

A large, stylized letter 'W' is formed using the characters 'S' and 'Y'. The 'S' characters are arranged in a grid-like pattern to create the vertical strokes and the central valley of the 'W'. The 'Y' characters are used for the diagonal strokes and the top horizontal bar. The overall shape is symmetrical and occupies most of the page.

```
CCCCCCCCC 000000 MM MM DDDDDDDD RRRRRRRR VV VV SSSSSSSS UU UU BBBB BBBB
CCCCCCCCC 000000 MM MM DDDDDDDD RRRRRRRR VV VV SSSSSSSS UU UU BBBB BBBB
CC CC 00 00 MMMM MMMM DD DD RR RR VV VV SS SSSSSSSS UU UU BB BB BB
CC CC 00 00 MMMM MMMM DD DD RR RR VV VV SS SSSSSSSS UU UU BB BB BB
CC CC 00 00 MM MM DD DD RR RR VV VV SS SSSSSS UU UU BB BB BB
CC CC 00 00 MM MM DD DD RRRRRRRR VV VV SSSSSS UU UU BBBB BBBB
CC CC 00 00 MM MM DD DD RRRRRRRR VV VV SSSSSS UU UU BBBB BBBB
CC CC 00 00 MM MM DD DD RR RR VV VV SS SS UU UU BB BB BB
CC CC 00 00 MM MM DD DD RR RR VV VV SS SS UU UU BB BB BB
CC CC 00 00 MM MM DD DD RR RR VV VV SS SS UU UU BB BB BB
CCCCCCCCC 000000 MM MM DDDDDDDD RR RR VV VV SSSSSSSS SSSSSSSS UUUUUUUUUU BBBB BBBB
CCCCCCCCC 000000 MM MM DDDDDDDD RR RR VV VV SSSSSSSS SSSSSSSS UUUUUUUUUU BBBB BBBB
.....
.....
.....
.....

LL LL IIIIII SSSSSSSS
LL LL IIIIII SSSSSSSS
LL LL II SS
LL LL II SS
LL LL II SS
LL LL II SS
LL LL II SSSSSS
LL LL II SSSSSS
LL LL II SS
LL LL II SS
LL LL II SS
LL LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
```


(2)	99	COM\$DELATTNAST - DELIVER ATTENTION ASTS
(3)	159	COM\$FLUSHATTNS - FLUSH ATTENTION AST LIST
(4)	202	COM\$POST - POST I.O COMPLETION INDEPENDENT OF UNIT STATUS
(5)	232	COM\$DRVDEALMEM - DEALLOCATE DRIVER MEMORY
(6)	292	COM\$SETATTNAST - SET UP ATTENTION AST
(7)	381	COM\$DELCTRLAST - DELIVER CONTROL ASTS
(8)	487	COM\$FLUSHCTRLS - FLUSH CONTROL AST LIST
(9)	539	COM\$SETCTRLAST - SET UP CONTROL AST
(10)	688	COM\$BLDCTRLAST - BUILD CONTROL AST
(11)	777	COM\$FILLCTRLAST - FILLIN A CONTROL AST CONTROL BLOCK


```
0000 1      .TITLE COMDRVSUB - COMMUNUCATION DRIVERS SUBROUTINES
0000 2      .IDENT 'V04-000'
0000 3
0000 4      *****
0000 5      *
0000 6      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8      *  ALL RIGHTS RESERVED.
0000 9      *
0000 10     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15     *  TRANSFERRED.
0000 16     *
0000 17     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19     *  CORPORATION.
0000 20     *
0000 21     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23     *
0000 24     *
0000 25     *****
0000 26
0000 27     ++
0000 28     FACILITY:
0000 29
0000 30     VAX/VMS I/O DRIVERS
0000 31
0000 32     ABSTRACT:
0000 33
0000 34     THIS MODULE CONTAINS SUBROUTINES FOR THE TERMINAL,MAILBOX AND DMC11 DRIVERS.
0000 35
0000 36     AUTHOR:
0000 37
0000 38     R.HEINEN 8-SEPT-1977
0000 39
0000 40     REVISION HISTORY:
0000 41
0000 42     V03-007 MHB0137      Mark Bramhall      12-Apr-1984
0000 43     Define minimum deallocatable block size as FKB$C_LENGTH.
0000 44     Restore stacked register after BADDALRQSZ bug check.
0000 45
0000 46     V03-006 RKS006      RICK SPITZ          2-MAR-1984
0000 47     Make sure that TAST is removed from UCB queue when
0000 48     TAST replaced and currently busy.
0000 49
0000 50     V03-005 DWT0157      David W. Thiel      30-DEC-1983
0000 51     Modify COM$DRVDEALMEM to avoid unnecessary forks.
0000 52
0000 53     V03-004 JLV0272      Jake VanNoy        14-JUN-1983
0000 54     Add abort I/O flag to out of band logic. Also
0000 55     add new entry points for checking PIDs to qualify
0000 56     AST delivery.
0000 57
```



```
0000 58 : V03-003 RKS0003 RICK SPITZ 23-SEP-1982
0000 59 : INSURE THAT TAST IS NOT BUSY WHEN A REPLACE CURRENT
0000 60 : BLOCK OPERATION OCCURS. IF SO, FLAG THE CURRENT BLOCK
0000 61 : FOR DELETE AND BUILD A NEW BLOCK.
0000 62 :
0000 63 : V03-002 KDM0002 Kathleen D. Morse 28-Jun-1982
0000 64 : Added $IODEF and $SSDEF.
0000 65 :
0000 66 :
0000 67 : --
0000 68 :
0000 69 : EXTERNAL SYMBOLS
0000 70 :
0000 71 : $ACBDEF : DEFINE AST CONTROL BLOCK
0000 72 : $CCBDEF : DEFINE CCB
0000 73 : $DYNDEF : DEFINE DYNAMIC MEMORY BLOCKS
0000 74 : $FKBDEF : DEFINE FORK BLOCK
0000 75 : $IODEF : DEFINE I/O FUNCTION CODES
0000 76 : $IPLDEF : DEFINE IPL LEVELS
0000 77 : $IRPDEF : DEFINE I/O PACKET
0000 78 : $PCBDEF : DEFINE PCB
0000 79 : $PRDEF : DEFINE PROCESSOR REGISTERS
0000 80 : $PRIDEF : DEFINE NEW PRIORITIES
0000 81 : $PRVDEF : DEFINE PRIVELEGE BITS
0000 82 : $PSLDEF : DEFINE PSL
0000 83 : $RSNDEF : DEFINE RESOURCES
0000 84 : $SSDEF : DEFINE SYSTEM STATUS CODES
0000 85 : $TASTDEF : DEFINE TERMINAL AST BLOCK
0000 86 : $UCBDEF : DEFINE UCB
0000 87 :
0000 88 : LOCAL DEFINITIONS
0000 89 :
00000000 0000 90 P1= 0
00000004 0000 91 P2= 4
00000008 0000 92 P3= 8
0000000C 0000 93 P4= 12
00000010 0000 94 P5= 16
00000014 0000 95 P6= 20
0000 96
00000000 0000 97 .PSECT WIONONPAGED
```



```
0000 99 .SBTTL COM$DELATTNAST - DELIVER ATTENTION ASTS
0000 100 :++
0000 101 : COM$DELATTNAST - DELIVER ATTENTION ASTS
0000 102 : COM$DELATTNASTP - DELIVER ATTENTION ASTS BY PID
0000 103 :
0000 104 : FUNCTIONAL DESCRIPTION:
0000 105 :
0000 106 : THIS ROUTINE IS USED BY THE TERMINAL AND MAILBOX DRIVERS TO DELIVER
0000 107 : ALL OF THE ASTS AWAITING ATTENTION. THE CONTROL BLOCKS ARE USED AS FORK BLOCKS
0000 108 : TO IPL$_QUEUEAST.
0000 109 :
0000 110 : INPUTS:
0000 111 :
0000 112 : R4 = ADDRESS OF LIST HEAD OF AST CONTROL BLOCKS
0000 113 : R5 = UCB OF UNIT
0000 114 : R6 = ACTIVE PID (AT ENTRY POINT COM$DELATTNASTP)
0000 115 :
0000 116 : OUTPUTS:
0000 117 :
0000 118 : R2,R3,R4,R5 ARE PRESERVED.
0000 119 :--
0000 120 COM$DELATTNAST:: : DELIVER ATTENTION ASTS
00F8 8F BB 0000 121 : PUSH R4,R5,R6,R7 : SAVE
56 D4 0004 122 : CLRL R6 : CLEAR SO ALL ASTS WILL BE DELIVERED
04 11 0006 123 : BRB ATTN2 : continue
0008 124
0008 125 COM$DELATTNASTP:: : DELIVER ATTENTION ASTS BY PID
00F8 8F BB 0008 126 : PUSH R4,R5,R6,R7 : SAVE
000C 127 ATTN2: :
57 54 D0 000C 128 : MOVL R4,R7 : R7 WILL TRACK LIST
000F 129
55 67 D0 000F 130 10$: : MOVL (R7),R5 : GET NEXT ENTRY
36 13 0012 131 : BEQL 50$ : IF EQL THEN NONE
56 D5 0014 132 : TSTL R6 : CHECKING PIDS?
0B 13 0016 133 : BEQL 30$ : NOT IF ZERO
56 24 A5 D1 0018 134 : CMPL ACB$_KAST+12(R5),R6 : COMPARE PIDS
05 13 001C 135 : BEQL 30$ : EQUAL, DELIVER AST
001E 136 :
001E 137 : PIDS NOT EQUAL, SKIP THIS ENTRY AND LEAVE IT IN QUEUE
001E 138 :
57 55 D0 001E 139 : MOVL R5,R7 : NEXT
EC 11 0021 140 : BRB 10$ : LOOP
0023 141 :
0023 142 : DELIVER AST AND CLOSE UP LIST
0023 143 :
67 65 D0 0023 144 30$: : MOVL (R5),(R7) : CLOSE LIST
E6 AF 9F 0026 145 : PUSHAB B*10$ : SET UP RETURN ADDRESS
0029 146 : FORK : CREATE FORK PROCESS
002F 147 :
002F 148 : AST QUEUE FORK PROCESS
002F 149 :
10 A5 18 A5 7D 002F 150 : MOVQ ACB$_KAST(R5),ACB$_AST(R5); REARRANGE ENTRIES
0B A5 20 A5 90 0034 151 : MOV B ACB$_KAST+8(R5),ACB$_RMOD(R5);
0C A5 24 A5 D0 0039 152 : MOVL ACB$_KAST+12(R5),ACB$_PID(R5);
18 A5 D4 003E 153 : CLRL ACB$_KAST(R5);
52 01 9A 0041 154 : MOVZBL #PRIS-IOCOM,R2 : SET UP PRIORITY INCREMENT
00000000'GF 17 0044 155 : JMP G$SCH$QAST : QUEUE THE AST
```


COMDRVSUB
V04-000

H 11
- COMMUNUCATION DRIVERS SUBROUTINES
COM\$DELATTNAST - DELIVER ATTENTION ASTS

15-SEP-1984 23:56:17
5-SEP-1984 03:40:49

VAX/VMS Macro V04-00
[SYS.SRC]COMDRVSUB.MAR;1

Page 4
(2)

00F8 8F BA 004A 156 50\$: POPR #^M<R3,R4,R5,R6,R7> ;
05 004E 157 RSB

```
004F 159 .SBTTL COM$FLUSHATTNS - FLUSH ATTENTION AST LIST
004F 160 :++
004F 161 : COM$FLUSHATTNS - FLUNS ATTENTION AST LIST
004F 162 :
004F 163 : THIS ROUTINE IS USED BY THE TERMINAL AND MAILBOX DRIVERS TO FLUSH
004F 164 : AN ATTENTION AST LIST. THIS IS DONE AT CANCEL I/O TIME AND WHEN A
004F 165 : QIO SPECIFIES A 0 AST ADDRESS ON A SET ATTENTION AST FUNCTION.
004F 166 : IF THE AST CONTROL BLOCK OWNER IS NO LONGER IN THE SYSTEM THE AST IS ALSO
004F 167 : FLUSHED.
004F 168 :
004F 169 :
004F 170 : INPUTS:
004F 171 :
004F 172 : R4 = PCB ADDRESS
004F 173 : R5 = UCB ADDRESS OF RELATED UNIT
004F 174 : R6 = CHANNEL NUMBER
004F 175 : R7 = LIST HEAD
004F 176 :
004F 177 : OUTPUTS:
004F 178 :
004F 179 : R0 = SS$-NORMAL
004F 180 : R1,R2,R7 ARE DESTROYED.
004F 181 :
004F 182 : --
004F 183 COM$FLUSHATTNS::
004F 184 DSBINT UCB$B DIPL(R5) ; FLUSH ATTENTION AST LIST
10$ 185 MOVL (R7),R0 ; DISABLE INTERRUPTS
0056 186 BEQL 50$ ; GET LIST ENTRY
24 A0 60 A4 D1 0058 187 CML PCBSL_PID(R4),ACBSL_KAST+12(R0); PID MATCH?
0060 188 BNEQ 40$ ; IF EQL THEN DONE
22 A0 56 B1 0062 189 CMPW R6,ACBSL_KAST+10(R0) ; IF NEQ THEN NO
0066 190 BNEQ 40$ ; CHANNEL MATCH?
67 60 D0 0068 191 MOVL (R0),(R7) ; IF NEQ THEN NO
0068 192 ENBINT ; CLOSE UP LIST TO REMOVE ENTRY
38 A4 B6 006E 193 INCW PCBSW_ASTCNT(R4) ; REENABLE INTERRUPTS
0071 194 BSBB COM$DRVDEALMEM ; RESTORE AST QUOTA
0073 195 BRB COM$FLUSHATTNS ; DEALLOCATE THE BLOCK
57 50 D0 0075 196 40$: MOVL R0,R7 ; CONTINUE
0078 197 BRB 10$ ; LOOK TO NEXT ENTRY
007A 198 50$: ENBINT ; CONTINUE
50 01 9A 007D 199 MOVZBL #SS$-NORMAL,R0 ; REENABLE INTERRUPTS
0080 200 RSB ; SET NORMAL RETURN
```



```
0081 202 .SBTTL COM$POST - POST I.O COMPLETION INDEPENDENT OF UNIT STATUS
0081 203 :++
0081 204 : COM$POST - POST I/O COMPLETION INDEPENDENT OF UNIT STATUS
0081 205 :
0081 206 : FUNCTIONAL DESCRIPTION:
0081 207 :
0081 208 : THIS ROUTINE IS USED BY THE TERMINAL, MAILBOX AND DMC DRIVER TO COMPLETE
0081 209 : I/O OPERATIONS INDEPENDENT OF THE STATUS OF THE UNIT. NO ATTEMPT IS MADE
0081 210 : TO DE-QUEUE ANOTHER PACKET OR CHANGE THE BUSY STATUS OF THE UNIT.
0081 211 :
0081 212 : INPUTS:
0081 213 :
0081 214 : R3 = I/O PACKET ADDRESS
0081 215 : R5 = UCB ADDRESS
0081 216 :
0081 217 : IMPLICIT INPUTS:
0081 218 :
0081 219 : CALLER AT DRIVER FORK IPL OR GREATER.
0081 220 : IRP$L_MEDIA AND IRP$L_MEDIA+4 ARE THE IOSB QUAD WORD.
0081 221 :
0081 222 : OUTPUTS:
0081 223 :
0081 224 : R0,R1 ARE DESTROYED.
0081 225 : --
0081 226 COM$POST::
0081 227 INCL UCB$L_OPCNT(R5) ; COMPLETE I/O
0081 228 INSQUE (R3),@IOC$GL_PSBL ; INCREMENT OPERATION COUNT
0081 229 SOFTINT #IPL$_IOPOST ; INSERT PACKET ON QUEUE
0081 230 RSB ; REQUEST FORK
; RETURN
```

00000000'FF 70 A5 D6 0081 227
63 0E 0084 228
05 008B 229
008E 230

```
008F 232 .SBTTL COM$DRVDEALMEM - DEALLOCATE DRIVER MEMORY
008F 233 :++
008F 234 : COM$DRVDEALMEM - DEALLOCATE DRIVER MEMORY
008F 235 :
008F 236 : FUNCTIONAL DESCRIPTION:
008F 237 :
008F 238 : THIS ROUTINE IS USED BY DRIVERS TO DEALLOCATE SYSTEM DYNAMIC MEMORY.
008F 239 :
008F 240 : IT CAN BE CALLED AT ANY IPL.
008F 241 :
008F 242 : INPUTS:
008F 243 :
008F 244 : R0 = ADDRESS OF THE BLOCK TO DEALLOCATE
008F 245 :
008F 246 : *****
008F 247 :
008F 248 : THE BUFFER MUST BE AT LEAST 'FKB$C_LENGTH' BYTES LONG
008F 249 : IF A FORK IS REQUIRED. THIS IS BECAUSE WE TURN THE
008F 250 : BUFFER INTO A FORK BLOCK FOR THE DELAYED DEALLOCATION.
008F 251 :
008F 252 : *****
008F 253 :
008F 254 : OUTPUTS:
008F 255 :
008F 256 : R0-R5 ARE PRESERVED.
008F 257 : --
008F 258 COM$DRVDEALMEM:: : DEALLOCATE DRIVER MEMORY
008F 259 : PUSH R3 : SAVE A REGISTER
0091 260 : SAVIPL R3 : FETCH CURRENT IPL LEVEL
0094 261 : CMPL R3,#IPL$_SYNCH : COMPARE CURRENT IPL TO SYNCH
0097 262 : BGTR 10$ : BRANCH IF IPL LEVEL > SYNCH
0099 263 : PUSHR #^M<R0,R1,R2> : SAVE MORE REGISTERS
009B 264 : JSB G^EXE$DEANONPAGED : DEALLOCATE MEMORY
00A1 265 : POPR #^M<R0,R1,R2,R3> : RESTORE REGISTERS
00A3 266 : RSB : RETURN, ALL REGISTERS PRESERVED
00A4 267 :
008F 268 10$: CMPW #FKB$C_LENGTH,FKB$W_SIZE(R0) : BIG ENOUGH FOR A FORK BLOCK?
00A8 269 : BGTRU 30$ : IF GTRU THEN NO - ERROR
00AA 270 : MOVQ R4,-(SP) : SAVE FORKING REGS
00AD 271 : MOVW #<DYN$C FRK!- : INSERT STRUCTURE TYPE
00AE 272 : <IPL$_QUEUEASTa8>>,- : AND PROPER IPL
00AE 273 : FKB$B_TYPE(R0) : FOR THIS TO BE A FORK BLOCK
00B3 274 : ASSUME FKB$B_FIPL EQ FKB$B_TYPE+1
00B3 275 : MOVL R0,R5 : COPY ADDRESS
00B6 276 : PUSHAB B^20$ : SET UP RETURN ADDRESS
00B9 277 : JSB G^EXE$FORK : CREATE FORK
00BF 278 :
00BF 279 : IPL$_QUEUEAST FORK ROUTINE
00BF 280 :
00BF 281 : MOVL R5,R0 : DEALLOCATE THE BLOCK
00C2 282 : JMP G^EXE$DEANONPAGED :
00C8 283 20$: MOVQ (SP)+,R4 : RESTORE REGISTERS
00CB 284 25$: POPL R3 : RESTORE R3
00CE 285 : RSB :
00CF 286 :
00CF 287 : BUGCHECK ON TOO SMALL A BUFFER
00CF 288 :
```

53 DD 008F 259
08 53 D1 0091 260
08 0B 14 0094 261
07 0B 14 0097 262
00000000'GF 07 BB 0099 263
0F BA 16 009B 264
05 BA 16 009B 264
05 00A1 265
00A3 266
00A4 267
08 A0 18 B1 00A4 268 10\$:
25 1A 00A8 269
7E 54 7D 00AA 270
B0 00AD 271
0A A0 0608 8F 00AE 272
55 50 D0 00AE 273
C8'AF 9F 00B3 274
00000000'GF 16 00B3 275
00B6 276
00B9 277
00BF 278
00BF 279
00BF 280
50 55 D0 00BF 281
00000000'GF 17 00C2 282
54 8E 7D 00C8 283 20\$:
53 8ED0 00CB 284 25\$:
05 00CE 285
00CF 286
00CF 287
00CF 288


```

- COMMUNUCATION DRIVERS SUBROUTINES L 11 15-SEP-1984 23:56:17 VAX/VMS Macro V04-00
COM$DRVDEALMEM - DEALLOCATE DRIVER MEMOR 5-SEP-1984 03:40:49 [SYS.SRC]COMDRVSUB.MAR;1

```

Page 8
(5)

```

F6  11  00CF  289 30$:  BUG-CHECK BADDALRQSZ      ; BUGCHECK
      00D3  290      BRB-25$      ; CONTINUE

```

C S A A A A A A A B C C C C C C C C D E E E F F F I I I I I P P P P P P

```
00D5 292 .SBTTL COM$SETATTNAST - SET UP ATTENTION AST
00D5 293 :++
00D5 294 :COM$SETATTNAST - SET UP ATTENTION AST
00D5 295 :
00D5 296 :FUNCTIONAL DESCRIPTION:
00D5 297 :
00D5 298 :THIS ROUTINE IS A SUBROUTINE USED BY THE TERMINAL AND MAILBOX DRIVERS
00D5 299 :TO PROCESS REQUESTS FOR ENABLE OR DISABLE OF ATTENTION ASTS.
00D5 300 :P1 IS THE ADDRESS OF THE AST SERVICE FOR ENABLES. P1 = 0 FOR DISABLE.
00D5 301 :FOR DISABLES, THE SPECIFIED LIST IS SEARCHED AND THE ENTRY EXTRACTED AND
00D5 302 :DEALLOCATED.
00D5 303 :FOR ENABLES, A CONTROL BLOCK IS SET UP THAT WILL DOUBLE AS THE AST CONTROL
00D5 304 :BLOCK WHEN THE AST IS DELIVERED. THE BLOCK IS FORMATTED AS FOLLOWS:
00D5 305 :
00D5 306 :ACB$B_RMOD = IPR$ QUEUEAST
00D5 307 :ACB$L_KAST = AST PC
00D5 308 :ACB$L_KAST+4 = AST PARAMETER (P2)
00D5 309 :ACB$L_KAST+8 = ACCESS MODE OF REQUEST
00D5 310 :ACB$L_KAST+10 = CHANNEL NUMBER
00D5 311 :ACB$L_KAST+12 = PID OF REQUEST
00D5 312 :
00D5 313 :THE NEW BLOCK IS PLACED AT THE HEAD OF THE CURRENT LIST.
00D5 314 :
00D5 315 :IN BOTH CASES THE I/O IS COMPLETED.
00D5 316 :
00D5 317 :INPUTS:
00D5 318 :
00D5 319 :R3 = I/O PACKET ADDRESS
00D5 320 :R4 = CURRENT PCB
00D5 321 :R5 = UCB ADDRESS
00D5 322 :R6 = ASSIGNED CCB
00D5 323 :R7 = ADDRESS OF THE CONTROL AST LIST HEAD TO CHANGE
00D5 324 :AP = ADDRESS OF THE QIO ARGLIST
00D5 325 :
00D5 326 :OUTPUTS:
00D5 327 :
00D5 328 :R0 = STATUS OF THE I/O
00D5 329 :R3 = PACKET ADDRESS
00D5 330 :R5 = UCB ADDRESS
00D5 331 :
00D5 332 :NO OTHER REGISTERS ARE PRESERVED.
00D5 333 :
00D5 334 :COMPLETION CODES:
00D5 335 :
00D5 336 :SS$ NORMAL
00D5 337 :SS$_EXQUOTA -- BUFFERED I/O OR AST QUOTA FAILURE
00D5 338 :SS$_INSUFMEM -- DYNAMIC MEMORY FAILURE
00D5 339 :--
00D5 340 COM$SETATTNAST::
00D5 341 :MOVZWL IRP$W_CHAN(R3),R6 ; SET UP ATTENTION AST
00D5 342 :MOVL P1(AP),R8 ; GET PACKET CHANNEL NUMBER
00D5 343 :BEQL 10$ ; GET USER AST ADDRESS
00D5 344 : ; IF EQL THEN DISABLE FUNCTION
00DE 345 :REQUEST TO ENABLE AST
00DE 346 :
00DE 347 :SET UP AST BLOCK
00DE 348 :
```

56 28 A3 3C 00D5 341
58 6C D0 00D9 342
5B 13 00DC 343


```
50 1C 3C 00DE 349 MOVZWL #SS$ EXQUOTA,R0 ; ASSUME AST QUOTA EXCEEDED
38 A4 B5 00E1 350 TSTW PCB$Q_ASTCNT(R4) ; AST QUOTA ALLO' CONTINUE?
56 15 00E4 351 BLEQ 20$ ; IF LEQ THEN NO
38 A4 B7 00E6 352 DECW PCB$W_ASTCNT(R4) ; ADJUST QUOTA
51 28 9A 00E9 353 MOVZBL #ACB$C_KAST+16,R1 ; SET SIZE OF NEEDED BLOCK
53 DD 00EC 354 PUSHL R3 ; SAVE PACKET ADDRESS
00000000'GF 16 00EE 355 JSB G^EXE$ALLOCBUF ; ALLOCATE THE BUFFERED BLOCK
53 8ED0 00F4 356 POPL R3 ; RESTORE PACKET ADDRESS
42 50 E9 00F7 357 BLBC R0,20$ ; IF LOW SET THEN ALLOCATED
0B A2 06 90 00FA 358 MOVB #IPL$ QUEUEAST,ACB$B_RMOD(R2); INSERT FORK IPL
18 A2 58 D0 00FE 359 MOVL R8,ACB$L_KAST(R2) ; INSERT AST ROUTINE ADDRESS
1C A2 04 AC D0 0102 360 MOVL P2(AP),ACB$L_KAST+4(R2) ; INSERT PARAMETER FOR AST
50 0B AC 02 00 EF 0107 361 EXTZV #0,#2,P3(AP),R0 ; GET REQUEST ACCESS MODE
00000000'GF 16 010D 362 JSB G^EXE$MAXACMODE ; MAXIMIZE ACCESS MODE
20 A2 50 9A 0113 363 MOVZBL R0,ACB$L_KAST+8(R2) ; INSERT IN CONTROL BLOCK
20 A2 40 8F 88 0117 364 BISB #ACB$M_QUOTA,ACB$L_KAST+8(R2); INSERT TARGET ACCESS MODE
22 A2 56 B0 011C 365 MCWV R6,ACB$L_KAST+10(R2) ; SAVE CHANNEL
24 A2 60 A4 D0 0120 366 MOVL PCB$L_PID(R4),ACB$L_KAST+12(R2); INSERT PID ADDRESS OF REQUESTOR
0125 367 ;
0125 368 ; LOCK OUT INTERRUPTS TO ENTER BLOCK ON UCB
0125 369 ;
0125 370 DSBINT UCB$B_DIPL(R5) ; LOCK OUT INTERRUPTS
62 67 D0 012C 371 MOVL (R7),(R2) ; MERGE WITH CURRENT ENTRY
67 52 D0 012F 372 MOVL R2,(R7) ; INSERT NEW ENTRY VALUE
0132 373 ENBINT ; LOWER IPL
50 01 9A 0135 374 MOVZBL #SS$_NORMAL,R0 ; SET NORMAL RETURN
05 0138 375 RSB ; RETURN VIA CALLER
0139 376
FF13 31 0139 377 10$: BRW COM$FLUSHATTNS ; DISABLE FUNCTION
013C 378
00000000'GF 17 013C 379 20$: JMP G^EXE$ABORTIO ; ABORT THE I/O
```



```
0142 381 .SBTTL COM$DELCTRLAST - DELIVER CONTROL ASTS
0142 382 :++
0142 383 :COM$DELCTRLAST - DELIVER CONTROL ASTS
0142 384 :COM$DELCTRLASTP - DELIVER CONTROL ASTS AND CHECK PIDS
0142 385
0142 386 :FUNCTIONAL DESCRIPTION:
0142 387
0142 388 :THIS ROUTINE IS USED BY THE TERMINAL SERVICES TO DELIVER
0142 389 :ALL OF THE ASTS AWAITING ATTENTION WHICH MATCH THE CONDITION.
0142 390 :THE AST BLOCK IS NOT DELETED AND IS REUSED.
0142 391
0142 392 :INPUTS:
0142 393
0142 394 :R3 = MATCH CHARACTER
0142 395 :R4 = ADDRESS OF LIST HEAD OF AST CONTROL BLOCKS
0142 396 :R5 = UCB OF UNIT
0142 397 :R6 = ACTIVE PID (AT ENTRY POINT COM$DELCTRLASTP)
0142 398
0142 399 :OUTPUTS:
0142 400
0142 401 :R3 = (Low byte) CHARACTER TO INCLUDE IN DATA STREAM OR NULL
0142 402 :TAST$V_ABO - set to signal driver to abort I/O in progress
0142 403 :TAST$V_INC - set to include this character in datastream
0142 404
0142 405 :R1,R4 DESTROYED
0142 406 :--
0142 407
0142 408 COM$DELCTRLAST:: : DELIVER CONTROL ASTS
0142 409 :PUSHR #^M<R2,R3,R5,R6> : SAVE REGISTERS
0142 410 :CLRL R6 : CLEAR SO ALL ASTs WILL BE DELIVERED
0142 411 :BRB CTRL2 : continue
0142 412
0142 413 COM$DELCTRLASTP:: : DELIVER CONTROL ASTS BY PID
0142 414 :PUSHR #^M<R2,R3,R5,R6> :
0142 415 CTRL2:
0142 416 :CLRL R2 : INIT RETURN CHARACTER
0142 417 :MOVL R4,R1 : LIST HEAD ADDRESS
0142 418 10$:
0142 419 :MOVL (R1),R1 : ADDRESS FIRST BLOCK
0142 420 :BEQL 50$ : NO MORE
0142 421 :SUBL3 #TAST$L FLINK,R1,R5 : ADDRESS START OF BLOCK
0142 422 :BBC 4(SP),TAST$L_MASK(R5),10$ : SKIP IF CHARACTER NOT IN MASK
0142 423 :
0142 424 : Check PID
0142 425 :
0142 426 :TSTL R6 : checking pids?
0142 427 :BEQL 11$ : Branch if not
0142 428 :CMPL TAST$L_PID(R5),R6 : compare
0142 429 :BNEQ 10$ : skip if not equal
0142 430 11$:
0142 431 :
0142 432 : This one matches, check include and abort
0142 433 :
0142 434 :BBC #TAST$V_ABORT, -
0142 435 :TAST$B_CTRL(R5),12$ : BRANCH IF NOT ABORT
0142 436 :MOVZBL 4(SP),R2 : FETCH CHARACTER
0142 437 :BISW #TAST$M_ABO,R2 : SET ABORT
```

006C 8F BB 0142 409
56 D4 0146 410
04 11 0148 411
014A 412
006C 8F BB 014A 413
51 52 D4 014E 415
51 54 D0 0150 416
51 61 D0 0153 418
51 69 13 0156 420
55 51 1C C3 0158 421
F1 30 A5 04 AE E1 015C 422
0162 423
0162 424
0162 425
56 56 D5 0162 426
06 13 0164 427
56 28 A5 D1 0166 428
E7 12 016A 429
016C 430
016C 431
016C 432
016C 433
0B 05 E1 016C 434
2D A5 016E 435
52 04 AE 9A 0171 436
4000 8F A8 0175 437


```

      OD 11 017A 438 BRB 15$ :
      01 017C 439 12$:
      08 2D 01 E1 017C 440 BBC #TAST$V INCLUDE,-
      52 04 A5 01 017E 441 TAST$B_CTRL(R5),15$ : SKIP IF STRIP CHARACTER
      00 52 0F 9A 0181 442 MOVZBL 4(SP),R2 : CHARACTER NOT STRIPPED
      03 03 E2 0185 443 BBSS #TAST$V_INC,R2,15$ : MARK CHARACTER PRESENT
      05 03 E2 0189 444 15$:
      C5 2D 03 0189 445 BBSS #TAST$V_BUSY,-
      24 A5 04 A5 018B 446 TAST$B_CTRL(R5),10$ : SKIP IF IN USE/ MARK IN USE
      06 9A 018E 447 MOVZBL 4(SP),TAST$L_ASTPRM(R5) : RETURN CHARACTER TO USER
      0B A5 06 90 0193 448 20$:
      B9 AF 9F 0197 449 MOVB #IPL$_QUEUEAST,ACB$B_RMOD(R5) : INSERT FORK IPL,TO ALLOW USE AS
      0197 450 : FORK BLOCK
      019A 451 PUSHAB B*10$ : SET UP RETURN ADDRESS
      01A0 452 FORK : CREATE FORK PROCESS
      01A0 453 :
      01A0 454 : AST QUEUE FORK PROCESS
      01A0 455 :
      01A0 456 ASSUME TAST$L_ASTPRM EQ <TAST$L_AST + 4>
      10 A5 20 A5 7D 01A0 457 ASSUME ACP$B_ASTPRM EQ <ACP$B_AST + 4>
      30 88 01A5 458 22$: MOVQ TAST$L_AST(R5),ACP$B_AST(R5) : REARRANGE ENTRIES
      0B A5 2C A5 90 01A7 459 BISB #<ACP$B_NODELETE!ACP$B_PKAST>,- : PK AST, NODELETE
      0C A5 28 A5 90 01A9 461 MOVB TAST$B_RMOD(R5),ACP$B_RMOD(R5) : MODE
      18 A5 CA AF DE 01AE 462 MOVL TAST$L_PID(R5),ACP$L_PID(R5) : PID
      52 01 9A 01B3 463 MOVAL B*ASTACNTNG,ACP$L_KAST(R5) : ADDRESS OF PIGGY-BACK KERNEL ROUTI
      00000000 GF 17 01B8 464 MOVZBL #PRI$ IOCOM,R2 : SET UP PRIORITY INCREMENT
      04 AE 52 DO 01C1 465 JMP G*SCH$QAST : QUEUE THE AST
      006C 8F 05 01BB 466 50$:
      01C1 467 MOVL R2,4(SP) : RETURN ANY CHARACTER
      01C5 468 POPR #*M<R2,R3,R5,R6> : RESTORE REGISTERS
      01C9 469 RSB
      01CA 470 :
      01CA 471 : PIGGY-BACK KERNEL MODE ROUTINE TO:
      01CA 472 : 1. MARK TAST AVAILABLE FOR USE
      01CA 473 : 2. DEALLOCATE "LOST" BLOCK(S) AND RETURN THEIR QUOTA
      01CA 474 :
      01CA 475 :
      01CA 476 ASTACNTNG:
      2D A5 08 8A 01CA 477 BICB #TAST$M_BUSY, TAST$B_CTRL(R5) ; Mark block available.
      15 2D A5 04 E1 01CE 478 BBC #TAST$V_LOST, - ; Branch if still using this block.
      01D3 479 TAST$B_CTRL(R5), 150$ ; Otherwise,
      50 28 A5 3C 01D3 480 MOVZWL TAST$L_PID(R5), R0 ; use PID index to locate PCB for
      00000000 FF40 DO 01D7 481 MOVL @L*SCH$GL_PCBVEC[R0], R0 ; this process.
      38 A0 B6 01DF 482 INCW PCB$W_ASTCNT(R0) ; Then return AST quota for and
      50 55 DO 01E2 483 MOVL R5,R0- ; deallocate this block.
      FE18 30 01E5 484 BSBW EXE$DEANONPAGED
      05 01E8 485 150$: RSB ; That completes piggy-back accounting.
```



```
01E9 487 .SBTTL COM$FLUSHCTRLS - FLUSH CONTROL AST LIST
01E9 488 :++
01E9 489 : COM$FLUSHCTRLS - FLUSH CONTROL AST LIST
01E9 490 :
01E9 491 : THIS ROUTINE IS USED BY THE TERMINAL SERVICES TO FLUSH
01E9 492 : THE CONTROL AST LIST. THIS IS DONE AT CANCEL I/O TIME AND WHEN A
01E9 493 : QIO SPECIFIES A 0 AST ADDRESS ON A SET AST FUNCTION.
01E9 494 : THE SUMMARY MASK POINTED TO BY R2 IS UPDATED.
01E9 495 :
01E9 496 : INPUTS:
01E9 497 :
01E9 498 : R2 = ADDRESS OF SUMMARY MASK
01E9 499 : R4 = PCB ADDRESS
01E9 500 : R5 = UCB ADDRESS OF RELATED UNIT
01E9 501 : R6 = CHANNEL NUMBER
01E9 502 : R7 = LIST HEAD
01E9 503 :
01E9 504 : OUTPUTS:
01E9 505 :
01E9 506 : R0 = SS$ NORMAL
01E9 507 : R1 AND R7 ARE DESTROYED.
01E9 508 :
01E9 509 : --
01E9 510 COM$FLUSHCTRLS::
01E9 511 DSBINT UCB$B_DIPL(R5) ; FLUSH CONTROL AST LIST
01E9 512 CLRL R1 ; DISABLE INTERRUPTS
01E9 513 5$: MOVL (R7),R0 ; INIT RETURN MASK
01E9 514 BEQL 50$ ; GET LIST ENTRY
01E9 515 50$ ; IF EQL THEN DONE
01E9 516 515: SUBL #TAST$L_FLINK,R0 ; COMPUTE START OF BLOCK
01E9 517 CMPL PCB$L_PID(R4),TAST$L_PID(R0); PID MATCH?
01E9 518 BNEQ 40$ ; IF NEQ THEN NO
01E9 519 CMPW R6,TAST$W_CHAN(R0) ; CHANNEL MATCH?
01E9 520 BNEQ 40$ ; IF NEQ THEN NO
01E9 521 MOVL TAST$L_FLINK(R0),(R7) ; CLOSE UP LIST TO REMOVE ENTRY
01E9 522 521: BBC #TAST$V_BUSY,TAST$B_CTRL(R0),20$; BLOCK CAN BE DELETED NOW?
01E9 523 BISB #TAST$M_LOST,TAST$B_CTRL(R0) ; NO, FLAG IT FOR LATER DELETION
01E9 524 BRB 5$ ; THAT'S ALL WE CAN DO RIGHT NOW
01E9 525 525:
01E9 526 20$: JSB G*COM$DRVDEALMEM ; DEALLOCATE THE BLOCK
01E9 527 INCW PCB$W_ASTCNT(R4) ; RESTORE AST QUOTA
01E9 528 BRB 5$ ; CONTINUE
01E9 529 529:
01E9 530 40$: BISL TAST$L_MASK(R0),R1 ; OR IN ACTIVE CONTROL CHARACTERS
01E9 531 MOVAL TAST$L_FLINK(R0),R7 ; LOOK TO NEXT ENTRY
01E9 532 BRB 5$ ; CONTINUE
01E9 533 533:
01E9 534 50$: MOVL R1,(R2) ; UPDATE SUMMARY MASK
01E9 535 ENBINT ; REENABLE INTERRUPTS
01E9 536 MOVZBL #SS$_NORMAL,R0 ; SET NORMAL RETURN
01E9 537 RSB
```

28	A0	60	A4	D1	01FA	516	CMPL	PCB\$L_PID(R4),TAST\$L_PID(R0);	PID MATCH?
2E	A0	56	B1	0201	518	CMPW	R6,TAST\$W_CHAN(R0)	CHANNEL MATCH?	
67	1C	A0	D0	0207	520	MOVL	TAST\$L_FLINK(R0),(R7)	CLOSE UP LIST TO REMOVE ENTRY	
06	2D	A0	03	E1	020B	521	BBC	#TAST\$V_BUSY,TAST\$B_CTRL(R0),20\$;	BLOCK CAN BE DELETED NOW?
	2D	A0	10	88	0210	523	BISB	#TAST\$M_LOST,TAST\$B_CTRL(R0)	; NO, FLAG IT FOR LATER DELETION
			DC	11	0214	524	BRB	5\$	; THAT'S ALL WE CAN DO RIGHT NOW
0000008F	'GF	16	0216	526	20\$:	JSB	G*COM\$DRVDEALMEM	DEALLOCATE THE BLOCK	
38	A4	B6	021C	527		INCW	PCB\$W_ASTCNT(R4)	RESTORE AST QUOTA	
	D1	11	021F	528		BRB	5\$	CONTINUE	
51	30	A0	C8	0221	530	40\$:	BISL	TAST\$L_MASK(R0),R1	OR IN ACTIVE CONTROL CHARACTERS
57	1C	A0	DE	0225	531		MOVAL	TAST\$L_FLINK(R0),R7	LOOK TO NEXT ENTRY
		C7	11	0229	532		BRB	5\$	CONTINUE
62	51	D0	022B	534	50\$:	MOVL	R1,(R2)	UPDATE SUMMARY MASK	
50	01	9A	0231	535		ENBINT		REENABLE INTERRUPTS	
		05	0234	536		MOVZBL	#SS\$_NORMAL,R0	SET NORMAL RETURN	
				537		RSB			


```
0235 539 .SBTTL COM$SETCTRLAST - SET UP CONTROL AST
0235 540 :++
0235 541 : COM$SETCTRLAST - SET CONTROL AST
0235 542 :
0235 543 : FUNCTIONAL DESCRIPTION:
0235 544 :
0235 545 : This routine is used by the terminal services FDT routines during the
0235 546 : processing of requests to enable out-of-band ASTs. The current list is
0235 547 : scanned for a block with a requestor matching that of the request being
0235 548 : processed. If a match is found, the specified mask and AST routine address
0235 549 : in the matching block are replaced. If no match is found, COM$BLDCTRLAST is
0235 550 : called to create a new block. The summary mask pointed to by R2 is updated
0235 551 : to be the inclusive or of all masks in the control AST list pointed to by R7.
0235 552 :
0235 553 : If either the AST routine address or the out-of-band enable mask --
0235 554 : a<P2(AP)+4> -- is zero, all out-of-band AST requests entered by this
0235 555 : requestor are flushed from queue pointed to by R7.
0235 556 :
0235 557 : INPUTS:
0235 558 :
0235 559 : R2 = CURRENT SUMMARY MASK ADDRESS
0235 560 : R3 = I/O PACKET ADDRESS
0235 561 : R4 = CURRENT PCB
0235 562 : R5 = UCB ADDRESS
0235 563 : R7 = ADDRESS OF THE CONTROL AST LIST HEAD TO CHANGE
0235 564 : AP = ADDRESS OF THE QIO ARGLIST
0235 565 :
0235 566 : P1(AP) = ADDRESS OF AST ROUTINE TO CALL WHEN OUT-OF-BAND CHARACTER IS
0235 567 : TYPED (0 ==> FLUSH QUEUE)
0235 568 : P2(AP) = ADDRESS OF SHORT FORM TERMINATOR MASK GIVING WHICH OUT-OF-
0235 569 : BAND CHARACTERS WILL PRECIPITATE DELIVERY OF AN AST
0235 570 : (0 MASK VALUE ==> FLUSH QUEUE: This address will be passed
0235 571 : as the AST parameter when the AST is delivered)
0235 572 : P3(AP) = ACCESS MODE IN WHICH THE AST IS TO BE DELIVERED
0235 573 : (This is maximized against the caller's access mode)
0235 574 :
0235 575 : IPL at entry is assumed to be IPL$_ASTDEL.
0235 576 :
0235 577 : OUTPUTS:
0235 578 :
0235 579 : R0 = STATUS OF THE I/O
0235 580 : (SS$ NORMAL only; all others return via EXE$ABORTIO)
0235 581 : R2 = ADDRESS OF TAST BLOCK
0235 582 : R3 = I/O PACKET ADDRESS
0235 583 : R4 = CURRENT PCB ADDRESS
0235 584 : R5 = UCB ADDRESS
0235 585 : NO OTHER REGISTERS ARE PRESERVED.
0235 586 :
0235 587 : IPL at exit is the same as IPL at entry.
0235 588 :
0235 589 : The summary mask pointed to by R2 is updated to be the inclusive or of
0235 590 : all masks in the control AST list pointed to by R7.
0235 591 :
0235 592 : The control AST list pointed to by R7 is flushed, extended, or an
0235 593 : entry in the list is updated.
0235 594 :
0235 595 : COMPLETION CODES:
```



```
0235 596 :  
0235 597 : SSS_NORMAL  
0235 598 :  
0235 599 : The following return codes are reported by a JMP to EX$ABORTIO:  
0235 600 : SSS_ACCVIO -- specified mask not accessible  
0235 601 : SSS_EXQUOTA -- buffered I/O or AST quota failure  
0235 602 : SSS_INSUFMEM -- dynamic memory failure  
0235 603 :--  
0235 604 :  
0235 605 : COM$SETCTRLAST, COM$BLDCTRLAST, and COM$FILLCTRLAST build and use the  
0235 606 : following information block. COM$SETCTRLAST builds the block on the stack  
0235 607 : and points R11 to it. COM$BLDCTRLAST and COM$FILLCTRLAST use information  
0235 608 : stored to build or update a TAST control block.  
0235 609 :  
0235 610 : $OFFSET 0, POSITIVE, <-  
0235 611 : ASTROUT, - ; User's AST routine address  
0235 612 : ASTPARM, - ; AST routine parameter  
0235 613 : USRMSKADR, - ; Pointer to user's OOB mask  
0235 614 : CURMSKADR, - ; Pointer to current summary mask  
0235 615 : <TINFO SIZE, 0> - ; Size of the block  
0235 616 : >  
0000 ASTROUT:  
0004 ASTPARM:  
0008 USRMSKADR:  
000C CURMSKADR:  
0010 TINFO SIZE:  
0235 617  
0235 618  
0235 619 COM$SETCTRLAST:: ; SET UP CONTROL AST  
0235 620  
56 28 A3 3C 0235 621 MOVZWL IRPSW CHAN(R3), R6 ; Get packet channel number.  
58 6C D0 0239 622 MOVL P1(AP), R8 ; Get user AST address.  
AB 13 023C 623 BEQL COM$FLUSHCTRLS ; If its zero, flush OOB requests.  
59 04 AC D0 023E 624 MOVL P2(AP), R9 ; Get address of user OOB mask.  
50 0C 3C 0242 625 MOVZWL #SS$ ACCVIO, R0 ; Assume no access  
50 14 3C 0245 626 IFNORD #8, (R9), 5$ ; Probe for read access.  
69 D5 024B 627 MOVZWL #SS$ BADPARAM, R0 ; Assume invalid mask format  
08 12 024E 628 TSTL (R9) ; Verify short form terminator  
04 A9 D5 0250 629 BNEQ 5$ ; mask format.  
09 12 0252 630 TSTL 4(R9) ; Any OOB characters specified?  
FF8F 31 0255 631 BNEQ 7$ ; If there are some, proceed.  
00000000'GF 17 0257 632 BRW COM$FLUSHCTRLS ; Otherwise, flush OOB requests.  
025A 633 5$: JMP G^EX$ABORTIO ; I/O failed, abort request  
0260 634 7$:  
0260 635  
0260 636  
0260 637 ; We are now holding a valid request of enable an out-of-band character AST  
0260 638  
0260 639  
0260 640 ASSUME ASTPARM EQ <ASTROUT + 4>  
SE F0 AE 9E 0260 641 MOVAB -TINFO SIZE(SP), SP ; Allocate info. block on the stack.  
58 5E D0 0264 642 MOVL SP, R11 ; Save info. block address.  
6B 58 7D 0267 643 MOVQ R8, ASTROUT(R11) ; Save AST routine and parameter info.  
08 AB 04 A9 DE 026A 644 MOVAL 4(R9), USRMSKADR(R11) ; Save address of user OOB mask.  
OC AB 52 D0 026F 645 MOVL R2, CURMSKADR(R11) ; Save current summary mask address.  
58 7C 0273 646 CLRQ R8 ; Clear accumulation summary mask and  
0275 647 ; a local flags longword.
```


5A	D4	0275	648	CLRL	R10	; TAST address		
		0277	649	DSBINT	UCB\$B_DIPL(R5)	; Interlock queue access.		
		027E	650					
52	67	D0	027E	651	10\$: MOVL	(R7), R2 ; Get list entry.		
	36	13	0281	652	BEQL	20\$; Branch if no more entries in list.		
52	E4	A2	9E	0283	MOVAB	-TAST\$FLINK(R2), R2 ; Compute start of block.		
28	A2	60	A4	D1	0287	654 CMPL	PCB\$PID(R4), TAST\$PID(R2) ; Do the PIDs match?	
	21	12	028C	655	BNEQ	15\$; Branch if PIDs don't match.		
2E	A2	56	B1	028E	656	CMPW	R6, TAST\$W_CHAN(R2) ; Is the channel right?	
	1B	12	0292	657	BNEQ	15\$; Branch if not the right channel.		
			0294	658				
OD	2D	A2	03	E1	0294	659	BBC	#TAST\$V_BUSY, TAST\$B_CTRL(R2), 12\$; Block can be reused now
	2D	A2	10	88	0299	660	BISB	#TAST\$M_LOST, TAST\$B_CTRL(R2) ; No, flag it for later deletion
67	1C	A2	D0	029D	661	MOVL	TAST\$FLINK(R2), (R7) ; Remove the entry from the queue	
	59	02	C8	02A1	662	BISL	#2, R9 ; Indicate that a match was found	
		13	11	02A4	663	BRB	20\$; And build a new one, since the old	
				02A6	664		; one is busy and will be deleted when	
				02A6	665		; the AST is delivered.	
5A	52	D0	02A6	666	12\$: MOVL	R2, R10 ; Save TAST address		
	006D	30	02A9	667	BSBW	COM\$FILLCTRLAST ; Update matching TAST control block.		
59	01	C8	02AC	668	BISL	#1, R9 ; Indicate that an update was done.		
			02AF	669				
58	30	A2	C8	02AF	670	15\$: BISL	TAST\$MASK(R2), R8 ; Accumulate OOB mask data for queue.	
57	1C	A2	DE	02B3	671	MOVAL	TAST\$FLINK(R2), R7 ; Advance to next queue entry.	
		C5	11	02B7	672	BRB	10\$; Loop until entire queue processed.	
				02B9	673			
OC	BB	58	D0	02B9	674	20\$: MOVL	R8, @CURMSKADR(R11) ; Make accumulated OOB mask for queue	
				02BD	675		; the current OOB mask for the queue.	
				02BD	676		; Return to caller's IPL.	
50	01	3C	02C0	677	ENBINT			
52	5A	D0	02C3	678	MOVZWL	#SS\$ NORMAL, R0 ; Assume work completed successfully.		
	02	59	E8	02C6	679	MOVL	R10, R2 ; Valid only if R9 LBS	
	0E	10	02C9	680	BLBS	R9, 30\$; If no blocks in the queue were		
5E	10	AE	9E	02CB	681	30\$: BSBB	COM\$BLDCTRLAST ; updated, build a new block.	
	01	50	E9	02CF	682	MOVAB	TINFO\$SIZE(SP), SP ; Restore stack.	
			05	02D2	683	BLBC	R0, 50\$; Was COM\$BLDCTRLAST successful?	
				02D3	684	RSB	; if so, return to caller.	
00000000'GF	17	02D3	685	50\$: JMP	G^EXE\$ABORTIO ; If it failed, abort I/O request.			
		02D9	686					


```
02D9 688 .SBTTL COM$BLDCTRLAST - BUILD CONTROL AST
02D9 689 :++
02D9 690 : COM$BLDCTRLAST - BUILD NEW CONTROL AST PACKET
02D9 691 :
02D9 692 : FUNCTIONAL DESCRIPTION:
02D9 693 :
02D9 694 : This routine builds a new terminal AST control block and inserts it after
02D9 695 : the TAST entry pointed to by R7. The control block will double as an AST
02D9 696 : control block when a AST is delivered. The block will be reused until the
02D9 697 : out-of-band AST request is canceled. The summary mask pointed to by
02D9 698 : CURMSKADR(R11) is inclusively or'ed with the user out-of-band mask pointed
02D9 699 : to by USRMSKADR(R11).
02D9 700 :
02D9 701 : INPUTS:
02D9 702 :
02D9 703 : R3 = I/O PACKET ADDRESS
02D9 704 : R4 = CURRENT PCB
02D9 705 : R5 = UCB ADDRESS
02D9 706 : R6 = CHANNEL ON WHICH OOB REQUEST IS BEING MADE
02D9 707 : R7 = ADDRESS OF THE CONTROL AST LIST ENTRY PRECEDING THE POINT WHERE
02D9 708 : THE NEW ENTRY IS TO BE ADDED.
02D9 709 : R11= THE ADDRESS OF A TAST INFORMATION LIST (SEE COM$SETCTRLAST)
02D9 710 : AP = ADDRESS OF THE QIO ARGLIST
02D9 711 :
02D9 712 : P3(AP) = ACCESS MODE IN WHICH THE AST IS TO BE DELIVERED
02D9 713 : (This is maximized against the caller's access mode)
02D9 714 :
02D9 715 : ASTROUT(R11) = ADDRESS OF AST ROUTINE TO CALL WHEN OUT-OF-BAND
02D9 716 : CHARACTER IS TYPED
02D9 717 : ASTPARAM(R11) = AST PARAMETER VALUE TO BE PASSED TO AST ROUTINE WHEN
02D9 718 : OUT-OF-BAND AST IS DELIVERED
02D9 719 : CURMSKADR(R11)= ADDRESS OF THE CURRENT OUT-OF-BAND SUMMARY MASK
02D9 720 : USRMSKADR(R11)= ADDRESS OF OUT-OF-BAND MASK SPECIFIED BY USER FOR
02D9 721 : THIS AST ENABLE
02D9 722 :
02D9 723 : IPL at entry is assumed to be IPL$_ASTDEL.
02D9 724 :
02D9 725 : OUTPUTS:
02D9 726 :
02D9 727 : R0 = STATUS OF THE I/O
02D9 728 : R1 & R2 DESTROYED
02D9 729 : ALL OTHER REGISTERS PRESERVED
02D9 730 :
02D9 731 : IPL at entry is assumed to be IPL$_ASTDEL.
02D9 732 :
02D9 733 : A TAST control block is allocated, filled in, and linked after the
02D9 734 : entry pointed to by R7.
02D9 735 :
02D9 736 : COMPLETION CODES:
02D9 737 :
02D9 738 : SSS_NORMAL
02D9 739 : SSS_EXQUOTA -- BUFFERED I/O OR AST QUOTA FAILURE
02D9 740 : SSS_INSUFMEM -- DYNAMIC MEMORY FAILURE
02D9 741 : --
02D9 742 :
02D9 743 : COM$BLDCTRLAST:
02D9 744 :
```



```
38 A4 B5 02D9 745 TSTW PCBSW_ASTCNT(R4) ; Is there enough AST quota?
31 15 02DC 746 BLEQ 91$ ; Branch if insufficient AST quota.
38 A4 B7 02DE 747 DECW PCBSW_ASTCNT(R4) ; Deduct from AST quota.
53 DD 02E1 748 PUSHL R3 ; Save reg. destroyed by EXE$ALLOCBUF.
51 34 9A 02E3 749 MOVZBL #TAST$C LENGTH, R1 ; Set size of TAST block.
00000000 GF 16 02E6 750 JSB G^EXE$ALLOCBUF ; Allocate for the TAST control block.
53 8ED0 02EC 751 POPL R3 ; Restore saved register.
21 50 E9 02EF 752 BLBC R0, 93$ ; Branch if allocation failed.
25 10 02F2 753 BSBB COM$FILLCTRLAST ; Fill in newly allocated block.
02F4 754 ;
02F4 755 ;
02F4 756 ; INSERT NEWLY BUILD TAST CONTROL BLOCK
02F4 757 ;
02F4 758 ;
02F4 759 DSBINT UCB$B_DIPL(R5) ; Interlock access to queue links.
1C A2 67 D0 02FB 760 MOVL (R7), TAST$L_FLINK(R2) ; Move list for. pointer to new entry.
67 1C A2 DE 02FF 761 MOVAL TAST$L_FLINK(R2), (R7) ; Link new entry to current list.
OC BB 30 A2 C8 0303 762 BISL TAST$L_MASK(R2), @CURMSKADR(R11) ; Update summary mask.
50 01 3C 0308 763 ENBINT ; Restore previous IPL.
05 030B 764 MOVZWL #SS$_NORMAL, R0 ; Indicate that build succeeded.
030E 765 RSB ; Return to caller.
030F 766 ;
030F 767 ;
030F 768 ; ERROR RETURNS:
030F 769 ;
030F 770 ;
50 1C 3C 030F 771 91$: MOVZWL #SS$_EXQUOTA, R0 ; AST quota exceeded.
05 0312 772 RSB
0313 773
50 0124 8F 3C 0313 774 93$: MOVZWL #SS$_INSFMEM, R0 ; Insufficient dynamic memory.
05 0318 775 RSB
```

```
0319 777 .SBTTL COMSFILLCTRLAST - FILLIN A CONTROL AST CONTROL BLOCK
0319 778 :++
0319 779 : COMSFILLCTRLAST - FILLIN A CONTROL AST CONTROL BLOCK
0319 780 :
0319 781 : FUNCTIONAL DESCRIPTION:
0319 782 :
0319 783 : This routine fills in the terminal AST control block pointed to by R2. The
0319 784 : block may be either a previously allocated block which is already linked to
0319 785 : a control AST queue, or a newly allocated block which is being filled in for
0319 786 : the first time.
0319 787 :
0319 788 : INPUTS:
0319 789 :
0319 790 : R2 = ADDRESS OF TAST CONTROL BLOCK TO BE FILLED IN
0319 791 : R3 = I/O PACKET ADDRESS
0319 792 : R4 = CURRENT PCB
0319 793 : R6 = CHANNEL ON WHICH OOB AST REQUEST IS BEING MADE
0319 794 : R11= THE ADDRESS OF A TAST INFORMATION LIST (SEE COM$SETCTRLAST)
0319 795 : AP = ADDRESS OF THE QIO ARGLIST
0319 796 :
0319 797 : P3(AP) = ACCESS MODE IN WHICH THE AST IS TO BE DELIVERED
0319 798 : (This is maximized against the caller's access mode)
0319 799 :
0319 800 : ASTROUT(R11) = ADDRESS OF AST ROUTINE TO CALL WHEN OUT-OF-BAND
0319 801 : CHARACTER IS TYPED
0319 802 : ASTPARAM(R11) = AST PARAMETER VALUE TO BE PASSED TO AST ROUTINE WHEN
0319 803 : OUT-OF-BAND AST IS DELIVERED
0319 804 : USRMSKADR(R11)= ADDRESS OF OUT-OF-BAND MASK SPECIFIED BY USER FOR
0319 805 : THIS AST ENABLE
0319 806 :
0319 807 : If this routine is called to operate on a TAST block which is already
0319 808 : linked to a control queue, it should be called at device IPL.
0319 809 : Otherwise, it can be called at IPL$_ASTDEL.
0319 810 :
0319 811 : OUTPUTS:
0319 812 :
0319 813 : R0 & R1 ARE DESTROYED.
0319 814 : ALL OTHER REGISTERS ARE PRESERVED.
0319 815 :
0319 816 : IPL at exit is the same as IPL at entry.
0319 817 :
0319 818 : The TAST control block pointed to by R2 is filled in.
0319 819 :
0319 820 : COMPLETION CODES:
0319 821 :
0319 822 : There is no completion status. This routine is always successful.
0319 823 :--
0319 824 :
0319 825 : COMSFILLCTRLAST:
0319 826 :
0319 827 : ASSUME TAST$L_ASTPRM EQ <TAST$L_AST + 4>
0319 828 : ASSUME ASTPRM EQ <ASTROUT + 4>
0319 829 : MOVQ ASTROUT(R11), - ; Plant AST routine and
0319 830 : TAST$L_AST(R2) ; parameter addresses.
0319 831 : MOVL @USRMSKADR(R11), - ;
0319 832 : TAST$L_MASK(R2) ; Plant OOB mask.
0319 833 : CLRL R0 ; Assume no flags
```

20 A2 6B 7D 0319 829
30 A2 08 BB D0 031D 830
50 D4 0322 832
0322 833


```
0324 834  
0324 835  
0324 836  
05 20 A3 0C E1 0324 837  
50 20 88 0329 838  
08 11 0329 839  
032C 840  
032E 841  
032E 842  
032E 843  
03 20 A3 0B E1 032E 844 5$:  
50 02 88 0333 845  
0333 846  
0336 847  
50 08 AC 2D A2 50 90 0336 848 10$:  
51 51 02 00 EF 033A 849  
51 51 02 51 DC 0340 850  
51 51 02 16 EF 0342 851  
51 51 02 50 91 0347 852  
51 51 03 18 034A 853  
50 51 90 034C 854  
2C A2 50 90 034F 855 20$:  
2E A2 56 B0 0353 856  
28 A2 60 A4 D0 0357 857  
05 035C 858  
035D 859  
035D 860  
; Check for ABORT  
BBC #IOSV-IT ABORT, -  
IRPSW-FUNC(R3), 5$ ; Branch if not abort  
BISB #TAST$M-ABORT, R0 ; Else set abort flag.  
BRB 10$ ; Ignore INCLUDE  
; Check for INCLUDE  
BBC #IOSV-INCLUDE, -  
IRPSW-FUNC(R3), 10$ ; BR if striping OOB chars.  
BISB #TAST$M-INCLUDE, R0 ; Else set no-strip flag.  
MOVB R0, TAST$B_CTRL(R2) ; Set TAST control field.  
EXTZV #0, #2, P3(AP), R0 ; Get requested delivery access mode.  
MOVPSL R1 ; Get access mode of requestor.  
EXTZV #PSL$V-PRVMOD, #PSL$S-PRVMOD, R1, R1 ; If requestor's access  
CMPB R0, R1 ; mode is bigger than delivery access  
BGEQ 20$ ; mode, then delivery AST in  
MOVB R1, R0 ; requestor's access mode.  
MOVB R0, TAST$B_RMOD(R2) ; Plant delivery access mode.  
MOVW R6, TAST$W-CHAN(R2) ; Plant requestor's channel.  
MOVL PCB$SL_PID(R4), TAST$SL_PID(R2) ; Plant requestor's PID.  
RSB ; Return to caller.  
.END
```

COMDRVSUB
Symbol table

L 12
- COMMUNUCATION DRIVERS SUBROUTINES

15-SEP-1984 23:56:17 VAX/VMS Macro V04-00
5-SEP-1984 03:40:49 [SYS.SRC]COMDRVSUB.MAR;1

Page 21
(11)

```
ACBSB_RMOD      = 0000000B
ACBSL_AST       = 00000010
ACBSL_ASTPRM    = 00000014
ACBSL_KAST      = 00000018
ACBSL_PID       = 0000000C
ACBSM_NODELETE  = 00000020
ACBSM_PKAST     = 00000010
ACBSM_QUOTA     = 00000040
ASTACNTNG      = 000001CA R      02
ASTPARM        = 00000004
ASTROUT        = 00000000
ATTN2          = 0000000C R      02
BUGS_BADDALRQSZ ***** X      02
COM$BLDCTRLAST = 000002D9 R      02
COM$DELATTNAST  = 00000000 RG     02
COM$DELATTNASTP = 00000008 RG     02
COM$DELCTRLAST  = 00000142 RG     02
COM$DELCTRLASTP = 0000014A RG     02
COM$DRVDEALMEM  = 0000008F RG     02
COM$FILLCTRLAST = 00000319 R      02
COM$FLUSHATTNS  = 0000004F RG     02
COM$FLUSHCTRLS  = 000001E9 RG     02
COM$POST        = 00000081 RG     02
COM$SETATTNAST  = 000000D5 RG     02
COM$SETCTRLAST  = 00000235 RG     02
CTRL2          = 0000014E R      02
CURMSKADR      = 0000000C
DIR...         = 00000001
DYN$C_FRK      = 00000008
EXE$ABORTIO     ***** X      02
EXE$ALLOCBUF    ***** X      02
EXE$DEANONPAGED ***** X      02
EXE$FORK        ***** X      02
EXE$MAXACMODE   ***** X      02
FKBSB_FIPL      = 0000000B
FKBSB_TYPE      = 0000000A
FKB$C_LENGTH    = 00000018
FKBSW_SIZE      = 00000008
IOSV_INCLUDE    = 0000000B
IOSV_TT_ABORT   = 0000000C
IO$GL_P$BL      ***** X      02
IPL$_IOPOST     = 00000004
IPL$_QUEUEAST   = 00000006
IPL$_SYNCH      = 00000008
IRP$B_CHAN      = 00000028
IRP$W_FUNC      = 00000020
P1              = 00000000
P2              = 00000004
P3              = 00000008
P4              = 0000000C
P5              = 00000010
P6              = 00000014
PCBSL_PID       = 00000060
PCBSW_ASTCNT    = 00000038
PR$_IPL         = 00000012
PR$_SIRR        = 00000014
PRI$_IOCOM      = 00000001
```

```
PSL$S_PRIVMOD  = 00000002
PSL$V_PRIVMOD  = 00000016
SAVABS...      = 00000010
SCH$GL_PCBVEC  ***** X      02
SCH$QAST       ***** X      02
SS$_ACCVIO     = 0000000C
SS$_BADPARAM   = 00000014
SS$_EXQUOTA    = 0000001C
SS$_INSMEM     = 00000124
SS$_NORMAL     = 00000001
TAST$B_CTRL    = 0000002D
TAST$B_RMOD    = 0000002C
TAST$C_LENGTH  = 00000034
TAST$L_AST     = 00000020
TAST$L_ASTPRM  = 00000024
TAST$L_FLINK   = 0000001C
TAST$L_MASK    = 00000030
TAST$L_PID     = 00000028
TAST$M_ABO     = 00004000
TAST$M_ABORT   = 00000020
TAST$M_BUSY    = 00000008
TAST$M_INCLUDE = 00000002
TAST$M_LOST    = 00000010
TAST$V_ABORT   = 00000005
TAST$V_BUSY    = 00000003
TAST$V_INC     = 0000000F
TAST$V_INCLUDE = 00000001
TAST$V_LOST    = 00000004
TAST$W_CHAN    = 0000002E
TINFO$SIZE     = 00000010
UCBSB_DIPL     = 0000005E
UCBSL_OPCNT    = 00000070
USRMSRADR      = 00000008
```


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

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000010 (16.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
WIONONPAGED	0000035D (861.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.04	00:00:02.07
Command processing	119	00:00:00.45	00:00:05.91
Pass 1	463	00:00:17.71	00:00:55.87
Symbol table sort	0	00:00:03.05	00:00:09.48
Pass 2	161	00:00:03.56	00:00:13.09
Symbol table output	12	00:00:00.10	00:00:00.55
Psect synopsis output	1	00:00:00.07	00:00:00.94
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	789	00:00:24.98	00:01:27.91

The working set limit was 1650 pages.
103255 bytes (202 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1887 non-local and 37 local symbols.
860 source lines were read in Pass 1, producing 15 object records in Pass 2.
34 pages of virtual memory were used to define 32 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	18
_\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	11
TOTALS (all libraries)	29

2060 GETS were required to define 29 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:COMDRVSUB/OBJ=OBJ\$:COMDRVSUB MSRC\$:COMDRVSUB/UPDATE=(ENH\$:COMDRVSUB)+EXECMLS/LIB

0373

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY