

```

PPPPPPPPPPPP      AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPPPP    AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPPPPPPPPPPPPP    AAAAAAAAAA      SSSSSSSSSSSS      RRRRRRRRRRRR      TTTTTTTTTTTTTT      LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPP                PPP  AAA          AAA  SSS                RRR                RRR                TTT                LLL
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT                LLL
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT                LLL
PPPPPPPPPPPPPP    AAA          AAA          SSSSSSSSSS      RRRRRRRRRRRR      TTT                LLL
PPP                AAAAAAAAAAAAAAAAAA          SSS                RRR      RRR                TTT                LLL
PPP                AAAAAAAAAAAAAAAAAA          SSS                RRR      RRR                TTT                LLL
PPP                AAAAAAAAAAAAAAAAAA          SSS                RRR      RRR                TTT                LLL
PPP                AAA          AAA          SSS                RRR                RRR      RRR                TTT                LLL
PPP                AAA          AAA          SSS                RRR                RRR      RRR                TTT                LLL
PPP                AAA          AAA          SSS                RRR                RRR                TTT                LLL
PPP                AAA          AAA          SSSSSSSSSSSS      RRR                RRR                TTT                LLLLLLLLLLLLLLLLLL
PPP                AAA          AAA          SSSSSSSSSSSS      RRR                RRR                TTT                LLLLLLLLLLLLLLLLLL
PPP                AAA          AAA          SSSSSSSSSSSS      RRR                RRR                TTT                LLLLLLLLLLLLLLLLLL

```

2

Sym

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAC

PAS

PAS

PAS
PAS

PAS

PAS

PAS

PAS

PAS
PASPAS
PAS

PAS

PAS
PAS

PAS


```
PPPPPPPP      AAAAAA      SSSSSSSS      SSSSSSSS      QQQQQQ      RRRRRRRR
PPPPPPPP      AAAAAA      SSSSSSSS      SSSSSSSS      QQQQQQ      RRRRRRRR
PP      PP      AA      AA      SS      SS      QQ      QQ      RR      RR
PP      PP      AA      AA      SS      SS      QQ      QQ      RR      RR
PP      PP      AA      AA      SS      SS      QQ      QQ      RR      RR
PP      PP      AA      AA      SS      SS      QQ      QQ      RR      RR
PPPPPPPP      AA      AA      SSSSSS      SSSSSS      QQ      QQ      RRRRRRRR
PPPPPPPP      AA      AA      SSSSSS      SSSSSS      QQ      QQ      RRRRRRRR
PP      AAAAAAAAAA      SS      SS      QQ      QQ      QQ      RR      RR
PP      AAAAAAAAAA      SS      SS      QQ      QQ      QQ      RR      RR
PP      AA      AA      SS      SS      QQ      QQ      QQ      RR      RR
PP      AA      AA      SS      SS      QQ      QQ      QQ      RR      RR
PP      AA      AA      SSSSSSSS      SSSSSSSS      QQQQ      QQ      RR      RR
PP      AA      AA      SSSSSSSS      SSSSSSSS      QQQQ      QQ      RR      RR
```

```
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)	76
(4)	113
(5)	150
(6)	193
(7)	236
(8)	277

DECLARATIONS
PASSQR_L - Return square of a signed longword
PASSQR_LU - Return square of an unsigned longword
PASSQR_F - Return square of an F_floating
PASSQR_D - Return square of a D_floating
PASSQR_G - Return square of a G_floating
PASSQR_H - Return square of an H_floating

PASSQR
1-001

- Return Square of Value

D 3

16-SEP-1984 01:26:47 VAX/VMS Macro V04-00
6-SEP-1984 11:33:40 [PASRTL.SRC]PASSQR.MAR;1

Page 1
(1)

```
0000 1 .TITLE PASSQR - Return Square of Value
0000 2 .IDENT /1-001/ ; File: PASSQR.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 six routines which return the square of
0000 35 : their argument.
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 $SFDEF ; Stack frame definitions
0000 51 $PSLDEF ; Processor status longword definitions
0000 52 :
0000 53 : EXTERNAL DECLARATIONS:
0000 54 :
0000 55 .DSABL GBL ; Force all external symbols to be declared
0000 56 : NONE
0000 57 :
0000 58 : MACROS:
0000 59 :
0000 60 : NONE
0000 61 :
0000 62 : EQUATED SYMBOLS:
0000 63 :
0000 64 : NONE
0000 65 :
0000 66 : OWN STORAGE:
0000 67 :
0000 68 : NONE
0000 69 :
0000 70 : PSECT DECLARATIONS:
0000 71 :
0000 72 .PSECT _PASSCODE PIC, USR, CON, REL, LCL, SHR, -
0000 73 EXE, RD, NOWRT, LONG
0000 74
```

```
0000 76 .SBTTL PAS$$SQR_L - Return square of a signed longword
0000 77 :++
0000 78 : FUNCTIONAL DESCRIPTION:
0000 79 :
0000 80 :     This routine returns the square of a signed longword
0000 81 :
0000 82 : CALLING SEQUENCE:
0000 83 :
0000 84 :     Result.wl.v = PAS$$SQR_L (Long.rl.r)
0000 85 :
0000 86 : FORMAL PARAMETERS:
0000 87 :
0000 88 :     Long      - Signed longword 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 :     The square of the argument
0000 101 :
0000 102 : SIDE EFFECTS:
0000 103 :
0000 104 :     SS$_INTOVF      - Integer overflow
0000 105 :
0000 106 :--
0000 107 :
4000 0000 108 .ENTRY PAS$$SQR_L, ^M<IV>      ; Integer overflow must be enabled
0002 109      MULL3 @4(AP), @4(AP), R0      ; Compute result
50 04 BC 04 BC C5 0002 110      RET      ; End of routine PAS$$SQR_L
04 0008 111
```



```
0009 113 .SBTTL PAS$$SQR_LU - Return square of an unsigned longword
0009 114 :++
0009 115 : FUNCTIONAL DESCRIPTION:
0009 116 :
0009 117 : This routine returns the square of an unsigned longword
0009 118 :
0009 119 : CALLING SEQUENCE:
0009 120 :
0009 121 : Result.wlu.v = PAS$$SQR_LU (Long.rlu.r)
0009 122 :
0009 123 : FORMAL PARAMETERS:
0009 124 :
0009 125 : Long - Unsigned longword argument
0009 126 :
0009 127 : IMPLICIT INPUTS:
0009 128 :
0009 129 : NONE
0009 130 :
0009 131 : IMPLICIT OUTPUTS:
0009 132 :
0009 133 : NONE
0009 134 :
0009 135 : ROUTINE VALUE:
0009 136 :
0009 137 : The square of the argument
0009 138 :
0009 139 : SIDE EFFECTS:
0009 140 :
0009 141 : NONE
0009 142 :
0009 143 :--
0000 0009 144
50 04 BC 04 BC C5 0009 145 .ENTRY PAS$$SQR_LU, ^M<> ; Integer overflow must be disabled
000B 146
000B 147 MULL3 @4(AP), @4(AP), R0 ; Compute result
0011 148 RET ; End of routine PAS$$SQR_LU
```

```
0012 150 .SBTTL PAS$$SQR_F - Return square of an F_floating
0012 151 :++
0012 152 : FUNCTIONAL DESCRIPTION:
0012 153 :
0012 154 : This routine returns the square of an F_floating value.
0012 155 :
0012 156 : CALLING SEQUENCE:
0012 157 :
0012 158 : Result.wf.v = PAS$$SQR_F (Single.rf.r)
0012 159 :
0012 160 : FORMAL PARAMETERS:
0012 161 :
0012 162 : Single - F_floating argument
0012 163 :
0012 164 : IMPLICIT INPUTS:
0012 165 :
0012 166 : NONE
0012 167 :
0012 168 : IMPLICIT OUTPUTS:
0012 169 :
0012 170 : NONE
0012 171 :
0012 172 : ROUTINE VALUE:
0012 173 :
0012 174 : The square of the argument
0012 175 :
0012 176 : SIDE EFFECTS:
0012 177 :
0012 178 : SS$_ROPRAND - if the argument is a reserved operand
0012 179 : SS$_FLTOVF - floating overflow trap
0012 180 : SS$_FLTOVF_F - floating overflow fault
0012 181 : SS$_FLTUND - floating underflow trap if caller has enabled FU
0012 182 : SS$_FLTUND_F - floating underflow fault if caller has enabled FU
0012 183 :
0012 184 :--
0012 185 :
0000 0012 186 .ENTRY PAS$$SQR_F, ^M<> ; Entry point
0014 187
0014 188 BBC #SF$V_FU, SF$W_SAVE_PSW(FP), 10$ ; Skip if FU disabled
0019 189 BISPSW #PSL$M_FU ; Enable FU because caller did
50 04 BC 04 BC 45 001D 190 10$: MULF3 @4(AP), @4(AP), R0 ; Compute square
04 04 AD 06 E1 0014 188 BBC #SF$V_FU, SF$W_SAVE_PSW(FP), 10$ ; Skip if FU disabled
0040 8F B8 0019 189 BISPSW #PSL$M_FU ; Enable FU because caller did
04 BC 04 BC 45 001D 190 10$: MULF3 @4(AP), @4(AP), R0 ; Compute square
04 0023 191 RET ; End of routine PAS$$SQR_F
```



```
0024 193 .SBTTL PAS$$QR_D - Return square of a D_floating
0024 194 :++
0024 195 : FUNCTIONAL DESCRIPTION:
0024 196 :
0024 197 : This routine returns the square of a D_floating value.
0024 198 :
0024 199 : CALLING SEQUENCE:
0024 200 :
0024 201 : Result.wd.v = PAS$$QR_D (Double.rd.r)
0024 202 :
0024 203 : FORMAL PARAMETERS:
0024 204 :
0024 205 : Double - D_floating argument
0024 206 :
0024 207 : IMPLICIT INPUTS:
0024 208 :
0024 209 : NONE
0024 210 :
0024 211 : IMPLICIT OUTPUTS:
0024 212 :
0024 213 : NONE
0024 214 :
0024 215 : ROUTINE VALUE:
0024 216 :
0024 217 : The square of the argument
0024 218 :
0024 219 : SIDE EFFECTS:
0024 220 :
0024 221 : SS$_ROPRAND - if the argument is a reserved operand
0024 222 : SS$_FLTOVF - floating overflow trap
0024 223 : SS$_FLTOVF_F - floating overflow fault
0024 224 : SS$_FLTUND - floating underflow trap if caller has enabled FU
0024 225 : SS$_FLTUND_F - floating underflow fault if caller has enabled FU
0024 226 :
0024 227 :--
0024 228
0000 0024 229 .ENTRY PAS$$QR_D, ^M<> ; Entry point
0026 230
0026 231 BBC #SF$V FU, SF$W_SAVE_PSW(FP), 10$ ; Skip if FU disabled
0026 232 BISPSW #PSL$M_FU ; Enable FU because caller did
50 04 BC 04 BC 65 002F 233 10$: MULD3 @4(AP), @4(AP), R0 ; Compute square
04 0035 234 RET ; End of routine PAS$$QR_D
```


PASSQR
1-001

- Return Square of Value

PASSQR_G - Return square of a G_floating

J 3

16-SEP-1984 01:26:47

6-SEP-1984 11:33:40

VAX/VMS Macro V04-00
[PASRTL.SRC]PASSQR.MAR;1

Page 7
(7)

```
0036 236 .SBTTL PASSQR_G - Return square of a G_floating
0036 237 :++
0036 238 : FUNCTIONAL DESCRIPTION:
0036 239 :
0036 240 : This routine returns the square of a G_floating value.
0036 241 :
0036 242 : CALLING SEQUENCE:
0036 243 :
0036 244 : Result.wg.v = PASSQR_G (Double.rg.r)
0036 245 :
0036 246 : FORMAL PARAMETERS:
0036 247 :
0036 248 : Double - G_floating argument
0036 249 :
0036 250 : IMPLICIT INPUTS:
0036 251 :
0036 252 : NONE
0036 253 :
0036 254 : IMPLICIT OUTPUTS:
0036 255 :
0036 256 : NONE
0036 257 :
0036 258 : ROUTINE VALUE:
0036 259 :
0036 260 : The square of the argument
0036 261 :
0036 262 : SIDE EFFECTS:
0036 263 :
0036 264 : SSS_ROPRAND - if the argument is a reserved operand
0036 265 : SSS_FLTOVF_F - floating overflow fault
0036 266 : SSS_FLTUND_F - floating underflow fault if caller has enabled FU
0036 267 :
0036 268 :--
0036 269 :
0000 0036 270 .ENTRY PASSQR_G, ^M<> ; Entry point
0038 271
0038 272 BBC #SF$V FU, SF$W_SAVE_PSW(FP), 10$ ; Skip if FU disabled
003D 273 BISPSW #PSL$M_FU ; Enable FU because caller did
50 04 BC 04 BC 45FD 0041 274 10$: MULG3 @4(AP), @4(AP), R0 ; Compute square
0048 275 RET ; End of routine PASSQR_G
```



```
0049 277 .SBTTL PAS$SQR_H - Return square of an H_floating
0049 278 :++
0049 279 : FUNCTIONAL DESCRIPTION:
0049 280 :
0049 281 : This routine returns the square of an H_floating value.
0049 282 :
0049 283 : CALLING SEQUENCE:
0049 284 :
0049 285 : CALL PAS$SQR_H (Result.wh.r, Quad.rh.r)
0049 286 :
0049 287 : FORMAL PARAMETERS:
0049 288 :
0049 289 : Result - H_floating result
0049 290 : Quad - H_floating argument
0049 291 :
0049 292 : IMPLICIT INPUTS:
0049 293 :
0049 294 : NONE
0049 295 :
0049 296 : IMPLICIT OUTPUTS:
0049 297 :
0049 298 : NONE
0049 299 :
0049 300 : ROUTINE VALUE:
0049 301 :
0049 302 : The square of the argument
0049 303 :
0049 304 : SIDE EFFECTS:
0049 305 :
0049 306 : SS$_R0PRAND - if the argument is a reserved operand
0049 307 : SS$_FLT0VF_F - floating overflow fault
0049 308 : SS$_FLTUND_F - floating underflow fault if caller has enabled FU
0049 309 :
0049 310 :--
0049 311 :
0000 0049 312 .ENTRY PAS$SQR_H, ^M<> ; Entry point
0048 313
0048 314 BBC #SF$V FU, SF$W_SAVE_PSW(FP), 10$ ; Skip if FU disabled
0050 315 BISPSW #PSL$M_FU ; Enable FU because caller did
04 BC 04 BC 08 BC 08 BC 65FD 0054 316 10$: MULH3 @8(AP), @8(AP), @4(AP) ; Compute square
005C 317 RET ; End of routine PAS$SQR_H
005D 318
005D 319 .END ; End of module PAS$SQR
```


PASSQR
Symbol table

- Return Square of Value

L 3

16-SEP-1984 01:26:47
6-SEP-1984 11:33:40

VAX/VMS Macro V04-00
[PASRTL.SRC]PASSQR.MAR;1

Page 9
(8)

PASSQR_D 00000024 RG 02
PASSQR_F 00000012 RG 02
PASSQR_G 00000036 RG 02
PASSQR_H 00000049 RG 02
PASSQR_L 00000000 RG 02
PASSQR_LU 00000009 RG 02
PSL\$M_F0 = 00000040
SF\$V_FU = 00000006
SF\$W_SAVE_PSW = 00000004

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
_PASSCODE	0000005D (93.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	10	00:00:00.08	00:00:01.20
Command processing	72	00:00:00.69	00:00:03.66
Pass 1	110	00:00:01.45	00:00:07.48
Symbol table sort	0	00:00:00.06	00:00:00.09
Pass 2	63	00:00:00.63	00:00:01.80
Symbol table output	2	00:00:00.02	00:00:00.02
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	262	00:00:02.95	00:00:14.27

The working set limit was 900 pages.
7203 bytes (15 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 74 non-local and 4 local symbols.
319 source lines were read in Pass 1, producing 27 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5

131 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

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

0297

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY