

|              |              |       |
|--------------|--------------|-------|
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAAAA |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAAAA |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAAAA |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSSSSSSSSS   | DDD          | AAA   |
| SSSSSSSSSS   | DDD          | AAA   |
| SSSSSSSSSS   | DDD          | AAA   |
| SSS          | DDD          | AAAAA |
| SSS          | DDD          | AAAAA |
| SSS          | DDD          | AAAAA |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAA   |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAA   |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAA   |

```

EEEEEEEEEE XX      XX      AAAAAA MM      MM      P P P P P P P P S S S S S S S S LL
EEEEEEEEEE XX      XX      AAAAAA MM      MM      P P P P P P P P S S S S S S S S LL
EE          XX      XX      AA      AA      M M M M M M M M P P      P P S S      LL
EE          XX      XX      AA      AA      M M M M M M M M P P      P P S S      LL
EE          XX      XX      AA      AA      M M      M M      P P      P P S S      LL
EEEEEEEEEE      XX      AA      AA      M M      M M      P P P P P P P P S S S S S S LL
EEEEEEEEEE      XX      AA      AA      M M      M M      P P P P P P P P S S S S S S LL
EE          XX      XX      AAAAAAAAAA MM      M M      P P      S S      LL
EE          XX      XX      AAAAAAAAAA MM      M M      P P      S S      LL
EE          XX      XX      AA      AA      M M      M M      P P      S S      LL
EEEEEEEEEE XX      XX      AA      AA      M M      M M      P P      S S S S S S S S LL
EEEEEEEEEE XX      XX      AA      AA      M M      M M      P P      S S S S S S S S LL

```

```

LL          I I I I I I S S S S S S S S
LL          I I I I I I S S S S S S S S
LL          I I      S S
LL          I I      S S
LL          I I      S S
LL          I I      S S
LL          I I      S S S S S S
LL          I I      S S S S S S
LL          I I      S S
LL          I I      S S
LL          I I      S S
LL          I I      S S
LLLLLLLLLLLL I I I I I I S S S S S S S S
LLLLLLLLLLLL I I I I I I S S S S S S S S

```



```

1 0001 0 MODULE FORMAT_PSL ( IDENT = 'V04-000' ) =
2 0002 1 BEGIN
3 0003 1
4 0004 1 *****
5 0005 1 *
6 0006 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
7 0007 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
8 0008 1 * ALL RIGHTS RESERVED.
9 0009 1 *
10 0010 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
11 0011 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
12 0012 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
13 0013 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
14 0014 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
15 0015 1 * TRANSFERRED.
16 0016 1 *
17 0017 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
18 0018 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
19 0019 1 * CORPORATION.
20 0020 1 *
21 0021 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
22 0022 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
23 0023 1 *
24 0024 1 *
25 0025 1 *****
26 0026 1
27 0027 1 FACILITY: SDA
28 0028 1
29 0029 1 ++
30 0030 1 FUNCTIONAL DESCRIPTION:
31 0031 1
32 0032 1 Given a longword value, format as a PSL.
33 0033 1
34 0034 1 Author:
35 0035 1 Jake VanNoy (taken from DEBUG sources)
36 0036 1
37 0037 1 Modified by:
38 0038 1
39 0039 1 V03-001 JLV0222 Jake VanNoy 21-JAN-1983
40 0040 1 Add module to SDA sources.
41 0041 1 --
42 0042 1
43 0043 1 LIBRARY
44 0044 1 'SYSS$LIBRARY:STARLET';
45 0045 1
46 0046 1 REQUIRE
47 0047 1 'SHRLIB$:UTILDEF';
48 0232 1
49 0233 1 EXTERNAL ROUTINE
50 0234 1 SYSS$FAOL : ADDRESSING_MODE (GENERAL),
51 0235 1 LIB$PUT_OUTPUT : ADDRESSING_MODE (GENERAL);
52 0236 1
53 0237 1 FORWARD ROUTINE
54 0238 1 fao_put : NOVALUE,
55 0239 1 format_psl : NOVALUE; ! ROUTINE TO OUTPUT PSL IN SPECIAL FORMAT
56 0240 1

