

DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

EEEEEEEEEE	XX	XX	AAAAAA	MM	MM	DDDDDDDD	EEEEEEEEEE	PPPPPPPP	
EEEEEEEEEE	XX	XX	AAAAAA	MM	MM	DDDDDDDD	EEEEEEEEEE	PPPPPPPP	
EE	XX	XX	AA	MMM	MMM	DD	EE	PP	PP
EE	XX	XX	AA	MMM	MMM	DD	EE	PP	PP
EE	XX	XX	AA	MM	MM	DD	EE	PP	PP
EE	XX	XX	AA	MM	MM	DD	EE	PP	PP
EEEEEEEEEE	XX	XX	AA	MM	MM	DD	EEEEEEEEEE	PPPPPPPP	
EEEEEEEEEE	XX	XX	AA	MM	MM	DD	EEEEEEEEEE	PPPPPPPP	
EE	XX	XX	AAAAAAAAAA	MM	MM	DD	EE	PP	
EE	XX	XX	AAAAAAAAAA	MM	MM	DD	EE	PP	
EE	XX	XX	AA	MM	MM	DD	EE	PP	
EE	XX	XX	AA	MM	MM	DD	EE	PP	
EEEEEEEEEE	XX	XX	AA	MM	MM	DDDDDDDD	EEEEEEEEEE	PP	
EEEEEEEEEE	XX	XX	AA	MM	MM	DDDDDDDD	EEEEEEEEEE	PP	

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SSSSSS
LL	II	SSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLLLL	IIIIII	SSSSSSSS

