


```

AAAAAA      CCCCCCCC  TTTTTTTTTT  RRRRRRRR  EEEEEEEEEE  FFFFFFFFFF
AAAAAA      CCCCCCCC  TTTTTTTTTT  RRRRRRRR  EEEEEEEEEE  FFFFFFFFFF
AA          AA      CC          TT          RR          RR  EE          FF
AA          AA      CC          TT          RR          RR  EE          FF
AA          AA      CC          TT          RR          RR  EE          FF
AA          AA      CC          TT          RR          RR  EE          FF
AA          AA      CC          TT          RRRRRRRR  EEEEEEEEE  FFFFFFFF
AA          AA      CC          TT          RRRRRRRR  EEEEEEEEE  FFFFFFFF
AAAAAAA     AA      CC          TT          RR  RR  EE          FF
AAAAAAA     AA      CC          TT          RR  RR  EE          FF
AA          AA      CC          TT          RR  RR  EE          FF
AA          AA      CC          TT          RR  RR  EE          FF
AA          AA      CCCCCCCC  TT          RR          RR  EEEEEEEEE  FF
AA          AA      CCCCCCCC  TT          RR          RR  EEEEEEEEE  FF

```

```

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)	100	DECLARATIONS
(4)	162	REFERENCE DIRECTIVE ROUTINES (.REFn)
(5)	223	OPERAND REFERENCE ROUTINES
(6)	269	DEFERRED REGISTER MODE OPERAND
(8)	334	INDIRECT DISPLACEMENT REFERENCES
(9)	371	CHECK FOR ILLEGAL USE OF PC
(10)	405	DISPLACEMENT REFERENCES
(11)	476	DISPLACEMENT OFF 'PC'
(12)	562	EXPLICIT DISPLACEMENTS AND PIC REFERENCES
(14)	665	LITERAL MODE
(15)	710	AUTO-INCREMENT MODE
(16)	737	IMMEDIATE MODE
(16)	750	AUTO-INCREMENT PC
(17)	836	DEFERRED INDIRECT REGISTER REFERENCE
(18)	865	INDEXED REFERENCES
(19)	924	SPESHL_MODE_CHK

```

0000 1      .TITLE MAC$ACTREF OPERAND REFERENCES
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 :
0000 29 :++
0000 30 : FACILITY:      VAX MACRO ASSEMBLER OBJECT LIBRARY
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000 35 : modules for input to the VAX-11 LINKER.
0000 36 :
0000 37 : ENVIRONMENT:  USER MODE
0000 38 :
0000 39 : AUTHOR: Benn Schreiber, CREATION DATE: 20-AUG-78
0000 40 :
0000 41 : MODIFIED BY:
0000 42 :
0000 43 :      V02.18  MTR0001      Mike Rhodes      28-Oct-1981
0000 44 :              Fix immediate mode H-float short literal optimization/
0000 45 :              generation code.  Also fixed broken BRBs to BRWs.
0000 46 :
0000 47 :      V02.17  PCG0011      Peter George    12-Oct-1981
0000 48 :              The previous fix only handles explicit short literals.
0000 49 :              Now extend that code to cover implicit short literals.
0000 50 :
0000 51 :      V02.16  PCG0010      Peter George    08-Sep-1981
0000 52 :              Correct generation of short floating point literals.
0000 53 :
0000 54 :      V02.15  CNH0042      Chris Hume      1-Dec-1980
0000 55 :              De-optimize boundary valued backward references if indexing
0000 56 :              requested.  Allow the architecturally legal immediate mode in
0000 57 :              address and yield contexts and also the practically useless

```

```

0000 58 : indexed immediate mode.
0000 59 : (ACTSTA.MAR 02.15, DEFINE.MAR 02.17, SYMTAB.MAR 02.18)
0000 60 :
0000 61 : V01.14 RN0030 R. Newland 17-Mar-1980
0000 62 : In explicit PC relative addressing test for displacement
0000 63 : expression being absolute. SPR 11-29354
0000 64 :
0000 65 : V01.13 RN0029 R. Newland 9-Mar-1980
0000 66 : In relative and relative deferred references always
0000 67 : optimize if displacement is known, and use default
0000 68 : displacement if not known. SPR 11-29062
0000 69 :
0000 70 : V01.12 RN0023 R. Newland 3-Nov-1979
0000 71 : New message codes to get error messages from system
0000 72 : message file.
0000 73 :
0000 74 : V01.11 RN0019 R. Newland 25-Oct-1979
0000 75 : Improve error pointer positioning
0000 76 :
0000 77 : V01.10 RN0017 R. Newland 20-Oct-1979
0000 78 : Support for G_floating, H_floating and Octaword data types.
0000 79 : Check all bits of operand for optimisation test.
0000 80 :
0000 81 : V01.08 RN0014 R. Newland 14-Oct-1979
0000 82 : Support for .REF16 directive
0000 83 :
0000 84 : V01.07 RN0005 R. Newland 12-Aug-1979
0000 85 : Remove .ALIGN LONG statements
0000 86 :
0000 87 : V01.09 RN0017 R. Newland 20-Oct-1979
0000 88 : Fix problem with optimising a quadword literal when
0000 89 : bits 0-31 were all zero.
0000 90 :
0000 91 : V01.08 RN0016 R. Newland 19-Oct-1979
0000 92 : Don't output error messages when .NTYPE operand
0000 93 : argument is the PC. SPR 11-26392
0000 94 :
0000 95 : V01.06 007 B. SCHREIBER 22-JAN-79
0000 96 : Fix problem with negative displacement from PC.
0000 97 :
0000 98 :--

```

```

0000 100      .SBTTL  DECLARATIONS
0000 101      :
0000 102      : INCLUDE FILES:
0000 103      :
0000 104      :
0000 105      :
0000 106      : MACROS:
0000 107      :
0000 108      :
0000 109      $MAC_ADRMODDEF      ;DEFINE ADDRESSING MODES
0000 110      $MAC_CTLFLGDEF     ;DEFINE CONTROL FLAGS
0000 111      $MAC_GENVALDEF     ;DEFINE GENERAL VALUES
0000 112      $MAC_INTCODDEF     ;DEFINE INT. FILE CODES
0000 113      $MAC_OPRDEF        ; Define operand descriptor bits
0000 114      $MAC_SYMBLKDEF     ;DEFINE SYMBOL BLOCK OFFSETS
0000 115      $MACMSGDEF        ; Define message codes
0000 116      :
0000 117      :
0000 118      : EQUATED SYMBOLS:
0000 119      :
0000 120      :
0000 121      :
0000 122      : OWN STORAGE:
0000 123      :
0000 124      :
0000 125      :
00000000 126      .PSECT  MAC$RO_DATA,NOWRT,NOEXE,GBL, LONG
0000 127      :
0000 128      PC_DISPL_DISP:      ; JUMP TABLE USED WHEN
0000 129      :                  ; .DEFAULT HAS BEEN SPECIFIED
0000 130      :                  ; FOR PC-DISPLACMENT REFS
00000000 0000 131      .LONG  0      ; 0--NOT USED
00000228' 0004 132      .LONG  DISPL3 ; 1--BYTE DISPLACEMENT
00000238' 0008 133      .LONG  DISPL5 ; 2--WORD DISPLACEMENT
00000230' 000C 134      .LONG  DISPL4 ; 3--LONGWORD DISPLACEMENT

```

```

0010 136 :++
0010 137 : OPERANDS OF MACHINE INSTRUCTIONS ARE SPECIFIED BY REFERENCES. BEFORE
0010 138 : THESE ROUTINES ARE CALLED, THE FOLLOWING FLAGS MUST BE SET:
0010 139 :
0010 140 :     MAC$GL_OPSIZE   SIZE OF OPERAND IN BYTES (DEPENDS ON INSTRUCTION)
0010 141 :
0010 142 :     MAC$GL_REFSIZ  USER SUPPLIED REF WIDTH (IE MOVL ^S#1,R0)
0010 143 :     THIS VALUE MUST BE SET TO THE DEFAULT SIZE
0010 144 :     IF THE USER SPECIFIED NO WIDTH.
0010 145 :
0010 146 :
0010 147 : THESE ROUTINES GENERATE INTERMEDIATE CODE SPECIFYING THE MODE(S)
0010 148 : AND REGISTER(S) USED BY THE REFERENCE.
0010 149 :
0010 150 : THE RESULTS ARE RETURNED AS FOLLOWS:
0010 151 :
0010 152 :     MAC$GB_VAL1    MAC$GB_MODE    MODE OF OPERAND
0010 153 :     MAC$GB_VAL2    MAC$GB_IMODE   ('E' MODE IF INDEX MODE)
0010 154 :     MAC$GB_VAL3    MAC$GB_REG    (REGISTER USED BY OPERAND)
0010 155 :     MAC$GB_VAL4    MAC$GB_IREG   ('E' REGISTER IF INDEX MODE)
0010 156 :
0010 157 : THE VALUE OF 'MAC$GL_PC' IS UPDATED BY THE SIZE OF THE REFERENCE
0010 158 : AND THE ASSOCIATED VALUES, IF ANY.
0010 159 :
0010 160 :--

```

```

0010 162 .SBTTL REFERENCE DIRECTIVE ROUTINES (.REFn)
0010 163
00000000 164 .PSECT MAC$RO_CODE_P1,NOWRT,GBL,LONG
0000 165
0000 166 :++
0000 167 : FUNCTIONAL DESCRIPTION:
0000 168 :
0000 169 : THIS ROUTINE IS INVOKED AFTER A REFERENCE DIRECTIVE HAS
0000 170 : BEEN PROCESSED. THE ROUTINES RFHD1/2/4/8/16 SET UP THE
0000 171 : INFORMATION, THE REFERENCE IS THEN PARSED, AND DIRREF
0000 172 : IS THEN CALLED TO FINISH THE PRODUCTION. THE REFERENCE
0000 173 : IS EMITTED TO THE INTERMEDIATE FILE.
0000 174 :
0000 175 :--
0000 176
0000 177 DIRREF:: ;DIRECTIVE = REF HEAD REF
0000 178 $INTOUT_LW INT$ REF,<W^MAC$GL_VALUE,W^MAC$GL_OPsize> ;OUTPUT REFERENCE
08 0000'CF 91 000E 179 CMPB W^MAC$GL_OPsize,#8 ;QUADWORD REFERENCE?
01 0000'CF 11 12 0013 180 BNEQ 10$ ;IF NEQ NO
01 0000'CF 91 0015 181 CMPB W^MAC$GB_VAL1,#ADMS_IMMEDIATE ;YES--IMMEDIATE?
0A 12 001A 182 BNEQ 10$ ;IF NEQ NO
001C 183 $INTOUT_LW INT$_STIL,<W^MAC$GL_VAL3> ;YES--STORE UPPER 32 BITS
05 0026 184 10$: RSB
0027 185
0027 186 :++
0027 187 : FUNCTIONAL DESCRIPTION:
0027 188 :
0027 189 : THESE ROUTINES (RFHD1/2/4/8) ARE INVOKED WHEN A REFERENCE
0027 190 : DIRECTIVE IS ENCOUNTERED. RFHDX SET THE SIZE OF THE OPERAND
0027 191 : AND INITIALIZE STORAGE FOR PARSING THE EXPRESSION TO FOLLOW.
0027 192 :
0027 193 :--
0027 194
0027 195 RFHD1:: ;REF HEAD = KREF1
0D 10 0027 196 BSBB RFHD ;GO TO COMMON ROUTINE
01 01 0029 197 .BYTE 1 ;OPERAND SIZE IS ONE BYTE
002A 198
002A 199 RFHD2:: ;REF HEAD = KREF2
0A 10 002A 200 BSBB RFHD ;GO TO COMMON ROUTINE
02 02 002C 201 .BYTE 2 ;SIZE OF OPERAND IS 2 BYTES
002D 202
002D 203 RFHD4:: ;REF HEAD = KREF4
07 10 002D 204 BSBB RFHD ;GO TO COMMON ROUTINE
04 04 002F 205 .BYTE 4 ;SIZE OF OPERAND IS 4 BYTES
0030 206
0030 207 RFHD8:: ;REF HEAD = KREF8
04 10 0030 208 BSBB RFHD ;GO TO COMMON ROUTINE
08 08 0032 209 .BYTE 8 ;SIZE OF OPERAND IS 8 BYTES
0033 210
0033 211 RFHD16:: ;REF HEAD = KREF16
01 10 0033 212 BSBB RFHD ;GOTO COMMON ROUTINE
10 10 0035 213 .BYTE 16 ;SIZE OF OPERAND IS 16 BYTES
0036 214
0036 215 RFHD:
0000'CF 9E 9A 0036 216 MOVZBL @(SP)+,W^MAC$GL_OPsize ;SET OPERAND SIZE
0000'CF 0000'CF 9E 003B 217 MOVAB W^MAC$GK_ZERO,W^MAC$GL_MOPPTR ;ALL MODES ARE LEGAL
0000'CF 0000'CF D4 0042 218 CLRL W^MAC$GB_MODE ;CLEAR MODE,IMODE,REG,IREG

```


MAC\$ACTREF
V04-000

OPERAND REFERENCES
REFERENCE DIRECTIVE ROUTINES (.REFn)

G 5

16-SEP-1984 02:00:48 VAX/VMS Macro V04-00
5-SEP-1984 01:47:09 [MACRO.SRC]ACTREF.MAR;1

Page 6
(4)

MA
VC

00000044	8F	C8	0046	219	BISL2	#.LGSM_EVALEXPR!FLGSM_COMPEXPR,-	;EVALUATE EXPRESSION
	6B		004C	220		(R11)	;AND ASSUME COMPILE TIME EXPR
		05	004D	221	RSB		;RETURN TO CALLER'S CALLER

```

004E 223      .SBTTL  OPERAND REFERENCE ROUTINES
004E 224
004E 225      :++
004E 226      : FUNCTIONAL DESCRIPTION:
004E 227      :
004E 228      : REF1 AND REF3 ARE INVOKED WHEN A BASIC REFERENCE OR
004E 229      : AN INDEX REFERENCE IS PROCESSED TO COPY THE MODE AND
004E 230      : REGISTER VALUES INTO MAC$GB_VAL1-MAC$GB_VAL4.
004E 231      :
004E 232      :--
004E 233
004E 234 REF1::      ;REF = BASIC_REF
004E 235 REF3::      ;REF = INDEX_REF
004E 236 REF_EXIT:  ;GENERAL REF_EXIT
0000'CF 0000'CF D0 004E 237      MOVL      W^MAC$GB_MODE,W^MAC$GB_VAL1 ;ENCODE MODES AND REGISTERS
0055 238      ;INTO GB_VAL1-GB_VAL4
0055 239 AUTOI::    ;BASIC_REF = AUTO INC
0055 240 DISP::     ;BASIC_REF = DISPLACEMENT
0055 241      RSB
0056 242
0056 243      :++
0056 244      : FUNCTIONAL DESCRIPTION:
0056 245      :
0056 246      : REF2 IS INVOKED WHEN A BASIC REGISTER REFERENCE IS ENCOUNTERED.
0056 247      : THE MODE (REGISTER MODE) IS ENCODED INTO MAC$GB_MODE, AND THE
0056 248      : REGISTER NUMBER IS SET IN MAC$GB_REG. THE FINAL RESULTS ARE
0056 249      : THEN ENCODED INTO MAC$GB_VAL1-MAC$GB_VAL4.
0056 250      :
0056 251      : INPUTS:
0056 252      :
0056 253      : MAC$GB_VAL1      REGISTER NUMBER
0056 254      :
0056 255      : OUTPUTS:
0056 256      :
0056 257      : MAC$GB_MODE      SET TO 'ADMS REGISTER'
0056 258      : MAC$GB_REG      REGISTER NUMBER
0056 259      : MAC$GB_VAL1-VAL4 ENCODED MODES AND REGISTERS
0056 260      :
0056 261      :--
0056 262
0056 263 REF2::      ;REF = RRREG
0000'CF 0000'CF 05 90 0056 264      MOVB      #ADMS_REGISTER,W^MAC$GB_MODE ;MODE IS REGISTER MODE
0000'CF 0000'CF 90 0058 265      MOVB      W^MAC$GB_VAL1,W^MAC$GB_REG ;GET REGISTER VALUE
0062 266      BSBB      ILL_REG_CHK      ;CHECK FOR ILLEGAL PC USAGE
0064 267      BRB      REF_EXIT      ;FINISH UP

```

```

0066 269 .SBTTL DEFERRED REGISTER MODE OPERAND
0066 270
0066 271 :++
0066 272 : FUNCTIONAL DESCRIPTION:
0066 273 :
0066 274 : THIS ROUTINE IS INVOKED WHEN A DEFERRED REGISTER MODE OPERAND
0066 275 : IS ENCOUNTERED. THE MODE IS SET TO INDIRECT REGISTER, AND
0066 276 : THE REGISTER NUMBER IS STORED.
0066 277 :
0066 278 : INPUTS:
0066 279 :
0066 280 : MAC$AL_VALSTACK-4[R7] REGISTER NUMBER
0066 281 :
0066 282 : OUTPUTS:
0066 283 :
0066 284 : MAC$GB_MODE 'ADMS RRIND'--INDIRECT REGISTER MODE
0066 285 : MAC$GB_REG REGISTER NUMBER
0066 286 :
0066 287 :--
0066 288
0066 289 RDEFER:: :BASIC_REF = DOPN RRREG DCLS
0066 290 MOVB #ADMS RRIND,W^MAC$GB_MODE ;SET MODE AS INDIRECT REG
0066 291 CVTLB W^MAC$AL_VALSTACK-4[R7],- ;STORE REGISTER NUMBER
0073 292 Q^MAC$GB_REG
0073 293 BRB ILL_REG_CHK ;CHECK ILLEGAL PC AND RETURN
0000'CF 06 90 0066 290
0000'CF FFFC'CF47 F6 0068 291
42 11 0073 292
0073 293

```

```

0075 295 :++
0075 296 : FUNCTIONAL DESCRIPTION:
0075 297 :
0075 298 : THIS ROUTINE IS INVOKED WHEN THE PRODUCTION 'DAT AUTO INC'
0075 299 : IS FOUND IN THE TEXT. IF LITERAL MODE IS USED IT IS FORCED
0075 300 : TO IMMEDIATE LONGWORD. THE MODE IS SET AS ABSOLUTE.
0075 301 :
0075 302 :--
0075 303 :
0075 304 AUTOI1:: :BASIC_REF = DAT AUTO INC
03F8 30 0075 305 BSBW SPESHL_MODE_CHK :FORCE_LONGWORD TO IMMEDIATE
03F5 30 0078 306 BSBW SPESHL_MODE_CHK :FORCE LONGWORD CONTEXT
0000'CF 96 0078 307 INCB W^MAC$GB_MODE :MAKE INDIRECT MODE
05 007F 308 RSB
0080 309 :
0080 310 :++
0080 311 : FUNCTIONAL DESCRIPTION:
0080 312 :
0080 313 : THIS ROUTINE IS INVOKED WHEN AN AUTO-DECREMENT ADDRESSING MODE
0080 314 : IS ENCOUNTERED. THE MODE AND REGISTER ARE SET UP, AND A CHECK
0080 315 : IS MADE FOR ILLEGAL USE OF 'PC'.
0080 316 :
0080 317 : INPUTS:
0080 318 :
0080 319 : MAC$AL_VALSTACK-4[R7] REGISTER NUMBER
0080 320 :
0080 321 : OUTPUTS:
0080 322 :
0080 323 : MAC$GB_MODE 'ADMS_REGAUTODEC'
0080 324 : MAC$GB_REG REGISTER NUMBER
0080 325 :
0080 326 :--
0080 327 :
0080 328 AUTODC:: :BASIC_REF = DMINUS DOPN RRREG DCLS
0000'CF 07 90 0080 329 MOVB #ADMS_REGAUTODEC,W^MAC$GB_MODE;SET MODE
0000'CF FFFC'CF47 F6 0085 330 CVTLB W^MAC$AL_VALSTACK-4[R7],-;PICK UP REGISTER NUMBER
008D 331 W^MAC$GB_REG
28 11 008D 332 BRB ILL_REG_CHK :CHECK ILLEGAL REGISTER AND RETURN

```

.SBTTL INDIRECT DISPLACEMENT REFERENCES

```

008F 334
008F 335
008F 336 :++
008F 337 : FUNCTIONAL DESCRIPTION:
008F 338 :
008F 339 : THIS ROUTINE PROCESSES INDIRECT DISPLACEMENT REFERENCES.
008F 340 : ABSOLUTE AND PIC MODES ARE CHANGED TO LONGWORD DISPLACEMENT,
008F 341 : LONGWORD DISPLACEMENTS ARE FORCE TO IMMEDIATE LONGWORD
008F 342 : CONTEXT. THE MODE IS THEN SET TO DEFERRED DISPLACEMENT MODE.
008F 343 :

```

```

008F 344 : INPUTS:
008F 345 :
008F 346 : MAC$GB_MODE MODE OF DISPLACEMENT
008F 347 :

```

```

008F 348 : OUTPUTS:
008F 349 :
008F 350 : MAC$GB_MODE DEFERRED DISPLACEMENT MODE
008F 351 : MAC$GL_PC UPDATED IF NECESSARY
008F 352 :

```

```

008F 353 :--
008F 354 :

```

```

50 0000'CF 9E 008F 355 DISPI::
02 60 91 0094 356 MOVAB W^MAC$GB_MODE,R0 ;BASIC REF = DAT DISPLACEMENT
05 12 0097 357 CMPB (R0),#ADMS_ABSOLUTE ;GET ADDRESS OF MODE
60 0E 90 0099 358 BNEQ 20$ ;ABSOLUTE MODE?
11 11 009C 359 10$: MOVB #ADMS_LONG_DISP,(R0) ;IF NEQ NO
03 60 91 009E 360 BRB 30$ ;YES--FORCE LONGWORD DISPLACEMENT
06 60 91 00A1 361 20$: CMPB (R0),#ADMS_PIC ;PIC CODE?
07 12 00A3 362 BEQL 10$ ;IF EQL YES
0A 90 00A6 363 CMPB (R0),#ADMS_RRIND ;IF (R) THEN DE-OPTIMIZE
03BE 30 00AF 364 BNEQ 30$ ;NOT (R)
0000'CF 96 00B2 365 $INC_PC ;ONE MORE BYTE
05 00B6 366 MOVB #ADMS_BYTE_DISP,(R0) ;USE BYTE DISPLACEMENT
05 00B6 367 30$: BSBW SPESHC_MODE_CHK ;CHECK MODES FOR DEFERRAL
05 00B6 368 INCB W^MAC$GB_MODE ;MAKE DEFERRED DISPLACEMENT
05 00B6 369 RSB

```

```

00B7 371      .SBTTL CHECK FOR ILLEGAL USE OF PC
00B7 372
00B7 373      :++
00B7 374      : FUNCTIONAL DESCRIPTION:
00B7 375      :
00B7 376      : THIS ROUTINE CHECKS FOR ILLEGAL USE OF PC. IF THE REGISTER
00B7 377      : IS PC, AN ERROR MESSAGE IS SENT TO PASS 2. THE LOCATION COUNTER
00B7 378      : IS BUMPED BY 1.
00B7 379
00B7 380      : INPUTS:
00B7 381
00B7 382      : MAC$GB_REG          REGISTER IN QUESTION
00B7 383      : MAC$GB_MODE        CURRENT MODE
00B7 384
00B7 385      : OUTPUTS:
00B7 386
00B7 387      : MAC$GL_PC          INCREMENTED BY 1
00B7 388
00B7 389      :--
00B7 390
00B7 391      ILL_REG_CHK:
05 0000'CF 91 00B7 392      CMPB  W^MAC$GB_REG,#REG$PC ;'PC' IS ILLEGAL REGISTER HERE
14 6B 25 E4 00BC 393      BNEQ  10$ ;IF NEQ THEN LEGAL REGISTER
00C2 394      BBSC  #FLG$V_NTYPEPC,(R11),10$ ; If .NTYPE directive no error
05 0000'CF 91 00C7 395      $MAC_ERR ILLREGHERE ; Load message code
05 0000'CF 91 00C7 396      CMPB  W^MAC$GB_MODE,#ADMS_REGISTER ; Is addressing mode register?
05 0000'CF 91 00C7 397      BNEQ  5$ ; No if NEQ
05 0000'CF 91 00C7 398      BSBW  MAC$ERRORPX ; Send error to pass-2
05 0000'CF 91 00C7 399      BRB  10$
05 0000'CF 91 00D3 400 5$:
05 0000'CF 91 00D3 401      BSBW  MAC$ERRORPT ; SEND ERROR TO PASS2
05 0000'CF 91 00D6 402 10$:
05 0000'CF 91 00DA 403      $INC_PC ;ADD ONE TO LOCATION COUNTER
05 0000'CF 91 00DA 403      RSB

```

```

00DB 405 .SBTTL DISPLACEMENT REFERENCES
00DB 406
00DB 407 :++
00DB 408 : FUNCTIONAL DESCRIPTION:
00DB 409 :
00DB 410 : THIS ROUTINE PROCESSES DISPLACEMENTS OFF OF A SPECIFIC
00DB 411 : REGISTER. IF THE REGISTER IS 'PC', THE OPERAND IS
00DB 412 : TREATED AS 'EXPR'. IF THE VALUE OF THE EXPRESSION IS 0
00DB 413 : AND WE CAN OPTIMIZE EXPRESSION, THE CODE FOR THE EXPRESSION
00DB 414 : IS DELETED FROM THE INTERMEDIATE BUFFER. THE DISPLACEMENT
00DB 415 : IS SET TO BYTE, WORD, OR LONGWORD WITH WORD DISPLACEMENT
00DB 416 : THE DEFAULT.
00DB 417 :
00DB 418 : INPUTS:
00DB 419 :
00DB 420 : MAC$AL_VALSTACK-12[R7] EXPRESSION VALUE TO CONSIDER
00DB 421 : MAC$AL_VALSTACK-4[R7] REGISTER BEING DISPLACED
00DB 422 :
00DB 423 : OUTPUTS:
00DB 424 :
00DB 425 : MAC$GB_MODE SET WITH CORRECT MODE
00DB 426 : MAC$GL_PC UPDATED BY SIZE OF OPERAND
00DB 427 :
00DB 428 :--
00DB 429 :
00DB 430 DISPL1::
51 FFF4'CF47 D0 00DB 431 MOVL W^MAC$AL_VALSTACK-12[R7],R1 ;DISPLACEMENT = EXPR DOPN RRREG DCLS
   56 51 D0 00E1 432 MOVL R1,R6 ;GET EXPRESSION TO OPTIMIZE
52 FFFC'CF47 D0 00E4 433 MOVL W^MAC$AL_VALSTACK-4[R7],R2 ;COPY VALUE IN CASE REGISTER IS PC
   0000'CF 52 90 00EA 434 MOVWB R2,W^MAC$GB_REG ;GET REGISTER BEING DISPLACED
   OF 52 91 00EF 435 CMPB R2,#REG$PC ;STORE FOR WORLD TO SEE
   09 12 00F2 436 BNEQ 5$ ;IS REGISTER 'PC'?
   0000'CF D5 00F4 437 TSTL W^MAC$GL_ABSFLAG ; No if NEQ
   03 13 00F8 438 BEQL 5$ ; Is expression absolute?
   0087 31 00FA 439 BRW DISPPC ; Yes if EQL
   00FD 440 5$:
   0000'CF 51 D0 00FD 441 MOVL R1,W^MAC$GL_EXPOPVL1 ;LOAD EXPR VALUE
   0000'CF 0C 90 0102 442 MOVWB #ADMS_WORD_DISP,W^MAC$GB_MODE ;SET DEFAULT DISPLACEMENT
   56 03 9A 0107 443 MOVZBL #3,R6 ;DEFAULT IS 3 BYTES LONG
   0000'CF D5 010A 444 TSTL W^MAC$GL_ABSFLAG ;ABSOLUTE EXPRESSION?
   0E 13 010E 445 BEQL 10$ ;IF EQL YES
   6B 04 CA 0110 446 BICL2 #FLG$M COMEXPR,(R11) ;NO--NOT COMPILE TIME EXPRESSION
51 FFFF0000 8F D3 0113 447 BITL #^XFFF0000,R1 ;REQUIRE LONGWORD?
   3E 13 011A 448 BEQL 50$ ;IF EQL NO--LINKER MAY COMPLAIN THO
   34 11 011C 449 BRB 40$ ;YES--SET LONGWORD AND EXIT
   51 D5 011E 450 10$: TSTL R1 ;VALUE 0?
   OF 12 0120 451 BNEQ 20$ ;IF NEQ NO
   OB 6B 07 E1 0122 452 BBC #FLG$V EXPOPT,(R11),20$ ;BRANCH IF CANNOT OPTIMIZE EXPR
   FED7' 30 0126 453 BSBW MAC$OPTIMIZEEXPR ;WE CAN--WIPE OUT EXPRESSION
   0000'CF 59 D0 0129 454 MOVL R9,W^MAC$GL_EXPEND ;SET NEW END OF EXPR POINTER
   FF35 31 012E 455 BRW W^RDEFER ;USE REGISTER DEFERRED MODE
   03 18 0131 456 20$: bgeq 30$ ; If negative
   51 51 D2 0133 457 mcoml r1,r1 ; then make it positive magnitude.
00007FFF 8F 51 D1 0136 458 30$: cmpl r1,#^x7fff ; Byte or Word Displacement?
   13 1A 013D 459 bgtru 40$ ; If GTRU no.
0000007F 8F 51 D1 013F 460 cmpl r1,#^x7f ; Yes: Byte Displacement?
   12 1A 0146 461 bgtru 50$ ; If GTRU no: Word Disp. and PC OK.

```



```

0228 562 .SBTTL EXPLICIT DISPLACEMENTS AND PIC REFERENCES
0228 563
0228 564 :++
0228 565 : FUNCTIONAL DESCRIPTION:
0228 566 :
0228 567 : THESE ROUTINES (DISPL3/4/5) ARE INVOKED WHEN AN EXPLICIT
0228 568 : DISPLACEMENT REFERENCE IS ENCOUNTERED. THE CORRECT MODE
0228 569 : IS SET, THE REGISTER IS SET, AND THE PC IS UPDATED BY
0228 570 : THE SIZE OF THE OPERAND.
0228 571 :
0228 572 : INPUTS:
0228 573 :
0228 574 : MAC$AL_VALSTACK-4[R7] REGISTER VALUE
0228 575 : MAC$AL_VALSTACK-12[R7] EXPRESSION VALUE
0228 576 :
0228 577 : OUTPUTS:
0228 578 :
0228 579 : MAC$GB_MODE SET TO CORRECT MODE
0228 580 : MAC$GB_REG SET TO REGISTER
0228 581 : MAC$GL_PC UPDATED BY SIZE OF OPERAND
0228 582 :
0228 583 :--
0228 584
0228 585 .ENABL LSB
0228 586
0228 587 DISPL3:: ;DISPLACEMENT = DBUP EXPR DOPN RRREG DCLS
50 0A 9A 0228 588 MOVZBL #ADMS_BYTE_DISP,R0 ;SET BYTE DISPLACEMENT
51 02 9A 0228 589 MOVZBL #2,R1 ;LENGTH OF OPERAND
0E 11 022E 590 BRB 10$
0230 591
0230 592 DISPL4:: ;DISPLACEMENT = DLUP EXPR DOPN RRREG DCLS
50 0E 9A 0230 593 MOVZBL #ADMS_LONG_DISP,R0 ;SET LONGWORD DISPLACEMENT
51 05 9A 0233 594 MOVZBL #5,R1 ;LENGTH OF OPERAND
06 11 0236 595 BRB 10$
0238 596
0238 597 DISPL5:: ;DISPLACEMENT = DWUP EXPR DOPN RRREG DCLS
50 0C 9A 0238 598 MOVZBL #ADMS_WORD_DISP,R0 ;SET WORD DISPLACEMENT
51 03 9A 023B 599 MOVZBL #3,R1 ;LENGTH OF OPERAND
0000'CF FFFC'CF47 F6 023E 600 10$: CVTLB W^MAC$AL_VALSTACK-4[R7],- ;GET THE REGISTER VALUE
0000'CF FFF4'CF47 D0 0246 601 W^MAC$GB_REG ;
0246 602 MOVL W^MAC$AL_VALSTACK-12[R7],- ;COPY EXPRESSION VALUE
024E 603 W^MAC$GL_EXPOPVL1
0000'CF D5 024E 604 TSTL W^MAC$GL_ABSFLAG ;ABSOLUTE EXPRESSION?
03 13 0252 605 BEQL 20$ ;IF EQL YES
6B 04 CA 0254 606 BICL2 #FLG$M_COMPEXPR,(R11) ;ELSE NOT COMPILE TIME EXPRESSION
0257 607 DISPLX_EXIT:
0000'CF 50 90 0257 608 20$: MOVB R0,W^MAC$GB_MODE ;SET MODE OF OPERAND
025C 609 $INC_PC R1 ;UPDATE PC BY SIZE OF OPERAND
0261 610 RSB
0262 611
0262 612 .DSABL LSB

```

```

0262 614 :++
0262 615 : FUNCTIONAL DESCRIPTION:
0262 616 :
0262 617 : THESE ROUTINES (DISPL6/7/8 AND GENLDS) ARE INVOKED WHEN
0262 618 : A SPECIFIC SIZE DISPLACEMENT REFERENCE OR A SPECIFIC PIC
0262 619 : REFERENCE IS ENCOUNTERED. THE REGISTER IS SET TO 'PC',
0262 620 : AND THE PC IS UPDATED BY THE SIZE OF THE OPERAND.
0262 621 :
0262 622 : INPUTS:
0262 623 :
0262 624 : MAC$AL_VALSTACK[R7] EXPRESSION VALUE
0262 625 :
0262 626 : OUTPUTS:
0262 627 :
0262 628 : MAC$GB_MODE SET TO MODE
0262 629 : MAC$GB_REG SET TO 'PC' (15.)
0262 630 : MAC$GL_PC UPDATED BY SIZE OF OPERAND
0262 631 : MAC$GL_EXPOPVL1 SET TO VALUE OF EXPRESSION
0262 632 :
0262 633 :--
0262 634 :
0262 635 : .ENABL LSB
0262 636 :
0262 637 DISPL6:: :DISPLACEMENT = DBUP EXPR
50 0A 9A 0262 638 MOVZBL #ADMS_BYTE_DISP,R0 :SET BYTE DISPLACEMENT
51 02 9A 0265 639 MOVZBL #2,R1 :SET SIZE OF OPERAND
16 11 0268 640 BRB 10$
026A 641
026A 642 DISPL7:: :DISPLACEMENT = DLUP EXPR
50 0E 9A 026A 643 MOVZBL #ADMS_LONG_DISP,R0 :SET LONGWORD DISPLACEMENT
51 05 9A 026D 644 MOVZBL #5,R1 :SIZE IS 5 BYTES
0E 11 0270 645 BRB 10$
0272 646
0272 647 DISPL8:: :DISPLACEMENT = DWUP EXPR
50 0C 9A 0272 648 MOVZBL #ADMS_WORD_DISP,R0 :SET WORD DISPLACEMENT
51 03 9A 0275 649 MOVZBL #3,R1 :SIZE IS 3 BYTES
06 11 0278 650 BRB 10$
027A 651
027A 652 GENLDS:: :DISPLACEMENT = DGUP EXPR
50 03 9A 027A 653 MOVZBL #ADMS_PIC,R0 :SET PIC MODE
51 05 9A 027D 654 MOVZBL #5,R1 :LENGTH IS 5 BYTES
0000'CF 0F 90 0280 655 10$: MOVB #REGS_PC,W^MAC$GB_REG :SET REGISTER TO PC
0000'CF 0000'CF47 D0 0285 656 MOVL W^MAC$AL_VALSTACK[R7],- :COPY EXPRESSION
028D 657 W^MAC$GL_EXPOPVL1
0000'CF D5 028D 658 TSTL W^MAC$GL_ABSFLAG :ABSOLUTE EXPRESSION?
C4 12 0291 659 BNEQ DISPLX_EXIT :IF NEQ NO--GO FINISH UP
6B 04 CA 0293 660 BICL2 #FLG$M_COMPEXPR,(R11) :YES--ABS REQUIRES LINKER ACTION
BF 11 0296 661 BRB DISPLX_EXIT :GO EXIT
0298 662
0298 663 :.DSABL LSB

```

