

[illegible]

```
MM      MM      CCCCCCCC  HH      HH  EEEEEEEEEEE  CCCCCCCC  KK      KK  77777777  5555555555  000000
MM      MM      CCCCCCCC  HH      HH  EEEEEEEEEEE  CCCCCCCC  KK      KK  77777777  5555555555  000000
MMM     MMM     CC      HH      HH  EEEEEEEEEEE  CC      CC  KK      KK  77      77  55      55  00      00
MMM     MMM     CC      HH      HH  EEEEEEEEEEE  CC      CC  KK      KK  77      77  55      55  00      00
MM      MM      CC      HH      HH  EEEEEEEEEEE  CC      CC  KK      KK  77      77  55      55  00      00
MM      MM      CC      HH      HH  EEEEEEEEEEE  CC      CC  KK      KK  77      77  55      55  00      00
MM      MM      CC      HH      HH  EEEEEEEEEEE  CC      CC  KK      KK  77      77  55      55  00      00
MM      MM      CC      HH      HH  EEEEEEEEEEE  CC      CC  KK      KK  77      77  55      55  00      00
MM      MM      CC      HH      HH  EEEEEEEEEEE  CC      CC  KK      KK  77      77  55      55  00      00
MM      MM      CC      HH      HH  EEEEEEEEEEE  CC      CC  KK      KK  77      77  55      55  00      00
MM      MM      CCCCCCCC  HH      HH  EEEEEEEEEEE  CCCCCCCC  KK      KK  77      77  55      55  000000
MM      MM      CCCCCCCC  HH      HH  EEEEEEEEEEE  CCCCCCCC  KK      KK  77      77  55      55  000000

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)	85	LOCAL SYMBOL DEFFINITIONS AND LOCAL DATA STORAGE
(3)	227	MACHINE CHECK ENTRY POINTS
(4)	317	CONTROL STORE PARITY ERRORS
(5)	342	ASYNCHRONOUS WRITE ERROR INTERRUPT
(6)	373	TB, BUS, CACHE PARITY
(7)	633	CACHE PARITY ERRORS
(8)	671	CORRECTED MEMORY DATA INTERRUPTS
(9)	808	ERROR LOGGING ROUTINES
(10)	980	REFLECT EXCETION TO USER
(12)	1035	TABLE OF RESUMABLE INSTRUCTIONS.

```
0000 1 .TITLE MCHECK750 - VAX 11/750 MACHINE CHECK HANDLER
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
0000 29 FACILITY:
0000 30
0000 31 EXECUTIVE, ERROR HANDLING
0000 32
0000 33 ABSTRACT:
0000 34
0000 35 MACHINE CHECK INTERRUPT AND ABORT HANDLER, MEMORY ECC ERROR LOGGER.
0000 36 LOGS ERRORS AND ATTEMPTS TO CONTINUE IF POSSIBLE.
0000 37
0000 38 ENVIRONMENT:
0000 39
0000 40 INTERRUPT STACK AT IPL 31 UNTIL ERROR IS IDENTIFIED. IPL SYNCH TO
0000 41 LOG ERRORS.
0000 42
0000 43 SIDE EFFECTS:
0000 44
0000 45 IF ERROR IS UNRECOVERABLE OR ERROR IS RESULT OF INSTRUCTION EXECUTION
0000 46 IN EXECUTIVE OR KERNAL MODE, THEN A FATAL BUG-CHECK IS TAKEN.
0000 47
0000 48 AUTHOR:
0000 49
0000 50 C. SAMUELSON, APRIL, 1979
0000 51
0000 52 MODIFIED BY:
0000 53
0000 54 V03-008 WMC0001 Wayne Cardoza 14-Jun-1984
0000 55 Preserve cache state from boot.
0000 56
0000 57 V03-007 RLRSBICONF Robert L. Rappaport 22-Mar-1984
```



```

0000 58 :      Test MMG$GL_SBICONF array elements for valid system
0000 59 :      virtual address (high bit set) before using.
0000 60 :
0000 61 :      V03-006 KPL0100      Peter Lieberwirth      10-Feb-1984
0000 62 :      Change to use CONFREGL.
0000 63 :
0000 64 :      V03-005 KDM0053      Kathleen D. Morse      11-Jul-1983
0000 65 :      Replace cpu-dependent IPR references with new
0000 66 :      $PR750DEF symbols.
0000 67 :
0000 68 :      V03-004 MSH0002      Maryann Hinden      23-Nov-1982
0000 69 :      Add entry MCHK$GL_LOG to LOGGER. Delete EXE$UBAERR_INT.
0000 70 :
0000 71 :      V03-003 KTA3018      Kerbey T. Altmann      01-Nov-1982
0000 72 :      Change name of data psect.
0000 73 :
0000 74 :      V03-002 MSH0001      Maryann Hinden      24-Sep-1982
0000 75 :      Change EXE$DW780_INT entry to EXE$UBAERR_INT.
0000 76 :
0000 77 :      V03-001 TCM0001      Trudy C. Matthews      5-Apr-1982
0000 78 :      Altered table of resumable instructions to make all the
0000 79 :      interlocked queue and the "reserved to DIGITAL" instructions
0000 80 :      non-resumable.
0000 81 :
0000 82 :--

```

```
0000 85 .SBTTL LOCAL SYMBOL DEFFINITIONS AND LOCAL DATA STORAGE
0000 86 :++
0000 87 : MACHINE CHECK INTERRUPT STACK OFFSETS
0000 88 :--
0000 89 $DEFINI MCK
0000 90 $DEF MCK_LENGTH .BLKL 1 ;LENGTH OF MACHINE CHECK STACK FRAME
0004 91 $DEF MCK_CODE .BLKL 1 ;MACHINE CHECK ERROR CODE
0008 92 $DEF MCK_VA .BLKL 1 ;VIRTUAL ADDRESS OF LAST FETCH/STORE
000C 93 $DEF MCK_ERROR_PC .BLKL 1 ;PROGRAM COUNTER AT TIME OF ABORT
0010 94 $DEF MCK_MDR .BLKL 1 ;MEMEORY DATA OF LAST FETCH/STORE
0014 95 $DEF MCK_SMR .BLKL 1 ;SAVED MODE REGISTER
0018 96 $VIELD SMR,0,<- ;BITS DEFINED IN SAVED MODE REGISTER
0018 97 <MODE,2,M>- ;ACCESS MODE AT TIME OF ABORT
0018 98 <PV,,M>- ;PHYSICAL/VIRTUAL FLAG
0018 99 >
0018 100 $DEF MCK_RLT .BLKL 1 ;READ LOCK TIMEOUT - WRITE VECTOR OCCURED
001C 101 $VIELD RLT,0,<- ;BITS DEFINED IN READ LOCK TIMEOUT REGISTER
001C 102 <VMDR,,M>- ;VECTOR IN MDR IF 1
001C 103 >
001C 104 $DEF MCK_TBP .BLKL 1 ;TRANSLATION BUFFER PARITY ERROR
0020 105 $VIELD TBP,0,<- ;BITS IN TB PARITY REGISTER
0020 106 <GRPOD,,M>- ;GROUP 0 DATA ERROR
0020 107 <GRP1D,,M>- ;GROUP 1 DATA ERROR
0020 108 <GRPOT,,M>- ;GROUP 0 TAG ERROR
0020 109 <GRP1T,,M>- ;GROUP 1 TAG ERROR
0020 110 >
0020 111 $DEF MCK_CER .BLKL 1 ;CACHE ERROR REGISTER
0024 112 $VIELD CER,0,<- ;BITS IN CACHE ERROR REGISTER
0024 113 <HIT,,M>- ;HIT IF 1
0024 114 <LOST,,M>- ;LOST ERROR
0024 115 <DATA,,M>- ;DATA ERROR
0024 116 <TAG,,M>- ;TAG ERROR
0024 117 >
0024 118 $DEF MCK_BER .BLKL 1 ;BUS ERROR REGISTER
0028 119 $VIELD BER,0,<- ;BITS IN BUS ERROR REGISTER
0028 120 <CD,,M>- ;CORRECTED DATA
0028 121 <LOST,,M>- ;LOST ERROR
0028 122 <UCD,,M>- ;UNCORRECTED DATA
0028 123 <NEX,,M>- ;NON EXISTANT MEMORY
0028 124 >
0028 125 $DEF MCK_ESR .BLKL 1 ;ERROR SUMMARY REGISTER
002C 126 $VIELD ESR,0,<- ;BITS IN SUMMARY REG
002C 127 <XB,,M>- ;INSTRUCTION BUFFER IF 1
002C 128 <,M>-
002C 129 <TB,,M>- ;TRANSLATION BUFFER ERROR
002C 130 <CMI,,M>- ;CMI BUS ERROR
002C 131 >
002C 132 $DEF MCK_PC .BLKL 1 ;PC OF ABORTED OP-CODE
0030 133 $DEF MCK_PSL .BLKL 1 ;PSL AT TIME OF ABORT
0034 134 $DEFEND MCK
0000 135
0000 136 :
0000 137 : BITS DEFINED IN OTHER MEMORY CONTROL AND STATUS REGISTERS
0000 138 :
0000 139 $VIELD TBDR,0,<- ;TB DISABLE REGISTER
0000 140 <DGO,,M>- ;DISABLE GROUP 0
0000 141 <DG1,,M>- ;DISABLE GROUP 1
```



```
0000 142          <GRDP,,M>-          ;REPLACE GROUP (0=GROUP 0,1=GROUP 1)
0000 143          <REPL,,M>-        ;REPLACE
0000 144          >
0000 145          $VIELD CADR,0,<-    ;CACHE DISABLE REGISTER
0000 146          <DIS,,M>-        ;DISABLE CACHE
0000 147          >
0000 148          ;
0000 149          ; 11/750 MEMORY CONTROLLER REGISTER DEFFINITIONS
0000 150          ;
0000 151          $DEFINI MEM
0000 152 $DEF     MEM$SL CSR0 .BLKL 1 ;MEMORY CSR ZERO
0004 153          $VIELD CSR0,0,<-   ;BITS IN CSR0
0004 154          <SYN,7,M>-        ;ERROR SYNDROME
0004 155          <,2,M>-
0004 156          <PAGE,15,M>-      ;PAGE WHERE ERROR OCCURED
0004 157          <,5,>-
0004 158          <COREF,,M>-       ;CORRECTABLE ERROR FLAG
0004 159          <EIL0S,,M>-       ;ERROR INFORMATION LOST FLAG
0004 160          <UNCER,,M>-       ;UNCORRECTABLE ERROR FLAG
0004 161          >
0004 162 $DEF     MEM$SL CSR1 .BLKL 1 ;MEMORY CSR 1
0008 163          $VIELD CSR1,0,<-   ;BITS IN MEMORY CSR 1
0008 164          <CHCK,7,M>-       ;DIAGNOSTIC CHECK BITS
0008 165          <,2,>-
0008 166          <PMA,15,M>-      ;PAGE MODE ADDRESS
0008 167          <,1,>-
0008 168          <ECCD,,M>-       ;ECC DISABLE MODE
0008 169          <DIAG,,M>-       ;DIAGNOSTIC CHECK MODE
0008 170          <PMOD,,M>-       ;DIAGNOSTIC PAGE MODE
0008 171          <IERP,,M>-       ;INHIBIT CORRECTABLE ECC ERROR REPORTS
0008 172          <,3,>-
0008 173          >
0008 174 $DEF     MEM$SL CSR2 .BLKL 1 ;MEMORY PRESENT MAP
000C 175          $DEFEND MEM
0000 176          ;
0000 177          ; INCLUDED SYSTEM SYMBOL DEFFINITIONS
0000 178          ;
0000 179          $EMBDEF <MC>       ;DEFINE EMB OFFSETS AND VALUES
0000 180          $IPLDEF           ;DEFINE PROCESSOR INTERRUPT LEVELS
0000 181          $MCHKDEF         ;DEFINE RECOVERY BLOCK MASK BITS
0000 182          $SMPMDEF        ;MULTIPORT MEMORY
0000 183          $PCBDEF         ;DEFINE PROCESS CONTROL BLOCK
0000 184          $PFNDEF         ;DEFINE PFN DATABASE
0000 185          $PRDEF          ;DEFINE PROCESSOR REGISTERS
0000 186          $PR750DEF       ;DEFINE 750-SPECIFIC PROCESSOR REGISTERS
0000 187          $PSLDEF         ;DEFINE PSL
0000 188          $PTEDEF         ;DEFINE PTE SYMBOLS
0000 189          $RPBDEF         ;DEFINE RPB
0000 190          ;
0000 191          ;
0000 192          ;
0000 193          ; LOCAL DATA STORAGE
0000 194          ;
00000000 195          .PSECT MCHK$DATA,QUAD,WRT
0000 196          ;
0000000A 0000 197 TB_THRESHOLD = 10 ;ALLOWABLE TIME BETWEEN TB ERRORS
0000000A 0000 198 CH_THRESHOLD = 10 ;ALLOWABLE TIME BETWEEN CACHE ERRORS
```

```

00000003 0000 199 CRDINTMAX = 3 ;NO. OF CORRECTED MEMORY ERRORS TO LOG
00000006 0000 200 CRDWATCHMAX = 6 ;NO. OF ERRORS TO LOG IN REENABLE TIME
0000003C 0000 201 SOMETIME = 60 ;SCAN FOR NON INTERRUPT ERRORS EVER 60 SECON
00000384 0000 202 REENABTIME = 60*15 ;REENABLE INTERRUPT ERROR LOGGING TIME
0000 203
0000 204 ; THE FOLLOWING SYMBOL IS DEFINED FOR A TRANSFER VECTOR IN SYSLOAVEC.
0000 205 ; IT IS NOT JUMPED TO !!! IT IS USED TO DEFINE A GLOBAL
0000 206 ; SYMBOL IN THE SYSTEM MAP (SYS.MAP) TO LOCATE THESE COUNTERS.
0000 207
0000 208 EXESMCHK_ERRCNT::
00000000 0000 209 EXESGL_TB1AOLD:: .LONG 0 ;TIME OF LAST TB ERROR, GROUP 0
00000000 0004 210 EXESGL_TB2AOLD:: .LONG 0 ;TIME OF NEXT TO LAST TB ERROR, GROUP 0
00000000 0008 211 EXESGL_TB1BOLD:: .LONG 0 ;TIME OF LAST TB ERROR, GROUP 1
00000000 000C 212 EXESGL_TB2BOLD:: .LONG 0 ;TIME OF NEXT TO LAST TB ERROR, GROUP 1
00000020 0010 213 ECC$AB_MEMERR:: .BLKB 16 ;MEMORY ERROR COUNTERS FOR 16 SLOTS
0000 0020 214
00000000 0020 215 EXESGL_CH1OLD:: .LONG 0 ;TIME OF LAST CACHE ERROR
00000000 0024 216 EXESGL_CH2OLD:: .LONG 0 ;TIME OF NEXT TO LAST CACHE ERROR
00000000 0028 217 EXESGL_CHSTATE:: .LONG 0 ;CURRENT STATE OF CACHE
0000 002C 218 ; 1 = DISABLED, 0 = ENABLED
0000 002C 219 ECC$GW_WATCH:: .WORD 0 ;SCAN MEMEORY CONTROLLER TIMER
0000 002E 220 ECC$GW_REENAB:: .WORD 0 ;REENABLE TIMER
00000000 0030 221 MMG$GL_CRDCNT:: .LONG 0 ;COUNT OF CORRECTED MEMORY ERRORS
0000 0034 222 EXESGL_BADTIMOUT::
00000000 0034 223 .LONG 0 ;TIME SINCE LAST BAD MCHK CODE
0000 0038 224 EXESGL_VECTIMOUT::
00000000 0038 225 .LONG 0 ;TIME SINCE LAST UNDEFINED VECTOR INT

```



```
003C 227 .SBTTL MACHINE CHECK ENTRY POINTS
0000 228 .PSECT WIONONPAGED,QUAD,RD,WRT
0000 229 :++
0000 230 : ALL MACHINE CHECK ABORTS ARE VECTORED HERE. 11/750
0000 231 : MACHINE CHECK FOR ALL CONDITIONS RESULTS IN THE SAME STACK FRAME.  THUS
0000 232 : WE CAN DISPATCH FROM HERE ACCORDING TO THE ERROR CODE.
0000 233 : ENTRY IPL=31=X1F FOR ABORTS
0000 234 :--
0000 235
0000 236 .ALIGN LONG
0000 237
0000 238 EXESMCHK:: ;MACHINE CHECK EXCEPTION
0000 239
0000 240 INVALID ;INVALIDATE TRANSLATION BUFFER
0003 241 MFPR #PR750$_CADR,W^EXESGL_CHSTATE ;SAVE CACHE STATE
0008 242 MTPR #CADRSM_DIS,#PR750$_CADR ;DISABLE CACHE
000B 243 PUSHL #MCHKSM_LOG ;MASK FOR PRCTEST
000D 244 PUSHL MCK_PC+4(SP) ;PC,PSL POINTER FOR PRCTEST
0010 245 PUSHR #^MZR0,R1,R2,R3,R4,R5,AP> ;GET SOME WORKING REGISTERS
0014 246 ADDL3 #<9*4>,SP,AP ;POINT AP TO MACHINE CHECK LOG FRAME
0018 247 ;EXTRACT ERROR CODE FIELD FROM STACK
0018 248 CASE MCK_CODE(AP),<- ;DISPATCH ON ERROR TYPE CODE
0018 249 BAD_TYPE,- ;NO CODE 0
0018 250 CS_PARITY,- ;CONTROL STORE PARITY ERROR
0018 251 TB_BUS,- ;TB ERROR, BUS ERROR (UNCORRECTED READ)
0018 252 BAD_TYPE,- ;NO CODE 3 (USED TO BE CACHE PARITY)
0018 253 BAD_TYPE,- ;NO CODE 4 (USED TO BE CRD)
0018 254 BAD_TYPE,- ;NO CODE 5 (USED TO BE AWE)
0018 255 FUBAR,- ;NOT SUPPOSED TO BE HERE - CS LOCATION
0018 256 IRD_SLOT>, TYPE=B ;UNUSED IRD ROM SLOT
002D 257 BAD_TYPE: ;UNDEFINED EXCEPTION ERROR CODE
002D 258 MTPR W^EXESGL_CHSTATE,#PR750$_CADR ;RE-ENABLE THE CACHE
0032 259 BISL #MCHKSM_MCK,-4(AP) ;MASK FOR PRCTEST
0036 260 MOVZWL #EMBSK_MC,R3 ;LOG TYPE FOR ERROR REPORTING
0039 261 PUSHL W^EXESGL_BADTIMOUT ;TIME OF LAST BAD CODE MCHK
003D 262 MFPR #PR750$_TODR,W^EXESGL_BADTIMOUT ;SET TIME OF THIS ONE
0042 263 CMPL (SP)+,W^EXESGL_BADTIMOUT ;COMMING TOO FAST?
0047 264 BEQL 100$ ;YES, SOMETHING IS WRONG
0049 265 BRW TRYRESUME ;LOG IT, CONTINUE AND HOPE FOR THE BEST
004C 266 100$:
004C 267 POPR #^M<R0,R1,R2,R3,R4,R5,AP> ;RESTORE REGISTERS
0050 268 JSB G^EXESMCHK_BUGCHK ;RECOVERY BLOCK IN EFFECT?
0056 269 ;RECOVERY FROM THIS TYPE OF MACHINE
0056 270 ;CHECK IS VERY DUBIOUS - SOMETHING
0056 271 ;MUST BE BROKEN TO KEEP GETTING THIS
0056 272
0056 273 BUG_CHECK BADMCKCOD,FATAL ;BAD MACHINE CHECK CODE
005A 274
005A 275 :
005A 276 : THE FOLLOWING VECTORS ARE VESTIGAL FROM THE 11/780 SBI
005A 277 : THEY ARE NOT DEFINED IN COMET.
005A 278 : IF WE VECTOR HERE, EITHER THE HARDWARE GLITCHED OR SOMETHING
005A 279 : IS VERY BROKEN. THUS, IGNORE INTERRUPTS IF THEY DON'T COME
005A 280 : VERY FAST. ELSE, CRASH - AND CALL YOUR LOCAL REPAIR MAN.
005A 281 :
005A 282
005A 283 .ALIGN LONG ;VECTORED TO
```

```
005C 284
005C 285 EXE$LOGSBF:: ;SBI FAULT
005C 286 EXE$LOGSBA:: ;SBI ALERT
005C 287 EXE$INT58::
005C 288 EXE$INT5C::
005C 289 EXE$RH780_INT:: ;DEFINED FOR SYSLOAVEC, NOT USED IN 750
005C 290
005C 291 PUSHL #MCHK$M_MCK ;MASK FOR PRTCTEST
005E 292 PUSHAL 4(SP) ;PC,PSL POINTER FOR PRTCTEST
0061 293 PUSHR #^M<R0,R1,R2,R3,R4,R5> ;SAVE SOME REGISTERS
0063 294 PUSHL W^EXE$GL_VECTIMOUT ;TIME SINCE LAST ERROR
0067 295 MFPR #PR750$ TODR,W^EXE$GL_VECTIMOUT ;SAVE TIME OF THIS ERROR
006C 296 CMLP (SP)+,W^EXE$GL_VECTIMOUT ;ERRORS COMMING TOO FAST?
0071 297 BNEQ 200$ ;NO, TRY AND CONTINUE
0073 298
0073 299 POPR #^M<R0,R1,R2,R3,R4,R5> ;RESTORE REGISTERS
0075 300 JSB G^EXE$MCHK_BUGCHK ;RECOVERY BLOCK IN EFFECT?
007B 301 BUG_CHECK BADMCKCOD,FATAL ;NO, CRASH
007F 302
007F 303 200$:
007F 304 MOVQ <6*4>(SP),R1 ;SET UP TO LOG THE ERROR
0083 305 MOVZWL #EMB$K_SI,R3 ;UNEXPECTED INTERRUPT
0086 306 MOVL #<2*4>,R4 ;LOG PC,PSL
0089 307 MOVAL <8*4>(SP),R5 ;POINT TO LOG ENTRY
008D 308 BSBW LOGGER ;LOG THE ERROR
0090 309 JSB G^EXE$MCHK_TEST ;RECOVERY BLOCK IN EFFECT?
0096 310 BLBS R0,201$ ;YES, DO NOT LOG THIS ERROR
0099 311 INCL G^EXE$GL_MCHKERRS ;BUMP THE GLOBAL MACHINE CHECK COUNTER
009F 312 201$: POPR #^M<R0,R1,R2,R3,R4,R5> ;RESTORE REGISTERS
00A1 313 ADDL #<2*4>,SP ;REMOVER PRTCTEST STUFF
00A4 314 REI ;LETS HOPE THINGS ARE OK
```



```

00A5 317 .SBTTL CONTROL STORE PARITY ERRORS
00A5 318 :
00A5 319 : CONTROL STORE PARITY ERROR ABORT
00A5 320 : NOT SUPPOSED TO BE HERE - UNUSED C/S LOCATION
00A5 321 : UNUSED IRD ROM SLOT ABORT
00A5 322 :
00A5 323 : LOG THE ERROR
00A5 324 : IF ERROR WAS IN KERNAL OR EXECUTIVE MODE, DECLARE A FATAL BUG-CHECK
00A5 325 : IF ERROR WAS IN USER OR SUPERVISOR MODE, PASS EXCEPTION TO PROCESS
00A5 326 :
00A5 327 :
00A5 328 CS_PARITY: ;CONTROL STORE PARITY ERROR ABORT
00A5 329 FUBAR: ;NOT SUPPOSED TO BE HERE ABORT
00A5 330 IRD_SLOT: ;UNUSED IRD ROM SLOB ABORT
00A5 331 :
25 0028'CF DA 00A5 332 MTPR W^EXES$GL CHSTATE,#PR750$ _CADR :CACHE OK, ENABLE IT
FC AC 02 C8 00AA 333 BISL #MCHK$M MCK,-4(AP) ;MASK FOR PRCTEST
2A AC 2C BC 9B 00AE 334 MOVZBW @MCK_PCT(AP),MCK_ESR+2(AP) ;SAVE OP-CODE THAT FAULTED
53 02 3C 00B3 335 MOVZWL #EMBSK_MC,R3 ;SET ERROR TYPE IN R3
035D 30 00B6 336 BSBW LOGIT ;LOG ERROR
01D7 31 00B9 337 BRW TRYRESUME ;CONTINUE IF POSSIBLE
00BC 338 :
00BC 339 :

```

```
00BC 342      .SBTTL ASYNCHRONOUS WRITE ERROR INTERRUPT
00BC 343      :
00BC 344      : THIS ERROR IS CAUSED WHEN CMI WRITE OPERATION DID NOT COMPLETE SUCCESSFULLY.
00BC 345      : THERE COULD BE ANY NUMBER OF REASONS FOR THIS. THE CMI COULD BE BROKEN
00BC 346      : (UNLIKELY, SINCE WE WOULDN'T HAVE GOTTEN THIS FAR) OR AN ADAPTER ON THE CMI
00BC 347      : COULD BE BROKEN. THERE IS NO SPECIFIC CODE HERE TO LOOK FOR A BROKEN
00BC 348      : DEVICE. SUSPECTED DEVICES COULD BE MEMORY, DR750, MA750, OR ICCS.
00BC 349      : ANOTHER CAUSE IS WRITE TO NON-EXISTANT ADDRESS.
00BC 350      :
00BC 351      :
00BC 352      .ALIGN LONG
00BC 353
00BC 354 EXES$LOGAWE::
00BC 355 EXES$INT60::
5E 28 C2 00BC 356      SUBL #<10*4>,SP ; KLUDGE UP STACK TO LOOK LIKE MACHINE CHECK
6E 05 9A 00BF 357      MOVZBL #5,(SP) ; THIS IS THE TYPE - REST OF FRAME GARBAGE
7E 28 9A 00C2 358      MOVZBL #^X28,-(SP) ; LENGTH OF MACHINE CHECK ERROR FRAME
30 07 DD 00C5 359      PUSHL #MCHK$M_LOG!MCHK$M_MCK!MCHK$M_NEXM ; MASK FOR PRCTEST
103F 8F BB 00CA 360      PUSHAL MCK_PC+4(SP) ; AND PC POINTER
5C 5E 24 C1 00CE 361      PUSHR #^MZR0,R1,R2,R3,R4,R5,AP> ; SAVE REGISTERS
53 07 3C 00D2 362      ADDL3 #<9*4>,SP,AP ; POINT AP TO FAKE MACHINE CHECK LOG
54 08 9A 00D5 363      MOVZWL #EMB$K_AW,R3 ; ERROR TYPE
55 2C AC 9E 00D8 364      MOVZBL #<2*4>,R4 ; SIZE OF LOG ENTRY (PC,PSL)
0352 30 00DC 365      MOVAB MCK_PC(AP),R5 ; ADDRESS OF LOG ENTRY (ON STACK)
00000000'GF 16 00DF 366      BSBW LOGGER ; LOG THE ERROR
06 50 E8 00E5 367      JSB G^EXES$MCHK_TEST ; RECOVERY BLOCK IN EFFECT?
00000000'GF D6 00E8 368      BLBS R0,10$ ; YES, DO NOT LOG THIS ERROR
0456 31 00EE 369      INCL G^EXES$GL MCHKERRS ; BUMP THE GLOBAL MACHINE CHECK COUNTER
370 10$ : BRW REFLECTCHK ; CONTINUE IF USER OR SUPER MODE
```



```
00F1 373      .SBTTL TB, BUS, CACHE PARITY
00F1 374      :
00F1 375      : TRANSLATION BUFFER ERROR ABORT
00F1 376      : CMI BUS ERROR ABORT (UNCORRECTED READ)
00F1 377      : CACHE PARITY ERRORS
00F1 378      :
00F1 379      : LOG ERROR
00F1 380      :
00F1 381      : HANDLE TRANSLATION BUFFER ERROR AS FOLLOWS
00F1 382      :
00F1 383      :     INVALDATE TB
00F1 384      :     IF MANY RECENT TB ERRORS, DISABLE HALF OF TB
00F1 385      :     IF THIS RESULTS IN BOTH HALVES DISABLED, BUG CHECK
00F1 386      :     ELSE IF THE INSTRUCTION IS RESUMABLE, TRY TO RESTART IT
00F1 387      :     ELSE IF THE ABORT WAS IN EXEC OR KERNAL MODE, BUG CHECK
00F1 388      :     ELSE REFLECT ERROR TO USER OR SUPERVISOR AS EXCEPTION
00F1 389      :
00F1 390      :
00F1 391      : HANDLE BUS ERRORS AS FOLLOWS
00F1 392      :
00F1 393      :     IF THE INSTRUCTION WAS RESUMABLE, TRY AND RESTART
00F1 394      :     ELSE IF THE ABORT WAS IN EXEC OR KERNAL MODE, BUG CHECK
00F1 395      :     ELSE REFLECT ERROR TO USER OR SUPERVISOR MODE AS EXCEPTION
00F1 396      :
00F1 397      : HANDLE CACHE ERRORS AS FOLLOWS
00F1 398      :
00F1 399      :     IF MANY RECENT CACHE ERRORS, DISABLE CACHE
00F1 400      :     IF ERROR WAS IN USER OR SUPERVISOR MODE, REFLECT ERROR TO PROCESS
00F1 401      :     ELSE ISSUE A FATAL BUG-CHECK
00F1 402      :
00F1 403      : .ENABLE LSB
00F1 404      :
00F1 405      : TB_BUS:
00F1 406      :
00F1 407      :     MTPR    MCK_ESR(AP),#PR750$ MCESR      ;CLEAR PROCESSOR ERROR BITS
00F1 408      :     BBC     #ESR$V TB,MCK_ESR(AP),BUS CACHE ;BRANCH IF NOT TB ERROR
00F1 409      :     MTPR    W^EXES$GL CHSTATE,#PR750$ CADR ;CACHE PROBABLY OK, ENABLE IT
00F1 410      :     BISL    #MCHK$M_MCK,-4(AP)      ;MASK FOR PRCTEST
00F1 411      :
00F1 412      : TRANSLATION BUFFER ERRORS HANDLED HERE
00F1 413      :
00F1 414      :     MOVZWL #EMBS$K_MC,R3      ;PLACE ERROR TYPE IN R3 FOR LOGGER
00F1 415      :     MFPR    #PR750$ TODR,R0      ;CURRENT TIME IN 10MS TICKS
00F1 416      :     MFPR    #PR750$ TBDR,R1      ;GET TB DISABLE REGISTER
00F1 417      :
00F1 418      : DISCOVER WHICH HALF OF TB THE ERROR IS IN
00F1 419      :
00F1 420      :     BITL    MCK_TBP(AP),#<TBP$M_GRP0D!TBP$M_GRP0T>
00F1 421      :     BEQLU   20$      ;HANDLE EACH HALF SEPARATELY
00F1 422      :     ;ERROR IN SECOND HALF
00F1 423      :
00F1 424      : ERROR IS IN FIRST HALF OF TB
00F1 425      :
00F1 426      :     SUBL3   W^EXES$GL TB1AOLD,R0,R2 ;HOW LONG SINCE LAST ERROR?
00F1 427      :     CMPL    R2,#TB_THRESHOLD ;ERRORS COMMING TOO FAST?
00F1 428      :     BGTRU   10$      ;NO, CONTINUE
00F1 429      :
```

26	28	AC	DA
69	28	AC	E1
25	0028	CF	DA
FC	AC	02	C8
53	02	3C	0103
50	18	DB	0106
51	24	DB	0109
05	1C	AC	D3
	25	13	0110
52	50	0000	C3
	0A	52	D1
	0B	1A	011B
			011D


```
011D 430 ; DISABLE FIRST HALF OF TB
011D 431
3B 51 01 E0 011D 432 BBS #TBDR$V DG1,R1,TB BAD ;BOTH HALVES BAD, FATAL ERROR
24 0D DA 0121 433 MTPR #<TBDR$M_DG0!TBDR$M_GRP!TBDR$M_REPL>,#PR750$ TBDR ;DISABLE
1E AC 01 9B 0124 434 ;AND FORCE REPLACEMENT IN OTHER GROUP
0124 435 MOVZBW #TBDR$M_DG0,MCK_TBP+2(AP) ;LOG THAT WE DID IT
0128 436
0128 437 ; REMEMBER HISTORY OF GROUP 0 TB ERRORS
0128 438
0004'CF 0000'CF D0 0128 439 10$: MOVL W^EXE$GL_TB1AOLD,W^EXE$GL_TB2AOLD ;TIME OF LAST TO NEXT TO LAST
0000'CF 50 D0 012F 440 MOVL RO,W^EXE$GL_TB1AOLD ;TIME OF THIS TO TIME OF LAST
015C 31 0134 441 BRW TRYRESUME ;TRY TO RECOVER
0137 442
0137 443 ; ERROR IS IN SECOND HALF OF TB
0137 444
52 50 0008'CF C3 0137 445 20$: SUBL3 W^EXE$GL_TB1BOLD,RO,R2 ;TIME SINCE LAST ERROR IN 2nd HALF
OA 52 D1 013D 446 CMPL R2,#TB_THRESHOLD ;ERRORS COMMING TOO FAST?
OB 1A 0140 447 BGTRU 25$ ;NO, CONTINUE
0142 448
0142 449 ; DISABLE 2ND HALF OF TB
0142 450
16 51 00 E0 0142 451 BBS #TBDR$V DG0,R1,TB BAD ;BRANCH IF OTHER HALF BAD ALSO
24 0A DA 0146 452 MTPR #<TBDR$M_DG1!TBDR$M_REPL>,#PR750$ TBDR ;DISABLE 2ND HALF
1E AC 02 9B 0149 453 MOVZBW #TBDR$M_DG1,MCK_TBP+2(AP) ;LOG THAT WE DID IT
014D 454
014D 455 ; REMEMBER HISTORY OF GROUP 1 TB ERRORS
014D 456
000C'CF 0008'CF D0 014D 457 25$: MOVL W^EXE$GL_TB1BOLD,W^EXE$GL_TB2BOLD ;TIME OF LAST TO NEXT TO LAST
0008'CF 50 D0 0154 458 MOVL RO,W^EXE$GL_TB1BOLD ;TIME OF THIS TO TIME OF LAST
0137 31 0159 459 BRW TRYRESUME ;TRY TO RECOVER
015C 460
015C 461 ; BOTH HALVES OF TB BAD, LOG ERROR AND CRASH
015C 462
02B7 30 015C 463 TB_BAD:
015C 464 BSBW LOGIT ;LOG THE ERROR
015F 465 BUG_CHECK MACHINECHK,FATAL ;FATAL ERROR - BOTH HALVES OF TB BAD
0163 466
0163 467 ; CMI BUS ERRORS HANDLED HERE
0163 468
0163 469 ; EXAMINE BUS ERROR REGISTER TO FIND OUT WHAT HAPPENED
0163 470
0163 471 BUS_CACHE:
05 20 AC 02 E0 0163 472 BBS #CER$V_DATA,MCK_CER(AP),26$ ;BRANCH IF CACHE DATA ERROR
03 20 AC 03 E1 0168 473 BBC #CER$V_TAG,MCK_CER(AP),27$ ;BRANCH IF NOT CACHE TAG ERROR
014C 31 016D 474 26$: BRW CH_PARITY ; HANDLE CACHE PARITY ERRORS
0170 475
1A 24 AC 02 E0 0170 476 27$: BBS #BER$V_UCD,MCK_BER(AP),UCD ;BRANCH IF UNCORRECTED DATA
03 24 AC 03 E0 0175 477 BBS #BER$V_NEX,MCK_BER(AP),BUS ;BRANCH IF NON-EXISTANT REFFERENCE
FEB0 31 017A 478 BRW BAD_TYPE ; CANNOT IDENTIFY ERROR TYPE
017D 479
017D 480 ; THE ERROR WAS CAUSED BY NON-EXISTANT MEMORY ON A READ
017D 481
017D 482 BUS:
25 0028'CF DA 017D 483 MTPR W^EXE$GL_CHSTATE,#PR750$ CADR ;CACHE PROBABLY OK, ENABLE IT
FC AC 06 C8 0182 484 BISL #MCHK$M_NEXM!MCHK$M_MCK,-4(AP) ;MASK FOR PRCTEST
53 02 3C 0186 485 MOVZWL #EMB$K_MC,R3 ;HANDLE AS A MACHINE CHECK
028A 30 0189 486 BSBW LOGIT ;LOG THE ERROR
```



```
03B8 31 018C 487 BRW REFLECTCHK ;CONTINUE IF USER OR SUPER MODE
018F 488
018F 489
018F 490 :++
018F 491 : UNCORRECTED MEMORY READ ERROR (READ DATA SUBSTITUTE)
018F 492 : ATTEMPT TO REPLACE PAGE WITH A GOOD ONE
018F 493 : PAGE CAN BE REPLACED IF
018F 494 : FAULT OCCURED AT PAGEABLE PRIORITY
018F 495 : PAGE WAS VALID FOR THE ACCESSING PROCESS
018F 496 : PAGE WAS NOT MODIFIED OR IS READ-ONLY
018F 497 : INSTRUCTION CAUSING PROBLEM IS RESUMABLE
018F 498 : NO I/O WAS IN PROGRESS FOR THE PAGE
018F 499 : PAGE IS SYSTEM OR PROCESS PRIVATE
018F 500 : BAD PAGE IS PLACED ON BAD PAGE LIST
018F 501 :--
018F 502
018F 503 UCD:
018F 504 MTPR W^EXES$GL CHSTATE,#PR750$ CADR ;CACHE PROBABLY OK, ENABLE IT
50 30 AC 25 0028'CF DA 0194 505 BLSL #MCHK$M MCK,-4(AP) ;MASK FOR PRTCTEST *****
FC AC 02 C8 0198 506 EXTZV #PSL$V IPL,#PSL$S_IPL,MCK PSL(AP),R0 ;GET IPL
05 10 EF 019E 507 CMPL R0,#IPC$ ASTDEL ;ARE WE AT NON-PAGEABLE PRIORITY?
02 50 D1 01A1 508 BGTR NO_RESUME ;NO, NOTHING WE CAN DO
5C 14 01A3 509
01A3 510 MOVL MCK_VA(AP),R2 ;GET VIRTUAL ADDRESS OF ERROR
54 52 08 AC DO 01A7 511 MOVL G^SCH$GL CURPCB,R4 ;CURRENT USER'S PCB ADDRESS
00000000'GF DO 01AE 512 MOVL PCB$S_PHD(R4),R5 ;PROCESS HEADER ADDRESS
55 6C A4 DO 01B2 513 JSB G^MMG$SVAPTECHK ;TURN VA INTO VA OF PTE
00000000'GF 16 01B8 514 MOVL (R3),R0 ;GET THE PTE WHICH MAPS THE BAD PAGE
50 63 DO 01BB 515 BLSS 28$ ;BRANCH IF PAGE VALID FOR THIS PROCESS
OE 19 01BD 516
01BD 517
01BD 518 ; PROCESS EXECUTED A READ TO A NON VALID PAGE, CRASH
01BD 519
01BD 520 RDSNONRES:
01BD 521
01BD 522 POPR #^M<R0,R1,R2,R3,R4,R5,AP> ;RESTORE REGISTERS
103F 8F BA 01C1 523 JSB G^EXES$MCHK_BUGCHK ;RECOVERY BLOCK IN EFFECT?
00000000'GF 16 01C7 524
01C7 525 BUG_CHECK RDSNONRES,FATAL ;FATAL BUG-CHECK
01CB 526
01CB 527
01CB 528 ASSUME PTE$V_PFN EQ 0
01CB 529
01CB 530 28$:
30 50 15 E0 01CB 531 BBS #PTE$V_WINDOW,R0,NO_RESUME
01CF 532 ;BRANCH IF PAGE IS PFN-MAPPED
50 50 15 00 EF 01CF 533 EXTZV #PTE$V_PFN,#PTE$S_PFN,R0,R0
01D4 534 ;ISOLATE PAGE FRAME NUMBER IN PTE
00000000'GF 50 D1 01D4 535 CMPL R0,G^MMG$GL_MAXPFN ;PFN DATA BASE FOR THIS PAGE?
22 1A 01DB 536 BGTRU NO_RESUME ;NO, CANNOT CONTINUE
7E 00000000'GF 20 88 01DD 537 MOVL G^PFNS$B TYPE,-(SP) ;GET ADDRESS OF PFN TYPE ARRAY *****
9E40 51 D4 01E4 538 BISB #PFNS$M_BADPAG,@(SP)+[R0] ;MARK PAGE BAD *****
51 51 D4 01E8 539 CLRL R1 ;CLEAR MODIFY BIT FLAG
04 63 1A E5 01EA 540 BBCC #PTE$V_MODIFY,(R3),30$ ;TEST AND CLEAR MODIFY BIT IN PTE
51 80 8F 9A 01EE 541 MOVZBL #PFNS$M_MODIFY,R1 ;SET MODIFY BIT FLAG
7E 00000000'GF 51 D0 01F2 542 30$: MOVL G^PFNS$B_STAT@,-(SP) ;ADDRESS OF PFN STATE *****
9E40 51 88 01F9 543 BISB R1,@(SP)+[R0] ;SET MODIFY BIT FLAG IN PFN DATABASE **
```



```

      01FD 544
      01FD 545
      01FD 546
      OF 18 01FD 547
      01FF 548
      01FF 549
      01FF 550 ; FOR ANY OF THE VARIOUS REASONS, CANNONT TRY TO RESUME INSTRUCTION STREAM
      01FF 551
      01FF 552 NO_RESUME:
      53 08 9A 01FF 553 MOVZBL #EMBSK_HE,R3 ;ERROR TYPE FOR ERROR LOGGER
      0257 30 0202 554 BSBW EXE$LOGMEM ;LOG THE MEMORY ERROR
      53 02 3C 0205 555 MOVZWL #EMBSK_MC,R3 ;LOG A MACHINE CHECK ERROR
      020B 30 0208 556 BSBW LOGIT
      0339 31 020B 557 BRW REFLECTCHK ;CONTINUE ONLY IF USER OR SUPER MODE
      020E 558
      7E 2C BC 9A 020E 559 33$: MOVZBL @MCK_PC(AP),-(SP) ;GET OPCODE FOR RESUMEABILITY CHECK
      E7 0583'CF 8E E1 0212 560 BBC (SP)+[R0],#1 ;BRANCH IF CANNOT RESUME INSTRUCTION
      0218 561
      7E 00000000'GF D0 0218 562 MOVL G^PFNSAW_REFCNT,-(SP) ;GET ADDRESS OF PFN REFCNT ARRAY *****
      01 9E40 B1 021F 563 CMPW @ (SP)+[R0],#1 ;CHECK FOR I/O IN PROGRESS *****
      DA 1A 0223 564 BGTRU NO_RESUME ;IF SO, DON'T MESS WITH PAGE
      0225 565
      7E 00000000'GF D0 0225 566 MOVL G^PFNSAB_TYPE,-(SP) ;ADDRESS OF PFN TYPE ARRAY *****
      01 9E40 03 00 EC 022C 567 CMPV #PFNSV_PAGTYP,#PFNS$_PAGTYP,@(SP)+[R0],#PFNS$C_SYSTEM ;*****
      0232 568 ;CHECK TO SEE IF SYSTEM OR GLOBAL PAGE
      0232 569
      0232 570 ASSUME PFNS$C_SYSTEM EQ 1
      0232 571 ASSUME PFNS$C_PROCESS EQ 0
      0232 572
      CB 1A 0232 573 BGTRU NO_RESUME ;BRANCH IF GLOBAL PAGE
      0234 574
      52 D5 0234 575 TSTL R2 ;TEST THE VIRTUAL ADDRESS
      08 14 0236 576 BGTR 35$ ;BRANCH IF PAGE IS PROCESS PRIVATE
      54 00000000'GF 9E 0238 577 MOVAB G^MMG$AL_SYSPCB,R4 ;SYSTEM WORKING SET LIST
      55 6C A4 D0 023F 578 MOVL PCB$P_PHD(R4),R5
      00 63 1F E5 0243 579 35$: BBCC #PTES$VALID,(R3),40$ ;CLEAR VALID BIT FROM PTE
      0247 580 40$: INVALID R2 ;INVALIDATE TRANSLATION BUFFER OF VA
      7E 00000000'GF D0 024A 581 MOVL G^PFNSAW_REFCNT,-(SP) ;ADDRESS OF PFN REFCNT ARRAY *****
      9E40 B7 0251 582 DECW @ (SP)+[R0] ;DECREMENT REFERENCE COUNT FOR PAGE ****
      06 18 0254 583 BGEQ 50$
      00000000'GF 16 0256 584 JSB G^MMG$REFCNTNEG
      7E 00000000'GF D0 025C 585 50$: MOVL G^PFNSAx_WSLX,-(SP) ;ADDRESS OF WORK SET LIST *****
      0263 586 PFN_REFERENCE - ;WORKING SET LIST INDEX FOR PAGE ***
      0263 587 MOVZWL <@ (SP)+[R0],R1>,-
      0263 588 LONG_OPCODE=MOVL,-
      0263 589 IMAGE=SYSLOA750.EXE
      00000000'GF 16 0275 591 JSB G^MMG$DELWSLEX ;DELETE PAGE FROM WORKING SET
      00000000'GF 16 027B 592 JSB G^MMG$DELCONPFN ;DELETE PAGE FROM PAGE TABLE
      52 02 9A 0281 593 MOVZBL #PFNS$BADPAGLST,R2
      00000000'GF 16 0284 594 JSB G^MMG$INSPFNT ;PUT PAGE ON BAD PAGE LIST
      028A 595
      53 08 3C 028A 596 MOVZWL #EMBSK_HE,R3 ;PLACE ERROR TYPE IN R3
      01CC 30 028D 597 BSBW EXE$LOGMEM ;LOG MEMORY ERROR
      53 02 3C 0290 598 MOVZWL #EMBSK_MC,R3 ;LOG MACHINE CHECK
      0293 599
      0293 600 ;
```



```
0293 601 : TRYRESUME - TRY TO RESTART AN INSTRUCTION AFTER A FAULT
0293 602 :
0293 603 : IF PROCESSOR MODE WAS NOT CHANGED DURING EXECUTION OF INSTRUCTION
0293 604 : IF INSTRUCTION IS LISTED AS RETRIABLE IN TABLE
0293 605 : IF INSTRUCTION IS SINGLE BYTE INSTRUCTION
0293 606 : THEN - RESTART
0293 607 :
0293 608 : INPUTS - R3 CONTAINS ERROR TYPE FOR ERROR LOGGING
0293 609 :
0293 610
0293 611 TRYRESUME: ; TRY TO RESUME DEPENDING ON OP-CODE
0293 612
0293 613 BSBW LOGIT ;MACHINE CHECK ERROR LOGGER
55 14 AC 02 00 EF 0296 614 EXTZV #SMR$V_MODE,#SMR$S_MODE,MCK_SMR(AP),R5
029C 615 ;GET MODE AT TIME OF FAULT
55 30 AC 02 18 ED 029C 616 CMPZV #PSL$V_CURMOD,#PSL$S_CURMOD,MCK_PSL(AP),R5
02A2 617 ;COMPARE WITH MODE WHEN FAULT WAS DISCOVERED
02A2 618 BNEQU 70$ ;IF NOT EQUAL, DON'T TRY TO RESUME
02A4 619 MOVZBL @MCK_PC(AP),R5 ;FETCH OP-CODE
OB 0583'CF 55 E1 02A8 620 BBC R5,W*RESUMABLE,70$ ;LOOK UP OP-CODE IN TABLE
02AE 621
02AE 622 POPR #^M<R0,R1,R2,R3,R4,R5,AP> ;IF BIT IS SET, RETRY INSTRUCTION
5E 08 C0 02B2 623 ADDL #<2*4>,SP ;REMOVE PRCTEST STUFF
5E 8E C0 02B5 624 ADDL (SP)+,SP ;CLEAR LOG OFF STACK
02 02B8 625 REI
02B9 626
02B9 627 70$:
028B 31 02B9 628 BRW REFLECTCHK ;ELSE CONTINUE DEPENDING ON ACCESS MODE
02BC 629
02BC 630 .DSABL LSB
```

```
02BC 633 .SBTTL CACHE PARITY ERRORS
02BC 634 :
02BC 635 : CACHE PARITY ERROR HANDLER
02BC 636 :
02BC 637 : LOG ERROR
02BC 638 :
02BC 639 : IF MANY RECENT CACHE ERRORS, DISABLE CACHE
02BC 640 : IF ERROR WAS IN USER OR SUPERVISOR MODE, REFLECT ERROR TO PROCESS
02BC 641 : ELSE ISSUE A FATAL BUG-CHECK
02BC 642 :
02BC 643 :
02BC 644 CH_PARITY: ;CACHE PARITY ERROR HANDLER
02BC 645 :
51 50 53 02 3C 02BC 646 MOVZWL #EMBSK MC,R3 ;PLACE ERROR TYPE IN R3 FOR LOGING
FC AC 02 C8 02BF 647 BISL #MCHKSM_MCK,-4(AP) ;MASK FOR PRTCTEST
50 50 1B DB 02C3 648 MFPR #PR750$_TODR,R0 ;CURRENT TIME IN 10MS TICKS
0020'CF C3 02C6 649 SUBL3 W^EXESGL_CH1OLD,R0,R1 ;TIME SINCE LAST ERROR
OA 51 D1 02CC 650 CMPL R1,#CH_THRESHOLD ;ERRORS COMING TOO FAST?
09 1A 02CF 651 BGTRU 30$ ;NO, CONTINUE
02D1 652 :
02D1 653 :
02D1 654 : TOO MANY CACHE ERRORS IN TIME INTERVAL, DISABLE CACHE PERMANENTLY
02D1 655 :
22 AC 01 9B 02D1 656 MOVZBW #CADDRSM_DIS,MCK CER+2(AP) ;LOG THAT WE DISABLED CACHE
0028'CF 01 9A 02D5 657 MOVZBL #CADDRSM_DIS,W^EXESGL_CHSTATE ;RESET CURRENT CACHE STATE
02DA 658 :
02DA 659 :
02DA 660 : LOG ERROR - CONTINUE IF IN USER OR SUPERVISOR MODE
02DA 661 :
02DA 662 :
02DA 663 30$:
0024'CF 25 0028'CF DA 02DA 664 MTPR W^EXESGL_CHSTATE,#PR750$_CAER ;RE-ENABLE CACHE IF OK
CF 0020'CF DO 02DF 665 MOVL W^EXESGL_CH1OLD,W^EXESGL_CH2OLD ;SAVE TIME OF LAST ERROR
0020'CF 50 DO 02E6 666 MOVL R0,W^EXESGL_CH1OLD ;SAVE TIME OF THIS ERROR
27 20 AC DA 02EB 667 MTPR MCK CER(AP),#PR750$_CAER ;CLEAR ERROR BITS
A2 11 02EF 668 BRB TRYRESUME ;TRY TO RESUME DEPENDING ON OP-CODE
```



```
02F1 671 .SBTTL CORRECTED MEMORY DATA INTERRUPTS
02F1 672 :
02F1 673 : CM_DATA - CORRECTED MEMORY DATA INTERRUPT HANDLER
02F1 674 :
02F1 675 : LOG ALL INTERRUPTS
02F1 676 : IF TOO MANY INTERRUPTS ARE RECEIVED, THE MEMORY ERROR LOGGER WILL TURN
02F1 677 : OFF THE CRD INTERRUPT BIT.
02F1 678 :
02F1 679 : CONTINUE ALWAYS
02F1 680 :
02F1 681 :
02F1 682 .ALIGN LONG ;VECTORED TO
02F4 683 :
02F4 684 EXESLOGCRD:: ;CORRECTED MEMORY DATA INTERRUPT
02F4 685 EXESINT54::
02F4 686 :
0030 3F BB 02F4 687 PUSHR #^M<R0,R1,R2,R3,R4,R5>
53 06 D6 02F6 688 INCL W^MMG$GL_CRDCNT ;COUNT CORRECTED MEMORY ERRORS
015C 3C 3C 02FA 689 MOVZWL #EMBSK_SE,R3 ;SOFT ERROR
3F BA 02FD 690 BSBW EXESLOGMEM ;LOG A MEMORY ERROR
02 0300 691 POPR #^M<R0,R1,R2,R3,R4,R5>
0302 692 REI ;AND RETURN
0303 693 :
0303 694 :
0303 695 : ECC$REENABLE - TIMER CALL FROM SYSTEM CLOCK ROUTINE
0303 696 :
0303 697 : THIS ROUTINE SCANS ALL MEMEORY CONTROLLERS FOR CORRECTED ECC ERRORS
0303 698 : WHICH HAVE EITHER BEEN CAUSED BY A READ ISSUED FROM AN I/O DEVICE OR
0303 699 : WHICH HAVE OCCURED WHEN CRD INTERRUPTS ARE TURNED OFF. IF A MEMORY
0303 700 : ERROR HAS OCCURED, ALL MEMORY CSR'S ARE LOGGED.
0303 701 :
0303 702 ECC$REENABLE:: ;RE-ENABLE CORRECTED MEMORY DATA INTERRUPTS
0303 703 :
002C 3F B7 0303 704 DECW W^ECC$GW_WATCH ;TIME TO SCAN MEMORY CONTROLLER?
7E 14 0307 705 BGTR 50$ ;NOT YET IF GTR
007F 8F BB 0309 706 PUSHR #^M<R0,R1,R2,R3,R4,R5,R6> ;SAVE WORKING REGISTERS
56 0F D0 030D 707 MOVL #15,R6 ;R0 INDEXES AND COUNTS SLOTS
53 00000000'GF D0 0310 708 MOVL G^EXES$GL_CONFREGL,R3 ;ARRAY OF NEXUS DEVICE TYPE CODES
55 00000000'GF D0 0317 709 MOVL G^MMG$GL_SBICONF,R5 ;ARRAY OF ADAPTER VA'S
031E 710 DSBINT ;DO SCAN AT IPL 31
002C 3F 3C B0 0324 711 MOVW #SOMETIME,W^ECC$GW_WATCH ;RESET SCAN TIMER
0329 712 10$:
52 6346 D0 0329 713 MOVL (R3)[R6],R2 ;GET ADAPTER TYPE
4E 13 032D 714 BEQL 40$ ;NO ADAPTER IN THIS SLOT
51 6546 D0 032F 715 MOVL (R5)[R6],R1 ;GET VA OF NEXT CMI SLOT
48 18 0333 716 BGEQ 40$ ;GEQ IMPLIES NO VALID SYSTEM VA.
52 E0 8F 93 0335 717 BITB #^B11100000,R2 ;LOCAL MEMORY?
27 13 0339 718 BEQL 20$ ;YES IF EQL
033B 719 :
40 8F 52 91 033B 720 CMPB R2,#^X40 ;MULTI-PORT MEMORY?
3C 1F 033F 721 BLSSU 40$ ;NO IF TYPE LSS 40
43 8F 52 91 0341 722 CMPB R2,#^X43
36 1A 0345 723 BGTRU 40$ ;NO IF TYPE GTR 43
0347 724 :
0347 725 $PRTCTINI W^15$,#<MCHK$M_LOG!MCHK$M_NEXM>
0354 726 :
54 10 A1 D0 0354 727 MOVL MPM$L_ERR(R1),R4 ;GET MULTI-PORT ERROR REGISTER
```



```

      21 50 E9 0358 728 $PRTCTEND 15$
      1D 54 1C E1 0359 729 BLBC R0,40$ ;NO MA750 ANY MORE
      07 11 035C 730
      54 61 D0 035C 731
      14 54 1D E1 0360 732 BBC #MPMSV_ERR_ELR,R4,40$ ;NO ERROR HERE
      06 91 0362 733 BRB 30$ ;ERROR, SEE IF WE SHOULD LOG IT
      54 61 D0 0362 734
      14 54 1D E1 0365 735 20$: MOVL MEMSL CSRO(R1),R4 ;LOCAL MEMORY CSR
      0030'CF D6 0369 736 30$: BBC #CSRO$V_COREF,R4,40$ ;CORRECTED ECC ERROR IN THIS MEMORY?
0010'CF46 06 91 0369 737 INCL W^MMG$GL CRDCNT ;COUNT TOTAL CRD ERRORS IN SYSTEM
      08 1B 036D 738 CMPB #CRDWATCHMAX,W^ECC$AB_MEMERR[R6] ;LOGGED ENOUGH ERRORS HERE?
      53 06 3C 0373 739 BLEQU 40$ ;YES
      00E1 30 0375 740 MOVZWL #EMBSK_SE,R3 ;ERROR TYPE IN R3
      03 11 0378 741 BSBW EXESLOGMEM ;LOG MEMORY ERRORS
      A9 56 F4 037B 742 BRB 45$ ;LOGGER LOGS ALL MEMORIES
      007F 8F BA 037D 743
      037D 744
      0380 745 40$: SOBGEQ R6,10$ ;LOOP THROUGH ALL POSSIBLE SLOT NOS.
      007F 8F BA 0380 746 45$: ENBINT ;RE-ENABLE INTERRUPTS
      0383 747 POPR #^M<R0,R1,R2,R3,R4,R5,R6> ;RESTORE REGS
      0387 748
      0387 749
      0387 750 50$: DECW W^ECC$GW_REENAB ;CHECK IF TIME TO REENABLE CRD INTS
      002E'CF B7 0387 751 BLEQ 52$ ;REENABLE TIME ELAPSED
      01 15 038B 752 RSB ;YES
      05 038D 753 ;NO, RETURN
002E'CF 0384 8F B0 038E 754 52$: MOVW #REENABTIME,W^ECC$GW_REENAB ;RESET REENABLE TIMER
      007F 8F BB 0395 755 PUSHR #^M<R0,R1,R2,R3,R4,R5,R6> ;SAVE WORKING REGISTERS
10 00 00000010'EF 00 2C 0399 756 MOVCS #0,ECC$AB_MEMERR,#0,#16,ECC$AB_MEMERR ;RESET ERROR COUNTERS TO ZERO
      00000010'EF 03A2
      53 00000000'GF D0 03A7 757 MOVL #15,R6 ;R0 INDEXES SLOT NOS.
      55 00000000'GF D0 03AA 758 MOVL G^EXES$GL_CONFREG,R3 ;ARRAY OF NEXUS DEVICE TYPE CODES
      00000000'GF D0 03B1 759 MOVL G^MMG$GL_SBICONF,R5 ;ARRAY OF ADAPTER VA'S
      52 6346 D0 03B8 760 55$: MOVL (R3)[R6],R2 ;ADAPTER TYPE CODE
      50 13 03BC 761 BEQL 65$ ;NO ADAPTER IN THIS SLOT
      51 6546 D0 03BE 762 MOVL (R5)[R6],R1 ;VA OF ADAPTER
      4A 18 03C2 763 BGEQ 65$ ;GEQ IMPLIES NO VALID SYSTEM VA.
      52 E0 8F 93 03C4 764 BITB #^B11100000,R2 ;LOCAL MEMORY?
      30 13 03C8 765 BEQL 60$ ;YES IF EQL
      40 8F 52 91 03CA 766 CMPB R2,#^X40 ;MULTIPOINT MEMORY?
      3E 1F 03CE 767 BLSSU 65$
      43 8F 52 91 03D0 768 CMPB R2,#^X43
      38 1A 03D4 769 BGTRU 65$
      03D6 770
      03D6 771 ; HANDLE MA750 ECC INTERRUPT ENABLE
      03D6 772
      03D6 773
      30 00000000'GF 00' E1 03D6 774 BBC S^#EXES$V_CRDENABL,G^EXES$GL_FLAGS,65$
      03DE 775 ;BRANCH IF CRD INTERRUPTS NOT WANTED
      03DE 776
      03DE 777 $PRTCTINI W^58$,#<MCHK$M_LOG!MCHK$M_NEXM>
      03EB 778
      52 10 A1 D0 03EB 779 MOVL MPMSL_ERR(R1),R2 ;GET MA750 ERROR REGISTER
      00 52 1E E2 03EF 780 BBSS #MPMS$V_ERR_ICRD,R2,57$ ;RE-ENABLE MA750 CRD INTERRUPTS
      10 A1 52 D0 03F3 781 57$: MOVL R2,MPMSL_ERR(R1)
      03F7 782
      03F7 783 $PRTCTEND 58$
```



```
14 11 03F8 784
      03F8 785 BRB 65$
      03FA 786
      03FA 787 : HANDLE MAIN MEMORY ECC INTERRUPT ENABLE
      03FA 788
      03FA 789 : ***** NOTE *****
      03FA 790 : IF ECC INTERRUPT BIT IS NOT SET (A SYSGEN PARAMATER), THEN NO
      03FA 791 : SOFT ECC ERROR REPORTING WILL TAKE PLACE, EVEN WHEN ECC$REENABLE
      03FA 792 : SCANS THE MAIN MEMORY CONTROLLER. THIS IS DIFFERENT THAN IN THE
      03FA 793 : 11/780 WHERE THE MEMORY CSR ERROR REPORTING BITS ARE SET WHETHER
      03FA 794 : OR NOT INTERRUPTS ARE ENABLED.
      03FA 795
OC 00000000'GF 00' E1 03FA 796 60$: BBC S^#EXE$V_CRDENABL,G^EXE$GL_FLAGS,65$
      0402 797 :BRANCH IF CRD INTERRUPTS NOT WANTED
      0402 798 MOVL MEM$[CSR1(R1),R2 :GET LOCAL MEMORY CSR
      0406 799 BBSS #CSR1$V_IERP,R2,62$ :SET INTERRUPT ENABLE
      040A 800 62$: MOVL R2,MEM$[CSR1(R1) :WRITE BACK TO MEMORY CSR
      040E 801
      040E 802 65$: SOBGEQ R6,55$ :LOOP THROUGH ALL POSSIBLE SLOTS
      007F 8F BA 0411 803 POPR #^M<R0,R1,R2,R3,R4,R5,R6>
      05 0415 804 70$: RSB :RETURN AT LAST
      0416 805
```

```
0416 808 .SBTTL ERROR LOGGING ROUTINES
0416 809 :
0416 810 : LOGIT - ERROR LOGGING FOR MACHINE CHECKS
0416 811 :
0416 812 : COMMON ERROR LOGGING CODE FOR NON MEMORY MACHINE CHECKS
0416 813 :
0416 814 : INPUT:
0416 815 :
0416 816 : R3 = ERROR LOG TYPE CODE
0416 817 : 0(SP) = RETURN ADDRESS
0416 818 : 4(SP) TO 28(SP) SAVED REGISTERS 0-5,AP
0416 819 : 32(SP) -40(SP) MASK, PC-PSL POINTER FOR PRTCTEST
0416 820 : (AP) = LENGTH OF HARDWARE ERROR LOG FRAME (28 HEX)
0416 821 : 4(AP) TO 44(AP) HARDWARE ERROR LOG INFORMATION
0416 822 : 48(AP) = INTERRUPT PC
0416 823 : 52(AP) = INTERRUPT PSL
0416 824 :
0416 825 :
0416 826 : LOGIT:
0416 827 : MOVQ -8(AP),R1 ;LOG MACHINE CHECK ERRORS
0416 828 : ADDL3 MCK_LENGTH(AP),#<2*4>,R4 ;MASK AND PC,PSL POINTER OF PRTCTEST
0416 829 : MOVAB MCK_CODE(AP),R5 ;# OF BYTES TO LOG + PC&PSL
0416 830 : JSB G^EXE$MCHK_TEST ;GET ADDRESS OF LOG
0416 831 : BLBS R0,20$ ;RECOVERY BLOCK IN EFFECT?
0416 832 : INCL G^EXE$GL_MCHKERRS ;YES, DO NOT LOG THIS ERROR
0416 833 : 20$: ;BUMP THE GLOBAL MACHINE CHECK COUNTER
0416 834 : ;FALL THROUGH TO ERROR LOG ROUTINE
0416 835 :
0416 836 : :
0416 837 : :
0416 838 : :
0416 839 : :
0416 840 : :
0416 841 : :
0416 842 : :
0416 843 : :
0416 844 : :
0416 845 : :
0416 846 : :
0416 847 : :
0416 848 : :
0416 849 : :
0416 850 : :
0416 851 : :
0416 852 : :
0416 853 : :
0416 854 : :
0416 855 : :
0416 856 : :
0416 857 : :
0416 858 : :
0416 859 : :
0416 860 : :
0416 861 : :
0416 862 : :
0416 863 : :
0416 864 : :

51 F8 AC 7D 0416 827 MOVQ -8(AP),R1 ;LOG MACHINE CHECK ERRORS
54 08 6C C1 041A 828 ADDL3 MCK_LENGTH(AP),#<2*4>,R4 ;MASK AND PC,PSL POINTER OF PRTCTEST
55 04 AC 9E 041E 829 MOVAB MCK_CODE(AP),R5 ;# OF BYTES TO LOG + PC&PSL
00000000'GF 16 0422 830 JSB G^EXE$MCHK_TEST ;GET ADDRESS OF LOG
06 50 E8 0428 831 BLBS R0,20$ ;RECOVERY BLOCK IN EFFECT?
00000000'GF D6 0428 832 INCL G^EXE$GL_MCHKERRS ;YES, DO NOT LOG THIS ERROR
0431 833 20$: ;BUMP THE GLOBAL MACHINE CHECK COUNTER
0431 834 ;FALL THROUGH TO ERROR LOG ROUTINE
0431 835 :
0431 836 :
0431 837 :
0431 838 :
0431 839 :
0431 840 :
0431 841 :
0431 842 :
0431 843 :
0431 844 :
0431 845 :
0431 846 :
0431 847 :
0431 848 :
0431 849 :
0431 850 :
0431 851 :
0431 852 :
0431 853 :
0431 854 :
0431 855 :
0431 856 :
0431 857 :
0431 858 :
0431 859 :
0431 860 :
0431 861 :
0431 862 :
0431 863 :
0431 864 :

00000000'GF 16 0431 857 JSB G^EXE$MCHK_TEST ;RECOVERY BLOCK IN EFFECT?
21 50 E8 0437 858 BLBS R0,EXIT ;YES, DO NOT LOG THIS ERROR
043A 859
043A 860 MCHK$GL_LOG::
51 54 10 C1 043A 861 ADDL3 #EMBSB_MC_SUMCOD,R4,R1 ;SPACE FOR LOG HEADER
00000000'GF 16 043E 862 JSB G^ERL$ALLOCEMB ;GET ERROR LOGGING BUFFER
14 50 E9 0444 863 BLBC R0,EXIT ;FAILED TO GET BUFFER
52 DD 0447 864 PUSHL R2 ;SAVE BUFFER ADDRESS
```



```

      04 A2 53 B0 0449 865      MOVW R3,EMB$W_MC_ENTRY(R2) ;SET ENTRY TYPE IN LOG
10 A2 65 54 28 044D 866      MOV3 R4,(R5),EMB$B_MC_SUMCOD(R2) ;TRANSFER STACK TO LOG
      52 8E D0 0452 867      MOVL (SP)+,R2 ;RETRIEVE BUFFER ADDRESS
      00000000'GF 16 0455 868      JSB G^ERL$RELEASEMB ;RELEASE BUFFER TO LOGGER
      045B 869      EXIT:
05 045B 870      RSB ;EXIT
      045C 871      :
      045C 872      : EXE$LOGMEM - ERROR LOGGING FOR MEMORY CONTROLLERS
      045C 873      :
      045C 874      : ERROR LOGGING FOR PROCESSOR MEMORY
      045C 875      : MEMORIES CAN INCLUDE LOCAL AND MULTIPOINT
      045C 876      :
      045C 877      : INPUTS:
      045C 878      :
      045C 879      : R3 = ERROR LOG TYPE CODE
      045C 880      : 0(SP) = RETURN ADDRESS
      045C 881      :
      045C 882      : OUTPUTS:
      045C 883      :
      045C 884      : ALL MEMORY CONTROLLERS ARE EXAMINED
      045C 885      : THEIR CSR'S ARE READ AND LOGGED TO THE ERRORLOGGER
      045C 886      : R0 = SUCCESS OR FAILURE INDICATION
      045C 887      : SUCCESS = .TRUE. - THE MEMORY CSR'S WERE LOGGED
      045C 888      : FAILURE = .FALSE. - NO DYNAMIC MEMORY AVAILABLE FOR LOG BUFFER
      045C 889      :
      045C 890      : R1-R5 ARE DESTROYED
      045C 891      :
      045C 892      : EXE$LOGMEM::
      045C 893      :
      00000000'GF D6 045C 894      INCL G^EXE$GL_MEMERRS ;BUMP THE GLOBAL MEMORY ERROR COUNTER
      07C0 8F BB 0462 895      PUSH R6,R7,R8,R9,R10 ;SAVE MORE REGISTERS
56 00000000'GF D0 0466 896      MOVL G^EXE$GL_CONFREG,R6 ;ARRAY OF NEXUS DEVICE TYPE CODES
5A 00000000'GF D0 046D 897      MOVL G^MMG$GL_SBICONF,R10 ;ARRAY OF ADAPTER VA'S
      54 7C 0474 898      CLRQ R4 ;COUNTER FOR NO. OF BYTES TO LOG
      58 0F D0 0476 899      MOVL #15,R8 ;INDEX INTO CMI SLOTS
      6648 D0 0479 900 10$: MOVL (R6)[R8],R2 ;ADAPTER TYPE
      03 12 047D 901      BNEQ 12$ ;PROCEED IF ADAPTER PRESENT
      00A2 31 047F 902      BRW 60$
      0482 903 12$: MOVL (R10)[R8],R1 ;VA OF ADAPTER REGISTER
      6A48 D0 0482 904      BGEQ 160$ ;GEQ IMPLIES NO VALID SYSTEM VA.
      63 18 0486 905      BITB #^B11100000,R2 ;MEMORY (BITS 5:7 ARE 0) ?
52 E0 8F 93 0488 906      BEQL 20$ ;YES
      66 13 048C 907      :
      048E 908      :
      40 8F 52 91 048E 909      CMPB R2,#^X40 ;MULTI-PORT MEMORY (3F<TYPE<44) ?
      57 1F 0492 910      BLSSU 160$ ;NO
      43 8F 52 91 0494 911      CMPB R2,#^X43 ;NO
      51 1A 0498 912      BGTRU 160$ ;NO
      57 07 9A 049A 913      MOVZBL #7,R7 ;MARK STACK IN R9
      59 5E D0 049D 914      MOVL SP,R9 ;SPACE FOR MA750 LOG
      5E 20 C2 04A0 915      SUBL #<8*4>,SP
      04A3 916      :
      04A3 917      :
      04B0 918      $PRTCTINI W^100$,#<MCHK$M_LOG!MCHK$M_NEXM>
      6147 D0 04B0 919      :
79 F9 57 F4 04B4 920 15$: MOVL (R1)[R7],-(R9) ;PUSH REGISTERS ON STACK
      57 57 F4 04B4 921      SOBGEQ R7,15$ ;LOOP THROUGH ALL
```

```
04B7 922
04B7 923 :
04B7 924 : CLEAR ERRORS FROM MA750
04B7 925 :
04B7 926 :
04 A1 FF000000 8F C8 04B7 927 BISL #^XFF000000,MPMSL_CR(R1) ;CLEAR PORT INTERFACE CONTROL REG
08 A1 D000C000 8F C8 04BF 928 BISL #^XD000C000,MPMSL_SR(R1) ;CLEAR PORT CONTROLLER STATUS REG
04C7 929
57 10 A1 D0 04C7 930 MOVL MPMSL_ERR(R1),R7 ;READ MULTIPOINT ERROR REGISTER
4D 57 1C E1 04CB 931 BBC #MPMSV_ERR_EL,R7,30$ ;IF CLEAR, NO ERRORS TO LOG NOW
0010'CF48 96 04CF 932 INCB W^ECC$AB_MEMERR[R8] ;ACCOUNT AN ERROR
0010'CF48 03 91 04D4 933 CMPB #CRDINTMAX,W^ECC$AB_MEMERR[R8] ;TOO MANY ERRORS?
00 57 08 1A 04DA 934 BGTRU 18$ ;NO, KEEP LOGGING THEM
10 A1 57 D0 04DC 935 BBSS #MPMSV_ERR_ICRD,R7,17$ ;SET INHIBIT LOGGING BIT
04E4 936 17$: MOVL R7,MPMSL_ERR(R1) ;WRITE IT BACK
04E4 937 18$:
04E4 938
04E4 939 $PRTCTEND 100$
04E5 940
05 50 E8 04E5 941 BLBS R0,110$ ;BRANCH IF MA750 STILL HERE
5E 1C C0 04E8 942 ADDL #<7*4>,SP ;REMOVE LOG SPACE FROM STACK
37 11 04EB 943 BRB 60$ ;CONTINUE TO LOOK FOR SLOTS
54 24 C0 04ED 944 ADDL #<9*4>,R4 ;COUNT NO OF BYTES TO LOG
2A 11 04F0 945 BRB 30$
04F2 946
54 85 11 04F2 947 BRB 10$
54 10 C0 04F4 948 ADDL #<4*4>,R4 ;MEMORY, LOG 3 LONGWORDS + SLOT INDEX
57 02 9A 04F7 949 MOVZBL #2,R7
61 47 DD 04FA 950 25$: PUSHL (R1)[R7] ;PUSH REGISTERS ON STACK
FA 57 F4 04FD 951 SOBGEQ R7,25$
61 6E D0 0500 952 MOVL (SP),(R1) ;WRITE BACK TO CLEAR MEMORY CSR
0010'CF48 96 0503 953 INCB W^ECC$AB_MEMERR[R8] ;ACCOUNT ERROR TO THIS CONTROLLER
0010'CF48 03 91 0508 954 CMPB #CRDINTMAX,W^ECC$AB_MEMERR[R8] ;TOO MANY ERRORS FOR THIS MEMORY?
00 57 04 A1 D0 0510 955 BGTRU 30$ ;NOT YET, KEEP LOGGING
04 A1 57 D0 0514 956 MOVL MEMSL_CSR1(R1),R7 ;DISABLE LOGGING OF CORRECTED ERRORS
051C 957 BBCC #CSR1$V_IERP,R7,28$ ;CLEAR CONDITIONAL LOG BIT IN CSR 1
0518 958 28$: MOVL R7,MEM$C_CSR1(R1) ;WRITE BACK TO MEMORY CSR 1
051C 959
02 AE 58 DD 051C 960 30$: PUSHL R8 ;SAVE SLOT INDEX
52 9B 051E 961 MOVZBW R2,2(SP) ;SAVE ADAPTOR TYPE
55 D6 0522 962 INCL R5 ;COUNT NUMBER OF ENTRIES PUSHED
0524 963
CB 58 F4 0524 964 60$: SOBGEQ R8,19$ ;LOOP THROUGH ALL 16 POSSIBLE SLOTS
55 DD 0527 965 PUSHL R5 ;SAVE COUNT OF NUMBER OF ENTRIES
54 04 C0 0529 966 ADDL #<4*1>,R4
55 5E D0 052C 967 MOVL SP,R5 ;POINT TO LOG INFO
54 DD 052F 968 PUSHL R4 ;SAVE LOG SIZE
7E DC 0531 969 MOVPSL -(SP) ;GET CURRENT PSL FOR PRTCTEST
51 FC AE DE 0533 970 MOVAL -4(SP),R1 ;ADDRESS OF PC,PSL PAIR (AFTER BSB)
52 D4 0537 971 CLRL R2 ;ALWAYS LOG MEMORY ERRORS
FEF5 30 0539 972 BSBW LOGGR ;PUT INFO INTO ERROR LOG
5E 04 C0 053C 973 ADDL #<1*4>,SP ;CLEAR PSL
5E 8E C0 053F 974 ADDL (SP)+,SP ;CLEAN UP THE STACK
07C0 8F BA 0542 975 POPR #^M<R6,R7,R8,R9,R10> ;RESTORE REGISTERS
05 0546 976 RSB ;RETURN
0547 977 ;
```



```
0547 980 .SBTTL REFLECT EXCETION TO USER
0547 981 :
0547 982 : REFLECT ERROR TO USER
0547 983 :
0547 984 : IF CURRENT MODE WAS USER OR SUPERVISOR MODE, SET UP EXCEPTION ON
0547 985 : USERS KERNEL STACK AND REI TO IT
0547 986 :
0547 987 : IF CURRENT MODE WAS KERNEL OR EXECUTIVE MODE, ISSUE A FATAL BUG-CHECK
0547 988 :
0547 989 :
0547 990 REFLECTCHK: ;REFLECT EXCEPTION TO CURRENT ACCESS MODE
0547 991 :
OE 30 AC 19 EO 0547 992 BBS #PSL$V_CURMOD+1,MCK_PSL(AP),20$ ;USER OR SUPER MODE?
103F 8F BA 054C 993 :
00000000'GF 16 054C 994 POPR #^M<R0,R1,R2,R3,R4,R5,AP> ;RESTORE REGISTERS
0550 995 JSB G^EXE$MCHK_BUGCHK ;RECOVERY BLOCK IN EFFECT?
0556 996 :
0556 997 BUG_CHECK MACHINECHK,FATAL ;BUG-CHECK IN EXEC OR KERNAL MODE
055A 998 :
50 00 DB 055A 999 20$: MFPR #PR$_KSP,R0 ;GET KERNEL MODE STACK POINTER
70 2C AC 7D 055D 1001 : ;FOR CURRENT PROCESS
00 50 DA 0561 1002 MOVQ MCK_PC(AP),-(R0) ;MOVE INTERRUPT PC AND PSL OF MACHINE
103F 8F BA 0561 1003 : ;CHECK TO PROCESS'S KERNEL STACK
5E 08 CO 0561 1004 MTPR R0,#PR$_KSP ;REPLACE NEW KERNAL STACK POINTER
5E 8E CO 0564 1005 POPR #^M<R0,R1,R2,R3,R4,R5,AP> ;RESTORE REGISTERS SAVED AT BEGINNING
0568 1006 ADDL #<2*4>,SP ;REMOVE PRTCTEST STUFF
056B 1007 ADDL (SP)+,SP ;POP HARDWARE MACHINE CHECK LOG FROM STACK
056E 1008 :
056E 1009 : THIS IS A SLIGHT PIECE OF MAGIC - NOTHING SPECTACULAR
056E 1010 : IF YOU HAVE RETRIEVED THE PIRATE'S CHEST, THEN THIS SHOULD BE EASY
056E 1011 :
056E 1012 : WE ARE NOW EXECUTING ON THE SYSTEM INTERRUPT STACK AT IPL 31
056E 1013 : THE PC AND PSL FROM THIS MACHINE CHECK IS PLACED ON THE CURRENT PROCESS'S
056E 1014 : KERNEL STACK.
056E 1015 : THE PC AND PSL FROM THIS MACHINE CHECK ON THE INTERRUPT STACK IS REPLACED
056E 1016 : BY A PC WHICH POINTS INTO THE MODULE 'EXCEPTION'. THE PSL IS REPLACED
056E 1017 : WITH A PSL INDICATING A KERNEL MODE EXCEPTION.
056E 1018 : WHEN THE FOLLOWING REI IS EXECUTED, WE END EXECUTING ON THE KERNEL
056E 1019 : STACK OF THE CURRENT PROCESS IN THE SYSTEM EXCEPTION INTERRUPT HANDLER.
056E 1020 : THE CODE THERE REPORTS AN EXCEPTION TO THE CURRENT PROCESS AND REI'S BACK
056E 1021 : TO WHERE THE ORIGONAL MACHINE CHECK HAPPENED.
056E 1022 : IF THE CURRENT PROCESS HAS AN EXCEPTION HANDLER FOR MACHINE CHECK'S THEN
056E 1023 : SOME KIND OF FORWARD ERROR RECOVERY CAN BE ENVOCKED. IF NOT, THE SYSTEM
056E 1024 : LAST CHANCE HANDLER WILL END UP DELEATING THE PROCESS.
056E 1025 :
6E 00000000'GF 9E 056E 1026 MOVAB G^EXE$MCKECHK,(SP) ;WITH MIRRORS - SET UP A PC AND
04 AE 04 AE 02 18 EF 0575 1027 : ;PSL FOR EXCEPTION
04 AE 04 AE 16 9C 0575 1028 EXTZV #PSL$V_CURMOD,#PSL$S_CURMOD,4(SP),4(SP) ;GET MODE WE WERE IN
057C 1029 ROTL #PSL$V_PVRMOD,4(SP),4(SP) ;CREATE A PSL OF CURRENT TO BE
0582 1030 : ;KERNAL WITH CORRECT PREVIOUS MODE
02 0582 1031 REI ;GET TO EXCEPTION HANDLER
```



```

0583 1035 .SBTTL TABLE OF RESUMABLE INSTRUCTIONS.
0583 1036 : EACH BIT IN THE TABLE IS A 1 IF THE INSTRUCTION IS RESUMABLE,
0583 1037 : AND A 0 IF IT IS NOT.
0583 1038
0583 1039 RESUMABLE:
OF6B 0583 1040 .WORD ^B0000111101101011 ;REI,RET,SVPCTX,PROBEX,INSQUE,REMQUE
FFBF 0585 1041 .WORD ^B1111111110111111 ;JSB
FFFF 0587 1042 .WORD ^B1111111111111111
FFFF 0589 1043 .WORD ^B1111111111111111
FFFF 058B 1044 .WORD ^B1111111111111111
002F 058D 1045 .WORD ^B0000000000101111 ;EMODF,CVTFD,INTERLOCKED INSTRUCTIONS
OF00 058F 1046 .WORD ^B0000111100000000 ;DOUBLE PRECISION FLOATING POINT
C14A 0591 1047 .WORD ^B1100000101001010 ;MORE DOUBLE PREC/QUAD, EMUL, EDIV
FFFF 0593 1048 .WORD ^B1111111111111111
FFFF 0595 1049 .WORD ^B1111111111111111
FFFF 0597 1050 .WORD ^B1111111111111111
03FF 0599 1051 .WORD ^B0000001111111111 ;PUSHR,POPR,CHMK,CHME,CHMS,CHMU
FFFF 059B 1052 .WORD ^B1111111111111111
FFFF 059D 1053 .WORD ^B1111111111111111
FFFF 059F 1054 .WORD ^B1111111111111111
01FF 05A1 1055 .WORD ^B0000000111111111 ;CVTLP,CALLG,CALLS,XFC,ESCD,ESCE,EXCF
05A3 1056

```


MCHECK750
V04-000

- VAX 11/750 MACHINE CHECK HANDLER^C 9
TABLE OF RESUMABLE INSTRUCTIONS.

05A3 1058 .END

16-SEP-1984 00:50:52 VAX/VMS Macro V04-00 Page 24
5-SEP-1984 04:10:23 [SYSLOA.SRC]MCHECK750.MAR;1 (13)

MCH
V04

MCHECK750
Symbol table

- VAX 11/750 MACHINE CHECK HANDLER

16-SEP-1984 00:50:52 VAX/VMS Macro V04-00
5-SEP-1984 04:10:23 [SYSLOA.SRC]MCHECK750.MAR;1

Page 25
(13)

BAD_TYPE	0000002D	R	03
BERSV_NEX	= 00000003		
BERSV_UCD	= 00000002		
BUGS_BADMCKCOD	*****	X	03
BUGS_MACHINECHK	*****	X	03
BUGS_RDSNONRES	*****	X	03
BUS	0000017D	R	03
BUS_CACHE	00000163	R	03
CADRSM_DIS	= 00000001		
CADRSV_DIS	= 00000000		
CERSV_DATA	= 00000002		
CERSV_TAG	= 00000003		
CH_PARITY	000002BC	R	03
CH_THRESHOLD	= 0000000A		
CRDINTMAX	= 00000003		
CRDWATCHMAX	= 00000006		
CSROSV_COREF	= 0000001D		
CSR1SV_IERP	= 0000001C		
CS_PARITY	000000A5	R	03
ECC\$AB_MEMERR	00000010	RG	02
ECC\$GW_REENAB	0000002E	RG	02
ECC\$GW_WATCH	0000002C	RG	02
ECC\$REENABLE	00000303	RG	03
EMBSB_MC_SUMCOD	= 00000010		
EMBSK_AW	= 00000007		
EMBSK_HE	= 00000008		
EMBSK_MC	= 00000002		
EMBSK_SE	= 00000006		
EMBSK_SI	= 0000000A		
EMBSW_MC_ENTRY	= 00000004		
ERLSA\$LOC\$EMB	*****	X	03
ERLS\$RELEASEMB	*****	X	03
ESRSV_TB	= 00000002		
EXESGL_BADTIMOUT	00000034	RG	02
EXESGL_CH1OLD	00000020	RG	02
EXESGL_CH2OLD	00000024	RG	02
EXESGL_CHSTATE	00000028	RG	02
EXESGL_CONFREGL	*****	X	03
EXESGL_FLAGS	*****	X	03
EXESGL_MCHKERRS	*****	X	03
EXESGL_MEMERRS	*****	X	03
EXESGL_TB1AOLD	00000000	RG	02
EXESGL_TB1BOLD	00000008	RG	02
EXESGL_TB2AOLD	00000004	RG	02
EXESGL_TB2BOLD	0000000C	RG	02
EXESGL_VECTIMOUT	00000038	RG	02
EXESINT54	000002F4	RG	03
EXESINT58	0000005C	RG	03
EXESINT5C	0000005C	RG	03
EXESINT60	000000BC	RG	03
EXESLOGAWE	000000BC	RG	03
EXESLOGCRD	000002F4	RG	03
EXESLOGMEM	0000045C	RG	03
EXESLOGSBA	0000005C	RG	03
EXESLOGSBF	0000005C	RG	03
EXESM\$CHECK	*****	X	03
EXESM\$CHK	00000000	RG	03

EXESM\$CHK_BUGCHK	*****	X	03
EXESM\$CHK_ERRCNT	00000000	RG	02
EXESM\$CHK_PRTCT	*****	X	03
EXESM\$CHK_TEST	*****	X	03
EXESRH780_INT	0000005C	RG	03
EXESV_CRDENABL	*****	X	03
EXIT	0000045B	R	03
FUBAR	000000A5	R	03
IPL\$ASTDEL	= 00000002		
IRD_SLOT	000000A5	R	03
LOGGER	00000431	R	03
LOGIT	00000416	R	03
MCHK\$GL_LOG	0000043A	RG	03
MCHK\$M_COG	= 00000001		
MCHK\$M_MCK	= 00000002		
MCHK\$M_NEXM	= 00000004		
MCK_BER	00000024		
MCK_CER	00000020		
MCK_CODE	00000004		
MCK_ERROR_PC	0000000C		
MCK_ESR	00000028		
MCK_LENGTH	00000000		
MCK_MDR	00000010		
MCK_PC	0000002C		
MCK_PSL	00000030		
MCK_RLT	00000018		
MCK_SMR	00000014		
MCK_TBP	0000001C		
MCK_VA	00000008		
MEM\$SL_CSRO	00000000		
MEM\$SL_CSR1	00000004		
MEM\$SL_CSR2	00000008		
MMGS\$AL_SYSPCB	*****	X	03
MMGS\$DE\$CONPFN	*****	X	03
MMGS\$DELWSLEX	*****	X	03
MMGS\$GL_CRDCNT	00000030	RG	02
MMGS\$GL_MAXPFN	*****	X	03
MMGS\$GL_SBICONF	*****	X	03
MMGS\$GW_BIGPFN	*****	X	03
MMGS\$IN\$SPFNT	*****	X	03
MMGS\$REFCNTNEG	*****	X	03
MMGS\$SVAPTECHK	*****	X	03
MPM\$SL_CR	= 00000004		
MPM\$SL_ERR	= 00000010		
MPM\$SL_SR	= 00000008		
MPM\$SV_ERR_EL	= 0000001C		
MPM\$SV_ERR_ICRD	= 0000001E		
NO_RESUME	000001FF	R	03
PCBSL_PHD	= 0000006C		
PFNS\$AB_STATE	*****	X	03
PFNS\$AB_TYPE	*****	X	03
PFNS\$AW_REFcnt	*****	X	03
PFNS\$AX_WSLX	*****	X	03
PFNSC_BADPAGLST	= 00000002		
PFNSC_PROCESS	= 00000000		
PFNSC_SYSTEM	= 00000001		
PFNSM_BADPAG	= 00000020		

MCHECK750
Symbol table

- VAX 11/750 MACHINE CHECK HANDLER ^{E 9}

16-SEP-1984 00:50:52 VAX/VMS Macro V04-00 Page 26
5-SEP-1984 04:10:23 [SYSLOA.SRC]MCHECK750.MAR;1 (13)

```

PFNSM_MODIFY      = 00000080
PFNSS_PAGTYP      = 00000003
PFNSV_PAGTYP      = 00000000
PRS_IPL           = 00000012
PRS_KSP           = 00000000
PRS_TBIA          = 00000039
PRS_TBIS          = 0000003A
PR750$_CADR       = 00000025
PR750$_CAER       = 00000027
PR750$_MCESR      = 00000026
PR750$_TBDR       = 00000024
PR750$_TODR       = 0000001B
PSLSS_CURMOD      = 00000002
PSLSS_IPL         = 00000005
PSLSV_CURMOD      = 00000018
PSLSV_IPL         = 00000010
PSLSV_PRVMOD      = 00000016
PTESS_PFN         = 00000015
PTESV_MODIFY      = 0000001A
PTESV_PFN         = 00000000
PTESV_VALID       = 0000001F
PTESV_WINDOW      = 00000015
RDSNONRES         = 000001BD R      03
REENABTIME        = 00000384
REFLECTCHK        = 00000547 R      03
RESUMABLE         = 00000583 R      03
SCH$GL_CURPCB     = ***** X    03
SIZ...            = 00000003
SMRSS_MODE        = 00000002
SMRSV_MODE        = 00000000
SOMETIME          = 0000003C
TBDRSM_DGO        = 00000001
TBDRSM_DG1        = 00000002
TBDRSM_GRPDP      = 00000004
TBDRSM_REPL       = 00000008
TBDRSV_DGO        = 00000000
TBDRSV_DG1        = 00000001
TBDRSV_GRPDP      = 00000002
TBDRSV_REPL       = 00000003
TBPSM_GRPDP       = 00000001
TBPSM_GRPOT       = 00000004
TB_BAD            = 0000015C R      03
TB_BUS            = 000000F1 R      03
TB_THRESHOLD      = 0000000A
TRYRESUME         = 00000293 R      03
UCD               = 0000018F R      03

```

+-----+
! 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	00000034 (52.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
MCHK\$DATA	0000003C (60.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC QUAD
WIONONPAGED	000005A3 (1443.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC QUAD

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

Phase	Page faults	CPU Time	Elapsed Time
-----	-----	-----	-----
Initialization	29	00:00:00.06	00:00:00.49
Command processing	111	00:00:00.34	00:00:02.69
Pass 1	308	00:00:07.24	00:00:25.85
Symbol table sort	0	00:00:00.87	00:00:03.97
Pass 2	195	00:00:02.07	00:00:10.20
Symbol table output	19	00:00:00.11	00:00:00.20
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	666	00:00:10.71	00:00:43.41

The working set limit was 1650 pages.
61858 bytes (121 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 844 non-local and 52 local symbols.
1058 source lines were read in Pass 1, producing 22 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	19
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	8
TOTALS (all libraries)	27

888 GETS were required to define 27 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:MCHECK750/OBJ=OBJ\$:MCHECK750 MSRC\$:MCHECK750/UPDATE=(ENH\$:MCHECK750)+EXECMLS/LIB

0397 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