(2) 103

EXAM/DEPO COMMAND

```
0000 1      .TITLE  EXAMDEP - EXAMINE AND DEPOSIT COMMANDS
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 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      EXAMINE / DEPOSIT DCLS COMMAND EXECUTION
0000 29
0000 30      W. H. BROWN 28-JUL-1977
0000 31
0000 32      MODIFIED BY:
0000 33
0000 34      V03-002 PCG0002      Peter George      14-Jul-1983
0000 35      Convert results to binary after evaluating expressions.
0000 36
0000 37      V03-001 PCG0001      Peter George      17-Feb-1983
0000 38      Remove SETBIT WRK_V_NOSTAT in EXAMINE and DEPOSIT.
0000 39      Remove reference to $CLIDEFQUALEXAM.
0000 40      Use PTR_B_NUMBER to get qualifier number.
0000 41      :---
0000 42
0000 43
0000 44      MACRO LIBRARY CALLS
0000 45
0000 46
0000 47      PRCDEF      : DEFINE PROCESS WORK AREA
0000 48      WRKDEF      : DEFINE COMMAND WORK AREA
0000 49      PTRDEF      : DEFINE TOKEN DESCRIPTOR
0000 50      $CLMSGDEF    : DEFINE ERROR/STATUS CODES
0000 51      $PSLDEF      : PROCESSOR STATUS LONG WORDS
0000 52
0000 53
0000 54      LCOAL DEFINITIONS
0000 55
0000 56      DISPLAY CONTROL MODE
0000 57
```

```
00000000 0000 58
00000001 0000 59      HEXMOD = 0      ; EXAMINE MODE IS HEX
00000002 0000 60      DECMOD = 1      ; EXAMINE MODE IS DECIMAL
FFFFFFF 0000 61      OCTMOD = 2      ; EXAMINE MODE IS OCTAL
        0000 62      ASCMOD = -1     ; EXAMINE MODE IS ASCII
        0000 63
        0000 64      ; WIDTH CONTROL VALUES
        0000 65
00000004 0000 66      LONGWID = 4      ; LONG WORD WIDTH
00000001 0000 67      BYTEWID = 1      ; BYTE WIDTH
00000002 0000 68      WORDWID = 2      ; WORD WIDTH
        0000 69
        0000 70      ; LOCAL DATA
        0000 71
        0000 72
00000000 0000 73      .PSECT DCL$ZCODE      BYTE, RD, NOWRT
        0000 74
        0000 75      ; DISPLAY WIDTH CONTROL ARRAYS FOR EACH MODE
        0000 76
01 03 03 02 0000 77 BW:      .BYTE 2,3,3,1      ; DISPLAY WIDTH FOR BYTE OF EACH MODE
02 06 05 04 0004 78 WW:      .BYTE 4,5,6,2      ; LIKEWISE FOR WORD WIDTH
04 08 0A 08 0008 79 LW:      .BYTE 8,10,8,4      ; AND THE SAME FOR LONGWORDS
        000C 80
        000C 81
        000C 82      ; RADIX TABLE FOR NUMERIC CONVERSION.      NOTE:      RADIX 0 IS ASCII CONVERT
        000C 83
08 0A 10 000C 84 RADIX:      .BYTE 16,10,8      ; RADIX CONVERT TABLE
        000F 85
        000F 86
        000F 87      ; BIT ARRAY TO DEFINE WHICH OPTIONS EFFECT MODE OR WIDTH.
        000F 88      ; BIT SET INDICATES MODE SPECIFIER.
        000F 89
        000F 90
        000F 91 TYPE:      .BYTE <1@CLISK_EXAM_ASCII>!--      ; ASCII IS A MODE SWITCH
        000F 92      <1@CLISK_EXAM_DECI>!--      ; DECIMAL IS A MODE SWITCH
        000F 93      <1@CLISK_EXAM_HEXA>!--      ; HEXIDECIMAL
01' 000F 94      <1@CLISK_EXAM_OCTA>      ; AND OCTAL
        0010 95
        0010 96
        0010 97      ; QUALIFIER VALUE TABLE.      THIS ARRAY HAS THE ASSOCIATED DATA FOR
        0010 98      ; THE RESPECTIVE OPTION TO BE LOADED INTO THE TABLE.
        0010 99
02 02 04 00 01 01 FF 0010 100 VALU:      .BYTE ASCMOD,BYTEWID,DECMOD,HEXMOD,LONGWID,OCTMOD,WORDWID
        0017 101
```

```
0017 103 .SBTTL EXAM/DEPO COMMAND
0017 104 :+
0017 105 : DCLSEXAMINE - EXAMINE MEMORY COMMAND
0017 106 :
0017 107 :
0017 108 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE EXAMINE
0017 109 : DCLS COMMAND.
0017 110 :
0017 111 : INPUTS:
0017 112 :
0017 113 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
0017 114 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
0017 115 :
0017 116 : WORK AREA LOCATIONS:
0017 117 :
0017 118 : PRC_L_EXMDEPADR HAS "DOT"
0017 119 : PRC_B_EXMDEPWID HAS CURRENT WIDTH, IE: BYTE, WORD, LONGWORD.
0017 120 : PRC_B_EXMDEPMOD HAS CURRENT MODE, IE: HEX, ASCII, DECIMAL, OCTAL.
0017 121 :
0017 122 : OUTPUTS:
0017 123 :
0017 124 : THE CURRENT DEFAULT RADIX AND WIDTH ARE SET TO THOSE SPECIFIED, IF ANY.
0017 125 : THE SPECIFIED LOCATION(S) ARE DISPLAYED IN THE CURRENT RADIX AND WIDTH.
0017 126 : -
0017 127 : .ENABL LSB
0017 128 :
0017 129 DCLSEXAMINE::
0017 130 BSBW GETIVL : GET LOCATION
001A 131 PUSHL R8 : SAVE VALUE
001C 132 TSTB R0 : END-OF-LINE?
001E 133 BEQL 20$ : BR IF YES
50 3A 91 0020 134 CMPB #^A/:/,R0 : LIST?
03 13 BEQL 10$ : BR IF OK
0144 31 0025 136 BRW EXPSYN : ALL DONE - SYNTAX ERROR
FFD5' 30 0028 137 10$: BSBW DCL$MOVCHAR : COPY COLON INTO COMMAND BUFFER
00CB 30 002B 138 BSBW GETVAL : FIND OTHER LIMIT
00A8 CB 88 9E 002E 139 20$: MOVAB (R8)+,PRC_L_EXMDEPADR(R11) : SET NEW "DOT"
57 8ED0 0033 140 POPL R7 : SET NEXT ADDRESS TO EXAMINE
50 57 D0 0036 141 EXAMIN: MOVL R7,R0 : GET A COPY OF CURRENT ADDRESS
52 6E 9E 0039 142 MOVAB (SP),R2 : BUFFER TO USE
01E2 30 003C 143 BSBW DCL$CNVHXL : FORMAT HEX LONGWORD
82 203A 8F B0 003F 144 MOVW #^A/: /,(R2)+ : SEPARATE ADDRESS FOR CONTENTS
51 D4 0044 145 CLRL R1 : BUFFER WIDTH ADJUSTMENT OF ZERO
52 51 C0 0046 146 40$: ADDL R1,R2 : POINT AT NEXT FREE BYTE
82 20 90 0049 147 MOVW #^A/ /,(R2)+ : SPACE OUT FOR NEATNESS
53 00AC CB 9A 004C 148 MOVZBL PRC_B_EXMDEPWID(R11),R3 : GET WIDTH OF CELL TO DISPLAY
0051 149 IFNORD R3,(R7),50$,#PSL$C_USER : CHECK FOR USER ACCESS TO DATA
18 00AF CB 03 E0 0057 150 BBS #PRC_V_EXEONLY,PRC_B_FLAGS2(R11),50$ : NOT READABLE IF EXECUTE ONLY
51 00AD CB 98 005D 151 CVTBL PRC_B_EXMDEPMOD(R11),R1 : MODE WE ARE DISPLAYING IN
40 19 0062 152 BLSS 90$ : BR IF ASCII MODE
02 53 D1 0064 153 CMPL R3,#2 : CHECK WIDTH OF DISPLAY
23 14 0067 154 BGTR 70$ : IF GTR THAN WIDTH IS LONGWORD
16 13 0069 155 BEQL 60$ : BR IF WORD WIDTH
50 87 9A 006B 156 MOVZBL (R7)+,R0 : BYTE WIDTH - GET VALUE
53 8E AF41 9A 006E 157 MOVZBL BW[R1],R3 : WIDTH OF FORMATED DISPLAY
20 11 0073 158 BRB 80$ :
58 57 D0 0075 159 50$: MOVL R7,R8 : TERMINATE DISPLAY AFTER THIS LINE
```

82	2A2A2A2A	8F	D0	0078	160	MOVL	#^A/****/, (R2)+	:	INDICATE ACCESS VIOLATION
		37	11	007F	161	BRB	110\$	:	SHOW WHAT WE HAVE TO SHOW
	50	87	3C	0081	162	MOVZWL	(R7)+, R0	:	VALUE
53	FF7B	CF41	9A	0084	163	MOVZBL	WW[R1], R3	:	GET WIDTH FOR WORD DISPLAY
		09	11	008A	164	BRB	80\$	:	
	50	87	D0	008C	165	MOVL	(R7)+, R0	:	VALUE
53	FF74	CF41	9A	008F	166	MOVZBL	LW[R1], R3	:	WIDTH OF FORMATTED DISPLAY
		08	BB	0095	167	PUSHR	#^M<R3>	:	SAVE WIDTH
51	FF70	CF41	9A	0097	168	MOVZBL	RADIX[R1], R1	:	RADIX
		0187	30	009D	169	BSBW	DCL\$CNVNUM	:	GET VALUE
		01	BA	00A0	170	POPR	#^M<R0>	:	GET WIDTH OF DISPLAY INTO R0
		14	11	00A2	171	BRB	110\$	:	
		52	D7	00A4	172	DECL	R2	:	BACK OVER SPACE
	50	53	D0	00A6	173	MOVL	R3, R0	:	COPY WIDTH OF DISPLAY
	62	87	90	00A9	174	MOVB	(R7)+, (R2)	:	COPY DATA INTO OUTPUT BUFFER
	20	82	91	00AC	175	CMPB	(R2)+, #^A/ /	:	CHARACTER PRINTABLE ?
		04	18	00AF	176	BGEQ	105\$	:	BR IF YES
	FF A2	2E	90	00B1	177	MOVB	#^A/ /, -1(R2)	:	IF NO-PRINT A 'DOT'
		53	F5	00B5	178	SOBGTR	R3, 100\$	:	MOVE ENOUGH BYTES?
51	52	5E	C3	00B8	179	SUBL3	SP, R2, R1	:	SET BYTE COUNT IN THE LINE
	52	5E	D0	00BC	180	MOVL	SP, R2	:	STARTING ADDRESS
	50	51	C0	00BF	181	ADDL	R1, R0	:	ADD COUNT FOR NEXT DISPLAY
4C	8F	50	91	00C2	182	CMPB	R0, #76	:	ENOUGH ROOM FOR STRING PLUS SPACE
		08	1E	00C6	183	BGEQU	120\$	:	BR IF NOT ENOUGH
	58	57	D1	00C8	184	CMPL	R7, R8	:	FINISHED?
		03	1E	00CB	185	BGEQU	120\$	:	BR IF NO MORE TO PRINT
	FF76	31	00CD	186	BRW	40\$		:	ELSE PRINT SOME MORE
	FF2D	30	00D0	187	BSBW	DCL\$MSGOUT		:	PRINT IT
58	57	D1	00D3	188	CMPL	R7, R8		:	DONE?
	03	1E	00D6	189	BGEQU	130\$		:	BR IF YES
	FF5B	31	00D8	190	BRW	EXAMIN		:	ELSE FORMAT ANOTHER LINE
			00DB	191	STATUS	NORMAL		:	
	008E	31	00E2	192	BRW	EXIT		:	EXIT
			00E5	193				:	

```
00E5 194 :+
00E5 195 : GETIVL - GET INITIAL VALUE FROM THE COMMAND LINE.
00E5 196 :
00E5 197 : GETVAL - GET NEXT VALUE FROM THE COMMAND LINE.
00E5 198 :
00E5 199 : INPUTS:
00E5 200 :
00E5 201 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
00E5 202 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
00E5 203 :
00E5 204 : WORK AREA LOCATIONS:
00E5 205 :
00E5 206 : PRC_L_EXMDEPADR HAS "DOT"
00E5 207 : PRC_B_EXMDEPWID HAS CURRENT WIDTH, IE: BYTE, WORD, LONGWORD.
00E5 208 : PRC_B_EXMDEPMOD HAS CURRENT MODE, IE: HEX, ASCII, DECIMAL, OCTAL.
00E5 209 :
00E5 210 : OUTPUTS:
00E5 211 :
00E5 212 : ANY QUALIFIERS IN THE COMMAND LINE PRECEEDING THE NEXT VALUE
00E5 213 : ARE PROCESSED AND THE NEXT EXPRESSION ON THE COMMAND LINE
00E5 214 : IS EVALUATED AND RETURN IN R8. REGISTERS R0 TO R8 ARE MODIFIED.
00E5 215 : WRK_L_EXPANDPTR IS UPDATED TO THE NEXT BYTE IN THE EXPANSION BUFFER.
00E5 216 : -
00E5 217 :
00E5 218 : .ENABL LSB
00E5 219 :
2C 3D 3A 00 00E5 220 10$: .ASCII <0>\:=,\ ; SPECIAL TERMINATORS FOR "DOT"
00E9 221 11$:
00E9 222 :
00E9 223 : RADIX CONVERT TABLE - COVERT DISPLAY MODE TO RADIX INDEX
00E9 224 :
00E9 225 : .BYTE PRC_K_HEX ; MINUS ONE INDEX(ASCII)-CONVERT IN HEX
02 01 00 00EA 226 15$: .BYTE PRC_K_HEX,PRC_K_DEC,PRC_K_OCT ; RADIX TRANSLATE
00ED 227 :
F4 AA 01 BA 00ED 228 GETIVL: POPR #^M<R0> ; GET RETURN ADDRESS
SE 5E 5E DO 00EF 229 MOVL SP,WRK_L_SAVSP(R10) ; MARK THE STACK FOR ERRORS
B0 AE 9E 00F3 230 MOVAB -80(SP),SP ; ALLOCATE A LINE BUFFER
50 DD 00F7 231 PUSHL R0 ; REPLACE RETURN ADDRESS
6D 10 00F9 232 GETVAL: BSBB 130$ ; PEEK AT NEXT CHARACTER IN INPUT
FF02' 30 00FB 233 BSBB DCL$MARK ; MARK CURRENT PARSE POSITION
50 20 91 00FE 234 CMPB #^A/ /,R0 ; BLANK?
05 12 0101 235 BNEQ 20$ ; BR IF NO AND CONTINUE
FEFA' 30 0103 236 BSBB DCL$MOVCHAR ; ELSE COPY SPACE TO BUFFER
F1 11 0106 237 BRB GETVAL ; TRY AGAIN
2F 50 91 0108 238 20$: CMPB R0,#^A\ /\ ; QUALIFIER COMMING?
24 12 010B 239 BNEQ 40$ ; BR IF NO QUALIFIER IN SIGHT
FEF0' 30 010D 240 BSBB DCL$PROCQUAL ; PROCESS THE QUALIFIER
60 50 E9 0110 241 BLBC R0,EXIT ; BR IF ERROR IN QUALIFIER
51 BA AA 0C C3 0113 242 SUBL3 #PTR_C_LENGTH,WRK_L_RSLNXT(R10),R1 ; ADDRESS OF PREVIOUS TOKEN DESCR
50 05 A1 9A 0118 243 MOVZBL PTR_B_NUMBER(R1),R0 ; GET QUALIFIER CODE
51 00AC CB 9E 011C 244 MOVAB PRC_B_EXMDEPWID(R11),R1 ; ASSUME CHANGING WIDTH
02 FEE9 CF 50 E1 0121 245 BBC R0,TYPE,30$ ; BR IF ASSUMED CORRECTLY
51 D6 0127 246 INCL R1 ; POINT AT MODE LOCATION
61 FEE1 CF40 90 0129 247 30$: MOVB VALU-1[R0],(R1) ; SET MODE OR WIDTH
C8 11 012F 248 BRB GETVAL ; TRY FOR MORE
58 D4 0131 249 40$: CLRL R8 ; ZERO ACCUMULATOR
50 2E 91 0133 250 CMPB #^A/ ./,R0 ; REFERENCE TO LAST LOCATION
```

58	00A8	CB	11	12	0136	251	BNEQ	50\$	:	BR IF NO
		FEC0'	D0	0138	252	MOVL	PRC_L_EXMDEPADR(R11),R8	:	:	GET THE PREVIOUS LOCATION
		26	30	013D	253	BSBW	DCL\$MOVCHAR	:	:	PUT DOT INTO COMMAND BUFFER
9E	AF	04	50	3A	0140	254	BSBB	130\$	:	LOOK AT NEXT CHAR
		1F	12	0142	255	LOCC	RO,#<11\$-10\$>,10\$	:	:	ANY OF THE SPECIAL TERMINATORS
		58	DD	0147	256	BNEQ	130\$	:	:	BR IF YES-NO EXPRESSION TO EVALUATE
51	00AD	CB	98	0149	257	PUSHL	R8	:	:	SAVE INITIAL VALUE
51	96	AF41	9A	014B	258	CVTBL	PRC_B_EXMDEPMOD(R11),R1	:	:	GET MODE OF OPERATION
		FEA8'	30	0150	259	MOVZBL	15\$[RT],R1	:	:	CONVERT TO SYSTEM STANDARD RADIX
		18 50	E9	0155	260	BSBW	DCL\$EXPRADIX	:	:	EXPRESSION WITH RADIX
		FEA2'	30	0158	261	BLBC	RO,EXIT	:	:	BR IF ERROR EVALUATING EXPRESSION
		12 50	E9	015B	262	BSBW	DCL\$CVT_BINARY	:	:	CONVERT RESULT TO BINARY
		FE9C'	30	015E	263	BLBC	RO,EXIT	:	:	BR IF ERROR CONVERTING
58	8E	51	C1	0161	264	BSBW	DCL\$MARK	:	:	MARK CURRENT PARSE POSITION
		FE95'	30	0164	265	ADDL3	R1,(SP)+,R8	:	:	GET FINAL VALUE TO WORK WITH
			05	0168	266	BSBW	DCL\$SETNBLK	:	:	PEEK AT NEXT NON-BLANK CHAR
				016B	267	RSB	:	:	:	:
				016C	268	:	:	:	:	:
5E	F4	AA	D0	016C	269	EXPSYN:	STATUS	EXPSYN	:	SYNTAX ERROR
			05	0173	270	EXIT:	MOVL	WRK_L_SAVSP(R10),SP	:	CLEAR THE STACK
				0177	271		RSB	:	:	RETURN TO CALLER
				0178	272		.DSABL	LSB	:	:
				0178	273		:	:	:	:
FO	AA	02	A8	0178	274	OVRFLW:	STATUS	OVRFLW	:	VALUE TO LARGE FOR CELL
		EE	11	017F	275		BISW	#WRK_M_COMMAND,WRK_W_FLAGS(R10)	:	INDICATE ERROR DURING EXECUTION
				0183	276		BRB	EXIT	:	SHOW ERROR

```
0185 278
0185 279 :+
0185 280 : DCL$DEPOSIT - DEPOSIT DATA IN MEMORY
0185 281 :
0185 282 : THIS ROUTINE IS CALLED AS AN INTERNAL COMMAND TO EXECUTE THE DEPOSIT
0185 283 : DCLS COMMAND.
0185 284 :
0185 285 : INPUTS:
0185 286 :
0185 287 : R10 = BASE ADDRESS OF COMMAND WORK AREA.
0185 288 : R11 = BASE ADDRESS OF PROCESS WORK AREA.
0185 289 :
0185 290 : WORK AREA LOCATIONS:
0185 291 :
0185 292 : PRC_L_EXMDEPADR HAS "DOT"
0185 293 : PRC_B_EXMDEPWID HAS CURRENT WIDTH, IE: BYTE, WORD, LONGWORD.
0185 294 : PRC_B_EXMDEPMOD HAS CURRENT MODE, IE: HEX, ASCII, DECIMAL, OCTAL.
0185 295 :
0185 296 : OUTPUTS:
0185 297 :
0185 298 : THE CURRENT DEFAULT RADIX AND WIDTH ARE SET TO THOSE SPECIFIED, IF ANY.
0185 299 : THE SPECIFIED LOCATION(S) ARE SET IN THE CURRENT RADIX AND WIDTH.
0185 300 :-
0185 301
0185 302 SHFCNT: .BYTE -8,-16,0,-31 ; SHIFT COUNT FOR SIGN EXTEND
0189 303
0189 304 DCL$DEPOSIT:: ; DEPOSIT DATA IN MEMORY
0189 305 BSBW GETIVL ; INIT AND GET VALUE
50 FF61 30 0189 306 CMPB #^A/=/,R0 ; VALID TERMINATOR
DB 12 018F 307 BNEQ EXPSYN ; BR ON SYNTAX ERROR
58 DD 0191 308 PUSHL R8 ; SAVE INITIAL VALUE
57 58 DO 0193 309 MOVL R8,R7 ; SET AS LAST LOCATION USED("DOT").
FE67' 30 0196 310 10$: BSBW DCL$MOVCHAR ; MOVE TERMINATOR
FF 8F 00AD CB 91 0199 311 CMPB PRC_B_EXMDEPMOD(R11),#ASCMOD ; DOING ASCII
27 12 019F 312 BNEQ 40$ ; BR IF NO
FE5C' 30 01A1 313 BSBW DCL$MARK ; MARK CURRENT POSITION IN BUFFER
FE59' 30 01A4 314 20$: BSBW DCL$MOVCHAR ; MOVE NEXT CHARACTER
FB 12 01A7 315 BNEQ 20$ ; UNTIL END OF LINE