```

0298 665 .SBTTL LITERAL MODE
0298 666
0298 667 :++
0298 668 : FUNCTIONAL DESCRIPTION:
0298 669 :
0298 670 : THIS ROUTINE IS INVOKED WHEN A LITERAL REFERENCE IS SCANNED.
0298 671 : IF THE OPERAND IS A FLOATING POINT OPERAND, IT IS CONVERTED
0298 672 : TO A SHORT FLOATING LITERAL.
0298 673 :
0298 674 : INPUTS:
0298 675 :
0298 676 : MAC$AL_VALSTACK[R7] VALUE OF EXPRESSION
0298 677 :
0298 678 :--
0298 679
0298 680 LITERL::
0298 681 TSTL W^MAC$GL_ABSFLAG ;REF = DSUP DPOUND EXPR
0298 682 BEQL 10$ ;ABSOLUTE?
0298 683 BBCC #FLGSV_EXPOPT,(R11),10$ ;IF EQL YES
0298 684 10$: MOVB #ADMS_LITERAL,W^MAC$GB_MODE ;NO--DO NOT ALLOW OPTIMIZATION
0298 685 $INC_PC ;SET LITERAL MODE
0298 686 MOVL W^MAC$AL_VALSTACK[R7],- ;COUNT 1 BYTE
0298 687 W^MAC$GL_EXPOPVL1 ;COPY THE VALUE
0298 688 CMPB W^MAC$GB_RDXNDX,- ;READING FLT. PT.?
0298 689 #RDXSV_FFLOAT
0298 690 BLSSU 40$ ;IF LESS THEN NO
0298 691 MOVL W^MAC$GL_EXPOPVL1,R0 ;MOVE FLOATING PT. NUMBER INTO R0
0298 692 CMPB W^MAC$GB_RDXNDX,- ;H FLOAT?
0298 693 #RDXSV_HFLOAT
0298 694 BNEQ 20$ ;NO, THEN SKIP TO OTHER FLT. PT. CODE
0298 695 EXTZV #29,#3,R0,R1 ;GET FRACTION
0298 696 EXTZV #0,#3,R0,R2 ;GET EXPONENT
0298 697 INSV R2,#3,#3,R1 ;COMBINE THEM
0298 698 MOVZBL R1,W^MAC$GL_EXPOPVL1 ;STORE RESULT
0298 699 BRB 40$ ;DONE
0298 700 20$: CMPB W^MAC$GB_RDXNDX,- ;G FLOAT?
0298 701 #RDXSV_GFLOAT
0298 702 BNEQ 30$ ;NO, SKIP TO F AND D FLOAT COMMON CODE
0298 703 EXTZV #1,#6,R0,- ;YES, STORE RESULT
0298 704 W^MAC$GL_EXPOPVL1
0298 705 BRB 40$
0298 706 30$: EXTZV #4,#6,R0,- ;F OR D FLOAT, STORE RESULT
0298 707 W^MAC$GL_EXPOPVL1
0298 708 40$: BRW REF_EXIT ;EXIT FROM REFERENCE
0298 709
0298 710
0298 711
0298 712
0298 713
0298 714
0298 715
0298 716
0298 717
0298 718
0298 719
0298 720
0298 721
0298 722
0298 723
0298 724
0298 725
0298 726
0298 727
0298 728
0298 729
0298 730
0298 731
0298 732
0298 733
0298 734
0298 735
0298 736
0298 737
0298 738
0298 739
0298 740
0298 741
0298 742
0298 743
0298 744
0298 745
0298 746
0298 747
0298 748
0298 749
0298 750
0298 751
0298 752
0298 753
0298 754
0298 755
0298 756
0298 757
0298 758
0298 759
0298 760
0298 761
0298 762
0298 763
0298 764
0298 765
0298 766
0298 767
0298 768
0298 769
0298 770
0298 771
0298 772
0298 773
0298 774
0298 775
0298 776
0298 777
0298 778
0298 779
0298 780
0298 781
0298 782
0298 783
0298 784
0298 785
0298 786
0298 787
0298 788
0298 789
0298 790
0298 791
0298 792
0298 793
0298 794
0298 795
0298 796
0298 797
0298 798
0298 799
0298 800
0298 801
0298 802
0298 803
0298 804
0298 805
0298 806
0298 807
0298 808
0298 809
0298 810
0298 811
0298 812
0298 813
0298 814
0298 815
0298 816
0298 817
0298 818
0298 819
0298 820
0298 821
0298 822
0298 823
0298 824
0298 825
0298 826
0298 827
0298 828
0298 829
0298 830
0298 831
0298 832
0298 833
0298 834
0298 835
0298 836
0298 837
0298 838
0298 839
0298 840
0298 841
0298 842
0298 843
0298 844
0298 845
0298 846
0298 847
0298 848
0298 849
0298 850
0298 851
0298 852
0298 853
0298 854
0298 855
0298 856
0298 857
0298 858
0298 859
0298 860
0298 861
0298 862
0298 863
0298 864
0298 865
0298 866
0298 867
0298 868
0298 869
0298 870
0298 871
0298 872
0298 873
0298 874
0298 875
0298 876
0298 877
0298 878
0298 879
0298 880
0298 881
0298 882
0298 883
0298 884
0298 885
0298 886
0298 887
0298 888
0298 889
0298 890
0298 891
0298 892
0298 893
0298 894
0298 895
0298 896
0298 897
0298 898
0298 899
0298 900
0298 901
0298 902
0298 903
0298 904
0298 905
0298 906
0298 907
0298 908
0298 909
0298 910
0298 911
0298 912
0298 913
0298 914
0298 915
0298 916
0298 917
0298 918
0298 919
0298 920
0298 921
0298 922
0298 923
0298 924
0298 925
0298 926
0298 927
0298 928
0298 929
0298 930
0298 931
0298 932
0298 933
0298 934
0298 935
0298 936
0298 937
0298 938
0298 939
0298 940
0298 941
0298 942
0298 943
0298 944
0298 945
0298 946
0298 947
0298 948
0298 949
0298 950
0298 951
0298 952
0298 953
0298 954
0298 955
0298 956
0298 957
0298 958
0298 959
0298 960
0298 961
0298 962
0298 963
0298 964
0298 965
0298 966
0298 967
0298 968
0298 969
0298 970
0298 971
0298 972
0298 973
0298 974
0298 975
0298 976
0298 977
0298 978
0298 979
0298 980
0298 981
0298 982
0298 983
0298 984
0298 985
0298 986
0298 987
0298 988
0298 989
0298 990
0298 991
0298 992
0298 993
0298 994
0298 995
0298 996
0298 997
0298 998
0298 999
0298 1000
0298 1001
0298 1002
0298 1003
0298 1004
0298 1005
0298 1006
0298 1007
0298 1008
0298 1009
0298 1010
0298 1011
0298 1012
0298 1013
0298 1014
0298 1015
0298 1016
0298 1017
0298 1018
0298 1019
0298 1020
0298 1021
0298 1022
0298 1023
0298 1024
0298 1025
0298 1026
0298 1027
0298 1028
0298 1029
0298 1030
0298 1031
0298 1032
0298 1033
0298 1034
0298 1035
0298 1036
0298 1037
0298 1038
0298 1039
0298 1040
0298 1041
0298 1042
0298 1043
0298 1044
0298 1045
0298 1046
0298 1047
0298 1048
0298 1049
0298 1050
0298 1051
0298 1052
0298 1053
0298 1054
0298 1055
0298 1056
0298 1057
0298 1058
0298 1059
0298 1060
0298 1061
0298 1062
0298 1063
0298 1064
0298 1065
0298 1066
0298 1067
0298 1068
0298 1069
0298 1070
0298 1071
0298 1072
0298 1073
0298 1074
0298 1075
0298 1076
0298 1077
0298 1078
0298 1079
0298 1080
0298 1081
0298 1082
0298 1083
0298 1084
0298 1085
0298 1086
0298 1087
0298 1088
0298 1089
0298 1090
0298 1091
0298 1092
0298 1093
0298 1094
0298 1095
0298 1096
0298 1097
0298 1098
0298 1099
0298 1100
0298 1101
0298 1102
0298 1103
0298 1104
0298 1105
0298 1106
0298 1107
0298 1108
0298 1109
0298 1110
0298 1111
0298 1112
0298 1113
0298 1114
0298 1115
0298 1116
0298 1117
0298 1118
0298 1119
0298 1120
0298 1121
0298 1122
0298 1123
0298 1124
0298 1125
0298 1126
0298 1127
0298 1128
0298 1129
0298 1130
0298 1131
0298 1132
0298 1133
0298 1134
0298 1135
0298 1136
0298 1137
0298 1138
0298 1139
0298 1140
0298 1141
0298 1142
0298 1143
0298 1144
0298 1145
0298 1146
0298 1147
0298 1148
0298 1149
0298 1150
0298 1151
0298 1152
0298 1153
0298 1154
0298 1155
0298 1156
0298 1157
0298 1158
0298 1159
0298 1160
0298 1161
0298 1162
0298 1163
0298 1164
0298 1165
0298 1166
0298 1167
0298 1168
0298 1169
0298 1170
0298 1171
0298 1172
0298 1173
0298 1174
0298 1175
0298 1176
0298 1177
0298 1178
0298 1179
0298 1180
0298 1181
0298 1182
0298 1183
0298 1184
0298 1185
0298 1186
0298 1187
0298 1188
0298 1189
0298 1190
0298 1191
0298 1192
0298 1193
0298 1194
0298 1195
0298 1196
0298 1197
0298 1198
0298 1199
0298 1200
0298 1201
0298 1202
0298 1203
0298 1204
0298 1205
0298 1206
0298 1207
0298 1208
0298 1209
0298 1210
0298 1211
0298 1212
0298 1213
0298 1214
0298 1215
0298 1216
0298 1217
0298 1218
0298 1219
0298 1220
0298 1221
0298 1222
0298 1223
0298 1224
0298 1225
0298 1226
0298 1227
0298 1228
0298 1229
0298 1230
0298 1231
0298 1232
0298 1233
0298 1234
0298 1235
0298 1236
0298 1237
0298 1238
0298 1239
0298 1240
0298 1241
0298 1242
0298 1243
0298 1244
0298 1245
0298 1246
0298 1247
0298 1248
0298 1249
0298 1250
0298 1251
0298 1252
0298 1253
0298 1254
0298 1255
0298 1256
0298 1257
0298 1258
0298 1259
0298 1260
0298 1261
0298 1262
0298 1263
0298 1264
0298 1265
0298 1266
0298 1267
0298 1268
0298 1269
0298 1270
0298 1271
0298 1272
0298 1273
0298 1274
0298 1275
0298 1276
0298 1277
0298 1278
0298 1279
0298 1280
0298 1281
0298 1282
0298 1283
0298 1284
0298 1285
0298 1286
0298 1287
0298 1288
0298 1289
0298 1290
0298 1291
0298 1292
0298 1293
0298 1294
0298 1295
0298 1296
0298 1297
0298 1298
0298 1299
0298 1300
0298 1301
0298 1302
0298 1303
0298 1304
0298 1305
0298 1306
0298 1307
0298 1308
0298 1309
0298 1310
0298 1311
0298 1312
0298 1313
0298 1314
0298 1315
0298 1316
0298 1317
0298 1318
0298 1319
0298 1320
0298 1321
0298 1322
0298 1323
0298 1324
0298 1325
0298 1326
0298 1327
0298 1328
0298 1329
0298 1330
0298 1331
0298 1332
0298 1333
0298 1334
0298 1335
0298 1336
0298 1337
0298 1338
0298 1339
0298 1340
0298 1341
0298 1342
0298 1343
0298 1344
0298 1345
0298 1346
0298 1347
0298 1348
0298 1349
0298 1350
0298 1351
0298 1352
0298 1353
0298 1354
0298 1355
0298 1356
0298 1357
0298 1358
0298 1359
0298 1360
0298 1361
0298 1362
0298 1363
0298 1364
0298 1365
0298 1366
0298 1367
0298 1368
0298 1369
0298 1370
0298 1371
0298 1372
0298 1373
0298 1374
0298 1375
0298 1376
0298 1377
0298 1378
0298 1379
0298 1380
0298 1381
0298 1382
0298 1383
0298 1384
0298 1385
0298 1386
0298 1387
0298 1388
0298 1389
0298 1390
0298 1391
0298 1392
0298 1393
0298 1394
0298 1395
0298 1396
0298 1397
0298 1398
0298 1399
0298 1400
0298 1401
0298 1402
0298 1403
0298 1404
0298 1405
0298 1406
0298 1407
0298 1408
0298 1409
0298 1410
0298 1411
0298 1412
0298 1413
0298 1414
0298 1415
0298 1416
0298 1417
0298 1418
0298 1419
0298 1420
0298 1421
0298 1422
0298 1423
0298 1424
0298 1425
0298 1426
0298 1427
0298 1428
0298 1429
0298 1430
0298 1431
0298 1432
0298 1433
0298 1434
0298 1435
0298 1436
0298 1437
0298 1438
0298 1439
0298 1440
0298 1441
0298 1442
0298 1443
0298 1444
0298 1445
0298 1446
0298 1447
0298 1448
0298 1449
0298 1450
0298 1451
0298 1452
0298 1453
0298 1454
0298 1455
0298 1456
0298 1457
0298 1458
0298 1459
0298 1460
0298 1461
0298 1462
0298 1463
0298 1464
0298 1465
0298 1466
0298 1467
0298 1468
0298 1469
0298 1470
0298 1471
0298 1472
0298 1473
0298 1474
0298 1475
0298 1476
0298 1477
0298 1478
0298 1479
0298 1480
0298 1481
0298 1482
0298 1483
0298 1484
0298 1485
0298 1486
0298 1487
0298 1488
0298 1489
0298 1490
0298 1491
0298 1492
0298 1493
0298 1494
0298 1495
0298 1496
0298 1497
0298 1498
0298 1499
0298 1500
0298 1501
0298 1502
0298 1503
0298 1504
0298 1505
0298 1506
0298 1507
0298 1508
0298 1509
0298 1510
0298 1511
0298 1512
0298 1513
0298 1514
0298 1515
0298 1516
0298 1517
0298 1518
0298 1519
0298 1520
0298 1521
0298 1522
0298 1523
0298 1524
0298 1525
0298 1526
0298 1527
0298 1528
0298 1529
0298 1530
0298 1531
0298 1532
0298 1533
0298 1534
0298 1535
0298 1536
0298 1537
0298 1538
0298 1539
0298 1540
0298 1541
0298 1542
0298 1543
0298 1544
0298 1545
0298 1546
0298 1547
0298 1548
0298 1549
0298 1550
0298 1551
0298 1552
0298 1553
0298 1554
0298 1555
0298 1556
0298 1557
0298 1558
0298 1559
0298 1560
0298 1561
0298 1562
0298 1563
0298 1564
0298 1565
0298 1566
0298 1567
0298 1568
0298 1569
0298 1570
0298 1571
0298 1572
0298 1573
0298 1574
0298 1575
0298 1576
0298 1577
0298 1578
0298 1579
0298 1580
0298 1581
0298 1582
0298 1583
0298 1584
0298 1585
0298 1586
0298 1587
0298 1588
0298 1589
0298 1590
0298 1591
0298 1592
0298 1593
0298 1594
0298 1595
0298 1596
0298 1597
0298 1598
0298 1599
0298 1600
0298 1601
0298 1602
0298 1603
0298 1604
0298 1605
0298 1606
0298 1607
0298 1608
0298 1609
0298 1610
0298 1611
0298 1612
0298 1613
0298 1614
0298 1615
0298 1616
0298 1617
0298 1618
0298 1619
0298 1620
0298 1621
0298 1622
0298 1623
0298 1624
0298 1625
0298 1626
0298 1627
0298 1628
0298 1629
0298 1630
0298 1631
0298 1632
0298 1633
0298 1634
0298 1635
0298 1636
0298 1637
0298 1638
0298 1639
0298 1640
0298 1641
0298 1642
0298 1643
0298 1644
0298 1645
0298 1646
0298 1647
0298 1648
0298 1649
0298 1650
0298 1651
0298 1652
0298 1653
0298 1654
0298 1655
0298 1656
0298 1657
0298 1658
0298 1659
0298 1660
0298 1661
0298 1662
0298 1663
0298 1664
0298 1665
0298 1666
0298 1667
0298 1668
0298 1669
0298 1670
0298 1671
0298 1672
0298 1673
0298 1674
0298 1675
0298 1676
0298 1677
0298 1678
0298 1679
0298 1680
0298 1681
0298 1682
0298 1683
0298 1684
0298 1685
0298 1686
0298 1687
0298 1688
0298 1689
0298 1690
0298 1691
0298 1692
0298 1693
0298 1694
0298 1695
0298 1696
0298 1697
0298 1698
0298 1699
0298 1700
0298 1701
0298 1702
0298 1703
0298 1704
0298 1705
0298 1706
0298 1707
0298 1708
0298 1709
0298 1710
0298 1711
0298 1712
0298 1713
0298 1714
0298 1715
0298 1716
0298 1717
0298 1718
0298 1719
0298 1720
0298 1721
0298 1722
0298 1723
0298 1724
0298 1725
0298 1726
0298 1727
0298 1728
0298 1729
0298 1730
0298 1731
0298 1732
0298 1733
0298 1734
0298 1735
0298 1736
0298 1737
0298 1738
0298 1739
0298 1740
0298 1741
0298 1742
0298 1743
0298 1744
0298 1745
0298 1746
0298 1747
0298 1748
0298 1749
0298 1750
0298 1751
0298 1752
0298 1753
0298 1754
0298 1755
0298 1756
0298 1757
0298 1758
0298 1759
0298 1760
0298 1761
0298 1762
0298 1763
0298 1764
0298 1765
0298 1766
0298 1767
0298 1768
0298 1769
0298 1770
0298 1771
0298 1772
0298 1773
0298 1774
0298 1775
0298 1776
0298 1777
0298 1778
0298 1779
0298 1780
0298 1781
0298 1782
0298 1783
0298 1784
0298 1785
0298 1786
0298 1787
0298 1788
0298 1789
0298 1790
0298 1791
0298 1792
0298 1793
0298 1794
0298 1795
0298 1796
0298 1797
0298 1798
0298 1799
0298 1800
0298 1801
0298 1802
0298 1803
0298 1804
0298 1805
0298 1806
0298 1807
0298 1808
0298 1809
0298 1810
0298 1811
0298 1812
0298 1813
0298 1814
0298 1815
0298 1816
0298 1817
0298 1818
0298 1819
0298 1820
0298 1821
0298 1822
0298 1823
0298 1824
0298 1825
0298 1826
0298 1827
0298 1828
0298 1829
0298 1830
0298 1831
0298 1832
0298 1833
0298 1834
0298 1835
0298 1836
0298 1837
0298 1838
0298 1839
0298 1840
0298 1841
0298 1842
0298 1843
0298 1844
0298 1845
0298 1846
0298 1847
0298 1848
0298 1849
0298 1850
0298 1851
0298 185
```

```

02F6 710 .SBTTL AUTO-INCREMENT MODE
02F6 711
02F6 712 :++
02F6 713 : FUNCTIONAL DESCRIPTION:
02F6 714 :
02F6 715 : THIS ROUTINE IS CALLED WHEN AN AUTO-INCREMENT REFERENCE IS
02F6 716 : SCANNED. THE MODE AND REGISTER ARE SET, AND A CHECK IS MADE
02F6 717 : FOR AN ILLEGAL REGISTER REFERENCE.
02F6 718 :
02F6 719 : INPUTS:
02F6 720 :
02F6 721 : MAC$AL_VALSTACK-8[R7] REGISTER NUMBER
02F6 722 :
02F6 723 : OUTPUTS:
02F6 724 :
02F6 725 : MAC$GB_MODE 'ADMS REGAUTOINC'
02F6 726 : MAC$GB_REG SET TO REGISTER
02F6 727 :
02F6 728 :--
02F6 729 :
02F6 730 AINC1:: :AUTO_INC = DOPN RRREG DCLS DPLUS
02F6 731
02F6 732 MOVB #ADMS REGAUTOINC,W^MAC$GB_MODE ;SET MODE
0000'CF 08 90 02FB 733 CVTLB W^MAC$AL_VALSTACK-8[R7],- ;GET REGISTER NUMBER
0000'CF FFF8'CF47 F6 0303 734 Q^MAC$GB_REG
FDB1 31 0303 735 BRW ILL_REG_CHK ;CHK REG ERROR AND RETURN

```

```

0306 737      .SBTTL IMMEDIATE MODE
0306 738
0306 739 :++
0306 740 : FUNCTIONAL DESCRIPTION:
0306 741 :
0306 742 : THIS ROUTINE IS INVOKED WHEN AN IMMEDIATE REFERENCE IS SCANNED.
0306 743 : (ALSO FOR AUTO-INCRMENT OF PC [#EXPR]).
0306 744 :
0306 745 :--
0306 746
0306 747 AINC3::      ;BASIC REF = DIUP DPOUND EXPR
0000'CF 01 9A 0306 748 MOVZBL #1,W^MAC$GL_ABSFLAG      ;DO NOT ALLOW OPTIMIZATION
030B 749
030B 750      .SBTTL AUTO-INCREMENT PC
030B 751
030B 752 AINC2::      ;AUTO_INC = DPOUND DEXPR
0000'CF 05 030B 753 TSTL W^MAC$GL_ABSFLAG      ;ABSOLUTE EXPRESSION?
0000'CF 04 13 030F 754 BEQL 10$      ;IF EQL YES
00 6B 07 E5 0311 755 BBCC #FLG$V_EXPOPT,(R11),10$      ;NO--DO NOT ALLOW OPTIMIZATION
66 0000'CF 9E 0315 756 10$: MOVAB W^MAC$GL_EXPOPVL1,R6      ;POINT TO RESULT
66 0000'CF 47 D0 031A 757 MOVL W^MAC$AL_VALSTACK[R7],(R6) ;SET RESULT
0000'CF 05 0320 758 TSTL W^MAC$GL_ABSFLAG      ;ABSOLUTE EXPRESSION?
0000'CF 2E 12 0324 759 BNEQ 17$      ;IF NEQ NO
10 0000'CF 91 0326 760 CMPB W^MAC$GL_OPSIZE,#16      ; Is operand octaword?
0000'CF 0E 12 032B 761 BNEQ 12$      ; No if NEQ
0004'CF 05 032D 762 TSTL W^MAC$GQ_HIGH_64+4      ; Are bits 96-127 all zero?
0000'CF 21 12 0331 763 BNEQ 17$      ; No if NEQ
0000'CF 05 0333 764 TSTL W^MAC$GQ_HIGH_64+0      ; Are bits 64-95 all zero?
0000'CF 1B 12 0337 765 BNEQ 17$      ; No if NEQ
0000'CF 07 11 0339 766 BRB 14$
08 0000'CF 91 033B 767 12$: CMPB W^MAC$GL_OPSIZE,#8      ; Is operand quadword?
0000'CF 06 12 0340 769 BNEQ 16$      ; No if NEQ?
0000'CF 0C 12 0342 770 14$: TSTL W^MAC$GL_HIGH_32      ; Are bits 32-63 all zero?
0000'CF 0C 12 0346 772 BNEQ 17$      ; No if NEQ
0000'CF 91 0348 773 16$: CMPB W^MAC$GB_RDXNDX,-      ;H FLOAT?
0000'CF 07 034C 775 #RDX$V_HFLOAT
0000'CF 08 13 034D 776 BEQL 18$      ;YES, THEN LEAVE UPPER WORD FOR LATER
02 A6 B5 034F 777 TSTW 2(R6)      ;NO, UPPER 16 BITS 0?
0000'CF 03 13 0352 778 BEQL 18$      ;IF EQL THEN YES
0081 31 0354 779 17$: BRW 40$
0000'CF 05 0357 780      ; Yes: Get Operand bits, isolate Mode.
50 0000'CF D0 0357 781 18$: movl W^MAC$GL_MOPPTR,r0
0000'CF 68 13 035C 782 beql 26$      ; If null pointer assume integer.
51 60 05 05 EE 035E 783 extv #OPD$V_MODE,#OPD$S_MODE,(r0),r1
51 51 00 B1 0363 784 cmpw #OPD$M_ADDR,r1      ; Can't allow short lit. if Addr. ref..
0080 8F B1 0366 785 beql 40$
51 0080 8F B1 0368 786 cmpw #OPD$M_VIELD,r1      ; ditto for Vield references.
0080 69 13 036D 787 beql 40$
0080 60 B5 036F 788 tstw (r0)      ; Floating Operand?
0080 53 18 0371 789 bgeq 26$      ; No if positive.
0373 790 :
0373 791 : ABSOLUTE FLOATING OPERAND--MAKE SHORT FLOATING IF WE CAN
0373 792 :
50 66 C000 8F A1 0373 793 ADDW3 #^XC000,(R6),R0      ;CLEAR HIGH ORDER BIT IN EXPONENT

```

				0000'CF	91	0379	794		CMPB	W^MAC\$GB RDXNDX,-	;H FLOAT?
				07		037D	795			#RDX\$V_HFLOAT	
				23	12	037E	796		BNEQ	20\$;NO, THEN SKIP TO OTHER FLT. PT. CODE
50	10	10	02	A6	F0	0380	797		INSV	2(R6), #16, #16, R0	;YES, MOVE FRACTION TO LOW WORD OF R0
	50			8F	D3	0386	798		BITL	#^X1FFFFFF8,R0	;ARE NECESSARY BITS ZERO?
				49	12	038D	799		BNEQ	40\$;IF NEQ NO
51	50	03		1D	EF	038F	800		EXTZV	#29, #3, R0, R1	;GET FRACTION
52	50	03		00	EF	0394	801		EXTZV	#0, #3, R0, R2	;GET EXPONENT
51	03	03		52	F0	0399	802		INSV	R2, #3, #3, R1	;COMBINE THEM
		66		51	9A	039E	803		MOVZBL	R1, (R6)	;STORE RESULT
				2A	11	03A1	804		BRB	30\$;DONE
						03A3	805				
				0000'CF	91	03A3	806	20\$.	CMPB	W^MAC\$GB RDXNDX,-	;G FLOAT?
				06		03A7	807			#RDX\$V_GFLOAT	
				0E	12	03A8	808		BNEQ	23\$;NO, SKIP TO F AND D FLOAT COMMON CODE
	50			8F	B3	03AA	809		BITW	#^XFF81,R0	;ARE NECESSARY BITS ZERO?
				27	12	03AF	810		BNEQ	40\$;IF NEQ NO
66	50	06		01	EF	03B1	811		EXTZV	#1, #6, R0, (R6)	;YES, STORE RESULT
				15	11	03B6	812		BRB	30\$	
						03B8	813				
	50			8F	B3	03B8	814	23\$:	BITW	#^XFCOF,R0	;ARE NECESSARY BITS ZERO?
				19	12	03BD	815		BNEQ	40\$;IF NEQ NO
66	50	06		04	EF	03BF	816		EXTZV	#4, #6, R0, (R6)	;F OR D FLOAT, STORE RESULT
				07	11	03C4	817		BRB	30\$	
						03C6	818				
						03C6	819				
						03C6	820				
66	FFC0	8F		8F	B3	03C6	821	26\$:	BITW	#^C<^X3F>,(R6)	;INTEGER LESS THAN 64?
		0B		0B	12	03CB	822		BNEQ	40\$;IF NEQ NO
	0000'CF	00		00	90	03CD	823	30\$:	MOVB	#ADMS_LITERAL,W^MAC\$GB_MODE	;SET LITERAL MODE
						03D2	824		\$INC_PC		;COUNT THE BYTE
		16			11	03D6	825		BRB	50\$	
						03D8	826				
						03D8	827				
						03D8	828				
	0000'CF	01		01	90	03D8	829	40\$:	MOVB	#ADMS_IMMEDIATE,W^MAC\$GB_MODE	;SET THE MODE
	0000'CF	0F		0F	90	03DD	830		MOVB	#REG\$PC,W^MAC\$GB_REG	;REGISTER IS PC
	50	0000'CF		50	9A	03E2	831		MOVZBL	W^MAC\$GL_OPSIZE,R0	;GET THE OPERAND SIZE
					D6	03E7	832		INCL	R0	;PLUS ONE BYTE
						03E9	833		\$INC_PC	R0	;UPDATE LOCATION COUNTER
					05	03EE	834	50\$:	RSB		


```

      03EF 836      .SBTTL DEFERRED INDIRECT REGISTER REFERENCE
      03EF 837
      03EF 838      :++
      03EF 839      : FUNCTIONAL DESCRIPTION:
      03EF 840      :
      03EF 841      : ROUTINE IS INVOKED WHEN A DEFERRED INDIRECT REGISTER REFERENCE
      03EF 842      : IS SCANNED. THE MODE AND REGISTER ARE SET, AND THE PC IS
      03EF 843      : INCREMENTED BY TWO BYTES.
      03EF 844      :
      03EF 845      : INPUTS:
      03EF 846      :
      03EF 847      : MAC$AL_VALSTACK-4[R7] REGISTER
      03EF 848      :
      03EF 849      : OUTPUTS:
      03EF 850      :
      03EF 851      : MAC$GB_MODE 'ADMS_DFBYTEDISP'
      03EF 852      : MAC$GB_REG SET TO REGISTER
      03EF 853      : MAC$GL_PC INCREMENTED BY 2
      03EF 854      :
      03EF 855      :--
      03EF 856
      03EF 857 AUTOI3::
      03EF 858 CLRL W^MAC$GL_EXPOPVL1 ;BASIC_REF = DAT DOPN RRREG DCLS
      03F3 859 MOVB #ADMS_DFBYTEDISP,W^MAC$GB_MODE ;VALUE IS 0
      03F8 860 CVTLB W^MAC$AL_VALSTACK-4[R7],- ;GET THE REGISTER
      0400 861 W^MAC$GB_REG
      0400 862 $INC_PC #2 ;EAT TWO BYTES
      05 0405 RSB
  
```

```

      0000'CF D4
      0000'CF 0B 90
      0000'CF F6
      FFC'CF47
  
```

```

0406 865 .SBTTL INDEXED REFERENCES
0406 866
0406 867 :++
0406 868 : FUNCTIONAL DESCRIPTION:
0406 869 :
0406 870 : THIS ROUTINE IS INVOKED WHEN AN INDEX REFERENCE IS FOUND. CHECKS
0406 871 : ARE MADE FOR ILLEGAL INDEX MODES, AND MESSAGES ISSUED TO PASS 2
0406 872 : IF ILLEGAL INDEX MODES ARE FOUND. THE MODES AND REGISTERS OF
0406 873 : THE BASIC REFERENCE AND THE INDEX REFERENCE ARE SET UP AND THE
0406 874 : PC IS INCREMENTED TO COUNT THE INDEX REGISTER REFERENCE.
0406 875 :
0406 876 : INPUTS:
0406 877 :
0406 878 : MAC$AL_VALSTACK-4[R7] INDEX REGISTER NUMBER
0406 879 : MAC$GB_MODE MODE OF BASIC REFERENCE
0406 880 : MAC$GB_REG REGISTER OF BASIC REFERENCE
0406 881 :
0406 882 : OUTPUTS:
0406 883 :
0406 884 : MAC$GB_MODE 'ADMS_INDEX'
0406 885 : MAC$GB_REG INDEX REGISTER NUMBER
0406 886 : MAC$GB_IMODE MODE OF BASIC REFERENCE
0406 887 : MAC$GB_REG REGISTER OF BASIC REFERENCE
0406 888 : MAC$GL_PC INCREMENTED BY 1 FOR INDEX REF.
0406 889 :
0406 890 :--
0406 891 :
0406 892 INDEX::
0406 893 MOVB W^MAC$GB_MODE,R0 ;INDEX_REF = BASIC_REF DSQOPN RRREG DSQCLS
0406 894 CMPB R0,#ADMS_LITERAL ;GET BASIC_REF MODE
0406 895 BNEQ 20$ ;SURELY LITERAL MODE IS ILLEGAL
0406 896 10$: $MAC_ERR MAYNOTINDX ;NO ERROR MESSAGE TODAY
0406 897 BSBW MAC$ERRORPT ;Set message code
0406 898 20$: MOVL W^MAC$AL_VALSTACK-4[R7],R6 ;REPORT ERROR TO PASS 2
0406 899 CMPB R6,#REGS_PC ;GET INDEX REGISTER NUMBER
0406 900 BEQL 30$ ;PC IS AN ILLEGAL REGISTER
0406 901 CMPB R6,W^MAC$GB_REG ;IF EQL ILLEGAL INDEX REG
0406 902 BNEQ 40$ ;SAME AS BASIC_REF REGISTER?
0406 903 CMPB R0,#ADMS_REGAUTODEC ;IF NEQ NO
0406 904 BEQL 30$ ;YES--IS MODE -(RX)[RX]
0406 905 CMPB R0,#ADMS_REGAUTOINC ;IF EQL ILLEGAL
0406 906 BEQL 30$ ;IS MODE (RX)+[RX]
0406 907 CMPB R0,#ADMS_DFRAUTOINC ;IF EQL YES
0406 908 BNEQ 40$ ;IS MODE @(RX)+[RX]
0406 909 30$: $MAC_ERR ILLINDEXREG ;IF NEQ NO
0406 910 BSBW MAC$ERRORPT ;Yes--set message code
0406 911 40$: bbc #FLG$V_OPTVFLIDX,(r11),60$ ;ISSUE MESSAGE TO PASS 2
0406 912 addb2 #2,r0 ;Occasionally, the presence of
0406 913 cmpb r0,#ADMS_LONG_DISP ;indexing forces de-optimization.
0406 914 bnequ 50$ ;If we de-optimize to Longword Disp.
0406 915 $inc_pc ;the increase is two bytes, otherwise
0406 916 50$: $inc_pc ;the increase is only one byte.
0406 917 60$: MOVB r0,W^MAC$GB_IMODE ;MOVE REGISTERS AND MODES
0406 918 MOVB #ADMS_INDEX,W^MAC$GB_MODE ;SET INDEX MODE
0406 919 MOVB W^MAC$GB_REG,W^MAC$GB_IREG ;...
0406 920 MOVB R6,W^MAC$GB_REG ;STORE INDEX REGISTER VALUE
0406 921 $INC_PC ;ALLOW ROOM FOR INDEX REG

```

MAC\$ACTREF
V04-000

OPERAND REFERENCES
INDEXED REFERENCES

05 046F 922

RSB

L 6

16-SEP-1984 02:00:48
5-SEP-1984 01:47:09

VAX/VMS Macro V04-00
[MACRO.SRC]ACTREF.MAR;1

Page 24
(18)

MA
VO

