

[illegible]

```

EEEEEEEEEE RRRRRRRR RRRRRRRR SSSSSSSS UU UU BBBB BBBB UU UU VV VV 11
EEEEEEEEEE RRRRRRRR RRRRRRRR SSSSSSSS UU UU BBBB BBBB UU UU VV VV 11
EE          RR          RR RR          SS          UU UU BB BB UU UU VV VV 1111
EE          RR          RR RR          SS          UU UU BB BB UU UU VV VV 1111
EE          RR          RR RR          SS          UU UU BB BB UU UU VV VV 11
EEEEEEEEEE RRRRRRRR RRRRRRRR SSSSSS UU UU BBBB BBBB UU UU VV VV 11
EEEEEEEEEE RRRRRRRR RRRRRRRR SSSSSS UU UU BBBB BBBB UU UU VV VV 11
EE          RR RR          RR RR          SS          UU UU BB BB UU UU VV VV 11
EE          RR RR          RR RR          SS          UU UU BB BB UU UU VV VV 11
EE          RR RR          RR RR          SS          UU UU BB BB UU UU VV VV 11
EEEEEEEEEE RRR          RR RR          SSSSSSSS UUUUUUUUU BBBB BBBB UUUUUUUUU VV VV 111111
EEEEEEEEEE RR          RR RR          SSSSSSSS UUUUUUUUU BBBB BBBB UUUUUUUUU VV VV 111111

```

```

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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```


(4)	257	EX\$INIBOOTADP - INITIALIZE THE BOOT DEVICE ADAPTER
(5)	391	EX\$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
(5)	392	EX\$STARTUPADP - STARTUP ANY ADAPTERS
(6)	461	EX\$DUMPCPUREG - DUMP CPU-SPECIFIC IPR'S
(7)	577	EX\$READ TODR (P) - READ TIME-OF-DAY CLOCK
(8)	666	EX\$WRITE TODR (P) - WRITES TIME-OF-DAY CLOCK
(9)	724	EX\$REGSAVE - SAVE CPU-SPECIFIC IPR'S
(10)	786	EX\$REGRESTOR - RESTORE CPU-SPECIFIC IPR'S
(11)	846	EX\$INIPROCREG - CPU-DEPENDENT INITIALIZATION OF IPR'S
(13)	985	SYSL\$CLRSBIA
(14)	1025	EX\$TEST_CSR
(15)	1197	ADPLINK = LINK ADAPTER CONTROL BLOCK INTO ADP LIST


```
0000 1      .NOSHOW CONDITIONALS
0000 5
0000 9
0000 13
0000 17
0000 19      .TITLE  ERRSUBUV1 - ERROR SUBROUTINES FOR MICRO-VAX I
0000 21
0000 22      .IDENT  'V04-002'
0000 23
0000 24
0000 25      *****
0000 26      *
0000 27      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 28      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 29      *  ALL RIGHTS RESERVED.
0000 30      *
0000 31      *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 32      *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 33      *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 34      *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 35      *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 36      *  TRANSFERRED.
0000 37      *
0000 38      *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 39      *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 40      *  CORPORATION.
0000 41      *
0000 42      *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 43      *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 44      *
0000 45      *
0000 46      *****
0000 47
0000 48      ++
0000 49
0000 50      FACILITY:
0000 51
0000 52      EXECUTIVE, LOADABLE SUBROUTINES USED BY POWERFAIL AND BUGCHECK.
0000 53
0000 54      ABSTRACT:
0000 55
0000 56      LOADABLE SUBROUTINES USED BY POWERFAIL AND BUGCHECK.
0000 57
0000 58      AUTHOR:
0000 59
0000 60      N. KRONENBERG, JULY 2, 1979.
0000 61
0000 62      MODIFIED BY:
0000 63
0000 64      V04-003 WMC00001      Wayne Cardoza      13-Sep-1984
0000 65      CRD reporting must not be turned off for VENUS.
0000 66
0000 67      V04-002 CWH4002      CW Hobbs      08-Sep-1984
0000 68      Correct typo in TCM0010, use "-" instead of "="
0000 69
0000 70      V04-001 TCM0010      Trudy C. Matthews      07-Sep-1984
0000 71      For the venus processor: move turning on cache from routine
```


0000 72 : EXE\$INIPROCREG to a new routine: INISCACHE. Correct the
0000 73 : order in which registers are saved on the stack in EXE\$REGSAVE.
0000 74 :
0000 75 : V03-022 TCM0009 Trudy C. Matthews 30-Jul-1984
0000 76 : When turning off CRD interrupts in EXE\$INIPROCREG for VENUS,
0000 77 : read the processor register and write it back to preserve
0000 78 : the state of other bits in the register.
0000 79 :
0000 80 : V03-021 TCM0008 Trudy C. Matthews 23-Jul-1984
0000 81 : Remove venus code that queries the console for how to set up
0000 82 : cache and FBOX state. Instead always turn the cache and
0000 83 : FBOX on (and let the normal error handling code turn it off
0000 84 : if its bad).
0000 85 :
0000 86 : V03-020 DWT0214 David W. Thiel 02-May-1984
0000 87 : Revise MicroVAX I TODR register simulation.
0000 88 :
0000 89 : V03-019 KDM0096 Kathleen D. Morse 27-Mar-1984
0000 90 : Add missing indirection in MicroVAX I memory CSR
0000 91 : CRD enabling.
0000 92 :
0000 93 : V03-018 KPL0101 Peter Lieberwirth 4-Mar-1984
0000 94 : Add extra vectors now defined in SYSLOAVEC. These vectors
0000 95 : are insurance for v4.x
0000 96 :
0000 97 : V03-017 KPL0100 Peter Lieberwirth 12-Feb-1984
0000 98 : Change RPB\$B_BOOTNDT to RPB\$W_BOOTNDT, since BI devices
0000 99 : will have 16-bit device types.
0000 100 :
0000 101 : V03-016 KDM0092 Kathleen D. Morse 23-Jan-1984
0000 102 : Correct the number of cpu-specific IPRs logged for the
0000 103 : 11/730 and MicroVAX I cpus.
0000 104 :
0000 105 : V03-015 CWH8001 CW Hobbs 5-Dec-1983
0000 106 : Add entry points for EXE\$READP_TODR and EXE\$WRITEP_TODR
0000 107 : to access physical TODR register for Nautilus CPU. For
0000 108 : other processors, these amount to duplicate labels on
0000 109 : EXE\$READ_TODR and EXE\$WRITE_TODR.
0000 110 :
0000 111 : V03-014 KTA3088 Kerbey T. Altmann 17-Oct-1983
0000 112 : Fix bug in 730 conditional for EXE\$INIBOOTADP.
0000 113 :
0000 114 : V03-013 KDM0081 Kathleen D. Morse 13-Sep-1983
0000 115 : Create Micro-VAX I version.
0000 116 :
0000 117 : V03-012 KDM0055 Kathleen D. Morse 12-Jul-1983
0000 118 : Move IPR PME into the cpu-dependent register save and
0000 119 : restore routines.
0000 120 :
0000 121 : V03-011 KDM0049 Kathleen D. Morse 07-Jul-1983
0000 122 : Add the following processor registers to the cpu-specific
0000 123 : dump IPRs routine: ICR, TODR, ACCS. Add usage of
0000 124 : register: EXE\$READ_TODR and EXE\$WRITE_TODR.
0000 125 :
0000 126 : V03-010 KDM0048 Kathleen D. Morse 07-Jul-1983
0000 127 : Add loadable routines for referencing the time-of-day
0000 128 : clock: EXE\$READ_TODR, EXE\$WRITE_TODR.


```
0000 129 :
0000 130 :
0000 131 :
0000 132 :
0000 133 :
0000 134 :
0000 135 :
0000 136 :
0000 137 :
0000 138 :
0000 139 :
0000 140 :
0000 141 :
0000 142 :
0000 143 :
0000 144 :
0000 145 :
0000 146 :
0000 147 :
0000 148 :
0000 149 :
0000 150 :
0000 151 :
0000 152 :
0000 153 :
0000 154 :
0000 155 :
0000 156 :
0000 157 :
0000 158 :
0000 159 :
0000 160 :
0000 161 :
0000 162 :
0000 163 :
0000 164 :
0000 165 :--

V03-009 TCM0007      Trudy C. Matthews      02-Jun-1983
Fix routine SYSLSCLRSBIA so that it calculates the address
of SBI adapter register space correctly.

V03-008 TCM0006      Trudy C. Matthews      9-Feb-1983
Store enable/disable state of 11/790 cache and FBOX in
EXESGB_CPUDATA cell during system initialization.

V03-007 TCM0005      Trudy C. Matthews      11-Jan-1983
Add routine SYSLSCLRSBIA. Add SBIA register initialization
to EXESINIPROCREG. Add 11/790 machine check handler to
EXESTEST CSR. Change 11/780 machine check handler to
write PR$ SBIFS back to itself to clear error bits.
Add labels for two "extra" routines, that can be patched
if extra vectors from SYS to SYSLOA are needed in between
major releases. Make EXESDUMPCPUREG log the SBI registers
from the SBI the 11/790 system disk is on.

V03-006 TCM0004      Trudy C. Matthews      3-Jan-1983
Add more 11/790-specific code.

V03-005 TCM0003      Trudy C. Matthews      17-Dec-1982
Add conditional assembly switch to the invocations
of 11/790-specific definition macros.

V03-004 TCM0002      Trudy C. Matthews      15-Dec-1982
Added 11/790-specific code to EXESINIPROCREG.

V03-003 TCM0001      Trudy C. Matthews      13-Dec-1982
Added 11/790-specific code to power down/power up
routines.

V03-002 KTA3018      Kerbey T. Altmann      30-Oct-1982
Remove CI and UBA routines to another module.
```



```
0000 167
0000 168 :
0000 169 : MACRO LIBRARY CALLS:
0000 170 :
0000 171
0000 172 $ADPDEF ;DEFINE ADAPTER OFFSETS
0000 173 $BQODEF ;DEFINE BOOT QIO OFFSETS
0000 174 $BTDDDEF ;DEFINE BOOT DEVICE TYPES
0000 175 $EMBCRDEF ;DEFINE ERROR MSG BUFFER OFFSETS
0000 176 $IDBDEF ;DEFINE INTERRUPT DISPATCH OFFSETS
0000 177 $IPLDEF ;DEFINE INTERRUPT PRIORITY LEVELS
0000 178 $MBADEF ;DEFINE MASSBUS ADAPTER OFFSETS
0000 179 $NDTDEF ;DEFINE NEXUS DEVICE TYPES
0000 180 $PRDEF ;DEFINE INTERNAL PROCESSOR REGISTERS
0000 181 $RPBDEF ;DEFINE RESTART PARAM BLOCK OFFSETS
0000 182 $SSDEF ;DEFINE SYSTEM STATUS CODES
0000 183 $UBADEF ;DEFINE UNIBUS ADAPTER OFFSETS
0000 195
0000 199
0000 203
0000 207
0000 209 $PRUV1DEF ;DEFINE MICRO-VAX I INTERNAL PROC REGS
0000 211 :
0000 212 : EQUATED SYMBOLS:
0000 213 :
0000 218
0000 223
0000 228
0000 233
00000000 0000 235 C780_LIKE = 0
00000001 0000 236 C750_LIKE = 1
0000 238
0000 239 :
0000 240 : Define labels for two "extra" routines. This reserves some vectors from
0000 241 : SYS.EXE into SYSLOAxxx.EXE that can be patched if another routine must
0000 242 : be added in between major releases.
0000 243 :
0000 244 EXE$EXTRA1:: : aligned
0000 245 EXE$EXTRA2:: : aligned
0000 246 EXE$EXTRA3:: : aligned
0000 247 EXE$EXTRA4:: : aligned
0000 248 EXE$EXTRA5:: : aligned
0000 249 EXE$EXTRA6:: : packed
0000 250 EXE$EXTRA7:: : packed
0000 251 EXE$EXTRA8:: : packed
0000 252 EXE$EXTRA9:: : packed
0000 253 EXE$EXTRA10:: : packed (think this is enough?)
0000 254
00 0000 255 HALT ; Error if these labels are used.
```



```

0001 257 .SBTTL EX$INIBOOTADP - INITIALIZE THE BOOT DEVICE ADAPTER
0001 258 :+
0001 259 : EX$INIBOOTADP - GET THE SYSTEM BOOT DEVICE ADAPTER AND INIT IT.
0001 260 : THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE BOOTDRIVER IS CALLED.
0001 261 :
0001 262 : INPUTS:
0001 263 :
0001 264 : R6 = RPB ADDRESS
0001 265 :
0001 266 : OUTPUTS:
0001 267 :
0001 268 : R0-R2 DESTROYED
0001 269 : OTHER REGISTERS PRESERVED
0001 270 :-
0001 271 :
0000 272 .PSECT SYSLOA, LONG
0000 273 .ENABLE LSB
0000 274
0000 275 EX$INIBOOTADP:: ;SUBROUTINE ENTRY
05 0000 329 RSB
0001 331
0001 332 INI_UBADP: ;INIT UBA
0001 333
0001 341
0001 343
0001 347
0001 351
0001 353 ;**temp broken**
0001 354 ; MTPR #0, #PRUV1$_IORESET ;INITIALIZE QBUS
0001 356
0001 358
05 0001 388 40$: RSB ;DONE WITH UBA INIT
0002 389 .DISABLE LSB

```



