


```

PPPPPPPP      AAAAAA      SSSSSSSS  UU      UU  NN      NN  DDDDDDDD  EEEEEEEEE  FFFFFFFF  IIIIII
PPPPPPPP      AAAAAA      SSSSSSSS  UU      UU  NN      NN  DDDDDDDD  EEEEEEEEE  FFFFFFFF  IIIIII
PP      PP  AA      AA  SS      UU      UU  NN      NN  DD      DD  EE      FF      II
PP      PP  AA      AA  SS      UU      UU  NN      NN  DD      DD  EE      FF      II
PP      PP  AA      AA  SS      UU      UU  NNNN     NN  DD      DD  EE      FF      II
PP      PP  AA      AA  SS      UU      UU  NNNN     NN  DD      DD  EE      FF      II
PPPPPPPP      AA      AA  SSSSSS  UU      UU  NN      NN  DD      DD  EEEEEEE  FFFFFFF  II
PPPPPPPP      AA      AA  SSSSSS  UU      UU  NN      NN  DD      DD  EEEEEEE  FFFFFFF  II
PP      AAAAAAAAAA      SS      UU      UU  NN      NN  DD      DD  EE      FF      II
PP      AAAAAAAAAA      SS      UU      UU  NN      NN  DD      DD  EE      FF      II
PP      AA      AA  SS      UU      UU  NN      NN  DD      DD  EE      FF      II
PP      AA      AA  SS      UU      UU  NN      NN  DD      DD  EE      FF      II
PP      AA      AA  SSSSSSSS  UUUUUUUUU  NN      NN  DDDDDDD  EEEEEEEEE  FF      IIIIII
PP      AA      AA  SSSSSSSS  UUUUUUUUU  NN      NN  DDDDDDD  EEEEEEEEE  FF      IIIIII

```

```

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
LLLLLLLLLLLL  IIIIII      SSSSSSSS
LLLLLLLLLLLL  IIIIII      SSSSSSSS

```



(2)	46
(3)	75
(4)	117
(5)	159
(6)	201

DECLARATIONS
PASSUNDEFINED_F - Return true if F_floating argument is reserved
PASSUNDEFINED_D - Return true if D_floating argument is reserved
PASSUNDEFINED_G - Return true if G_floating argument is reserved
PASSUNDEFINED_H - Return true if H_floating argument is reserved

.....

```
0000 1 .TITLE PASSUNDEFINED - Return true if argument is reserved operand
0000 2 .IDENT /1-001/ ; File: PASUNDEFI.MAR Edit: SBL1001
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: Pascal Language Support
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : This module contains four routines, one for each floating data
0000 35 : type, which return true if their argument is a reserved operand.
0000 36 :
0000 37 : ENVIRONMENT: Runs at any access mode, AST Reentrant
0000 38 :
0000 39 : AUTHOR: Steven B. Lionel, CREATION DATE: 6-Nov-1980
0000 40 :
0000 41 : MODIFIED BY:
0000 42 :
0000 43 : 1-001 - Original. SBL 6-Nov-1980
0000 44 :--
```

```
0000 46 .SBTTL DECLARATIONS
0000 47 :
0000 48 : LIBRARY MACRO CALLS:
0000 49 :
0000 50 : NONE
0000 51 :
0000 52 : EXTERNAL DECLARATIONS:
0000 53 :
0000 54 : .DSABL GBL ; Force all external symbols to be declared
0000 55 : NONE
0000 56 :
0000 57 : MACROS:
0000 58 :
0000 59 : NONE
0000 60 :
0000 61 : EQUATED SYMBOLS:
0000 62 :
0000 63 : NONE
0000 64 :
0000 65 : OWN STORAGE:
0000 66 :
0000 67 : NONE
0000 68 :
0000 69 : PSECT DECLARATIONS:
0000 70 :
00000000 71 : .PSECT _PASSCODE PIC,USR,CON,REL,LCL,SHR,-
0000 72 : EXE,RD,NOWRT, LONG
0000 73 :
```

```

0000 75 .SBTTL PASSUNDEFINED_F - Return true if F_floating argument is reserved
0000 76 :++
0000 77 : FUNCTIONAL DESCRIPTION:
0000 78 :
0000 79 : This procedure returns BOOLEAN TRUE (1) if its argument is a reserved
0000 80 : floating operand, otherwise BOOLEAN FALSE (0).
0000 81 :
0000 82 : CALLING SEQUENCE:
0000 83 :
0000 84 : Result.wv.v = PASSUNDEFINED_F (Single.rf.r)
0000 85 :
0000 86 : FORMAL PARAMETERS:
0000 87 :
0000 88 : Single - F_floating argument
0000 89 :
0000 90 : IMPLICIT INPUTS:
0000 91 :
0000 92 : NONE
0000 93 :
0000 94 : IMPLICIT OUTPUTS:
0000 95 :
0000 96 : NONE
0000 97 :
0000 98 : ROUTINE VALUE:
0000 99 :
0000 100 : 1 if the argument is a reserved operand
0000 101 : 0 otherwise
0000 102 :
0000 103 : SIDE EFFECTS:
0000 104 :
0000 105 :
0000 106 :--
0000 107 :
0000 108 : .ENTRY PASSUNDEFINED_F, ^M<> ; Entry point
0002 109 :
0002 110 : CLRL R0 ; Initially FALSE
0004 111 : TSTL @4(AP) ; Test for longword access
0007 112 : CMPZV #7, #9, @4(AP), #^X100 ; Is it reserved?
0011 113 : BNEQ 10$ ; No
0013 114 : INCL R0 ; Yes
0015 115 10$: RET ; End of routine PASSUNDEFINED_F
  
