

|                  |           |           |                  |     |
|------------------|-----------|-----------|------------------|-----|
| FFFFFFFFFFFFFFFF | 111       | 111       | AAAAAAAAA        |     |
| FFFFFFFFFFFFFFFF | 111       | 111       | AAAAAAAAA        |     |
| FFFFFFFFFFFFFFFF | 111       | 111       | AAAAAAAAA        |     |
| FFF              | 111111    | 111111    | AAA              | AAA |
| FFF              | 111111    | 111111    | AAA              | AAA |
| FFF              | 111111    | 111111    | AAA              | AAA |
| FFF              | 111       | 111       | AAA              | AAA |
| FFF              | 111       | 111       | AAA              | AAA |
| FFF              | 111       | 111       | AAA              | AAA |
| FFFFFFFFFFFFFF   | 111       | 111       | AAA              | AAA |
| FFFFFFFFFFFFFF   | 111       | 111       | AAA              | AAA |
| FFFFFFFFFFFFFF   | 111       | 111       | AAA              | AAA |
| FFF              | 111       | 111       | AAAAAAAAAAAAAAAA |     |
| FFF              | 111       | 111       | AAAAAAAAAAAAAAAA |     |
| FFF              | 111       | 111       | AAAAAAAAAAAAAAAA |     |
| FFF              | 111       | 111       | AAA              | AAA |
| FFF              | 111       | 111       | AAA              | AAA |
| FFF              | 111       | 111       | AAA              | AAA |
| FFF              | 111       | 111       | AAA              | AAA |
| FFF              | 111111111 | 111111111 | AAA              | AAA |
| FFF              | 111111111 | 111111111 | AAA              | AAA |
| FFF              | 111111111 | 111111111 | AAA              | AAA |

```
DDDDDDDD  I I I I I  SSSSSSSS  PPPPPPPP  AAAAAA  TTTTTTTTTT
DDDDDDDD  I I I I I  SSSSSSSS  PPPPPPPP  AAAAAA  TTTTTTTTTT
DD      DD      SS      PP      PP      AA      AA      TT
DD      DD      SS      PP      PP      AA      AA      TT
DD      DD      SS      PP      PP      AA      AA      TT
DD      DD      SS      PP      PP      AA      AA      TT
DD      DD      SS      PP      PP      AA      AA      TT
DD      DD      SS      PP      PP      AA      AA      TT
DD      DD      SS      PP      PP      AA      AA      TT
DD      DD      SS      PP      PP      AA      AA      TT
DD      DD      SS      PP      PP      AA      AA      TT
DDDDDDDD  I I I I I  SSSSSSSS  PPPPPPPP  AAAAAA  TTTTTTTTTT
DDDDDDDD  I I I I I  SSSSSSSS  PPPPPPPP  AAAAAA  TTTTTTTTTT
```

```
LL      I I I I I  SSSSSSSS
LL      I I I I I  SSSSSSSS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LL      I I      SS
LLLLLLLLLL  I I I I I  SSSSSSSS
LLLLLLLLLL  I I I I I  SSSSSSSS
```



```
0001 0 MODULE DISPAT (  
0002 0     LANGUAGE (BLISS32),  
0003 0     MAIN = STARTUP,  
0004 0     IDENT = 'V04-000',  
0005 0 ) =  
0006 1 BEGIN  
0007 1  
0008 1 |  
0009 1 |*****  
0010 1 |*  
0011 1 |*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0012 1 |*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0013 1 |*  ALL RIGHTS RESERVED.  
0014 1 |*  
0015 1 |*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0016 1 |*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0017 1 |*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0018 1 |*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0019 1 |*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0020 1 |*  TRANSFERRED.  
0021 1 |*  
0022 1 |*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0023 1 |*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0024 1 |*  CORPORATION.  
0025 1 |*  
0026 1 |*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0027 1 |*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0028 1 |*  
0029 1 |*  
0030 1 |*****  
0031 1 |  
0032 1 |++  
0033 1 |  
0034 1 |FACILITY: F11ACP Structure Level 1  
0035 1 |  
0036 1 |ABSTRACT:  
0037 1 |  
0038 1 |    This module is the main routine of FCP. It dequeues a request,  
0039 1 |    executes it, and signals completion to the user.  
0040 1 |  
0041 1 |ENVIRONMENT:  
0042 1 |  
0043 1 |    STARLET operating system, including privileged system services  
0044 1 |    and internal exec routines.  
0045 1 |  
0046 1 |--  
0047 1 |  
0048 1 |  
0049 1 |AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 14:33  
0050 1 |  
0051 1 |MODIFIED BY:  
0052 1 |  
0053 1 |    V02-001 LMP0005      L. Mark Pilant,      13-Jan-1982 13:10  
0054 1 |    Adding support for Cathedral windows.  
0055 1 |  
0056 1 |    A0100  ACG0001      Andrew C. Goldstein, 10-Oct-1978 20:01  
0057 1 |    Previous revision history moved to f11A.REV
```

```
58 0058 1 !
59 0059 1 !**
60 0060 1
61 0061 1
62 0062 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
63 0063 1 REQUIRE 'SRC$FCPDEF.B32';
64 0378 1
65 0379 1
66 0380 1 ! Establish the max and min function codes for the function dispatch.
67 0381 1 !
68 0382 1
69 0383 1 LITERAL
70 0384 1     LOW_FUNCTION = MINU (
71 0385 1         IOS_ACCESS,
72 0386 1         IOS_CREATE,
73 0387 1         IOS_DEACCESS,
74 0388 1         IOS_DELETE,
75 0389 1         IOS_MODIFY,
76 0390 1         IOS_ACPCONTROL,
77 0391 1         IOS_MOUNT
78 0392 1     ),
79 0393 1
80 0394 1     HIGH_FUNCTION = MAXU (
81 0395 1         IOS_ACCESS,
82 0396 1         IOS_CREATE,
83 0397 1         IOS_DEACCESS,
84 0398 1         IOS_DELETE,
85 0399 1         IOS_MODIFY,
86 0400 1         IOS_ACPCONTROL,
87 0401 1         IOS_MOUNT
88 0402 1     );
89 0403 1
90 0404 1 FORWARD ROUTINE
91 0405 1     STARTUP      : NOVALUE,
92 0406 1     DISPATCHER : NOVALUE,
93 0407 1     MAIN_HANDLER : NOVALUE;
```



```

: 95      0408 1 GLOBAL ROUTINE STARTUP : NOVALUE =
: 96      0409 1
: 97      0410 1 ++
: 98      0411 1
: 99      0412 1 FUNCTIONAL DESCRIPTION:
100      0413 1
101      0414 1 This routine is the startup point for FCP. It locks all of FCP
102      0415 1 into memory and then calls the dispatcher loop in exec mode.
103      0416 1
104      0417 1 CALLING SEQUENCE:
105      0418 1 UNDEFINED
106      0419 1
107      0420 1 INPUT PARAMETERS:
108      0421 1 NONE
109      0422 1
110      0423 1 IMPLICIT INPUTS:
111      0424 1 NONE
112      0425 1
113      0426 1 OUTPUT PARAMETERS:
114      0427 1 NONE
115      0428 1
116      0429 1 IMPLICIT OUTPUTS:
117      0430 1 NONE
118      0431 1
119      0432 1 ROUTINE VALUE:
120      0433 1 NONE
121      0434 1
122      0435 1 SIDE EFFECTS:
123      0436 1 FCP locked into memory, dispatcher started.
124      0437 1
125      0438 1 --
126      0439 1
127      0440 2 BEGIN
128      0441 2
129      0442 2
130      0443 2 EXEC_CALL (DISPATCHER);
131      0444 2
132      0445 1 END;
```

! end of routine STARTUP

```

.TITLE  DISPAT
.IDENT  \V04-000\
.EXTRN  SYSS$CMEXEC
.PSECT  $CODE$,NOWRT,2
.ENTRY  STARTUP, Save nothing
CLRL    -(SP)
PUSHL   SP
PUSHAB  DISPATCHER
CALLS   #3, @#SYSS$CMEXEC
RET
```

```

: 0408
: 0443
:
: 0445
```

```

0000 00000
7E D4 00002
5E DD 00004
CF 9F 00006
03 FB 0000A
04 00011
```

00000000G 9F 0000V

; Routine Size: 18 bytes, Routine Base: \$CODE\$ + 0000

```
134 0446 1 GLOBAL ROUTINE DISPATCHER : NOVALUE =
135 0447 1
136 0448 1 ++
137 0449 1
138 0450 1 FUNCTIONAL DESCRIPTION:
139 0451 1
140 0452 1 This routine is the main routine of FCP. It dequeues a request,
141 0453 1 executes it, and signals completion to the user.
142 0454 1
143 0455 1 CALLING SEQUENCE:
144 0456 1 DISPATCHER ()
145 0457 1
146 0458 1 INPUT PARAMETERS:
147 0459 1 NONE
148 0460 1
149 0461 1 IMPLICIT INPUTS:
150 0462 1 NONE
151 0463 1
152 0464 1 OUTPUT PARAMETERS:
153 0465 1 NONE
154 0466 1
155 0467 1 IMPLICIT OUTPUTS:
156 0468 1 NONE
157 0469 1
158 0470 1 ROUTINE VALUE:
159 0471 1 NONE
160 0472 1
161 0473 1 SIDE EFFECTS:
162 0474 1 FCP functions executed
163 0475 1
164 0476 1 --
165 0477 1
166 0478 2 BEGIN
167 0479 2
168 0480 2 LOCAL
169 0481 2 FUNCTION; ! function being executed
170 0482 2
171 0483 2 EXTERNAL
172 0484 2 IO_PACKET : REF BBLOCK; ! current I/O packet
173 0485 2
174 0486 2 EXTERNAL ROUTINE
175 0487 2 INIT_FCP, ! one time initialization
176 0488 2 INIT_COMMON, ! per call initialization
177 0489 2 PMS_START, ! start performance metering
178 0490 2 PMS_END, ! end performance metering
179 0491 2 GET_REQUEST, ! get next I/O request
180 0492 2 READ_WRITEVB, ! process read/write virtual
181 0493 2 ACCESS, ! ACCESS function routine
182 0494 2 CREATE, ! CREATE function routine
183 0495 2 DEACCESS, ! DEACCESS function routine
184 0496 2 DELETE, ! DELETE function routine
185 0497 2 MODIFY, ! MODIFY function routine
186 0498 2 ACPCONTROL, ! ACPCONTROL function routine
187 0499 2 MOUNT, ! MOUNT function routine
188 0500 2 ERR_CLEANUP, ! error cleanup routine
189 0501 2 CLEANUP, ! general cleanup routine
190 0502 2 IO_DONE; ! I/O completion processing
```



```
191 0503 2
192 0504 2
193 0505 2 ! Do the one time initialization. Then for each request, do the per
194 0506 2 request initialization, get the next request, and process it. If
195 0507 2 the request fails, call the error cleanup before returning
196 0508 2 completion.
197 0509 2
198 0510 2
199 0511 2 ! ENABLE MAIN_HANDLER;
200 0512 2 BEGIN
201 0513 2 BUILTIN FP;
202 0514 2 .FP = MAIN_HANDLER;
203 0515 2 END;
204 0516 2 KERNEL_CALL (INIT_FCP);
205 0517 2
206 0518 2 WHILE 1 DO
207 0519 2 BEGIN
208 0520 2 PMS_START ();
209 0521 2 INIT_COMMON ();
210 0522 2 IO_PACKET = KERNEL_CALL (GET_REQUEST);
211 0523 2 FUNCTION = .IO_PACKET[IRPSV_FCODE];
212 0524 2
213 0525 2 IF .FUNCTION EQL IOS_READPBLK
214 0526 2 OR .FUNCTION EQL IOS_WRITEPBLK
215 0527 2 THEN
216 0528 2 BEGIN
217 0529 2 IF NOT READ_WRITEVB ()
218 0530 2 THEN
219 0531 2 BEGIN
220 0532 2 IF ERR_CLEANUP () THEN CLEANUP () ELSE (ERR_CLEANUP (); CLEANUP ());
221 0533 2 PMS_END ();
222 0534 2 KERNEL_CALL (IO_DONE, .IO_PACKET);
223 0535 2 END;
224 0536 2 END
225 0537 2 ELSE
226 0538 2 BEGIN
227 0539 2 IF
228 0540 2 (
229 0541 2 CASE .FUNCTION FROM LOW_FUNCTION TO HIGH_FUNCTION OF
230 0542 2 SET
231 0543 2 [IOS_ACCESS]: ACCESS ();
232 0544 2 [IOS_CREATE]: CREATE ();
233 0545 2 [IOS_DEACCESS]: DEACCESS ();
234 0546 2 [IOS_DELETE]: DELETE ();
235 0547 2 [IOS_MODIFY]: MODIFY ();
236 0548 2 [IOS_ACPCONTROL]: ACPCONTROL ();
237 0549 2 [IOS_MOUNT]: MOUNT ();
238 0550 2 [INRANGE]: (ERR_STATUS (SS$_ILLIOFUNC); 0);
239 0551 2 [OUTRANGE]: (ERR_STATUS (SS$_ILLIOFUNC); 0);
240 0552 2 TES
241 0553 2 )
242 0554 2 )
243 0555 2 THEN ! successful completion
244 0556 2 BEGIN
245 0557 2 IF NOT CLEANUP () THEN (ERR_CLEANUP (); CLEANUP ());
246 0558 2 END
247 0559 2 ELSE ! error completion
```

M 5  
16-Sep-1984 00:45:27 VAX-11 Bliss-32 V4.0-742 Page 6  
14-Sep-1984 12:29:30 DISK\$VMSMASTER:[F11A.SRC]DISPAT.B32:1 (3)

```
! end of routine DISPATCHER
```

|       |    |           |    |      |       |        |                           |   |      |
|-------|----|-----------|----|------|-------|--------|---------------------------|---|------|
|       | 54 | 00000000G | 00 | 001C | 00000 | .ENTRY | DISPATCHER, Save R2,R3,R4 | : | 0446 |
|       | 53 | 00000000G | 9F | 9E   | 00002 | MOVAB  | USER STATUS, R4           | : |      |
|       | 6D | 0000V     | CF | 9E   | 00009 | MOVAB  | @#SYSSCMKRNL, R3          | : |      |
|       |    |           | 7E | D4   | 00010 | MOVAB  | MAIN HANDLER, (FP)        | : | 0514 |
|       |    |           | 5E | DD   | 00015 | CLRL   | -(SP)                     | : | 0516 |
|       |    | 0000G     | CF | 9F   | 00017 | PUSHL  | SP                        | : |      |
|       | 63 |           | 03 | FB   | 00019 | PUSHAB | INIT_FCP                  | : |      |
| 0000G | CF |           | 00 | FB   | 0001D | CALLS  | #3, SYSSCMKRNL            | : |      |
| 0000G | CF |           | 00 | FB   | 00020 | CALLS  | #0, PMS_START             | : | 0520 |
|       |    |           | 00 | FB   | 00025 | CALLS  | #0, INIT_COMMON           | : | 0521 |
|       |    |           | 7E | D4   | 0002A | CLRL   | -(SP)                     | : | 0522 |
|       |    |           | 5E | DD   | 0002C | PUSHL  | SP                        | : |      |
|       |    | 0000G     | CF | 9F   | 0002E | PUSHAB | GET_REQUEST               | : |      |
|       | 63 |           | 03 | FB   | 00032 | CALLS  | #3, -SYSSCMKRNL           | : |      |
| 0000G | CF |           | 50 | D0   | 00035 | MOVL   | R0, IO_PACKET             | : |      |
|       | 06 |           | 00 | E8   | 0003A | EXTZV  | #0, #6, 32(R0), FUNCTION  | : | 0523 |
|       | 0C |           | 52 | D1   | 00040 | CML    | FUNCTION, #12             | : | 0525 |
|       |    |           | 05 | 13   | 00043 | BEQL   | 2\$                       | : |      |
|       | 0B |           | 52 | D1   | 00045 | CML    | FUNCTION, #11             | : | 0526 |
|       |    |           | 0A | 12   | 00048 | BNEQ   | 3\$                       | : |      |
| 0000G | CF |           | 00 | FB   | 0004A | CALLS  | #0, READ_WRITEVB          | : | 0529 |
|       | CE |           | 50 | E8   | 0004F | BLBS   | R0, 1\$                   | : |      |
|       |    |           | 5D | 11   | 00052 | BRB    | 15\$                      | : | 0532 |
|       | 32 |           | 52 | CF   | 00054 | CASEL  | FUNCTION, #50, #7         | : | 0542 |
| 0019  |    | 0012      |    |      | 00058 | .WORD  | 5\$-4\$, -                | : |      |
| 0046  |    | 002E      |    |      | 00060 |        | 6\$-4\$, -                | : |      |
|       |    |           |    |      |       |        | 7\$-4\$, -                | : |      |
|       |    |           |    |      |       |        | 8\$-4\$, -                | : |      |
|       |    |           |    |      |       |        | 9\$-4\$, -                | : |      |
|       |    |           |    |      |       |        | 13\$-4\$, -               | : |      |
|       |    |           |    |      |       |        | 10\$-4\$, -               | : |      |
|       |    |           |    |      |       |        | 11\$-4\$, -               | : |      |
|       |    |           |    |      |       |        | 13\$                      | : |      |
| 0000G | CF |           | 34 | 11   | 00068 | BRB    |                           | : | 0552 |
|       |    |           | 00 | FB   | 0006A | CALLS  | #0, ACCESS                | : | 0544 |

S  
R  
E  
L  
M  
C

.....



|       |    |       |    |       |        |                   |               |      |
|-------|----|-------|----|-------|--------|-------------------|---------------|------|
| 0000G | CF | 28    | 11 | 0006F | BRB    | 12\$              |               |      |
|       |    | 00    | FB | 00071 | CALLS  | #0                | CREATE        | 0545 |
| 0000G | CF | 21    | 11 | 00076 | BRB    | 12\$              |               |      |
|       |    | 00    | FB | 00078 | CALLS  | #0                | DEACCESS      | 0546 |
| 0000G | CF | 1A    | 11 | 0007D | BRB    | 12\$              |               |      |
|       |    | 00    | FB | 0007F | CALLS  | #0                | DELETE        | 0547 |
| 0000G | CF | 13    | 11 | 00084 | BRB    | 12\$              |               |      |
|       |    | 00    | FB | 00086 | CALLS  | #0                | MODIFY        | 0548 |
| 0000G | CF | 0C    | 11 | 0008B | BRB    | 12\$              |               |      |
|       |    | 00    | FB | 0008D | CALLS  | #0                | ACPCONTROL    | 0549 |
| 0000G | CF | 05    | 11 | 00092 | BRB    | 12\$              |               |      |
|       |    | 00    | FB | 00094 | CALLS  | #0                | MOUNT         | 0550 |
| 0000G | CF | 15    | 50 | E9    | BLBC   | R0, 15\$          |               |      |
|       |    | 09    | 11 | 0009C | BRB    | 14\$              |               |      |
|       | 10 | 64    | E9 | 0009E | BLBC   | USER_STATUS, 15\$ |               | 0551 |
|       | 64 | F4    | 8F | 9B    | MOVZBW | #244, USER_STATUS |               |      |
|       |    |       | 0A | 11    | 000A5  | BRB               | 15\$          |      |
| 0000G | CF | 00    | FB | 000A7 | CALLS  | #0                | CLEANUP       | 0557 |
|       | 14 | 50    | E8 | 000AC | BLBS   | R0, 18\$          |               |      |
|       |    | 08    | 11 | 000AF | BRB    | 16\$              |               |      |
| 0000G | CF | 00    | FB | 000B1 | CALLS  | #0                | ERR_CLEANUP   | 0561 |
|       | 05 | 50    | E8 | 000B6 | BLBS   | R0, 17\$          |               |      |
| 0000G | CF | 00    | FB | 000B9 | CALLS  | #0                | ERR_CLEANUP   |      |
| 0000G | CF | 00    | FB | 000BE | CALLS  | #0                | CLEANUP       |      |
| 0000G | CF | 00    | FB | 000C3 | CALLS  | #0                | PMS_END       | 0564 |
|       |    | 0000G | CF | DD    | 000C8  | PUSHL             | IO_PACKET     | 0565 |
|       |    |       | 01 | DD    | 000CC  | PUSHL             | #1            |      |
|       |    |       | 5E | DD    | 000CE  | PUSHL             | SP            |      |
|       | 63 | 0000G | CF | 9F    | 000D0  | PUSHAB            | IO_DONE       |      |
|       |    |       | 04 | FB    | 000D4  | CALLS             | #4, SYSSCMKRN |      |
|       |    | FF46  | 31 | 000D7 | BRW    | 1\$               |               | 0525 |

; Routine Size: 218 bytes, Routine Base: \$CODE\$ + 0012

```
258 0570 1 GLOBAL ROUTINE MAIN_HANDLER (SIGNAL, MECHANISM) : NOVALUE =
259 0571 1
260 0572 1 ++
261 0573 1
262 0574 1 FUNCTIONAL DESCRIPTION:
263 0575 1
264 0576 1 This routine is the main level condition handler. It stores the
265 0577 1 condition value (FCP error code) in the user status block, unwinds
266 0578 1 and returns from the function that was executing.
267 0579 1
268 0580 1 CALLING SEQUENCE:
269 0581 1 MAIN_HANDLER (ARG1, ARG2)
270 0582 1
271 0583 1 INPUT PARAMETERS:
272 0584 1 ARG1: address of signal array
273 0585 1 ARG2: address of mechanism array
274 0586 1
275 0587 1 IMPLICIT INPUTS:
276 0588 1 NONE
277 0589 1
278 0590 1 OUTPUT PARAMETERS:
279 0591 1 NONE
```

```
280 0592 1 |
281 0593 1 | IMPLICIT OUTPUTS:
282 0594 1 |     USER_STATUS: receives signal code
283 0595 1 |
284 0596 1 | ROUTINE VALUE:
285 0597 1 |     NONE
286 0598 1 |
287 0599 1 | SIDE EFFECTS:
288 0600 1 |     stack unwound to main level to return to dispatcher
289 0601 1 |
290 0602 1 | --
291 0603 1 |
292 0604 2 | BEGIN
293 0605 2 |
294 0606 2 | MAP
295 0607 2 |     SIGNAL          : REF BBLOCK,    ! signal array arg
296 0608 2 |     MECHANISM       : REF BBLOCK;    ! mechanism array arg
297 0609 2 |
298 0610 2 | EXTERNAL
299 0611 2 |     USER_STATUS    : WORD;          ! I/O status to user
300 0612 2 |
301 0613 2 | EXTERNAL ROUTINE
302 0614 2 |     SYSSUNWIND     : ADDRESSING_MODE (ABSOLUTE);
303 0615 2 |
304 0616 2 |
305 0617 2 | ! Check the signal code. The only permissible ones are SSS_UNWIND, which
306 0618 2 | ! is ignored, and SSS_CMODUSER. The error status is the 16 bit CHMU code.
307 0619 2 | ! If the error value is non-zero, store it in the user status (zero
308 0620 2 | ! means just exit). Set up a return value of 0, unwind to the current
309 0621 2 | ! depth, and return, causing the invoked function to return with failure
310 0622 2 | ! to the dispatcher.
311 0623 2 |
312 0624 2 |
313 0625 2 | IF .SIGNAL[CHFSL_SIG_NAME] EQL SSS_UNWIND THEN RETURN;
314 0626 2 | IF .SIGNAL[CHFSL_SIG_NAME] NEQ SSS_CMODUSER
315 0627 2 | THEN BUG_CHECK (ONXSIGNAL, FATAL, 'Unexpected signal name in ACP');
316 0628 2 |
317 0629 2 | IF .SIGNAL[CHFSL_SIG_ARG1] NEQ 0
318 0630 2 | THEN USER_STATUS = .SIGNAL[CHFSL_SIG_ARG1];
319 0631 2 |
320 0632 2 | MECHANISM[CHFSL_MCH_SAVRO] = 0;
321 0633 2 |
322 0634 2 | SYSSUNWIND (MECHANISM[CHFSL_MCH_DEPTH], 0);
323 0635 2 |
324 0636 2 | RETURN;
325 0637 2 |
326 0638 1 | END;                                     ! end of routine MAIN_HANDLER
```