```

0470 924      .SBTTL SPESHL_MODE_CHK
0470 925
0470 926      :++
0470 927      : FUNCTIONAL DESCRIPTION:
0470 928      :
0470 929      : THIS ROUTINE CHECKS FOR SPECIAL MODES THAT MUST BE HANDLED
0470 930      : SPECIALLY SINCE THEY ARE USED AS ADDRESSES IN INDEX MODE.
0470 931      : LITERAL MODE REFERENCES ARE CHANGED TO IMMEDIATE MODE
0470 932      : REFERENCES AND THE PC IS UPDATED BY THE SIZE OF THE OPERAND.
0470 933      : IMMEDIATE MODE REFERENCES ARE FORCED TO LONGWORD SIZE.
0470 934      :
0470 935      : INPUTS:
0470 936      :
0470 937      :     MAC$GB_MODE           MODE OF OPERAND
0470 938      :
0470 939      : OUTPUTS:
0470 940      :
0470 941      :     MAC$GB_MODE           NEW MODE OF OPERAND
0470 942      :     MAC$GB_REG           SET TO PC IF ORIGINAL REF WAS LITERAL
0470 943      :     MAC$GL_PC           UPDATED.
0470 944      :
0470 945      :--
0470 946
0470 947 SPESHL_MODE_CHK:
50 0000'CF 90 0470 948      MOVB     W^MAC$GB_MODE,RO           ;GET THE MODE SPECIFIER
      16 12 0475 949      BNEQ     10$                ;BRANCH IF NOT LITERAL (0)
0000'CF 01 90 0477 950      MOVB     #ADMS_IMMEDIATE,W^MAC$GB_MODE ;YES--SET MODE TO IMEDIATE
0000'CF 0F 90 047C 951      MOVB     #REG$-PC,W^MAC$GB_REG ;USE (PC)+
50 0000'CF 9A 0481 952      MOVZBL  W^MAC$GL_OPSIZE,RO       ;GET OPERAND SIZE
      18 11 0486 953      $INC_PC  RO                ;UPDATE PC
      50 97 048B 954      BRB      20$                ;EXIT ROUTINE
      14 12 048D 955 10$:  DECB     RO                ;IS MODE IMMEDIATE (1)?
50 0000'CF 9A 0491 956      BNEQ     20$                ;IF NEQ NO
      0496 957      MOVZBL  W^MAC$GL_OPSIZE,RO       ;YES--GET OPERAND SIZE
      049B 958      $DEC_PC  RO                ;SUBTRACT DEFAULT SIZE
0000'CF 04 90 04A0 959      $INC_PC  #4                ;FORCE TO LONGWORD
      05 04A5 960      MOVB     #4,W^MAC$GL_OPSIZE ;...
      04A6 961 20$:  RSB
      04A6 962
      04A6 963      .END

```

\$COUNT	=	0000003B			DIRREF	00000000	RG	04	FLGSM_SEQFIL	=	02000000
AB	=	00000001			DISP	00000055	RG	04	FLGSM_SKAN	=	00008000
AD	=	0000C008			DISP1	0000008F	RG	04	FLGSM_SPECOP	=	00000004
ADMS_ABSOLUTE	=	00000002			DISPL1	000000DB	RG	04	FLGSM_SPLALL	=	04000000
ADMS_BYTE_DISP	=	0000000A			DISPL2	0000016E	RG	04	FLG_STOIMF	=	00040000
ADMS_DFBYTEDISP	=	0000000B			DISPL3	00000228	RG	04	FLGSM_SYM2COL	=	00000400
ADMS_DFLONGDISP	=	0000000F			DISPL4	00000230	RG	04	FLGSM_TOCF LG	=	00080000
ADMS_DFRAUTOINC	=	00000009			DISPL5	00000238	RG	04	FLGSM_UPAF LG	=	00000010
ADMS_DFWORDDISP	=	0000000D			DISPL6	00000262	RG	04	FLGSM_UPDFIL	=	00000080
ADMS_IMMEDIATE	=	00000001			DISPL7	0000026A	RG	04	FLGSM_UPMARG	=	00000040
ADMS_INDEX	=	00000004			DISPL8	00000272	RG	04	FLGSM_XCRF	=	80000000
ADMS_LITERAL	=	00000000			DISPLX_EXIT	00000257	R	04	FLGSV_ALLCHR	=	00000000
ADMS_LONG_DISP	=	0000000E			DISPPC	00000184	R	04	FLGSV_BOL	=	00000001
ADMS_MAXMOD	=	0000000F			ENBSG_ABSADDR	*****	X	04	FLGSV_CHKLPND	=	00000014
ADMS_PIC	=	00000003			ERR	=	00000000		FLGSV_COMPEXPR	=	00000002
ADMS_REGAUTODEC	=	00000007			F	=	00008004		FLGSV_CONT	=	00000003
ADMS_REGAUTOINC	=	00000008			FF	=	0000000C		FLGSV_CRF	=	0000001E
ADMS_REGISTER	=	00000005			FLGSM_ALLCHR	=	00000001		FLGSV_CRSEEN	=	00000020
ADMS_RRIND	=	00000006			FLGSM_BOL	=	00000002		FLGSV_DATRPT	=	00000004
ADMS_WORD_DISP	=	0000000C			FLGSM_CHKLPND	=	00100000		FLGSV_DBGOUT	=	0000002E
AF	=	00008004			FLGSM_COMPEXPR	=	00000004		FLGSV_DLIMSTR	=	0000002F
AG	=	0000A008			FLGSM_CONT	=	00000008		FLGSV_ENDMCH	=	00000005
AH	=	00009010			FLGSM_CRF	=	40000000		FLGSV_EVALEXPR	=	00000006
AINC1	=	000002F6	RG	04	FLGSM_CRSEEN	=	00000001		FLGSV_EXPOPT	=	00000007
AINC2	=	0000030B	RG	04	FLGSM_DATRPT	=	00000010		FLGSV_EXTERR	=	00000030
AINC3	=	00000306	RG	04	FLGSM_DBGOUT	=	00004000		FLGSV_EXTWRN	=	00000031
AL	=	00000004			FLGSM_DLIMSTR	=	00008000		FLGSV_FIRSTLN	=	00000029
AO	=	00000010			FLGSM_ENDMCH	=	00000020		FLGSV_IFSTAT	=	00000017
AQ	=	00000008			FLGSM_EVALEXPR	=	00000040		FLGSV_IIF	=	00000016
ARG\$K_SIZE	=	000003E8			FLGSM_EXPOPT	=	00000080		FLGSV_INSERT	=	00000008
AUD\$K_SIZE	=	00000010			FLGSM_EXTERR	=	00010000		FLGSV_IRPC	=	0000001D
AUTODE	=	00000080	RG	04	FLGSM_EXTWRN	=	00020000		FLGSV_LEXOP	=	00000021
AUTOI	=	00000055	RG	04	FLGSM_FIRSTLN	=	00000200		FLGSV_LSTXST	=	00000009
AUTOI3	=	000003EF	RG	04	FLGSM_IFSTAT	=	00800000		FLGSV_MAC2COL	=	0000002B
AUTOI1	=	00000075	RG	04	FLGSM_IIF	=	00400000		FLGSV_MACL	=	0000000B
AW	=	00000002			FLGSM_INSERT	=	00000100		FLGSV_MACLTB	=	0000001B
B	=	00000001			FLGSM_IRPC	=	20000000		FLGSV_MACTXT	=	00000010
BLNK	=	00000020			FLGSM_LEXOP	=	00000002		FLGSV_MEBLST	=	0000000C
CHRSM_COMMA_CR	=	00000020			FLGSM_LSTXST	=	00000200		FLGSV_MOREARG	=	0000002D
CHRSM_ILL CHR	=	00000040			FLGSM_MAC2COL	=	00000800		FLGSV_MOREINP	=	00000023
CHRSM_NUM BER	=	00000010			FLGSM_MACL	=	00000800		FLGSV_NEWPND	=	0000000A
CHRSM_SPA_MSK	=	00000001			FLGSM_MACLTB	=	08000000		FLGSV_NOREF	=	00000018
CHRSM_SYM_CH1	=	00000008			FLGSM_MACT	=	00010000		FLGSV_NTTYPEPC	=	00000025
CHRSM_SYM_CHR	=	00000004			FLGSM_MERL	=	00001000		FLGSV_NULCHR	=	00000032
CHRSM_SYM_DLM	=	00000002			FLGSM_MOREARG	=	00002000		FLGSV_OBJXST	=	00000015
CHR\$V_COMMA CR	=	00000005			FLGSM_MOREINP	=	00003008		FLGSV_OPNDCHK	=	00000028
CHR\$V_CVTLWC	=	00000061			FLGSM_NEWPND	=	00004000		FLGSV_OPRND	=	0000000D
CHR\$V_ILL CHR	=	00000006			FLGSM_NOREF	=	00000000		FLGSV_OPTVFLIDX	=	0000002C
CHR\$V_NOCVT	=	0000007F			FLGSM_NTTYPEPC	=	00000020		FLGSV_ORDLST	=	00000011
CHR\$V_NUM BER	=	00000004			FLGSM_NULCHR	=	00040000		FLGSV_P2	=	0000000E
CHR\$V_SPA_MSK	=	00000000			FLGSM_OBJXST	=	00200000		FLGSV_RPTIRP	=	0000001C
CHR\$V_SYM_CH1	=	00000003			FLGSM_OPNDCHK	=	00000100		FLGSV_SEQFIL	=	00000019
CHR\$V_SYM_CHR	=	00000002			FLGSM_OPRND	=	00002000		FLGSV_SKAN	=	0000000F
CHR\$V_SYM_DLM	=	00000001			FLGSM_OPTVFLIDX	=	00001000		FLGSV_SPECOP	=	00000022
CNT	=	00000001			FLGSM_ORDLST	=	00020000		FLGSV_SPLALL	=	0000001A
CR	=	0000000D			FLGSM_P2	=	00004000		FLGSV_STOIMF	=	00000012
D	=	0000C008			FLGSM_RPTIRP	=	10000000		FLGSV_SYM2COL	=	0000002A

MAC\$ACTREF
Symbol table

OPERAND REFERENCES

B 7

16-SEP-1984 02:00:48
5-SEP-1984 01:47:09

VAX/VMS Macro V04-00
[MACRO.SRC]ACTREF.MAR;1

Page 27
(19)

FLGSV_TOCLG = 00000013
 FLGSV_UPAFGL = 00000024
 FLGSV_UPDFIL = 00000027
 FLGSV_UPMARG = 00000026
 FLGSV_XCRF = 0000001F
 G = 0000A008
 GENLDS = 000027A RG 04
 H = 00009010
 HASHSZ = 0000007F
 HYPHEN = 0000002D
 ILL_REG_CHK = 000000B7 R 04
 INDEX = 00000406 RG 04
 INPSK_BUFSIZ = 000003E6
 INT\$K_BUFSIZ = 000013F4
 INT\$K_BUFWRN = 00001390
 INT\$_ADD = 00000001
 INT\$_AND = 000000C2
 INT\$_ASH = 00000003
 INT\$_ASN = 0000000C
 INT\$_AUGPC = 0000000D
 INT\$_BDST = 0000000E
 INT\$_CHKL = 0000000F
 INT\$_DIV = 00000004
 INT\$_END = 00000010
 INT\$_EPT = 00000011
 INT\$_ERR = 00000012
 INT\$_ETX = 00000013
 INT\$_FNEWL = 00000014
 INT\$_ILG = 00000000
 INT\$_INFO = 0000003A
 INT\$_LGLAB = 00000015
 INT\$_MACL = 00000016
 INT\$_MUL = 00000005
 INT\$_NEG = 00000006
 INT\$_NEWL = 00000017
 INT\$_NEWP = 00000018
 INT\$_NOT = 00000007
 INT\$_OP = 00000019
 INT\$_OR = 00000008
 INT\$_PRIL = 0000001A
 INT\$_PRT = 0000001B
 INT\$_PSECT = 0000001C
 INT\$_REDEF = 0000001D
 INT\$_REF = 0000001E
 INT\$_REST = 0000001F
 INT\$_SAME = 00000009
 INT\$_SAVE = 00000020
 INT\$_SBTTL = 00000021
 INT\$_SETFLAG = 00000022
 INT\$_SETLONG = 00000023
 INT\$_SPIC = 00000024
 INT\$_SPID = 00000025
 INT\$_STIB = 00000026
 INT\$_STIL = 00000028
 INT\$_STIW = 00000027
 INT\$_STKEPT = 00000029
 INT\$_STKG = 0000002A

INT\$_STKL = 0000002B
 INT\$-STKPC = 0000002C
 INT\$-STKS = 0000002D
 INT\$-STOB = 00000034
 INT\$-STOL = 0000002E
 INT\$-STOW = 00000035
 INT\$-STRB = 0000002F
 INT\$-STRL = 00000031
 INT\$-STRSB = 00000032
 INT\$-STRSW = 00000033
 INT\$-STRW = 00000030
 INT\$-STSB = 00000036
 INT\$-STSW = 00000037
 INT\$_SUB = 0000000A
 INT\$_SUME = 00000039
 INT\$_WRN = 00000038
 INT\$_XOR = 0000000B
 L = 00000004
 LITERL = 00000298 RG 04
 LST\$K_BUFSIZ = 00000086
 LST\$K_L_P_PAGE = 0000003C
 LST\$K_TITLE_SIZE = 00000028
 MAC\$AL_VALSTACK ***** X 04
 MAC\$ERRORPT ***** X 04
 MAC\$ERRORPX ***** X 04
 MAC\$GB_IMODE ***** X 04
 MAC\$GB_IREG ***** X 04
 MAC\$GB_MODE ***** X 04
 MAC\$GB_RDXNDX ***** X 04
 MAC\$GB_REG ***** X 04
 MAC\$GB_VAL1 ***** X 04
 MAC\$GK_ZERO ***** X 04
 MAC\$GL_ABSFLAG ***** X 04
 MAC\$GL_DFPC_DSP ***** X 04
 MAC\$GL_EXPEND ***** X 04
 MAC\$GL_EXPOPVL1 ***** X 04
 MAC\$GL_HIGH_32 ***** X 04
 MAC\$GL_MOPPTR ***** X 04
 MAC\$GL_OPsize ***** X 04
 MAC\$GL_PC ***** X 04
 MAC\$GL_VAL3 ***** X 04
 MAC\$GL_VALUE ***** X 04
 MAC\$GQ_HIGH_64 ***** X 04
 MAC\$INTOUT_1_LW ***** X 04
 MAC\$INTOUT_2_LW ***** X 04
 MAC\$OPTIMIZE\$PR ***** X 04
 MAC\$_ILLINDEXREG = 007D90E2
 MAC\$_ILLREGHERE = 007D911A
 MAC\$_MAYNOTINDEX = 007D9142
 MAC\$_SUBSYS = 0000007D
 MB = 00000041
 MD = 0000C048
 MF = 00008044
 MG = 0000A048
 MH = 00009050
 ML = 00000044
 MO = 00000050

MQ = 00000048
 MW = 00000042
 O = 00000010
 OBJ\$K_BUFSIZ = 00000200
 OPD\$M_ADDR = 00000000
 OPD\$M_BB = 000000A1
 OPD\$M_BW = 000000C2
 OPD\$M_D_FLOAT = 0000C000
 OPD\$M_FFLOAT = 00008000
 OPD\$M_G_FLOAT = 0000A000
 OPD\$M_H_FLOAT = 00009000
 OPD\$M_MODE = 000003E0
 OPD\$M_MODIFY = 00000040
 OPD\$M_NOT_32F = 00007000
 OPD\$M_READ = 00000020
 OPD\$M_VFIELD = 00000080
 OPD\$M_WRITE = 00000060
 OPD\$S_MODE = 00000005
 OPD\$S_SIZE = 00000005
 OPD\$V_D_FLOAT = 0000000E
 OPD\$V_FFLOAT = 0000000F
 OPD\$V_G_FLOAT = 0000000D
 OPD\$V_H_FLOAT = 0000000C
 OPD\$V_MODE = 00000005
 OPD\$V_SIZE = 00000000
 OPF\$M_LASTOPR = 00002000
 OPF\$M_OPTEXP = 00001000
 OPF\$V_LASTOPR = 0000000D
 OPF\$V_OPTEXP = 0000000C
 PC_DISPL_DISP = 00000000 R 03
 PSC\$B_NAME = 00000004
 PSC\$B_SEG = 0000000C
 PSC\$B_UNUSED = 0000G00B
 PSC\$K_BLKSIZE = 00000013
 PSC\$K_NO_OPTNS = 0000000A
 PSC\$L_CURLOC = 0000000F
 PSC\$L_LINK = 00000000
 PSC\$L_MAXLGTH = 00000005
 PSC\$M_ABS = FFFFFFFF7
 PSC\$M_ALIGNFLG = 00004000
 PSC\$M_ALLOPTNS = 000003FF
 PSC\$M_BYTE = 00004000
 PSC\$M_CON = FFFFFFFFB
 PSC\$M_DEFAULT = 000001C8
 PSC\$M_EXE = 000000C0
 PSC\$M_GBL = 00000010
 PSC\$M_LCL = FFFFFFFEF
 PSC\$M_LIB = 00000002
 PSC\$M_LONG = 00004800
 PSC\$M_NOEXE = FFFFFFFBF
 PSC\$M_NOPIC = FFFFFFFFE
 PSC\$M_NORD = FFFFFFFF7
 PSC\$M_NOSHR = FFFFFFFDF
 PSC\$M_NOVEC = FFFFFFFDF
 PSC\$M_NOWRT = FFFFFFFEF
 PSC\$M_OVR = 00000004
 PSC\$M_PAGE = 00006400

MAC\$ACTREF
Symbol table

OPERAND REFERENCES

C 7

16-SEP-1984 02:00:48
5-SEP-1984 01:47:09

VAX/VMS Macro V04-00
[MACRO.SRC]ACTREF.MAR;1

Page 28
(19)

MAC
V04

PSC\$M_PIC = 00000001
PSC\$M_QUAD = 00004C00
PSC\$M_RD = 00000080
PSC\$M_REL = 00000008
PSC\$M_SHR = 00000020
PSC\$M_USR = FFFFFFFD
PSC\$M_VEC = 00000200
PSC\$M_WORD = 00004400
PSC\$M_WRT = 00000180
PSC\$S_ALIGNMENT = 00000004
PSC\$V_ALIGNFLG = 0000000E
PSC\$V_ALIGNMENT = 0000000A
PSC\$V_EXE = 00000006
PSC\$V_GBL = 00000004
PSC\$V_LIB = 00000001
PSC\$V_OVR = 00000002
PSC\$V_PIC = 00000000
PSC\$V_RD = 00000007
PSC\$V_REL = 00000003
PSC\$V_SHR = 00000005
PSC\$V_VEC = 00000009
PSC\$V_WRT = 00000008
PSC\$W_FLAG = 00000009
PSC\$W_OPTIONS = 0000000D
Q = 00000008
RB = 00000021
RD = 0000C028
RDEFER = 00000066 RG 04
RDX\$V_BINARY = 00000000
RDX\$V_DECIMAL = 00000002
RDX\$V_DOUBLE = 00000005
RDX\$V_FLOAT = 00000004
RDX\$V_GFLOAT = 00000006
RDX\$V_HEX = 00000003
RDX\$V_HFLOAT = 00000007
RDX\$V_OCTAL = 00000001
REF1 = 0000004E RG 04
REF2 = 00000056 RG 04
REF3 = 0000004E RG 04
REF_EXIT = 0000004E R 04
REG\$PC = 0000000F
RF = 00008024
RFHD = 00000036 R 04
RFHD1 = 00000027 RG 04
RFHD16 = 00000033 RG 04
RFHD2 = 0000002A RG 04
RFHD4 = 0000002D RG 04
RFHD8 = 00000030 RG 04
RG = 0000A028
RH = 00009030
RL = 00000024
RO = 00000030
RQ = 00000028
RW = 00000022
SEMI = 0000003B
SPESHL_MODE_CHK = 00000470 R 04
STB\$K_PG_MISS = 0000000A

SYMSB_NAME = 000000C4
SYMSB_SEG = 0000000C
SYMSB_TOKEN = 0000000B
SYMSK_BLKSIZE = 0000000D
SYMSK_MAXLEN = 0000001F
SYMSK_TWOCOL = 00000010
SYMSL_LINK = 00000000
SYMSL_VAL = 00000005
SYMSM_ABS = 00000010
SYMSM_ASN = 00000100
SYMSM_CRFO = 00002000
SYMSM_DEBUG = 00000020
SYMSM_DEF = 00000001
SYMSM_DELMAC = 00C00200
SYMSM_EPT = 00000200
SYMSM_EXTRN = 00000008
SYMSM_GLOBL = 00000004
SYMSM_LOCAL = 00000040
SYMSM_ODBG = 00000400
SYMSM_REF = 00000080
SYMSM_RELPSECT = 00000800
SYMSM_SUPR = 00004000
SYMSM_WEAK = 00000002
SYMSM_XCRF = 00001000
SYMSV_ABS = 00000004
SYMSV_ASN = 00000008
SYMSV_CRFO = 0000000D
SYMSV_DEBUG = 00000005
SYMSV_DEF = 00000000
SYMSV_DELMAC = 00000009
SYMSV_EPT = 00000009
SYMSV_EXTRN = 00000003
SYMSV_GLOBL = 00000002
SYMSV_LOCAL = 00000006
SYMSV_ODBG = 0000000A
SYMSV_REF = 00000007
SYMSV_RELPSECT = 0000000B
SYMSV_SUPR = 0000000E
SYMSV_WEAK = 00000001
SYMSV_XCRF = 0000000C
SYMSW_FLAG = 00000009
TAB = 00000009
VB = 00000081
VD = 0000C088
VF = 00008084
VG = 0000A088
VH = 00009090
VL = 00000084
VO = 00000090
VQ = 00000088
VW = 00000082
W = 00000002
WB = 00000061
WD = 0000C068
WF = 00008064
WG = 0000A068
WH = 00009070

WL = 00000064
WO = 00000070
WQ = 00000068
WW = 00000062
X = 00000010
X1 = 00000400
X2 = 0000000F

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS :	0C000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK :	00000000 (0.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$ABSS	00000013 (19.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MACSRO_DATA	00000010 (16.)	03 (3.)	NOPIC USR CON REL GBL NOSHR NOEXE RD NOWRT NOVEC LONG
MACSRO_CODE_P1	000004A6 (1190.)	04 (4.)	NOPIC USR CON REL GBL NOSHR EXE RD NCWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:01.43
Command processing	103	00:00:00.38	00:00:01.98
Pass 1	238	00:00:04.18	00:00:18.71
Symbol table sort	0	00:00:00.54	00:00:01.13
Pass 2	183	00:00:01.51	00:00:03.59
Symbol table output	39	00:00:00.21	00:00:00.67
Psect synopsis output	1	00:00:00.02	00:00:00.24
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	595	00:00:06.90	00:00:27.75

The working set limit was 1650 pages.
41361 bytes (81 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 540 non-local and 53 local symbols.
963 source lines were read in Pass 1, producing 24 object records in Pass 2.
19 pages of virtual memory were used to define 15 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[MACRO.OBJ]MACRO.MLB;1	13
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	3
TOTALS (all libraries)	16

616 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:ACTREF/OBJ=OBJ\$:ACTREF MSRCS:ACTREF/UPDATE=(ENHS:ACTREF)+LIBS:MACRO/LIB