```

```

58 0241 1 GLOBAL ROUTINE format_psl (value, buffer) : NOVALUE =
59 0242 1 ++
60 0243 1 FUNCTIONAL DESCRIPTION:
61 0244 1     Formats and outputs two lines of specially
62 0245 1     formatted data contained in the PSL. The fields shown are
63 0246 1
64 0247 1     CMP      - compatibility mode
65 0248 1     TP      - trace trap pending
66 0249 1     FPD     - first part done
67 0250 1     IS     - interrupt stack
68 0251 1     CURMOD  - current access mode
69 0252 1     PRVMOD  - previous access mode
70 0253 1     IPL     - interrupt priority level
71 0254 1     DV     - decimal overflow trap enable
72 0255 1     FU     - floating underflow trap enable
73 0256 1     IV     - integer overflow trap enable
74 0257 1     T      - trace trap
75 0258 1     N      - condition code
76 0259 1     Z      - condition code
77 0260 1     V      - condition code
78 0261 1     C      - condition code
79 0262 1
80 0263 1 CALLING SEQUENCE:
81 0264 1     FORMAT_PSL ( .psl, buffer)
82 0265 1
83 0266 1 INPUTS:
84 0267 1     value    -The current contents of the PSL
85 0268 1
86 0269 1 IMPLICIT INPUTS:
87 0270 1     NONE
88 0271 1
89 0272 1 OUTPUTS:
90 0273 1     buffer is returned with length and address of psl line.
91 0274 1
92 0275 1 IMPLICIT OUTPUTS:
93 0276 1     NONE
94 0277 1
95 0278 1 ROUTINE value:
96 0279 1     NOVALUE
97 0280 1
98 0281 1 SIDE EFFECTS:
99 0282 1     --
100 0283 1
101 0284 2 BEGIN
102 0285 2     MAP
103 0286 2         buffer : REF vector[2],
104 0287 2         value  : block;
105 0288 2     MACRO
106 0289 2         position_field = 0, 8, 1%,
107 0290 2         size_field    = 8, 8, 1%,
108 0291 2         mode_field    = 16, 4, 1%,
109 0292 2         blanks_field  = 20, 4, 1%,
110 0293 2         width_field   = 24, 8, 1%,
111 0294 2
112 M 0295 2         psl_field (name, position, size, mode, width, leading_blanks) =
113 0296 2             position, size, mode OR (leading_blanks ^ 4), width%;
114 0297 2

```



```

115 0298
116 0299
117 0300
118 0301
119 0302
120 0303
121 0304
122 0305
123 0306
124 0307
125 0308
126 0309
127 0310
128 0311
129 0312
130 0313
131 0314
132 0315
133 0316
134 0317
135 0318
136 0319
137 0320
138 0321
139 0322
140 0323
141 0324
142 0325
143 0326
144 0327
145 0328
146 0329
147 0330
148 0331
149 0332
150 0333
151 0334
152 0335
153 0336
154 0337
155 0338
156 0339
157 0340
158 0341
159 0342
160 0343
161 0344
162 0345
163 0346
164 0347
165 0348
166 0349
167 0350
168 0351
169 0352
170 0353
171 0354

```

```

LITERAL
    decimal      = 0;
    max_psl_field = 15;

BIND
    psl_table = UPLIT BYTE (
        psl_field (CMP, 31, 1, 0, 1, 1),
        psl_field (TP, 30, 1, 0, 1, 3),
        psl_field (FPD, 27, 1, 0, 1, 2),
        psl_field (IS, 26, 1, 0, 1, 3),
        psl_field (CURMOD, 24, 2, 1, 4, 2),
        psl_field (PRVMOD, 22, 2, 1, 4, 3),
        psl_field (IPL, 16, 5, 0, 2, 3),
        psl_field (DV, 7, 1, 0, 1, 2),
        psl_field (FU, 6, 1, 0, 1, 2),
        psl_field (IV, 5, 1, 0, 1, 2),
        psl_field (T, 4, 1, 0, 1, 1),
        psl_field (N, 3, 1, 0, 1, 1),
        psl_field (Z, 2, 1, 0, 1, 1),
        psl_field (V, 1, 1, 0, 1, 1),
        psl_field (C, 0, 1, 0, 1, 1)

        : BLOCK,

        hex_number      = UPLIT BYTE (%ASCIC '!AD!#XB'),
        stg_desc         = UPLIT BYTE (%ASCIC '!AD!AD'),
        blanks           = UPLIT BYTE (%ASCII ' '),

        priv_modes       = UPLIT BYTE (
            %ASCII 'KERN',
            %ASCII 'EXEC',
            %ASCII 'SUPR',
            %ASCII 'USER')

        : VECTOR;

LOCAL
    SAVE_BUFFER : VECTOR[2];    ! save buffer descriptor

    save_buffer[0] = .buffer[0];
    save_buffer[1] = .buffer[1];

    ! Write out the standard title which describes the PSL fields.

    fao_put (.buffer,
        UPLIT (%ASCIC '!_CMP TP FPD IS CURMOD PRVMOD IPL DV FU IV T N Z V C!/_!_'));
    INCR count FROM 0 TO max_psl_field - 1 DO
        IF .psl_table [.count, mode_field] EQL decimal
            THEN
                BEGIN
                    fao_put (.buffer, hex_number,
                        .psl_table [.count, blanks_field], blanks,
                        .psl_table [.count, width_field],
                        .value [0, .psl_table [.count, position_field],
                        .psl_table [.count, size_field], 0]);
                END
            END

```

```

: 172      0355      3
: 173      0356      3
: 174      0357      3
: 175      0358      3
: 176      0359      3
: 177      0360      3
: 178      0361      3
: 179      0362      3
: 180      0363      3
: 181      0364      3
: 182      0365      3
: 183      0366      3
: 184      0367      3
: 185      0368      1 END;

ELSE
    BEGIN
        fao_put (.buffer, stg_desc,
                .psl_table [.count, blanks_field], blanks,
                .psl_table [.count, width_field],
                priv_modes [.value [0, .psl_table [.count, position_field],
                .psl_table [.count, size_field], 0]]);
    END;

buffer[0] = .save_buffer[0] - .buffer[0];      ! length is starting length minus remaining length
buffer[1] = .save_buffer[1]                    ! address is reset

! EXAM_PSL

```

```

: 30 01 1A 01 20 01 1B 01 30 01 1E 01 10 01 1F 00000 P.AAA: .BYTE 31, 1, 16, 1, 30, 1, 48, 1, 27, 1, 32, 1, -
: 01 07 02 30 05 10 04 31 02 16 04 21 02 18 01 0000F 26, 1, 48, 1, 24, 2, 33, 4, 22, 2, 49, 4, -
: 03 01 10 01 04 01 20 01 05 01 20 01 06 01 20 0001E 16, 5, 48, 2, 7, 1, 32, 1, 6, 1, 32, 1, -
: 01 10 01 00 01 10 01 01 01 10 01 02 01 10 01 0002D 5, 1, 32, 1, 4, 1, 16, 1, 3, 1, 16, 1, 2, -
: 42 58 23 21 44 41 21 07 0003C P.AAB: .ASCII <7>\!AD!\XB\
: 44 41 21 06 00044 P.AAC: .ASCII <6>\!AD!AD\
: 20 20 20 20 0004B P.AAD: .ASCII \
: 4E 52 45 42 0004F P.AAE: .ASCII \KERN\
: 43 45 58 45 00053 .ASCII \EXEC\
: 52 50 55 53 00057 .ASCII \SUPR\
: 52 45 53 55 0005B .ASCII \USER\
: 0005F .BLKB 1
: 49 20 44 50 46 20 50 54 20 50 4D 43 5F 21 38 00060 P.AAF: .ASCII \8!_CMP TP FPD IS CURMOD PRVMOD IPL DV FU\
: 44 4F 4D 56 52 50 20 44 4F 4D 52 55 43 20 53 0006F
: 55 46 20 56 44 20 4C 50 49 20 0007E
: 2F 21 43 20 56 20 5A 20 4E 20 54 20 56 49 20 00088 .ASCII \ IV T N Z V C!/_\<0><0><0>
: 00 00 00 5F 21 00097

```

```

PSL_TABLE= P.AAA
HEX_NUMBER= P.AAB
STG_DESC= P.AAC
BLANKS= P.AAD
PRIV_MODES= P.AAE
.EXTRN SYS$FAOL, LIB$PUT_OUTPUT

```

```

.PSECT $CODE$,NOWRT,2

.ENTRY FORMAT_PSL, Save R2,R3,R4,R5      : 0241
MOVAB BLANKS, R5
SUBL2 #4, SP
MOVL BUFFER, R3
PUSHL (R3)
MOVL 4(R3), SAVE_BUFFER+4
PUSHAB P.AAF
PUSHL R3

```

```

55 0000' CF 9E 00002
5E 04 C2 00007
53 08 AC D0 0000A
04 AE 04 63 DD 0000E
15 A3 D0 00010
A5 9F 00015
53 DD 00018

```



FORMAT\_PSL  
V04-000

M 2  
16-Sep-1984 01:54:11  
14-Sep-1984 13:08:56

VAX-11 Bliss-32 V4.0-742  
DISK\$VMMASTER:[SDA.SRC]EXAMPSL.B32;1

Page 5  
(2)

\*\*\*F

|    |    |    |       |    |    |    |    |       |        |                         |                      |      |      |
|----|----|----|-------|----|----|----|----|-------|--------|-------------------------|----------------------|------|------|
|    |    |    | 0000V | CF |    | 02 | FB | 0001A | CALLS  | #2, FAO_PUT             | :                    |      |      |
|    |    |    |       | 54 |    | 52 | D4 | 0001F | CLRL   | COUNT                   | :                    | 0346 |      |
|    |    |    |       | 51 | B5 | A5 | DE | 00021 | 1\$:   | PSL_TABLE[COUNT], R4    | :                    | 0347 |      |
|    |    |    |       | A4 |    | 64 | 98 | 00026 | MOVAL  | (R4), R1                | :                    | 0354 |      |
| 50 | 04 | AC | 01    | OF |    | 51 | EF | 00029 | CVTBL  | R1, 1(R4), VALUE, R0    | :                    |      |      |
|    |    |    |       |    | 02 | A4 | 93 | 00030 | EXTZV  | 2(R4), #15              | :                    | 0347 |      |
|    |    |    |       |    |    | 12 | 12 | 00034 | BITB   | 2\$                     | :                    |      |      |
|    |    |    |       |    |    | 50 | DD | 00036 | BNEQ   | R0                      | :                    | 0353 |      |
|    |    |    | 7E    |    | 03 | A4 | 98 | 00038 | PUSHL  | 3(R4), -(SP)            | :                    | 0352 |      |
|    |    |    |       |    |    | 55 | DD | 0003C | CVTBL  | R5                      | :                    | 0350 |      |
| 7E |    | 64 | 04    |    |    | 14 | EE | 0003E | PUSHL  | #20, #4, (R4), -(SP)    | :                    | 0351 |      |
|    |    |    |       |    | F1 | A5 | 9F | 00043 | EXTV   | HEX_NUMBER              | :                    | 0350 |      |
|    |    |    |       |    |    | 12 | 11 | 00046 | PUSHAB | 3\$                     | :                    |      |      |
|    |    |    |       |    | 04 | A5 | 40 | DF    | 00048  | 2\$:                    |                      | 0362 |      |
|    |    |    | 7E    |    | 03 | A4 | 98 | 0004C | PUSHAL | PRIV_MODES[R0]          | :                    |      |      |
|    |    |    |       |    |    | 55 | DD | 00050 | CVTBL  | 3(R4), -(SP)            | :                    |      |      |
|    |    |    |       |    |    | 14 | EE | 00052 | PUSHL  | R5                      | :                    | 0358 |      |
| 7E |    | 64 | 04    |    |    | 53 | DD | 0005A | EXTV   | #20, #4, (R4), -(SP)    | :                    | 0362 |      |
|    |    |    |       |    | F9 | A5 | 9F | 00057 | PUSHAB | STG_DESC                | :                    | 0358 |      |
|    |    |    |       |    |    | 53 | DD | 0005A | 3\$:   |                         | :                    | 0362 |      |
|    |    |    | 0000V | CF |    | 06 | FB | 0005C | PUSHL  | R3                      | :                    |      |      |
|    |    | BC |       | 52 |    | 0E | F3 | 00061 | CALLS  | #6, FAO_PUT             | :                    |      |      |
|    |    | 63 |       | 6E |    | 63 | C3 | 00065 | AOBLEQ | #14, COUNT, 1\$         | :                    | 0347 |      |
|    |    |    | 04    | A3 |    | 04 | AE | 00069 | SUBL3  | (R3), SAVE_BUFFER, (R3) | :                    | 0365 |      |
|    |    |    |       |    |    |    |    | 04    | 0006E  | MOV                     | SAVE_BUFFER+4, 4(R3) | :    | 0366 |
|    |    |    |       |    |    |    |    |       | RET    |                         | :                    | 0368 |      |

; Routine Size: 111 bytes, Routine Base: \$CODE\$ + 0000

: 186 0369 1  
: 187 0370 1

```

: 189 0371 1 ROUTINE fao_put( buffer, STRING, ARGUMENTS ) : NOVALUE =
: 190 0372 1 ++
: 191 0373 1 FUNCTIONAL DESCRIPTION
: 192 0374 1
: 193 0375 1 INPUTS:
: 194 0376 1     STRING      - A COUNTED STRING WHICH CONTAINS THE DIRECTIVES FOR $FAO.
: 195 0377 1     ARGUMENTS - THE ARGS FOR $FAO.
: 196 0378 1     buffer    - The address of the beginning of the current output buffer.
: 197 0379 1
: 198 0380 1 OUTPUTS:
: 199 0381 1     THE $FAO OUTPUT IS PUT INTO THE OUTPUT BUFFER, and
: 200 0382 1     the buffer pointers are updated.
: 201 0383 1
: 202 0384 1 IMPLICIT OUTPUTS:
: 203 0385 1
: 204 0386 1 ROUTINE VALUE:
: 205 0387 1     NONE.
: 206 0388 1
: 207 0389 1 --
: 208 0390 1
: 209 0391 2 BEGIN
: 210 0392 2 MAP
: 211 0393 2     buffer : REF VECTOR[2],      ! OUTPUT DESC FOR $FAO.
: 212 0394 2     string : REF VECTOR[BYTE]; ! The string to be formatted
: 213 0395 2                                     ! for output
: 214 0396 2 LOCAL
: 215 0397 2     INP_DESC : VECTOR[2],      ! INPUT DESC FOR $FAO.
: 216 0398 2     STR_SIZE : WORD;          ! $FAO RETURNS OUTPUT SIZE HERE
: 217 0399 2
: 218 0400 2
: 219 0401 2     inp_desc[0] = .string[0];
: 220 0402 2     inp_desc[1] = string[1];
: 221 0403 2
: 222 0404 2     Signal_if_error (sys$faol( inp_desc, str_size, .buffer, arguments));
: 223 0405 2
: 224 0406 2     buffer[0] = .buffer[0] - .str_size;
: 225 0407 2     buffer[1] = .buffer[1] + .str_size;
: 226 0408 2
: 227 0409 1 END;

```

|    |           |       |    |             |                |                        |        |
|----|-----------|-------|----|-------------|----------------|------------------------|--------|
|    |           |       |    | 000C 0000   | FAO_PUT: .WORD | Save R2,R3             | : 0371 |
|    |           | 5E    |    | 0C C2 00002 | SUBL2          | #12, SP                |        |
|    |           | 04 AE | 08 | BC 9A 00005 | MOVZBL         | @STRING, INP_DESC      | : 0401 |
| 08 | AE        | 08 AC |    | 01 C1 0000A | ADDL3          | #1, STRING, INP_DESC+4 | : 0402 |
|    |           |       | 0C | AC 9F 00010 | PUSHAB         | ARGUMENTS              | : 0404 |
|    |           | 52    | 04 | AC D0 00013 | MOVL           | BUFFER, R2             |        |
|    |           |       |    | 52 DD 00017 | PUSHL          | R2                     |        |
|    |           |       | 08 | AE 9F 00019 | PUSHAB         | STR_SIZE               |        |
|    |           |       | 10 | AE 9F 0001C | PUSHAB         | INP_DESC               |        |
|    | 00000000G | 00    |    | 04 FB 0001F | CALLS          | #4, SYSSFAOL           |        |
|    |           | 53    |    | 50 D0 00026 | MOVL           | R0, STATUS             |        |
|    |           | 0A    |    | 53 E8 00029 | BLBS           | STATUS, 1\$            |        |
|    |           |       |    | 53 DD 0002C | PUSHL          | STATUS                 |        |



```

00000000G 00      01 FB 0002E      CALLS #1, LIB$SIGNAL
                  04 00035      RET
                  50      6E 3C 00036 1$: MOVZWL STR_SIZE, R0
                  62      50 C2 00039      SUBL2 R0, -(R2)
                  50      6E 3C 0003C      MOVZWL STR_SIZE, R0
                  04 A2      50 C0 0003F      ADDL2 R0, -4(R2)
                  04 00043      RET

```

; Routine Size: 68 bytes, Routine Base: \$CODE\$ + 006F

```

: 228      0410 1
: 229      0411 1
: 230      0412 1 END
: 231      0413 0 ELUDOM

```

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

| Name     | Bytes | Attributes                                                |
|----------|-------|-----------------------------------------------------------|
| \$PLITS  | 156   | NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2) |
| \$CODE\$ | 179   | NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)  |

Library Statistics

| File                               | Symbols |        | Percent | Pages Mapped | Processing Time |
|------------------------------------|---------|--------|---------|--------------|-----------------|
|                                    | Total   | Loaded |         |              |                 |
| _S255\$DUA28:[SYSLIB]STARLET.L32;1 | 9776    | 5      | 0       | 581          | 00:00.8         |

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:EXAMPSL/OBJ=OBJ\$:EXAMPSL MSRC\$:EXAMPSL/UPDATE=(ENH\$:EXAMPSL)

```

: Size:      179 code + 156 data bytes
: Run Time:   00:04.9
: Elapsed Time: 00:17.0
: Lines/CPU Min: 5098
: Lexemes/CPU-Min: 33135
: Memory Used: 76 pages
: Compilation Complete

```



0352

**DIGITAL  
CONFIDE**