```

```

0000100 8F 04 BC 09 04 50 D4 0002 110
0000100 8F 04 BC 09 07 ED 0007 112
0000100 8F 04 BC 09 02 12 0011 113
0000100 8F 04 BC 09 50 D6 0013 114
0000100 8F 04 BC 09 04 0015 115
  
```



```

002D 159 .SBTTL PASSUNDEFINED_G - Return true if G_floating argument is reserved
002D 160 :++
002D 161 : FUNCTIONAL DESCRIPTION:
002D 162 :
002D 163 : This procedure returns BOOLEAN TRUE (1) if its argument is a reserved
002D 164 : floating operand, otherwise BOOLEAN FALSE (0).
002D 165 :
002D 166 : CALLING SEQUENCE:
002D 167 :
002D 168 : Result.wv.v = PASSUNDEFINED_G (Double.rg.r)
002D 169 :
002D 170 : FORMAL PARAMETERS:
002D 171 :
002D 172 : Double - G_floating argument
002D 173 :
002D 174 : IMPLICIT INPUTS:
002D 175 :
002D 176 : NONE
002D 177 :
002D 178 : IMPLICIT OUTPUTS:
002D 179 :
002D 180 : NONE
002D 181 :
002D 182 : ROUTINE VALUE:
002D 183 :
002D 184 : 1 if the argument is a reserved operand
002D 185 : 0 otherwise
002D 186 :
002D 187 : SIDE EFFECTS:
002D 188 :
002D 189 :
002D 190 :--
002D 191 :
0000 002D 192 .ENTRY PASSUNDEFINED_G, ^M<> ; Entry point
002F 193
002F 194 MOVQ @4(AP), R0 ; Test for quadword access
0033 195 CLRL R0 ; Initially FALSE
0035 196 CMPZV #4, #12, @4(AP), #^X800 ; Is it reserved?
003F 197 BNEQ 10$, ; No
0041 198 INCL R0 ; Yes
0043 199 10$: RET ; End of routine PASSUNDEFINED_G
  
```

```

50 04 BC 7D
0000800 8F 04 BC 0C 04 ED
02 12
50 D6
04 0043
  
```



```

0044 201      .SBTTL  PASSUNDEFINED_H - Return true if H_floating argument is reserved
0044 202      :++
0044 203      : FUNCTIONAL DESCRIPTION:
0044 204      :
0044 205      : This procedure returns BOOLEAN TRUE (1) if its argument is a reserved
0044 206      : floating operand, otherwise BOOLEAN FALSE (0).
0044 207      :
0044 208      : CALLING SEQUENCE:
0044 209      :
0044 210      : Result.wv.v = PASSUNDEFINED_H (Quad.rh.r)
0044 211      :
0044 212      : FORMAL PARAMETERS:
0044 213      :
0044 214      : Quad      - H_floating argument
0044 215      :
0044 216      : IMPLICIT INPUTS:
0044 217      :
0044 218      : NONE
0044 219      :
0044 220      : IMPLICIT OUTPUTS:
0044 221      :
0044 222      : NONE
0044 223      :
0044 224      : ROUTINE VALUE:
0044 225      :
0044 226      : 1 if the argument is a reserved operand
0044 227      : 0 otherwise
0044 228      :
0044 229      : SIDE EFFECTS:
0044 230      :
0044 231      :
0044 232      :--
0000 0044 233      .ENTRY  PASSUNDEFINED_H, ^M<>      ; Entry point
51   04 AC   D0 0046 234      MOVL   4(AP), R1      ; Get address of argument
      OC A1   D5 004A 235      TSTL   12(R1)     ; Test last longword for access
      50 D4 004D 236      CLRL   R0      ; Initially FALSE
8000 8F 61 B1 004F 237      CMPW   (R1), #^X8000 ; Is it reserved?
      02 12 0054 238      BNEQ   10$      ; No
      50 D6 0056 239      INCL   R0      ; Yes
      04 0058 240      RET     10$:      ; End of routine PASSUNDEFINED_H
0059 241      .END      ; End of module PASSUNDEFINED
0059 242
0059 243
0059 244

```

PASSUNDEFINED
Symbol table

PASSUNDEFINED_D 00000016 RG 01
PASSUNDEFINED_F 00000000 RG 01
PASSUNDEFINED_G 0000002D RG 01
PASSUNDEFINED_H 00000044 RG 01

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

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_PASSCODE	00000059 (89.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	10	00:00:00.07	00:00:01.02
Command processing	72	00:00:00.62	00:00:04.69
Pass 1	67	00:00:00.51	00:00:01.53
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	54	00:00:00.42	00:00:02.34
Symbol table output	2	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	209	00:00:01.65	00:00:09.61

The working set limit was 900 pages.
2721 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 4 non-local and 4 local symbols.
244 source lines were read in Pass 1, producing 19 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:PASUNDEFI/OBJ=OBJ\$:PASUNDEFI MSRC\$:PASUNDEFI/UPDATE=(ENH\$:PASUNDEFI)