|          |    |    |             |        |                             |        |
|----------|----|----|-------------|--------|-----------------------------|--------|
|          |    |    |             | .EXTRN | SYSSUNWIND, BUG\$_UNXSIGNAL |        |
|          |    |    | 0000 0000   | .ENTRY | MAIN_HANDLER, Save nothing  | : 0570 |
|          |    |    | AC D0 00002 | MOVL   | SIGNAL, R0                  | : 0625 |
| 00000920 | 50 | 04 | A0 D1 00006 | CMPL   | 4(R0), #2336                |        |
|          | 8F | 04 | 30 13 0000E | BEQL   | 3\$                         |        |
| 00000424 | 8F | 04 | A0 D1 00010 | CMPL   | 4(R0), #1060                | : 0626 |



|  |           |    |       |             |       |                     |                  |
|--|-----------|----|-------|-------------|-------|---------------------|------------------|
|  |           | 04 | 13    | 00018       | BEQL  | 1\$                 |                  |
|  |           |    | FEFF  | 0001A       | BUGW  |                     | 0627             |
|  |           |    | 0000* | 0001C       | .WORD | <BUG\$ UNXSIGNAL!4> |                  |
|  | 50        | 04 | AC    | D0 0001E    | 1\$:  | MOVL                | SIGNAL, R0       |
|  |           | 08 | A0    | D5 00022    |       | TSTL                | 8(R0)            |
|  |           |    | 06    | 13 00025    | BEQL  | 2\$                 |                  |
|  | 0000G     | CF | 08    | A0 B0 00027 | MOVW  | 8(R0), USER STATUS  | 0630             |
|  |           | 50 | 08    | AC D0 0002D | 2\$:  | MOVL                | MECHANISM, R0    |
|  |           |    | 0C    | A0 D4 00031 |       | CLRL                | 12(R0)           |
|  |           |    | 7E    | D4 00034    |       | CLRL                | -(SP)            |
|  |           |    | 08    | A0 9F 00036 |       | PUSHAB              | 8(R0)            |
|  | 00000000G | 9F | 02    | FB 00039    |       | CALLS               | #2, @#SYSSUNWIND |
|  |           |    | 04    | 00040       | 3\$:  | RET                 | 0638             |

; Routine Size: 65 bytes, Routine Base: \$CODE\$ + 00EC

|   |     |      |          |
|---|-----|------|----------|
| : | 327 | 0639 | 1        |
| : | 328 | 0640 | 1 END    |
| : | 329 | 0641 | 0 ELUDOM |

## PSECT SUMMARY

| Name     | Bytes | Attributes                                                |
|----------|-------|-----------------------------------------------------------|
| \$CODE\$ | 301   | NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2) |

## Library Statistics

| File                            | Symbols |                | Pages<br>Mapped | Processing<br>Time |
|---------------------------------|---------|----------------|-----------------|--------------------|
|                                 | Total   | Loaded Percent |                 |                    |
| _\$255\$DUA28:[SYSLIB]LIB.L32;1 | 18619   | 18 0           | 1000            | 00:01.9            |

## COMMAND QUALIFIERS

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

|                    |                         |
|--------------------|-------------------------|
| : Size:            | 301 code + 0 data bytes |
| : Run Time:        | 00:09.6                 |
| : Elapsed Time:    | 00:26.7                 |
| : Lines/CPU Min:   | 4010                    |
| : Lexemes/CPU-Min: | 12938                   |

DISPAT  
V04-000

<sup>0 6</sup>  
16-Sep-1984 00:45:27 VAX-11 Bliss-32 V4.0-742

Page 10

: Memory Used: 117 pages  
: Compilation Complete

EXT  
V04



0165 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

