNODRIVE01	
PRINTNSPN6025	
PRINTNSPN8081	
PRINTPART	
PRINTPASS	
PRINTPROPER	
PRINTRAO	
PRINTRAS0	
PRINTRAS1	
PRINTRC25	
PRINTRET	
PRINTSPNEND	
PRINTTEND	
PRINTTEST	
PRINT_ROUTINE	
PROMPT	
PROMPT1	
PROMPT_MESS	
RA60_MESS-ST	
RA60-TEST	
RA8081_MESS-ST	
RA8081-TEST	
RA81-PRES	
RC25_MESS-ST	
RC25-TEST	
RFB_BUFF	
RUBOUT	
SCB	
SCB0	
SCB20	
SCB24	
SCB4	
SCB60	
SCB64	
SCB8	
SEC_REV	
SIZE	
SPIN_DWN	
SS\$NORMAL	
START	
T1-S1	
T1-S2	
T2-S1	
T2-S2	
T3-S1	
T4-S1	
T4-S2	

000004BC	R	01
000004C2	R	01
000002D0	R	01
00000486	R	01
00000267	R	01
00000245	R	01
00000232	R	01
0000031C	R	01
00000465	R	01
00000351	R	01
00000376	R	01
000003DB	R	01
00000427	R	01
0000046B	R	01
00000223	R	01
000003AD	R	01
000004C8	R	01
000004CE	R	01
000004D4	R	01
000004AA	R	01
00000293	R	01
00000456	R	01
00000285	R	01
00000475	R	01
00000F0E	R	01
= 00000004		
00000E9B	R	01
00000ED3	R	01
00000217	R	01
00000FE0	RG	01
0000021B	R	01
0000109E	RG	01
*****	X	01
0000021F	R	01
00001121	RG	01
00000C00	RG	01
= 0000007F		
00000600	RG	01
00000600	RG	01
00000620	RG	01
00000624	RG	01
00000604	RG	01
00000660	RG	01
00000664	RG	01
00000608	RG	01
= 00000003		
= 00C00008		
*****	X	01
= 00000001		
00000000	R	03
00000000	R	03
00000032	R	03
00000064	R	03
00000091	R	03
000000BE	R	03
000000E9	R	03
000000FF	R	03

ZZ-ENKAZ-1.3 Symbol table
ENKAZ01
Symbol table

CRD HARDWARE INSTRUCTION TESTS

D 4
14-JUN-1984

Fiche 1 Frame D4

Sequence 42

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 30
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A(12)

T5 S1	0000011E	R	03
TEST_1	00000000	R	03
TEST_2	00000064	R	03
TEST_3	000000BE	R	03
TEST_4	000000E9	R	03
TEST_5	0000011E	R	03
TEST_6	*****	X	03
TEST_8	*****	X	01
TOP_OF_ISTACK	00000BFC	RG	01
TOP_OF_KSTACK	000009FC	RG	01
TXCS	= 00000022		
TXDB	= 00000023		
UDA_LESI	*****	X	01
UDA_LESI_CSR	= 40FFF468	G	
UD_DRIVER1	*****	X	01
UNEX_E_I	000011C8	RG	01
V RUB	= 00000000		
WHAT_DRIVE	*****	X	01
WHAT_MOVE	00000203	RG	01

+-----+
! Psect synopsis !
+-----+

PSECT name

. ABS
A. DPM
FABS
A. HARDWARE

Allocation	PSECT No.	Attributes
00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
00001260 (4704.)	01 (1.)	PIC USR CON REL LCL NOSHR EXE RD WRT NOVEC PAGE
00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
00000178 (376.)	03 (3.)	PIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

+-----+
! Symbol Cross Reference !
+-----+

SYMBOL	VALUE	DEFINITION	REFERENCES...
BAD_DISK	00000000-XR		#-548 (4) #-629 (4) #-688 (4) #-724 (4)
BOO\$C_COMBUF\$Z	=00000084	219 (4)	#-449 (4)
BOO\$G_FREEEM	0000055E-R	259 (4)	#-476 (4)
BOO\$GT_COMBUF	000004DA-R	258 (4)	448 (4) 491 (4)
BOO\$READPROMPT	0000CE00-R	370 (4)	451 (4) 481 (4)
BOO\$TYPE_ASCII	00000EB7-R	473 (4)	#-492 (4)
BUF	=0000000C	360 (4)	#-377 (4)
CLEAN_UP	00000174-R	1070 (12)	1059 (12) 598 (4)
CONTROL_U	=00000015	327 (4)	#-402 (4)
CPU_BAD	0000020F-R	223 (4)	#-763 (4)
CPU_ERR	00000F93-R	572 (4)	#-550 (4)
CR	=0000000D	325 (4)	#-381 (4) #-409 (4)
DISK_ERR	00000F88-R	567 (4)	#-548 (4)
DRIVE_NFND	00000F74-R	556 (4)	#-544 (4)
DSSGB_UDASO_INIT	00000000-XR		#-493 (4)
END_PASS	00000150-R	1058 (12)	
ERROR_ROUTINE	00000F29-R	528 (4)	1002 (10) 1012 (10) 1015 (10) 1041 (11) 1044 (11) 1047 (11) 630 (4) 689 (4) 725 (4) 764 (4) 767 (4) 796 (4) 883 (7) 888 (7) 899 (7) 904 (7) 929 (8) 934 (8) 945 (8) 950 (8) 974 (9) 977 (9)
ERR_END	00000F9C-R	576 (4)	#-554 (4) #-560 (4) #-566 (4) #-571 (4)
EMBEK_ERR	00000000-XR		#-647 (4) #-736 (4)
IDC_CSR	=00F26200	217 (4)	
INITI	00001220-R	827 (6)	#-204 (4)
INIT_DRVO	00000000-XR		#-668 (4) #-710 (4)
INTERRUPT_STACK	00000A00-R	281 (4)	
IPL	=00000012	211 (4)	#-1071 (12)
KERNEL_STACK	00000800-R	276 (4)	
LES1_UDA_PARAM	00000000-XR		#-692 (4)
LI	=0000000A	326 (4)	#-411 (4)
LOAD_BAD	00000F7E-R	562 (4)	#-546 (4)
LOAD_ERROR	0000020B-R	222 (4)	#-546 (4) #-766 (4)
MCHK_IDC	000011F4-R	778 (4)	
MCHK_LES_UDA	00001208-R	792 (4)	
MCKADD	=00000004	212 (4)	
NOCTRL_END	00000207-R	221 (4)	#-542 (4) #-795 (4)
NODRIVE_Q_1	00000000-XR		#-544 (4)
NO_LOAD	00000F6A-R	551 (4)	#-542 (4)
OUTBSL\$H	00000E7E-R	418 (4)	#-391 (4) #-401 (4)
OUTCHAR	00000E87-R	422 (4)	#-382 (4) #-412 (4) #-420 (4) #-424 (4)
			#-430 (4)
OUTEX	00000E91-R	426 (4)	#-429 (4)
OUTR8	00000E84-R	421 (4)	#-397 (4) #-407 (4)
OUTZSTR	00000E92-R	427 (4)	#-373 (4) #-431 (4)
PASS_COUNT	00000213-R	224 (4)	#-1060 (12) #-1061 (12) #-761 (4)
PR\$TPI	=00000012		#-848 (6)
PR\$ISP	=00000004		#-833 (6)
PR\$KSP	=00000000		#-836 (6)

PRS_MAPEN	=00000038		#-827	(6)				
PRS_MCESR	=00000026		#-759	(4)	#-779	(4)	#-793	(4)
PRS_RXCS	=00000020		#-384	(4)				
PRS_RXDB	=00000021		#-386	(4)				
PRS_SCBB	=00000011		#-840	(6)				
PRS_TXCS	=00000022		#-423	(4)				
PRS_TXDB	=00000023		#-425	(4)				
PRM_REV	=00000001	65	(3)					
PRINTABORT	00000257-R	231	(4)	539	(4)			
PRINTAUTO	0000045F-R	244	(4)	533	(4)	585	(4)	
PRINTCPU	0000049F-R	248	(4)	573	(4)			
PRINTDAAG	000004B0-R	250	(4)	720	(4)	732	(4)	
PRINTDISKERR	0000030F-R	236	(4)	568	(4)			
PRINTDJA0	000004BC-R	252	(4)	625	(4)	643	(4)	
PRINTDJA1	000004C2-R	253	(4)	622	(4)	640	(4)	
PRINTDRVERR	000002D0-R	235	(4)	648	(4)	737	(4)	
PRINTDUAO	000004B6-R	251	(4)	684	(4)			
PRINTEND	00000267-R	232	(4)	580	(4)			
PRINTFALL	00000245-R	230	(4)	531	(4)			
PRINTFIELD	00000232-R	229	(4)	577	(4)	650	(4)	739 (4)
PRINTLOADMCK	0000031C-R	237	(4)	563	(4)			
PRINTMENU	00000465-R	245	(4)	536	(4)	589	(4)	
PRINTNOCNTLR	00000351-R	238	(4)	552	(4)			
PRINTNODRIVE01	00000376-R	239	(4)	557	(4)			
PRINTNSPN6025	000003DB-R	241	(4)	635	(4)	730	(4)	
PRINTNSPN8081	00000427-R	242	(4)	695	(4)			
PRINTPART	0000046B-R	246	(4)	619	(4)	682	(4)	718 (4)
PRINTPASS	00000223-R	228	(4)	614	(4)	671	(4)	713 (4)
PRINTPROPER	000003AD-R	240	(4)	633	(4)	693	(4)	728 (4)
PRINTRA60	000004C8-R	254	(4)	617	(4)			
PRINTRAS0	000004CE-R	255	(4)	680	(4)			
PRINTRA81	000004D4-R	256	(4)	676	(4)			
PRINTRC25	000004AA-R	249	(4)	716	(4)			
PRINTRET	00000293-R	234	(4)	595	(4)			
PRINTSPNEND	00000456-R	243	(4)	645	(4)	652	(4)	734 (4) 741 (4)
PRINTTEND	00000285-R	233	(4)	592	(4)			
PRINTTEST	00000475-R	247	(4)	627	(4)	686	(4)	722 (4)
PRINT_ROUTINE	00000FOE-R	505	(4)	532	(4)	534	(4)	537 (4) 540 (4)
				553	(4)	559	(4)	565 (4) 570 (4)
				574	(4)	578	(4)	581 (4) 586 (4)
				590	(4)	593	(4)	596 (4) 615 (4)
				618	(4)	620	(4)	623 (4) 626 (4)
				628	(4)	634	(4)	636 (4) 641 (4)
				644	(4)	646	(4)	649 (4) 651 (4)
				653	(4)	672	(4)	677 (4) 681 (4)
				683	(4)	685	(4)	687 (4) 694 (4)
				696	(4)	714	(4)	717 (4) 719 (4)
				721	(4)	723	(4)	729 (4) 731 (4)
				733	(4)	735	(4)	738 (4) 740 (4)
				742	(4)			
PROMPT	=00000004	352	(4)	#-372	(4)			
PROMPT1	00000E9B-R	446	(4)	#-489	(4)			
PROMPT_MESS	00000ED3-R	488	(4)	654	(4)	697	(4)	743 (4)
RAND_MESS_ST	00000217-R	225	(4)	#-613	(4)	#-616	(4)	
RAND_TEST	00000FE0-R	611	(4)					
REBOR1_MESS_ST	00000218-R	226	(4)	#-670	(4)	#-673	(4)	
REBOR1_TEST	0000109F-R	666	(4)					

#-726 (4)

#-667 (4)
#-612 (4)

-----+
! Macros Cross Reference !
-----+

MACRO	SIZE	DEFINITION	REFERENCES...				
\$DEFINE	1	313 (4)	313 (4)	314	(4)	315	(4)
\$PP/3ODEF	1	314 (4)	314 (4)				
\$PRIEF	4	313 (4)	313 (4)				
\$SSDEF	21	315 (4)	315 (4)				
ABORT MESSAGE	1	89 (4)	231 (4)				
AUTO MESSAGE	1	152 (4)	244 (4)				
CPU MESSAGE	1	93 (4)	248 (4)				
DAAG MESSAGE	1	184 (4)	250 (4)				
DISKERR MESSAGE	1	113 (4)	236 (4)				
DJAC MESSAGE	1	188 (4)	252 (4)				
DIAL MESSAGE	1	192 (4)	253 (4)				
DRVERR MESSAGE	1	139 (4)	235 (4)				
DUAG MESSAGE	1	196 (4)	251 (4)				
END MESSAGE	1	117 (4)	232 (4)				
FAIL MESSAGE	1	85 (4)	230 (4)				
FIELD MESSAGE	1	97 (4)	229 (4)				
LOADMCK MESSAGE	1	109 (4)	237 (4)				
MENU MESSAGE	1	156 (4)	245 (4)				
NOCNTLR MESSAGE	1	101 (4)	238 (4)				
NODRIVE01 MESSAGE	1	105 (4)	239 (4)				
NSPN6025 MESSAGE	1	134 (4)	241 (4)				
NSPN8081 MESSAGE	1	148 (4)	242 (4)				
PART MESSAGE	1	164 (4)	246 (4)				
FAST MESSAGE	1	81 (4)	228 (4)				
PROPER MESSAGE	1	130 (4)	240 (4)				
RA60 MESSAGE	1	172 (4)	254 (4)				
RA80 MESSAGE	1	176 (4)	255 (4)				
RA81 MESSAGE	1	180 (4)	256 (4)				
RC25 MESSAGE	1	168 (4)	249 (4)				
RET MESSAGE	1	125 (4)	234 (4)				
SPNEND MESSAGE	1	144 (4)	243 (4)				
TEND MESSAGE	1	121 (4)	233 (4)				
TEST MESSAGE	1	160 (4)	247 (4)				

-----+
! Performance indicators !
-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	112	00:00:00.44	00:00:04.21
Command processing	110	00:00:00.40	00:00:02.94
Pass 1	814	00:00:10.00	00:01:14.44
Symbol table sort	6	00:00:01.08	00:00:12.34
Pass 2	499	00:00:03.33	00:00:32.67
Symbol table output	18	00:00:00.13	00:00:01.27
Print synopsis output	6	00:00:00.03	00:00:00.03
Cross-reference output	132	00:00:00.90	00:00:08.37
Assembler run totals	1701	00:00:16.33	00:02:16.45

22-ENKAZ-1.3 Cross reference

ENKAZ01

CRD HARDWARE INSTRUCTION TESTS

14-JUN-1984

Fiche 1 Frame 14

Sequence 47

14-JUN-1984 16:01:36 VAX-11 Macro V03-01 Page 35
14-JUN-1984 15:53:29 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ01A(12)

The working set limit was 500 pages.

72173 bytes (141 pages) of virtual memory were used to buffer the intermediate code.

There were 40 pages of symbol table space allocated to hold 620 non-local and 69 local symbols.

1074 source lines were read in Pass 1, producing 21 object records in Pass 2.

48 pages of virtual memory were used to define 38 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

SYSSYSROOT:[SYSLIB]STARLET.MLB;1

6

568 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

MACRO/LIST/CROSS/OBJECT ENKAZ01A

Table of contents

(2)	54	DECLARATIONS
(2)	92	UDA50 Bootstrap device initialization
(3)	210	UDA50 Bootstrap driver QIO
(3)	300	TEST 6 DETERMINING CRD BASE CONTROLLER
(3)	373	TEST 7 SIZE CRD DISK OPTIONS AND CHECK DRIVE STATUS
(3)	441	TEST 8 RA60 AND RC25 VERIFY PACK LABEL CRDPACK

ZZ-ENKAZ-1.3
ENKAZ02
01-30

- UDA50, LES1, IDC SIZING TESTS
- UDA50, LES1, IDC SIZING TESTS

K 4
14-JUN-1984

Fiche 1 Frame K4

Sequence 49

14-JUN-1984 16:04:00 VAX-11 Macro V03-01 Page 1
14-JUN-1984 13:28:26 DRB0:[BAIL.ENKAZ.ENKAZ_1]ENKAZC2A.(1)

```
0000 1 .TITLE ENKAZ02 - UDA50, LES1, IDC SIZING TESTS
0000 2 .IDENT '01-30'
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1981, 1983 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28
0000 29 **
0000 30 FACILITY: BOOTS
0000 31
0000 32 ABSTRACT:
0000 33 This module contains the bootstrap device driver for the
0000 34 UDA 50 and RCF25/RC25 disks. It will perform some sizing
0000 35 for supported CRD 11/730 disk options and read the label
0000 36 off RA60 and RC25 disks.
0000 37
0000 38 ENVIRONMENT: IPL 31, kernel mode, code must be PIC
0000 39
0000 40 AUTHOR: Kerbey T. Altmann, CREATION DATE: 20-Nov-1981
0000 41
0000 42 MODIFIED BY:
0000 43
0000 44 Peter Green 1-April-1983
0000 45 Modified to run with ENKAZ level 4 program in CRD package.
0000 46
0000 47 Darryl L. Ball 14-June-1984
0000 48
0000 49 Added 730/725 Processor register definitions. Also modified existing
0000 50 boot routine to Boot the system disk if no RL02 is present.
0000 51
0000 52 --
```

```

0000 54 .SBTTL DECLARATIONS
0000 55
0000 56 : INCLUDE FILES:
0000 57 :
0000 58
0000 59 $BTDEF ; Boot device types
0000 60 $IODEF ; I/O function codes
0000 61 $MSCPDEF ; MSCP definitions
0000 62 $PRDEF ; Processor registers
0000 63 $PR730DEF ; 730/725 Processor registers
0000 64 $PTEDEF ; Page table entries
0000 65 $RPBDEF ; RPB offsets
0000 66 $SSDEF ; Status codes
0000 67 $UBADEF ; UBA definitions
0000 68 $UBIDEF ; 11/750 UBA definitions
0000 69 $VADEF ; Virtual addresses
0000 70
0000 71
0000 72 : EQUATED SYMBOLS:
0000 73 :
00000000 0000 74 UDAIP = 0
00000002 0000 75 UDASA = 2
00000001 0000 76 GO = 1
00008000 0000 77 OWN = 1015
0000000B 0000 78 S1 = 11
0000000E 0000 79 S4 = 14
000001EE 0000 80 UMR = 494 ; Last-1 UNIBUS Mapping Register
0000 81
0000 82 :
0000 83 : OWN STORAGE:
0000 84 :
0000 85 :
0000 86 :
0000 87 : Boot driver table entry
0000 88 :
0000 89
00000000 90 .Psect SIZING , page ; Define psect allocation
0000 91
0000 92 .SBTTL UDA50 Bootstrap device initialization
0000 93
0000 94 :++
0000 95 :
0000 96 : Inputs:
0000 97 :
0000 98 : R9 --> RPB
0000 99 :
0000 100 : Outputs:
0000 101 :
0000 102 : R0 - status code
0000 103 :
0000 104 :--
0000 105
01FC 0000 106 .Entry UD_INIT, *M<R2,R3,R4,R5,R6,R7,R8>
0002 107
0002 108 .ENABLE LSB
0002 109
50 38 DB 0002 110 MFPR #PR$_MAPEN, R0 ; Get the mapping status

```

```

0005 111 ;
0005 112 ; Set up a UNIBUS mapping register(s) to cover the ring and buffers.
0005 113 ; To make things easy, we will grab the last two register (494 & 495).
0005 114 ; These registers are necessary since the ring area will be accessed by
0005 115 ; the controller which is a UNIBUS device that does not do any mapping.
0005 116 ;
56 0003DC00 8F DO 0005 117 ; Set up a constant
52 0200'CF 9E 000C 118 ; Get the address of the ring
51 52 77'AF CB 0011 119 ; Get the byte offset in page
02 A2 51 50 C9 0016 120 ; Set in the UNIBUS addr
52 52 02 A2 10' C0 0018 121 ; Make it the ring address
52 52 15 09 EF 001F 122 ; Get the page frame
0024 123 ;
53 5C A940 DO 0024 124 ; Get correct pointer to UBA reg
0029 125 ;
57 54 A940 DO 0029 126 ; Get correct address of device CSR
0C 50 E9 002E 127 ; If clr, then physical
52 50 B942 DO 0031 128 ; Virtual, get physical
52 FFE00000 8F CA 0036 129 ; Now a physical PFN
54 0F88 C3 DE 003D 130 20$: MOVAL UBA$ MAP<UMR*4>(R3),R4 ; Get the last two UMR's
84 52 AC'AF C9 0042 131 ; Set as valid w/PFN
52 52 D6 0047 132 ; Set next page just in case
64 52 AC'AF C9 0049 133 ; Set as valid w/PFN
004E 134 ;
004E 135 ; Now go thru the ridiculously complicated startup sequence. This is a
004E 136 ; fugue in four parts.
004E 137 ;
53 58 02 DO 004E 138 ; Make two tries at this
52 0200'CF 9E 0051 139 RETRY: MOVAB W^INTTBL, R3
50 0B DO 0056 140 ; Step flag
67 B4 0059 141 ; Poke the controller's CSR
005B 142 ;
005B 143 ; Wait 100 microseconds.
005B 144 ;
52 52 1B DB 005B 145 TIME: MFPR #PR730$ TODR, R2 ; Current time
54 03E8 C2 9E 005E 146 ; Set for 10 seconds later
02 A7 B0 0063 147 LOOP: MOVW UDASA(R7),R4 ; Check the status register
39 19 0067 148 ; Bit 15 set is the error indicator
0E 54 50 E0 0069 149 ; Done with this step?
51 1B DB 006D 150 ; No, pick up time
52 51 D1 0070 151 ; Are we past due time?
2D 1A 0073 152 ; Yep, error
EC 11 0075 153 ; No, try again
0077 154 ;
0077 155 BYTE_OFF: .LONG ^C<^X1FF> ; Mask for byte offset in page
0078 156 ;
02 A7 83 B0 007B 157 30$: MOVW (R3)+,UDASA(R7) ; Send the controller the next step
DB 50 0E F3 007F 158 ; Set for next step
0083 159 ;
0083 160 ; Initialization complete. Write the packet address in the ring.
0083 161 ;
0083 162 ;
51 0250'CF 9E 0083 163 ;
51 EC AF CA 0088 164 ; Get the address of response packet
0210'CF 51 56 C9 008C 165 ; Set byte offset in ring desc
51 021C'CF 9E 0092 166 ; Get the address of command packet
MOVAB W^RSPPKT, R1
BICL BYTE_OFF, R1
BISL3 R6, R1, W^RD
MOVAB W^CMDPKT, R1

```

```

0214'CF 51 DD AF CA 0097 168 BICL BYTE OFF, R1
51 56 C9 009B 169 BISL3 R6, R1, W^CD ; Set byte offset in ring desc
04 00A1 170 RET
00A2 171
00A2 172 ERROR: SOBGTR R8, RETRY ; Try once again
50 AC 58 F5 00A2 173 MOVZWL #SS$_CTRLERR, R0 ; Set failure status
0054 8F 3C 00A5 173 RET
04 00AA 174
00AB 175
00AB 176 .DISABLE LSB
000000AC 00AB 177 .=<. +1>8-2
80000000 00AC 178 VALID: .LONG ^X80000000 ; Sign bit set
00B0 179
00B0 180 Ds$Gb_Uda50_Init:: ; Global label
00B0 181 Init: ; convenient local label
00 00B0 182 .Byte 0 ; .. init flag
00B1 183 ;
00B1 184 ; RINGS
00B1 185 ;
00B1 186 ;
00B1 187 ;
00B1 188 .Align Page
0200 189
8000 0200 190 INTTBL: .WORD OWN ; Step 1 pattern
000C0000 0202 191 .LONG 0 ; Step 2 & 3 pattern
0001 0206 192 .WORD 60
0208 193
0000 0000 0208 194 .WORD 0,0 ; Reserved
0000 020C 195 CMDINT: .WORD 0 ; Command status word
0000 020E 196 RSPINT: .WORD 0 ; Response status word
0210 197 RING:
00000000 0210 198 RD: .LONG 0 ; UNIBUS address of response ring
00000000 0214 199 CD: .LONG 0 ; UNIBUS address of command ring
0218 200 ;
0030 0218 201 .WORD 48 ; Length of message
0001 021A 202 .WORD 1 ; ID
0001 021C 203 CMDPKT: .WORD 1
0000024C 021E 204 .BLKW 23 ; Full envelope
024C 205 ;
00000250 024C 206 .BLKW 2
0250 207 UDA_Response:: ; Global label for CONFIG
00000280 0250 208 RSPPKT: .BLKW 24

```

```

0280 210 .SBTTL UDA50 Bootstrap driver Q10
0280 211
0280 212
0280 213
0280 214 Inputs:
0280 215
0280 216 R3 - base address of adapter's register space
0280 217 R5 - lbn for current piece of transfer
0280 218 R6 - contains 0
0280 219 R7 - address of the device's CSR
0280 220 R8 - size of transfer in bytes
0280 221 R9 - address of the RPB
0280 222 R10 - starting address of transfer (byte offset in first
0280 223 page ORed with starting map register number)
0280 224 R11 - LBN at start of transfer
0280 225
0280 226 FUNC(AP)- I/O operation (IO$ READBLK)
0280 227 SIZE(AP)- Size of transfer in bytes
0280 228 MODE(AP)- Address interpretation mode (0 = physical, 1 = virtual)
0280 229
0280 230 Implicit inputs:
0280 231
0280 232 RPB$W_UNIT - RPB field containing boot device unit number
0280 233
0280 234 Outputs:
0280 235
0280 236 R0 - status code
0280 237
0280 238 SS$ _NORMAL - successful transfer
0280 239 SS$ _NOSUCHDEV - unsupported device
0280 240 SS$ _CTRLERR - fatal controller error
0280 241
0280 242 R3 - must be preserved
0280 243
0280 244
0280 245
0280 246
00000010 0280 247 FUNC = 16
00000014 0280 248 MODE = 20
0280 249
50445243 0280 250 CRDPACK1 = ^X50445243 ; CRDP
204B4341 0280 251 CRDPACK2 = ^X204B4341 ; ACK_
0280 252
00000000 0280 253 WHAT_DRIVE:: .LONG 0
00 0284 254 RAB1_PRES:: .BYTE 0
00 0285 255 RC25_PRES:: .BYTE 0
00 0286 256 NODRIVE_0_1:: .BYTE 0
00 0287 257 HMBLK_ERR:: .BYTE 0
00 0288 258 BAD_DISK:: .BYTE 0
00 0289 259 SPIN_DWN:: .BYTE 0
028A 260
028A 261
028A 262
028A 263
00000081 028A 264 C_D_RDY ==^X81 ; Controller and Drive ready bits
00000100 028A 265 D51 ==^X100 ; Drive 1 select
00000000 028A 266 IDC_HLD:: .LONG 0 ; Temporary storage

```

ZZ-ENKAZ-1.3
ENKAZ02
01-30

UDA50 Bootstrap driver Q10
- UDA50, LESI, IDC SIZING TESTS
UDA50 Bootstrap driver Q10

C 5
14-JUN-1984

Fiche 1 Frame C5

Sequence 54

14-JUN-1984 16:04:00 VAX-11 Macro V03-01
14-JUN-1984 13:28:26 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

Page 6

```
00000000 028E 267 IDC_TST:: .LONG 0 ; Temporary storage
0292 268
0292 269 ;
0292 270 ;
0292 271 ;
01FC 0292 272 .ENTRY UD_DRIVER1, ^M<R2,R3,R4,R5,R6,R7,R8> ; UDA50 device driver.
0294 273
0294 274 ;
0294 275 ; Start out by re-initing the UDA if it hasn't been inited yet,
0294 276 ; or if there was an error previously.
0294 277 ;
00000000'EF 00000000'EF DE 0294 278 MOVAL UNEX_E_1,SCB4 ; UNEXPECTED MACHINE CHECK HANDLER
59 00000000'EF 9E 029F 279 MOVAB RPB_BUFF,R9 ; ADDRESS OF RPB
53 00F26000 8F D0 02A6 280 MOVL #<^XF26000>,R3 ; BASE ADDRESS OF 11/730 UNIBUS
02AD 281 ; ADAPTER
5C A9 53 D0 02AD 282 MOVL R3,RPB$_ADPPHY(R9) ; SET ADAPTER ADDRESS
60 A9 53 D0 02B1 283 MOVL R3,RPB$_ADPVIR(R9) ; SET ADAPTER ADDRESS
02B5 284
57 40FFF468 8F D0 02B5 285 MOVL #<^X40FFF468>,R7 ; FIXED ADDRESS OF UDA50
02B5 286 ; AND LESI CONTROLLERS
54 A9 57 D0 02BC 287 MOVL R7,RPB$_CSRPHY(R9) ; SET CSR ADDRESS
58 A9 57 D0 02C0 288 MOVL R7,RPB$_CSRVR(R9) ; SET CSR ADDRESS
02C4 289
03 FDE8 CF E9 02C4 290 Blbc W^Init, 3$ ; Branch if already set
00E0 31 02C9 291 BRW TEST 7
FD2F CF 00 FB 02C9 292 3$: CALLS #0, Od_Init ; Call init code
0A 50 E8 02D1 293 Blbs R0, 5$
00000000'EF 01 D0 02D4 294 MOVL #1,NOCNTRLR_FND
FD22' 31 02DB 295 BRW ERROR_ROUTINE ; Exit if error
FDCD CF 01 90 02DE 296 5$: MovB #1, W^Init ; Set init byte
02E3 297
02E3 298
```

```

02E3 300 .SBTTL TEST 6 DETERMINING CRD BASE CONTROLLER
02E3 301 :++
02E3 302 : TEST DESCRIPTION:
02E3 303 :
02E3 304 : THIS TEST DETERMINES THE BASE SUPPORTED CRD CONTROLLER.
02E3 305 :
02E3 306 :--
02E3 307 :
02E3 308 TEST_6::
00000000'EF 00000000'EF DE 02E3 309 MOVAL MCHK_IDC,SCB4 ; SET UP MACHINE CHECK
02EE 310 ; HANDLER FOR IDC
02E3 311 IDC:
00000000'9F 00000100 8F D0 02EE 312 MOVL #DS1,@#IDC_CSR ; Drive 1 select
0000028A'9F 00000000'9F D0 02F9 313 MOVL @#IDC_CSR,@#IDC_HLD ; Move status to temp storage
0000028E'9F 0000028A'9F 9A 0304 314 MOVZBL @#IDC_HLD,@#IDC_TST ; get rid of unnecessary bits
0000028E'9F 81 8F 91 030F 315 CMPB #C_D_RDY,@#IDC_TST ; Controller and Drive ready set?
03 12 0317 316 BNEG 1$ ; No goto 1$
C040 31 0319 317 BRW IDC_PARAM_1 ; Yes Boot RL02
0065 31 031C 318 1$: BRW IDC_PARAM_0 ; No Boot R80 System Disk
031F 319
031F 320 UDA_LESI::
00000000'EF 00000000'EF 9E 031F 321 MOVAB MCHK_LES_UDA,SCB4 ; SET UP MACHINE CHECK
032A 322 ; HANDLER FOR KLESI AND
032A 323 ; UDA50
032A 324
50 00000000'9F B0 032A 325 MOVW @#UDA_LESI_CSR,R0 ; SET UP CSR OF KLESI AND
0331 326 ; UDA50 ON 730 UNIBUS
0331 327 ; IF EITHER KLESI OR
0331 328 ; UDA50 THERE....
0331 329
FF5C CF 00 FB 0331 330 CALLS #0,UD_DRIVER1 ; CALL UDA50 & LESI
0336 331 ; INITIALIZATION AND
0336 332 ; STATUS
0336 333
0336 334 LESI_UDA_PARAM::
50 11 D0 0336 335 ; BOOT UDA OR LESI
51 03 D0 0336 336 ; DRIVE TO BE DETERMINED
0003F468 8F D0 0336 337 MOVL #X11,R0 ; D/G/L 0 11
53 FF39 CF D0 0339 338 MOVL #3,R1 ; D/G 1 3
54 D4 0348 340 MOVL #X03F468,R2 ; D/G 2 3F468
00000000'EF D0 0343 339 MOVL WHAT_DRIVE,R3 ; R3 = DRIVE 0 OR 1
5D D4 034A 341 CLRI R4 ; D/G 4 0
5E 00000200 8F D0 0351 342 MOVI WHAT_MODE,R5 ; R5 = CRD MODE
6E 17 D0 0353 343 CLRL FP ; D/G D 0
035A 344 MOVL #X200,SP ; D/G E 200
035C 345 JMP (SP)
035C 346
035C 347 IDC_PARAM_1::
50 00A80003 8F D0 035C 348 MOVL #X0A80003,R0 ; BOOT IDC RL02 CRDPACK
51 03 D0 0363 350 MOVL #3,R1 ; DRIVE 1 - SQ1B00.CMD
52 00FFFB86 8F D0 0366 351 MOVL #X0FFFB86,R2 ; D/G/L 0 00A80003
53 01 D0 036D 352 MOVL #1,R3 ; D/G 1 3
54 D4 0370 353 CLRL R4 ; D/G 2 FFFB86
55 00000000'EF D0 0372 354 MOVL WHAT_MODE,R5 ; D/G 3 1
5D D4 0379 355 CLRL FP ; D/G 4 0
5E 00000200 8F D0 037B 356 MOVL #X200,SP ; D/G 5 8010
; D/G D 0
; D/G E 200

```

22-ENKAZ-1.3
ENKAZ02
01-30

TEST 6 DETERMINING CRD BASE CONTROLLER
- UDA50, LESI, IDC SIZING TESTS
TEST 6 DETERMINING CRD BASE CONTROLLER

E 5
14-JUN-1984

Fiche 1 Frame E5

Sequence 56

14-JUN-1984 16:04:00 VAX-11 Macro V03-01 Page 8
14-JUN-1984 13:28:26 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

	6E	17	0382	357	JMP	(SP)	
			0384	358			
			0384	359	IDC_PARAM_0::		: BOOT IDC R80 CRDPACK
			0384	360			: DRIVE 0 - SQ0B00.CMD
50	00A80003	8F	D0	0384	361	MOVL	: D/G/L 0 00A80003
	51	03	D0	038B	362	MOVL	: D/G 1 3
52	00FFFB86	8F	D0	038E	363	MOVL	: D/G 2 FFB86
	53	00	D0	0395	364	MOVL	: D/G 3 0 Drive No. '0'
		54	D4	0398	365	CLRL	: D/G 4 0
55	00000000	EF	D0	039A	366	MOVL	: D/G 5 8010 Crd Mode
		5D	D4	03A1	367	CLRL	: D/G D 0
5E	00000200	8F	D0	03A3	368	MOVL	: D/G E 200
		6E	17	03AA	369	JMP	
				03AC	370		
				03AC	371		

```

373 .SBITL TEST 7 SIZE CRD DISK OPTIONS AND CHECK DRIVE STATUS
374 :++
375 : TEST DESCRIPTION:
376 :
377 : Bring the device on-line and check status
378 :
379 :--
380 TEST_7::
381     MOVL    #1,RPB$W_UNIT(R9)        ; TEST UNIT 1 FIRST
382     MOVL    #1,WHAT_DRIVE
383 DRIVE0::
384     CLRB    NODRIVE_0_1
385     CLRB    SPIN_DWN
386     CLRB    BAD_DISK
387     MOVAB    W^CMDPKT, R5              ; Get the address of command packet
388     MOVL    #1,(R5)+                  ; Set command ref number
389     MOVZBL    RPB$W_UNIT(R9),(R5)+    ; Put unit number in cmd packet field
390     MOVZBL    #MSCP$K_OP_ONLIN,(R5)+ ; Set opcode to bring drive online
391     CLRQ    (R5)+                     ; Clear byte count, buff desc
392     CLRQ    (R5)                      ; buff desc and LBN
393     BSBW    IO                        ; Send it out
394
395     CLRL    R2
396     MOVW    RSPPKT+MSCP$W_STATUS,R2  ; CHECK STATUS
397     CMPB    R2,MSCP$K_ST_SUCC        ; IF SUCCESS
398     BEQL    NORMAL
399     CMPW    #^X23,R2                 ; IF NOT SPINNING
400     BNEQ    7$
401     MOVB    #1,SPIN_DWN               ; SET A BIT TO TST LATER
402     BRB     NORMAL
403
404 7$:    MOVB    #1,BAD_DISK            ; DRIVE OR CONTROLLER
405         ; ERROR
406
407 NORMAL:
408     CLRL    R2
409     MOVL    RSPPKT+MSCP$Q_UNIT_ID+8,R2 ; UDA50 DISK TYPE
410     CMPL    #^X22A4103C,R2           ; RA60 ?
411     BNEQ    20$
412     BRW     RA60_TEST                 ; IF YES PRINT MESSAGE
413
414 20$:    CMPL    #^X25641050,R2        ; RA80 ?
415     BNEQ    30$
416     BRW     RA8081_TEST
417
418 30$:    CMPL    #^X25641051,R2        ; RA81 ?
419     BNEQ    40$
420     MOVB    #1,RA81 PRES
421     BRW     RA8081_TEST
422
423 40$:    CMPL    #^X20643019,R2        ; RC25 ?
424     BNEQ    50$
425     MOVB    #1,RC25 PRES
426     BRW     RC25_TEST
427
428 50$:    TSTL    WHAT_DRIVE
429         BNEQ    INIT_DRV0            ; NO DRIVE 1 TRY DRIVE 0

```

22-ENKAZ-1.3
ENKAZ02
01-30

TEST 7 SIZE CRD DISK OPTIONS AND CHECK D
- UDA50, LESI, IDC SIZING TESTS
TEST 7 SIZE CRD DISK OPTIONS AND CHECK D

G 5
14-JUN-1984

Fiche 1 Frame G5

Sequence 58

Page 10

14-JUN-1984 16:04:00 VAX-11 Macro V03-01
14-JUN-1984 13:28:26 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

FE42 CF	01	90	043F	430	MOVB	#1,NODRIVE 0 1	; NO SUPPORTED DRIVE OPTION FOUND
	FBB9	31	0444	431	BRW	ERROR_ROUTINE	; DRIVE 0 OR 1
			0447	432			
			0447	433			
			0447	434	INIT_DRVO::		
64 A9	00	D0	0447	435	MOVL	#0,RPBSW UNIT(R9)	; TEST UNIT 0
FE30 CF	00	D0	0448	436	MOVL	#0,WHAT_DRIVE	
	FF62	31	0450	437	BRW	DRIVE0	; NOT A CRD SUPPORTED
			0453	438			; DISK OPTION TRY DRIVE 0
			0453	439			

```
0453 441 .SBTTL TEST 8 RA60 AND RC25 VERIFY PACK LABEL CRDPACK
0453 442 :++
0453 443 : TEST DESCRIPTION:
0453 444 :
0453 445 : READS RA60 (DRIVE 0 OR 1) AND RC25 (DRIVE 0) HOME BLOCK TO DETERMINE IF THE
0453 446 : PACK LABEL IS CORRECT AND THE HOME BLOCK IS VALID.
0453 447 :
0453 448 :--
0453 449 .ALIGN PAGE
0600 450
00000000 0600 451 HMBLK_BUFF: .LONG 0
00000000 0604 452 VALID1: .LONG 0
00000000 0608 453 VALID2: .LONG 0
000007D8 060C 454 .BLKL 115
00000000 07D8 455 CRDP: .LONG 0
00000000 07DC 456 ACK_: .LONG 0
00000800 07E0 457 .BLKL 8
0800 458
0800 459 TEST_8::
10 AC 53 00F26000 8F D0 0800 460 CLR B HMBLK_ERR
58 55 01 D0 0804 461 CLR L R1
10 AC 58 00000200 8F D0 0806 462 MOVL #<^XF26000>,R3
10 AC 00000021 EF D0 080D 463 MOVL #1,R5 ; LBN = 1
FA00 CF 21 90 0810 464 MOVL #512,R8 ; BYTE COUNT = 512
FA0F CF 55 D0 0817 465 MOVL IO$ READLBLK,FUNC(AP) ; IO OPERATION
F9FA CF 58 D0 081F 466 MOV B #MSCP$K OP READ, - ; READ COMMAND
0824 467 CMDPKT+MSCP$B_OPCODE
0824 468 MOVL R5,CMDPKT+MSCP$L_LBN ; SET THE LOGICAL BLOCK NUMBER
F9FA CF 58 D0 0829 469 MOVL R8,CMDPKT+MSCP$L_BYTE_CNT ; SET THE BYTE COUNT
082E 470
082E 471 : SETTING UP 2 UNIBUS MAPPING REGISTERS AND HOME BLOCK BUFFER
082E 472 :
51 5A FDCE CF DE 082E 473 MOVAL HMBLK_BUFF,R10 ; ADDRESS OF TRANSFER BUFFER
51 5A 15 09 EF 0833 474 EXTZV #VAS$VPN,#VASS_VPN,R10,R1 ; GET PFN
51 54 0E40 C3 DE 0838 475 MOVAL UBAS$MAP+<400*4>(R3),R4 ; POINT TO UNIBUS MAP
84 51 80000000 8F C9 083D 476 ; REGISTER 400
51 51 D6 083D 477 BISL3 #UBAS$BRRVR_AIR,R1,(R4)+ ; SET AS VALID W/PFN
64 51 80000000 8F C9 0847 478 INCL R1 ; NEXT PFN
5A 15 09 00000190 8F F0 084F 479 BISL3 #UBAS$BRRVR_AIR,R1,(R4) ; SET AS VALID W/PFN
F9CF CF 5A D0 0858 480 INSV #400,#VAS$VPN,#VASS_VPN,R10 ; SET MAP REGISTER TO US.
43 10 085D 481 MOVL R10,CMDPKT+MSCP$B_BUFFER ; SET THE UNIBUS MAP REGISTER
085D 482
085F 483 BSBB IO ;
085F 484
F0A1 CF D5 085F 485 TSTL VALID1 ; TEST IF THE HOME BLOCK IS VALID
07 12 0863 486 BNEQ V_HMBLK ; IF EQ 0 THEN POSSIBLY CONTROLLER.
FA1D CF 01 90 0865 487 MOV B #T,HMBLK_ERR ; DRIVE OR DISK ERROR
26 11 086A 488 BRB PREP_ERR
086C 489 V_HMBLK:
FD98 CF D5 086C 490 TSTL VALID2
07 12 0870 491 BNEQ LABEL
FA10 CF 01 C0 0872 492 MOVL #1,HMBLK_ERR
19 11 0877 493 BRB PREP_ERR
FF56 CF 50445243 8F D1 0879 494 LABEL:
FF4F CF 204B4341 8F D1 0882 495 CMPL #CRDPACK1,CRDP ; COMPARE LABEL
0E 12 0882 496 BNEQ PREP_ERR ; IF NOT CORRECT
0884 497 CMPL #CRDPACK2,ACK_ ; PRINT ERROR PREP
```

```

03 12 088D 498 BNEQ PREP_ERR
FAA4 31 088F 499 BRW LESI_UDA_PARAM
0892 500
0892 501 PREP_ERR:
F9F2 CI 01 90 0892 502 MOVW #1,SPIN_DWN ; PREPARATION ERROR WRONG PACK
03 F9EA CF E8 0897 503 BLBS RC25_PRES,20$ ; TEST IF RC25
F761 31 089C 504 BRW RA60_TEST
F75E 31 089F 505 20$: BRW RC25_TEST
08A2 506
08A2 507
54 02 A7 B0 08A2 508 10: MOVW UDASA(R7),R4 ; Controller offline?
27 12 08A6 509 BNEQ ERROR1 ; Yep, error out
F967 CF 8000 8F A8 08A8 510 BLSW #OWN, CD+2 ; Set controller ownership
F95C CF 8000 8F A8 08AF 511 BLSW #OWN, RD+2 ; Ditto
54 67 B0 08B6 512 MOVW UDAIP(R7),R4 ; Tell controller to read
54 02 A7 B0 08B9 513 30$: MovW UdaSa(R7), R4 ; Any problems?
10 12 08BD 514 BNEQ Error1 ; Yes
F94F CF B5 08BF 515 TSTW RD+2 ; Any response back?
F4 19 08C3 516 BLSS 30$ ; No, spin until there is
F991 CF B5 08C5 517 TSTW RSPPKT+MSCPSW_STATUS ; Any drive errors?
04 12 08C9 518 BNEQ ERROR1 ; Yes
08CB 519
08CB 520
08CB 521 ; Transfer is complete. Return with success status code.
08CB 522
50 01 3C 08CB 523 MOVZWL #SS$_NORMAL,R0 ; SET COMPLETION CODE
05 08CE 524 RSB ; AND RETURN
08CF 525
08CF 526 ; Error occurred during transfer. Return and retry.
08CF 527
08CF 528
08CF 529 ERROR1:
50 F7DD CF 94 08CF 530 ClrB W^Init ; Must re-init next time
0054 8F 3C 08D3 531 MOVZWL #SS$_CTRLERR,R0 ; Set failure status
05 08D8 532 RSB ; Return to BOOTDRVR
08D9 533
08D9 534
08D9 535 .END

```

ZZ-ENKAZ-1.3 Symbol table
ENKAZ02
Symbol table

- UDA50, LESI, IDC SIZING TESTS

J 5
14-JUN-1984

Fiche 1 Frame J5

Sequence 61

14-JUN-1984 16:04:00 VAX-11 Macro V03-01 Page 13
14-JUN-1984 13:28:26 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

ACK	000007DC	R	02
BAD_DISK	00000288	RG	02
BYTE_OFF	00000077	R	02
CD	00000214	R	02
CMDINT	0000020C	R	02
CMDPKT	0000021C	R	02
CRDP	000007D8	R	02
CRDPAK1	= 50445243		
CRDPAK2	= 20484341		
C D RDY	= 00000081	G	
DRIVE0	000003B5	RG	02
DS\$GB_UDA50_INIT	000000B0	RG	02
DS1	= 00000100	G	
ERROR	000000A2	R	02
ERROR1	000008CF	R	02
ERROR_ROUTINE	*****	X	02
FUNC	= 00000010		
GO	= 00000001		
HMBLK_BUFF	00000600	R	02
HMBLK_ERR	00000287	RG	02
IDC	000002EE	R	02
IDC_CSR	*****	X	02
IDC_HLD	0000028A	RG	02
IDC_PARAM_0	00000384	RG	02
IDC_PARAM_1	0000035C	RG	02
IDC_TST	0000028E	RG	02
INIT	000000B0	R	02
INIT_DRVO	00000447	RG	02
INTTBL	00000200	R	02
IO	000008A2	R	02
IO\$ READLBLK	= 00000021		
LABEL	00000879	R	02
LESI_UDA_PARAM	00000336	RG	02
LOOP	00000063	R	02
MCHK_IDC	*****	X	02
MCHK_LES_UDA	*****	X	02
MODE	= 00000014		
MSCP\$B_BUFFER	= 00000010		
MSCP\$B_OPCODE	= 00000008		
MSCP\$K_OP_ONLIN	= 00000009		
MSCP\$K_OP_READ	= 00000021		
MSCP\$K_ST_SUCC	= 00000000		
MSCP\$L_BYTE_CNT	= 0000000C		
MSCP\$L_LBN	= 0000001C		
MSCP\$Q_UNIT_ID	= 00000014		
MSCP\$W_STAT05	= C000C00A		
NOCNTLR_FND	*****	X	02
NODRIVE_O_1	00000286	RG	02
NORMAL	000003F8	R	02
OWN	= 00000800		
PR\$ MAPEN	= 00000038		
PR730\$ TODR	= 0000001B		
PREP_ERR	00000892	R	02
PTE\$M PFN	= 001FFFFF		
RA60 TEST	*****	X	02
RA8081 TEST	*****	X	02
RA81_PRES	00000284	RG	02

RC25_PRES	00000285	RG	02
RC25_TEST	*****	X	02
RD	00000210	R	02
RETRY	00000051	R	02
RING	00000210	R	02
RPB\$L_ADPPHY	= 0000005C		
RPB\$L_ADPVIR	= 00000060		
RPB\$L_CSRPHY	= 00000054		
RPB\$L_CSRVIR	= 00000058		
RPB\$L_SVASPT	= 00000050		
RPB\$W_UNIT	= 00000064		
RPB_BUFF	*****	X	02
RSPINT	0000020E	R	02
RSPPKT	00000250	R	02
S1	= 0000000B		
S4	= 0000000E		
SCB4	*****	X	02
SPIN_DWN	00000289	RG	02
SS\$ CTRLERR	= 00000054		
SS\$ NORMAL	= 00000001		
TEST_6	000002F3	RG	02
TEST_7	000003AC	RG	02
TEST_8	00000800	RG	02
TIME	0000005B	R	02
UBA\$L_MAP	= 00000800		
UBA\$M-BRRVR_AIR	= 80000000		
UDAIP	= 00000000		
UDASA	= 00000002		
UDA_LESI	0000031F	RG	02
UDA_LESI_CSR	*****	X	02
UDA_RESPONSE	00000250	RG	02
UD_DRIVER1	00000292	RG	02
UD_INIT	00000000	RG	02
UMR	= 000001EE		
UNEX_E_I	*****	X	02
VAS\$ VPN	= 00000015		
VASV VPN	= C0000009		
VALID	000000AC	R	02
VALID1	00000604	R	02
VALID2	00000608	R	02
V_HMBLK	0000086C	R	02
WHAT_DRIVE	00000280	RG	02
WHAT_MODE	*****	X	02

22-ENKAZ-1.3
ENKAZ02
Psect synopsis

Psect synopsis

- UDA50, LESI, IDC SIZING TESTS

K 5
14-JUN-1984

Fiche 1 Frame K5

Sequence 62

14-JUN-1984 16:04:00

VAX-11 Macro V03-01

Page 14

14-JUN-1984 13:28:26

DRDO:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

+-----+
! Psect synopsis !
+-----+

PSECT name

Allocation

PSECT No.

Attributes

ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
\$ABS\$	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
SIZING	000008D9 (2265.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	PAGE

ZZ-ENKAZ-1.3 Cross reference
ENKAZ02
Cross reference

- UDA50, LESI, IDC SIZING TESTS

L 5
14-JUN-1984

Fiche 1 Frame L5

Sequence 63

14-JUN-1984 16:04:00 VAX-11 Macro V03-01 Page 15
14-JUN-1984 13:28:26 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

+-----+
! Symbol Cross Reference !
+-----+

SYMBOL	VALUE	DEFINITION	REFERENCES...
ACK	000007DC-R	456 (3)	#-497 (3)
BAD_DISK	00000288-R	258 (3)	#-386 (3) #404 (3)
BYTE_OFF	00000077-R	155 (2)	#-119 (2) #165 (2) #168 (2)
CD	00000214-R	199 (2)	#-169 (2) #510 (3)
CMDINT	0000020C-R	195 (2)	
CMDPKT	0000021C-R	203 (2)	167 (2) 387 (3) #467 (3) #468 (3)
			#-469 (3) #481 (3)
CRDP	000007D8-R	455 (3)	#-495 (3)
CRDPACK1	=50445243	250 (3)	#-495 (3)
CRDPACK2	=20484341	251 (3)	#-497 (3)
C D RDY	=00000081	264 (3)	#-315 (3)
DRIVE0	00000385-R	383 (3)	#-437 (3)
DS\$GB_UDA50_INIT	00000080-R	180 (2)	
DS1	=00000100	265 (3)	#-312 (3)
ERROR	000000A2-R	172 (2)	#-148 (2) #152 (2)
ERROR1	000008CF-R	529 (3)	#-509 (3) #514 (3) #518 (3)
ERROR_ROUTINE	00000000-XR		#-295 (3) #431 (3)
FUNC	=00000010	247 (3)	#-465 (3)
GO	=00000001	76 (2)	192 (2)
HMBLK_BUFF	00000600-R	451 (3)	473 (3)
HMBLK_ERR	00000287-R	257 (3)	#-460 (3) #487 (3) #492 (3)
IDC	000002EE-R	311 (3)	
IDC_CSR	00000000-XR		#-312 (3) #313 (3)
IDC_HLD	0000028A-R	266 (3)	#-313 (3) #314 (3)
IDC_PARAM_0	00000384-R	359 (3)	#-318 (3)
IDC_PARAM_1	0000035C-R	347 (3)	#-317 (3)
IDC_TST	0000028E-R	267 (3)	#-314 (3) #315 (3)
INIT	00000080-R	181 (2)	#-290 (3) #296 (3) #530 (3)
INIT_DRVO	00000447-R	434 (3)	#-429 (3)
INTIBL	00000200-R	190 (2)	118 (2) #121 (2) 139 (2)
IO	000008A2-R	508 (3)	#-393 (3) #483 (3)
IOS_READLBLK	=00000021		#-465 (3)
LABEL	00000879-R	494 (3)	#-491 (3)
LESI_UDA_PARAM	00000336-R	334 (3)	#-499 (3)
LOOP	00000063-R	147 (2)	#-153 (2)
MCHK_IDC	00000000-XR		309 (3)
MCHK_LES_UDA	00000000-XR		321 (3)
MODE	=00000014	248 (3)	
MSCP\$B_BUFFER	=00000010		#-481 (3)
MSCP\$B_OPCODE	=00000008		#-467 (3)
MSCP\$K_OP_ONLIN	=00000009		#-390 (3)
MSCP\$K_OP_READ	=00000021		#-466 (3)
MSCP\$K_ST_SUCC	=00000000		#-397 (3)
MSCP\$L_BYTE_CNT	=0000000C		#-469 (3)
MSCP\$L_LBN	=0000001C		#-468 (3)
MSCP\$Q_UNIT_ID	=00000014		#-409 (3)
MSCP\$W_STATDS	=0000000A		#-396 (3) #517 (3)
NOCNTRL_FND	00000000-XR		#-294 (3)
NODRIVE_0_1	00000286-R	256 (3)	#-384 (3) #430 (3)
NORMAL	000003F8-R	407 (3)	#-398 (3) #402 (3)

ZZ-ENKAZ-1.3 Cross reference
ENKAZ02
Cross reference

- UDA50, LESI, IDC SIZING TESTS

M 5
14-JUN-1984

Fiche 1 Frame M5

Sequence 64

14-JUN-1984 16:04:00 VAX-11 Macro V03-01 Page 16
14-JUN-1984 13:28:26 DRB0:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

OWN	=00008000	77	(2)	190	(2)	#-510	(3)	#-511	(3)		
PR\$ MAPEN	=00000038			#-110	(2)						
PR730\$ TODR	=00000018			#-145	(2)	#-150	(2)				
PREP ERR	00000892-R	501	(3)	#-488	(3)	#-493	(3)	#-496	(3)	#-498	(3)
PTESM PFN	=001FFFFF			#-129	(2)						
RA60 TEST	00000000-XR			#-412	(3)	#-504	(3)				
RA8081 TEST	00000000-XR			#-416	(3)	#-421	(3)				
RA81 PRES	00000284-R	254	(3)	#-420	(3)						
RC25 PRES	00000285-R	255	(3)	#-425	(3)	#-503	(3)				
RC25 TEST	00000000-XR			#-426	(3)	#-505	(3)				
RD	00000210-R	198	(2)	#-166	(2)	#-511	(3)	#-515	(3)		
RETRY	00000051-R	139	(2)	#-172	(2)						
RING	00000210-R	197	(2)	#-121	(2)						
RP3\$L ADPPHY	=0000005C			123	(2)	#-124	(2)	#-282	(3)		
RPB\$L ADPVIR	=00000060			123	(2)	#-283	(3)				
RPB\$L CSRPHY	=00000054			125	(2)	#-126	(2)	#-287	(3)		
RPB\$L CSR VIR	=00000058			125	(2)	#-288	(3)				
RPB\$L SVASPT	=00000050			#-128	(2)						
RPB\$W UNIT	=00000064			#-381	(3)	#-389	(3)	#-435	(3)		
RPB BUFF	00000000-XR			279	(3)						
RSPINT	0000020E-R	196	(2)								
RSPPKT	00000250-R	208	(2)	164	(2)	#-396	(3)	#-409	(3)	#-517	(3)
S1	=00000008	78	(2)	#-140	(2)						
S4	=0000000E	79	(2)	#-159	(2)						
SCB4	00000000-XR			#-278	(3)	#-309	(3)	#-321	(3)		
SPIN DWN	00000289-R	259	(3)	#-385	(3)	#-401	(3)	#-502	(3)		
SS\$ CTRLERR	=00000054			#-173	(2)	#-531	(3)				
SS\$ NORMAL	=00000001			#-523	(3)						
TEST 6	000002E3-R	308	(3)								
TEST 7	000003AC-R	380	(3)	#-291	(3)						
TEST 8	00000800-R	459	(3)								
TIME	00000058-R	145	(2)	#-159	(2)						
UBA\$L MAP	=00000800			130	(2)	475	(3)				
UBA\$M BRRVR AIR	=80000000			#-477	(3)	#-479	(3)				
UDAIP	=00000000	74	(2)	#-141	(2)	#-512	(3)				
UDASA	=00000002	75	(2)	#-147	(2)	#-158	(2)	#-508	(3)	#-513	(3)
UDA LESI	0000031F-R	320	(3)								
UDA LESI CSR	00000000-XR			#-325	(3)						
UDA RESPONSE	00000250-R	207	(2)								
UD DRIVER1	00000292-R	272	(3)	330	(3)						
UD INIT	00000000-R	106	(2)	292	(3)						
UMR	=000001EE	80	(2)	#-117	(2)	130	(2)				
UNEX E I	00000000-XR			278	(3)						
VAS\$ VPN	=00000015			#-122	(2)	#-474	(3)	#-480	(3)		
VASV VPN	=00000009			#-122	(2)	#-474	(3)	#-480	(3)		
VALID	000000AC-R	178	(2)	#-131	(2)	#-133	(2)				
VALID1	00000604-R	452	(3)	#-485	(3)						
VALID2	00000608-R	453	(3)	#-490	(3)						
V HMBLK	0000086C-R	489	(3)	#-486	(3)						
WHAT DRIVE	00000280-R	253	(3)	#-339	(3)	#-382	(3)	#-428	(3)	#-436	(3)
WHAT MODE	00000000-XR			#-341	(3)	#-354	(3)	#-366	(3)		

+-----+
 ! Macros Cross Reference !
 +-----+

MACRO	SIZE	DEFINITION	REFERENCES...
\$BTDDF	1	59 (2)	59 (2)
\$DEFINI	1	59 (2)	59 (2) 60 (2) 61 (2) 62 (2) 63 (2)
			64 (2) 65 (2) 66 (2) 67 (2) 68 (2)
			69 (2)
! IODEF	17	60 (2)	60 (2)
\$MSCPDEF	18	61 (2)	61 (2)
\$PR730DEF	1	63 (2)	63 (2)
\$PRDEF	4	62 (2)	62 (2)
\$PTEDEF	3	64 (2)	64 (2)
\$RPBDEF	5	65 (2)	65 (2)
\$SSDEF	21	66 (2)	66 (2)
\$UBADEF	6	67 (2)	67 (2)
\$UBIDEF	2	68 (2)	68 (2)
\$VADEF	1	69 (2)	69 (2)
ASSUME	1	123 (2)	123 (2) 125 (2)

OPCODE	VALUE	REFERENCES...
ADDL	00C0	121 (2)
AOBLEQ	00F3	159 (2)
BBS	00E0	149 (2)
BEQL	0013	398 (3)
BGTRU	001A	152 (2)
BICL	00CA	129 (2) 165 (2) 168 (2)
BICL3	00CB	119 (2)
BISL3	00C9	120 (2) 131 (2) 133 (2) 166 (2) 169 (2) 477 (3) 479 (3)
BISW	00A8	510 (3) 511 (3)
BLBC	00E9	127 (2) 290 (3)
BLBS	00E8	293 (3) 503 (3)
BLSS	0019	148 (2) 516 (3)
BNEQ	0012	316 (3) 400 (3) 411 (3) 415 (3) 419 (3) 424 (3) 429 (3)
		486 (3) 491 (3) 496 (3) 498 (3) 509 (3) 514 (3) 518 (3)
BRB	0011	153 (2) 402 (3) 488 (3) 493 (3)
BRW	0031	291 (3) 295 (3) 317 (3) 318 (3) 412 (3) 416 (3) 421 (3)
		426 (3) 431 (3) 437 (3) 499 (3) 504 (3) 505 (3)
BSBB	0010	483 (3)
BSBW	0030	393 (3)
CALIS	00FB	292 (3) 330 (3)
CLRB	0094	384 (3) 385 (3) 386 (3) 460 (3) 530 (3)
CLRL	00D4	340 (3) 342 (3) 353 (3) 355 (3) 365 (3) 367 (3) 395 (3)
		408 (3) 461 (3)
CLRQ	007C	391 (3) 392 (3)
CLRW	00B4	141 (2)
CMPB	0091	315 (3) 397 (3)
CMPL	00D1	151 (2) 410 (3) 414 (3) 418 (3) 423 (3) 495 (3) 497 (3)
CMPW	00B1	399 (3)
EXTZV	00EF	122 (2) 474 (3)
INCL	00D6	132 (2) 478 (3)
INSV	00F0	480 (3)
JMP	0017	344 (3) 357 (3) 369 (3)
MFPR	00DB	110 (2) 145 (2) 150 (2)
MOVAB	009E	118 (2) 139 (2) 146 (2) 164 (2) 167 (2) 279 (3) 321 (3)
		387 (3)
MOVAL	00DE	130 (2) 278 (3) 309 (3) 473 (3) 475 (3)
MOVB	0090	296 (3) 401 (3) 404 (3) 420 (3) 425 (3) 430 (3) 466 (3)
		487 (3) 492 (3) 502 (3)
MOVL	00D0	117 (2) 124 (2) 126 (2) 128 (2) 138 (2) 140 (2) 280 (3)
		282 (3) 283 (3) 285 (3) 287 (3) 288 (3) 294 (3) 312 (3)
		313 (3) 336 (3) 337 (3) 338 (3) 339 (3) 341 (3) 343 (3)
		349 (3) 350 (3) 351 (3) 352 (3) 354 (3) 356 (3) 361 (3)
		362 (3) 363 (3) 364 (3) 366 (3) 368 (3) 381 (3) 382 (3)
		388 (3) 409 (3) 435 (3) 436 (3) 462 (3) 463 (3) 464 (3)
		465 (3) 468 (3) 469 (3) 481 (3)
MOVW	00B0	147 (2) 158 (2) 325 (3) 396 (3) 508 (3) 512 (3) 513 (3)
MOVZBL	009A	314 (3) 389 (3) 390 (3)
MOVZWL	003C	173 (2) 523 (3) 531 (3)
RET	0004	170 (2)
RSB	0005	524 (3) 532 (3)

ZZ-ENKAZ-1.3 Cross reference
ENKAZ02
Cross reference

- UDA50, LESI, IDC SIZING TESTS

C 6
14-JUN-1984

Fiche 1 Frame C6

Sequence 67

14-JUN-1984 16:04:00

VAX-11 Macro V03-01

Page 19

14-JUN-1984 13:28:26

DRBO:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

SOBGIR	00F5	172	(2)				
TSTL	00D5	428	(3)	485	(3)	490	(3)
TSTW	00B5	515	(3)	517	(3)		

ZZ-ENKAZ-1.3 Cross reference
 ENKAZ02
 Cross reference

- UDA50, LES1, IDC SIZING TESTS

D 6
 14-JUN-1984

Fiche 1 Frame D6

Sequence 68

14-JUN-1984 16:04:00

VAX-11 Macro V03-01

Page 20

14-JUN-1984 13:28:26

DRBO:[BALL.ENKAZ.ENKAZ_1]ENKAZ02A.(3)

+-----+
 ! Directives Cross Reference !
 +-----+

DIRECTIVE	REFERENCES...															
.ALIGN	188	(2)	449	(3)												
.BLKL	454	(3)	457	(3)												
.BLKW	204	(2)	206	(2)	208	(2)										
.BYTE	182	(2)	254	(3)	255	(3)	256	(3)	257	(3)	258	(3)	259	(3)		
.DISABLE	176	(2)														
.ENABLE	108	(2)														
.END	535	(3)														
.ENDC	123	(2)	125	(2)												
.ENDM	123	(2)	59	(2)	60	(2)	61	(2)	62	(2)	63	(2)	64	(2)	65	(2)
	66	(2)	67	(2)	68	(2)	69	(2)								
.ENTRY	106	(2)	272	(3)												
.IDENT	2	(1)														
.IF	123	(2)	125	(2)												
.IFF	123	(2)	125	(2)												
.LONG	156	(2)	178	(2)	191	(2)	198	(2)	199	(2)	253	(3)	266	(3)	267	(3)
	451	(3)	452	(3)	453	(3)	455	(3)	456	(3)						
.NOCROSS	59	(2)	60	(2)	61	(2)	62	(2)	63	(2)	64	(2)	65	(2)	66	(2)
	67	(2)	68	(2)	69	(2)										
.PAGE	299	(3)	372	(3)	440	(3)										
.PSECT	90	(2)														
.RESTORE	59	(2)	60	(2)	61	(2)	62	(2)	63	(2)	64	(2)	65	(2)	66	(2)
	67	(2)	68	(2)	69	(2)										
.SAVE	59	(2)	60	(2)	61	(2)	62	(2)	63	(2)	64	(2)	65	(2)	66	(2)
	67	(2)	68	(2)	69	(2)										
.SBTTL	210	(3)	300	(3)	373	(3)	441	(3)	54	(2)	92	(2)				
.TITLE	1	(1)														
.WORD	190	(2)	192	(2)	194	(2)	195	(2)	196	(2)	201	(2)	202	(2)	203	(2)

+-----+
 ! Register Cross Reference !
 +-----+

REGISTER	NO. REFERENCES	REFERENCES...
AP	1	#-465 (3)
FP	3	#-342 (3)
R0	15	#-110 (2) #355 (3) #-367 (3) #127 (2) #-140 (2) #-149 (2) #-159 (2) #-124 (2) #-126 (2) #-293 (3) #-325 (3) #-336 (3) #-349 (3) #-361 (3) #-523 (3) #-531 (3)
R1	18	#-119 (2) #-120 (2) #-150 (2) #-151 (2) #-164 (2) #-165 (2) #-166 (2) #-167 (2) #-168 (2) #-169 (2) #-337 (3) #-350 (3) #-362 (3) #-461 (3) #-474 (3) #-477 (3) #-478 (3) #-479 (3)
R10	4	#-473 (3) 474 (3) 480 (3) #-481 (3)
R2	31	106 (2) #-118 (2) #-119 (2) #-120 (2) #-121 (2) 122 (2) #-128 (2) #-129 (2) #-131 (2) #-132 (2) #-133 (2) #-145 (2) 146 (2) #-151 (2) 272 (3) #-338 (3) #-351 (3) #-363 (3) #-395 (3) #-396 (3) #-397 (3) #-399 (3) #-408 (3) #-409 (3) #-410 (3) #-414 (3) #-418 (3)
R3	14	106 (2) #-124 (2) 130 (2) #-139 (2) #-158 (2) 272 (3) #-280 (3) #-282 (3) #-283 (3) #-339 (3) #-352 (3) #-364 (3) #-462 (3) 475 (3)
R4	16	106 (2) #-130 (2) #-131 (2) #-133 (2) #-147 (2) 149 (2) 272 (3) #-340 (3) #-353 (3) #-365 (3) #-475 (3) #-477 (3) #-479 (3) #-508 (3) #-512 (3) #-513 (3)
R5	13	106 (2) 272 (3) #-341 (3) #-354 (3) #-366 (3) #-387 (3) #-388 (3) #-389 (3) #-390 (3) #-391 (3) #-392 (3) #-463 (3) #-468 (3)
R6	6	106 (2) #-117 (2) #-120 (2) #-166 (2) #-169 (2) 272 (3)
R7	12	106 (2) #-126 (2) #-141 (2) #-147 (2) #-158 (2) 272 (3) #-285 (3) #-287 (3) #-288 (3) #-508 (3) #-512 (3) #-513 (3)
R8	6	106 (2) #-138 (2) #-172 (2) 272 (3) #-464 (3) #-469 (3)
R9	11	#-124 (2) #-126 (2) #-128 (2) #-279 (3) #-282 (3) #-283 (3) #-287 (3) #-288 (3) #-381 (3) #-389 (3) #-435 (3)
SP	6	#-343 (3) 344 (3) #-356 (3) 357 (3) #-368 (3) 369 (3)

+-----+
 ! Performance indicators !
 +-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	96	00:00:00.49	00:00:02.89
Command processing	143	00:00:00.70	00:00:04.11
Pass 1	1775	00:00:20.87	00:02:39.91
Symbol table sort	38	00:00:02.60	00:00:32.38
Pass 2	392	00:00:03.11	00:00:28.54
Symbol table output	40	00:00:00.11	00:00:00.95
Psect synopsis output	6	00:00:00.03	00:00:00.29
Cross-reference output	188	00:00:01.02	00:00:07.56
Assembler run totals	2683	00:00:28.95	00:03:56.65

The working set limit was 500 pages.
 82237 bytes (161 pages) of virtual memory were used to buffer the intermediate code.
 There were 90 pages of symbol table space allocated to hold 1624 non-local and 12 local symbols.

535 source lines were read in Pass 1, producing 22 object records in Pass 2.
60 pages of virtual memory were used to define 18 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
DRB1:[DS.WORK]DS.MLB;218	0
SYS\$SYSROOT:[SYSLIB]LIB.MLB;1	7
SYS\$SYSROOT:[SYSLIB]STARLET.MLB;1	8
TOTALS (all libraries)	15

1657 GETS were required to define 15 macros.

There were no errors, warnings or information messages.

MACRO/OBJ/LIST/CROSS=ALL ENKAZ02A+SYS\$LIBRARY:LIB/LIB+DRB1:[DS.WORK]DS.MLB/LIB

Fiche	Frame	Sequence		
1	B1	1	Documentation	
1	E1	4	Map	
1	M1	12	CRD HARDWARE INSTRUCTION TESTS	14-JUN-1984
1	D2	16	DECLARATIONS	14-JUN-1984
1	K2	23	BOO\$READPROMPT - Prompt and read input s	14-JUN-1984
1	M2	25	PROMPT1 - Routine to prompt console for	14-JUN-1984
1	N2	26	BOO\$TYPE ASCII - Type a counted ASCII st	14-JUN-1984
1	B3	27	PRINT ROUTINE	14-JUN-1984
1	C3	28	ERROR ROUTINE	14-JUN-1984
1	F3	30	RA60 PRINT ROUTINE	14-JUN-1984
1	F3	31	RA80 AND RA81 PRINT ROUTINE	14-JUN-1984
1	G3	32	RC25 PRINT ROUTINE	14-JUN-1984
1	I3	34	EXCEPTION HANDLER ROUTINES	14-JUN-1984
1	J3	35	INITIALIZATION ROUTINE	14-JUN-1984
1	K3	36	TEST 2: SOBGEQ, SOBGTR Test	14-JUN-1984
1	L3	37	TEST 3: BBS, BBC Test	14-JUN-1984
1	M3	38	TEST 4: BBSS, BBSS Test	14-JUN-1984
1	N3	39	TEST 5: BLBS, BLBC Test	14-JUN-1984
1	B4	40	Diagnostic Subroutines	14-JUN-1984
1	C4	41	Symbol table	14-JUN-1984
1	E4	43	Cross reference	14-JUN-1984
1	J4	48	- UDA50, LESI, IDC SIZING TESTS	14-JUN-1984
1	L4	50	DECLARATIONS	14-JUN-1984
1	M4	51	UDA50 Bootstrap device initialization	14-JUN-1984
1	B5	53	UDA50 Bootstrap driver QIO	14-JUN-1984
1	D5	55	TEST 6 DETERMINING CRD BASE CONTROLLER	14-JUN-1984
1	F5	57	TEST 7 SIZE CRD DISK OPTIONS AND CHECK D	14-JUN-1984
1	H5	59	TEST 8 RA60 AND RC25 VERIFY PACK LABEL C	14-JUN-1984
1	J5	61	Symbol table	14-JUN-1984
1	K5	62	Psect synopsis	14-JUN-1984
1	L5	63	Cross reference	14-JUN-1984

B	1	Documentation
C	1	Documentation
D	1	Documentation
E	1	Map
F	1	Map
G	1	Map
H	1	Map
I	1	Map
J	1	Map
K	1	Map
L	1	Map
M	1	CRD HARDWARE INSTRUCTION TESTS
N	1	CRD HARDWARE INSTRUCTION TESTS
B	2	CRD HARDWARE INSTRUCTION TESTS
C	2	CRD HARDWARE INSTRUCTION TESTS
D	2	DECLARATIONS
E	2	DECLARATIONS
F	2	DECLARATIONS
G	2	DECLARATIONS
H	2	DECLARATIONS
I	2	DECLARATIONS
J	2	DECLARATIONS
K	2	BOOT\$READPROMPT - Prompt and re
L	2	BOOT\$READPROMPT - Prompt and re
M	2	PROMPT1 - Routine to prompt co
N	2	BOOT\$TYPE_ASCII - Type a counte
B	3	PRINT ROUTINE
C	3	ERROR ROUTINE
D	3	ERROR ROUTINE
E	3	RA60 PRINT ROUTINE
F	3	RA80 AND RA81 PRINT ROUTINE
G	3	RC25 PRINT ROUTINE
H	3	RC25 PRINT ROUTINE
I	3	EXCEPTION HANDLER ROUTINES
J	3	INITIALIZATION ROUTINE
K	3	TEST 2: SOBGEQ, SOBGTR Test
L	3	TEST 3: BBS, BBC Test
M	3	TEST 4: BBSS, BBBS Test
N	3	TEST 5: BLBS, BLBC Test
B	4	Diagnostic Subroutines
C	4	Symbol table
D	4	Symbol table
E	4	Cross reference
F	4	Cross reference
G	4	Cross reference
H	4	Cross reference
I	4	Cross reference
J	4	- UDA50, LESI, IDC SIZING TEST
K	4	- UDA50, LESI, IDC SIZING TEST
L	4	DECLARATIONS
M	4	UDA50 Bootstrap device initial
N	4	UDA50 Bootstrap device initial
B	5	UDA50 Bootstrap driver QIO
C	5	UDA50 Bootstrap driver QIO
D	5	TEST 6 DETERMINING CRD BASE CO
E	5	TEST 6 DETERMINING CRD BASE CO
F	5	TEST 7 SIZE CRD DISK OPTIONS A
G	5	TEST 7 SIZE CRD DISK OPTIONS A
H	5	TEST 8 RA60 AND RC25 VERIFY PA
I	5	TEST 8 RA60 AND RC25 VERIFY PA

J	5	Symbol table
K	5	Psect synopsis
L	5	Cross reference
M	5	Cross reference
N	5	Cross reference
B	6	Cross reference
C	6	Cross reference
D	6	Cross reference
E	6	Cross reference
F	6	Cross reference
G	6	Directory