FE54' 30 01A9 316 BSBW DCL$MARKEDTOKEN ; GET DESCRIPTOR OF STRING
51 D7 01AC 317 DECL R1 ; DISCOUNT EOL CHARACTER
18 13 01AE 318 BEQL 40$ ; BR IF NULL STRING
FE4D' 30 01B0 319 BSBW DCL$COMPSTRING ; COMPRESS THE STRING
01B3 320 IFNOWRT R1,(R7),90$,#PSL$C_USER ; VERIFY PROPER ACCESS
54 00AF CB 03 E0 01B9 321 BBS #PRC_V_EXEONLY,PRC_B_FLAGS2(R11),90$ ; NOT WRITEABLE IF EXECUTE ONLY
67 62 51 28 01BF 322 MOVCL R1,(R2),(R7) ; STORE THE DATA
57 53 DO 01C3 323 MOVL R3,R7 ; SET ENDING ADDRESS
4B 11 01C6 324 BRB 90$ ; SHOW THE DAMAGE
52 00AC CB 9A 01C8 325 40$: MOVZBL PRC_B_EXMDEPWID(R11),R2 ; SET WIDTH OF OPERATION
01CD 326 IFNOWRT R2,(R7),90$,#PSL$C_USER ; VERIFY PROPER ACCESS
3A 00AF CB 03 E0 01D3 327 BBS #PRC_V_EXEONLY,PRC_B_FLAGS2(R11),90$ ; NOT WRITEABLE IF EXECUTE ONLY
57 DD 01D9 328 PUSHL R7 ; SAVE ADDRESS TO DEPOSIT
FF1B 30 01DB 329 BSBW GETVAL ; GET VALUE
57 8ED0 01DE 330 POPL R7 ; RESTORE THE ADDRESS
52 00AC CB 9A 01E1 331 MOVZBL PRC_B_EXMDEPWID(R11),R2 ; SET WIDTH
51 9A AF42 9A 01E6 332 MOVZBL SHFCNT-1[R2],R1 ; GET SHIFT COUNT
51 58 51 78 01EB 333 ASHL R1,R8,R1 ; GET THE SIGN OF THE VALUE
07 13 01EF 334 BEQL 50$ ; BR IF POSITIVE NUMBER
```

```
51 D6 01F1 335 INCL R1 ; NEGATIVE SIGN?
03 13 01F3 336 BEQL 50$ ; BR IF NO-OVERFLOW HAS OCCURED
FF80 31 01F5 337 BRW OVRFLW
52 02 C2 01F8 338 50$: ; DECODE WIDTH
0C 14 01FB 339 BGTR 70$ ; BR IF LONGWORD
05 19 01FD 340 BLSS 60$ ; BR IF BYTE
87 58 B0 01FF 341 MOVW R8,(R7)+ ; STORE DATA
08 11 0202 342 BRB 80$ ;
87 58 90 0204 343 60$: MOVW R8,(R7)+ ; STORE DATA
03 11 0207 344 BRB 80$ ;
87 58 D0 0209 345 70$: MOVL R8,(R7)+ ; ETC
50 95 020C 346 80$: TSTB R0 ; END OF LINE
03 13 020E 347 BEQL 90$ ; BR IF YES
FF83 31 0210 348 BRW 10$
58 67 9E 0213 349 90$: MOVAB (R7),R8 ; SET UPPER LIMIT
00A8 CB 57 D0 0216 350 MOVL R7,PRC_L_EXMDEPADR(R11) ; SET "DOT"
57 8ED0 021B 351 POPL R7 ; GET FIRST VALUE
FE15 31 021E 352 BRW EXAMIN ; PRINT THE RESULT
353
354 :+ DCL$CNVNUM - CONVERT NUMBER
355
356 THIS ROUTINE CONVERTS A BINARY NUMBER INTO A ASCII STRING IN ANY RADIX
357 FROM 2-9 OR HEX.
358
359 INPUTS:
360 R0 = NUMBER
361 R1 = RADIX
362 R2 = BUFFER TO STORE RESULT
363 R3 = NUMBER OF CHARACTERS
364
365 ALTERNATE ENTRY DCL$CNVHXL - FOR CONVERSION OF HEX LONG WORDS
366
367
368 OUTPUTS:
369 THE NUMBER IS CONVERTED TO STRING, AND THE BUFFER POINTER
370 IS ADVANCE TO BEYOND THE LAST CHARACTER.
371
372
373 DCL$CNVHXL:: ; CONVERT HEX LONG WORD
374 MOVL #8,R3 ; SET NUMBER OF CHARATERS TO OUTPUT
375 MOVL #16,R1 ; AND RADIX
376
377 DCL$CNVNUM:: ; CONVERT NUMBER TO STRING
378 MOVL R1,AP ; SAVE RADIX IN SCRATCH REGISTER
379 CLRL R1 ; SET EXTENDED WORD OF ZERO
380 10$: EDIV AP,R0,R0,-(SP) ; REMOVE THE RADIX
381 ADDB #^A/0/,(SP) ; CONVERT TO ASCII
382 CMPB (SP),#^A/9/ ; DECIMAL NUMBER
383 BLEQU 20$ ; BR IF YES
384 ADDB #7,(SP) ; SET TO HEX CHARACTER
385 20$: DECL R3 ; COUNT DOWN THE NUMBER OF CHARACTERS
386 BEQL 40$ ; BR IF DONE
387 BGTR 30$ ; BR IF NO ZERO SUPPRESSION
388 TSTL R0 ; ANY MORE DATA?
389 BEQL 40$ ; BR IF NO
390 30$: BSBB 10$ ; CONVERT NEXT DIGIT
82 8E F6 0248 391 40$: CVTLB (SP)+,(R2)+ ; STORE A DIGIT
```

EXAMDEP
V04-000

- EXAMINE AND DEPOSIT COMMANDS
EXAM/DEPO COMMAND

D 16

15-SEP-1984 23:45:07 VAX/VMS Macro V04-00
4-SEP-1984 23:40:24 [DCL.SRC]EXAMDEP.MAR;1

Page 9
(3)

05	024B	392	RSB
	024C	393	
	024C	394	.END

; UNWIND RECURSIVE LOOP

EXAMDEP
Symbol table

- EXAMINE AND DEPOSIT COMMANDS

E 16

15-SEP-1984 23:45:07 VAX/VMS Macro V04-00
4-SEP-1984 23:40:24 [DCL.SRC]EXAMDEP.MAR;1

Page 10
(3)

```

ASCMOD          = FFFFFFFF
BW              = 00000000 R      02
BYTEWID         = 00000001
CLISK_EXAM_ASCII ***** X      02
CLISK_EXAM_DEC  ***** X      02
CLISK_EXAM_HEXA ***** X      02
CLISK_EXAM_OCTA ***** X      02
CLIS_EXPSYN     = 00038038
CLIS_NORMAL     = 00030001
CLIS_OVRFLW     = 00038160
DCL$CNVHXL      = 00000221 RG     02
DCL$CNVNUM      = 00000227 RG     02
DCL$COMPSTRING ***** X      02
DCL$CVT_BINARY ***** X      02
DCL$DEPOSIT     = 00000189 RG     02
DCL$EXAMINE     = 00000017 RG     02
DCL$EXPRADIX    ***** X      02
DCL$MARK        ***** X      02
DCL$MARKEDTOKEN ***** X      02
DCL$MOVCHAR     ***** X      02
DCL$MSGOUT      ***** X      02
DCL$PROCQUAL    ***** X      02
DCL$SETNBLK     ***** X      02
DECMOD          = 00000001
EXAMIN          = 00000036 R      02
EXIT            = 00000173 R      02
EXPSYN          = 0000016C R      02
GETIVL          = 000000ED R      02
GETVAL          = 000000F9 R      02
HEXMOD          = 00000000
LONGWID         = 00000004
LW              = 00000008 R      02
OCTMOD          = 00000002
OVRFLW          = 00000178 R      02
PRC_B_CONTINUE  = 000000F3
PRC_B_DEFRADIX  = 000000AE
PRC_B_EXMDEPMOD = 000000AD
PRC_B_EXMDEPWID = 000000AC
PRC_B_EXONLYL   = 0000012D
PRC_B_FLAGS2    = 000000AF
PRC_B_IMGFLAG   = 00000078
PRC_B_OUTFLAGS  = 0000012C
PRC_B_PROMPTLEN = 000000F0
PRC_C_LENGTH    = 00000534
PRC_G_COMMANDS  = 00000133
PRC_G_PROMPT    = 000000F4
PRC_K_DEC       = 00000001
PRC_K_HEX       = 00000000
PRC_K_LENGTH    = 00000534
PRC_K_OCT       = 00000002
PRC_L_CURRKEY   = 00000048
PRC_L_EXMDEPADR = 000000A8
PRC_L_EXTARG    = 00000094
PRC_L_EXTBLK    = 0000008C
PRC_L_EXTCOD    = 0000009C
PRC_L_EXTHND    = 00000090
PRC_L_EXTPRM    = 00000098

```

```

PRC_L_IDFLNK    0000008C
PRC_L_IMGACTSTS 00000080
PRC_L_INDCLOCK  0000007C
PRC_L_INDEPTH   0000005C
PRC_L_INDFAB    0000001C
PRC_L_INDINPRAB 00000014
PRC_L_INDOUSTRAB 00000018
PRC_L_INPRAB    00000008
PRC_L_LASTKEY   0000004C
PRC_L_LSTSTATUS 00000080
PRC_L_ONCTLY    00000088
PRC_L_ONERROR    0000006C
PRC_L_OUTOFBAND 00000084
PRC_L_OUTRAB     0000000C
PRC_L_OUTRABCTX 00000118
PRC_L_PPFLIST    00000070
PRC_L_RECALLPTR 0000012F
PRC_L_RESTART    00000058
PRC_L_SAVAP      00000000
PRC_L_SAVFP      00000004
PRC_L_SEVERITY   00000050
PRC_L_SPWN       000000C0
PRC_L_STACKLM    000000A4
PRC_L_STACKPT    000000A0
PRC_L_STATUS     00000054
PRC_L_STS        00000084
PRC_L_STV        00000088
PRC_L_SYMBOL     00000060
PRC_L_TMBX       00000074
PRC_L_TRMLIST    00000010
PRC_Q_ALLOCREG   00000020
PRC_Q_COMMAND    000000E0
PRC_Q_FLUSHTIME  000000D0
PRC_Q_GLOBAL     00000028
PRC_Q_IMAGENAME  000000D8
PRC_Q_KEYPAD     00000040
PRC_Q_LABEL      00000030
PRC_Q_LOCAL      00000038
PRC_Q_SAVEPRIV   000000E8
PRC_T_OUTDVI     0000011C
PRC_V_EXEONLY    = 00000003
PRC_W_ASTIOSB    000000C6
PRC_W_ASTRETN    000000C8
PRC_W_ASTSTATUS  000000C4
PRC_W_ATTMBX     0000007A
PRC_W_FLAGS      00000068
PRC_W_INPCHAN    00000064
PRC_W_ONLEVEL    0000006A
PRC_W_OUTIFI     00000114
PRC_W_OUTISI     00000116
PRC_W_OUTMBXCHN  000000CA
PRC_W_OUTMBXREF  000000CE
PRC_W_OUTMBXSIZ  000000CC
PRC_W_PMPCTRL    000000F1
PRC_W_WAITIOSB   = 00000066
PSL$C_USER       = 00000003
PTR_B_LEVEL      00000004

```

EXAMDEP
Symbol table

- EXAMINE AND DEPOSIT COMMANDS

F 16

15-SEP-1984 23:45:07

VAX/VMS Macro V04-00

Page 11
(3)

4-SEP-1984 23:40:24

[DCL.SRC]EXAMDEP.MAR;1

PTR_B_NUMBER	00000005		
PTR_B_PARMCNT	00000006		
PTR_B_VALUE	00000000		
PTR_C_LENGTH	0000000C		
PTR_K_LENGTH	0000000C		
PTR_L_DESCR	00000000		
PTR_L_ENTITY	00000008		
RADIX	0000000C	R	02
SHFCNT	00000185	R	02
TYPE	0000000F	R	02
VALU	00000010	R	02
WORDWID	= 00000002		
WRK_B_CMDOPT	FFFFFFFFC3		
WRK_B_MAXPARM	FFFFFFFFD0		
WRK_B_MINPARM	FFFFFFFFD1		
WRK_B_PARMCNT	FFFFFFFFCE		
WRK_B_PARMSUM	FFFFFFFFCF		
WRK_B_RECALLCNT	FFFFFFFFC5		
WRK_B_VALLEV	FFFFFFFFC4		
WRK_B_VERBTYP	FFFFFFFFC2		
WRK_C_LENGTH	FFFFFF486		
WRK_G_BUFFER	FFFFFF492		
WRK_G_INPBUF	FFFFFF896		
WRK_G_RESULT	FFFFFF9B6		
WRK_K_LENGTH	FFFFFF486		
WRK_L_CHARPTR	FFFFFF48E		
WRK_L_DISALLOW	FFFFFFE6		
WRK_L_ERRORRTN	FFFFFF9AE		
WRK_L_EXPANDPTR	FFFFFF486		
WRK_L_IMAGE	FFFFFFE2		
WRK_L_MARKPTR	FFFFFF48A		
WRK_L_PAROUT	FFFFFFFD2		
WRK_L_PMPTADDR	FFFFFF9A2		
WRK_L_PROMPTRTN	FFFFFF9A6		
WRK_L_PROPTR	FFFFFFFC6		
WRK_L_QUABLK	FFFFFFCA		
WRK_L_READRTN	FFFFFF9AA		
WRK_L_RECALLPTR	FFFFFFEA		
WRK_L_RSLND	FFFFFFB6		
WRK_L_RSLNXT	FFFFFFBA		
WRK_L_SAVAP	FFFFFFF8		
WRK_L_SAVFP	FFFFFFFC		
WRK_L_SAVSP	FFFFFFF4		
WRK_L_SIGNALRTN	FFFFFFD6		
WRK_L_SPECRTN	FFFFFF9B2		
WRK_L_TAB_VEC	FFFFFFDE		
WRK_L_VERB	FFFFFFBE		
WRK_M_COMMAND	= 00000002		
WRK_W_FLAGS	FFFFFFF0		
WRK_W_FLAGS2	FFFFFFF2		
WRK_W_IMGCHAN	FFFFFFEE		
WRK_W_PMPTLEN	FFFFFF99E		
WW	00000004	R	02

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

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	FFFFFFFFC (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
DCL\$ZCODE	0000024C (588.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	15	00:00:00.05	00:00:01.66
Command processing	103	00:00:00.71	00:00:06.98
Pass 1	225	00:00:06.78	00:00:19.20
Symbol table sort	0	00:00:00.74	00:00:02.08
Pass 2	71	00:00:01.30	00:00:03.65
Symbol table output	20	00:00:00.15	00:00:00.44
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	436	00:00:09.76	00:00:34.04

The working set limit was 1200 pages.

31123 bytes (61 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 500 non-local and 34 local symbols.

394 source lines were read in Pass 1, producing 14 object records in Pass 2.

32 pages of virtual memory were used to define 18 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
-\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	6
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	2
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	12

594 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:EXAMDEP/OBJ=OBJ\$:EXAMDEP MSRC\$:EXAMDEP/UPDATE=(ENH\$:EXAMDEP)+EXECMLS/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSBLDMLB/LIB

0069 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INTDEF
SOL

COMMAND
LIS

DESCRVAL
LIS

CLIMAC
MAR

CLIMSG
LIS

CONNECT
LIS

DCLPARSE
LIS

CHARMANIP
LIS

EXAMDEP
LIS

EXIT
LIS

INTIMAGES
MAR

DCXSTART
LIS

CLIGBL
LIS

DISALLOW
LIS

CLINT
LIS

CANCEL
LIS