```
0002 391 .SBTTL EX$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
0002 392 .SBTTL EX$STARTUPADP - STARTUP ANY ADAPTERS
0002 393 :+
0002 394 : EX$SHUTDWNADP - SHUTDOWN ANY ADAPTERS DURING BUGCHECK
0002 395 : THIS ROUTINE IS CALLED FROM BUGCHECK BEFORE THE DUMP IS TAKEN TO
0002 396 : ENSURE THAT ALL ADAPTERS THAT NEED TO BE QUIESENT ARE.
0002 397 :
0002 398 : INPUTS:
0002 399 :
0002 400 : IPL = 31
0002 401 :
0002 402 : OUTPUTS:
0002 403 :
0002 404 : OTHER REGISTERS PRESERVED
0002 405 :-
0002 406 : .ENABLE LSB
0002 407 :
0002 408 EX$STARTUPADP::
0002 409 PUSH R0,R1,R2,R4 ; Save a register
51 49'AF DE 0004 410 MOVAL B^ADP_TBL_UP,R1 ; Address of startup table
0002 411 BRB 5$ ; Join common code
0002 412 :
0002 413 EX$SHUTDWNADP::
0002 414 PUSH R0,R1,R2,R4 ; Save a register
51 31'AF DE 000C 415 MOVAL B^ADP_TBL_DWN,R1 ; Address of shutdown table
FFFFF7FC'9F DE 0010 416 5$: MOVAL @#<IO$GL_ADPLIST- -
52 04 A2 DO 0016 417 ADP$L_LINK>,R2 ; Get pointer to head of adapter list
0017 418 10$: MOVL ADP$L_LINK(R2),R2 ; Flink onward
001B 419 BEQL 20$ ; Branch if at end of list
54 62 DO 001D 420 MOVL ADP$L_CSR(R2),R4 ; Get address of CSR
50 0E A2 3C 0020 421 MOVZWL ADP$W_ADPTYPE(R2),R0 ; Get adapter type code
50 6140 DE 0024 422 MOVAL (R1)[R0],R0 ; Get table entry of adap shutdown
00 B040 16 0028 423 JSB @<R0>[R0] ; Call adapter shutdown
002C 424 BRB 10$ ; Next adapter
002E 425 :
002E 426 20$: POP R0,R1,R2,R4
0030 427 30$: RSB
0031 428 :
0031 429 :
0031 430 : Table of addresses of adapter shutdown routines ordered
0031 431 : by adapter type in ADP$W_ADPTYPE.
0031 432 :
0031 433 :
0031 434 ADP_TBL_DWN: ; Address table start
FFFFF7FF 0031 435 .LONG 30$- ; 0-MBA
FFFFF7CC 0035 437 .LONG INI_UBADP- ; 1-UBA (QBUS is different than UNIBUS)
FFFFF7F7 0039 441 .LONG 30$- ; 2-DR32
FFFFF7F3 003D 442 .LONG 30$- ; 3-MA780
FFFFF7BF 0041 443 .LONG C1$SHUTDOWN- ; 4-C1
FFFFF7EB 0045 444 .LONG 30$- ; Rsvrd for future expansion
0049 445 :
0049 446 :
0049 447 : Table of addresses of adapter startup routines ordered
0049 448 : by adapter type in ADP$W_ADPTYPE.
0049 449 :
0049 450 :
0049 451 ADP_TBL_UP: ; Address table start
```

ERRSUBUV1
V04-002

K 13
- ERROR SUBROUTINES FOR MICRO-VAX I
EXES\$STARTUPADP - STARTUP ANY ADAPTERS

16-SEP-1984 01:07:13 VAX/VMS Macro V04-00
13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5

Page 7
(5)

FFFFFFFFB7'	0049	452	.LONG	MB\$INITIAL-	:	0-MBA
FFFFFFFFB3'	004D	453	.LONG	UB\$INITIAL-	:	1-UBA
FFFFFFFFDF	0051	454	.LONG	30\$-	:	2-DR32
FFFFFFFFAB'	0055	455	.LONG	MA\$INITIAL-	:	3-MA780
FFFFFFFFD7	0059	456	.LONG	30\$-	:	4-CI
FFFFFFFFD3	005D	457	.LONG	30\$-	:	Rsvrd for future expansion
	0061	458				
	0061	459	.DISABLE	LSB		


```
0061 461 .SBTTL EXESDUMPCPUREG - DUMP CPU-SPECIFIC IPR'S
0061 462 :+
0061 463 : DUMP CPU-SPECIFIC IPR'S INTO ERROR MESSAGE BUFFER.
0061 464 :
0061 465 : TWENTY-FOUR LONGWORDS ARE RESERVED IN THE EMB FOR CPU-SPECIFIC
0061 466 : IPR'S. THE FORMATS FOR VARIOUS CPU'S ARE:
0061 467 :
0061 468 : 11/780: 11/750: 11/730: 11/790: uVAX I:
0061 469 :
0061 470 : ICR ICR ICR ICR UNUSED(0)
0061 471 : TODR TODR TODR TODR APPROX TODR
0061 472 : ACCS ACCS ACCS ACCS UNUSED(0)
0061 473 : SBIFS TBDR 21 UNUSED(0) SBISTS (1st SBI) 21 UNUSED(0)
0061 474 : SBISC CADR
0061 475 : SBIMT MCESR
0061 476 : SBIER CAER
0061 477 : SBIS CMERR
0061 478 : 16 SBI SILO 16 UNUSED(0) 16 SBI SILO
0061 479 :
0061 480 : INPUTS:
0061 481 :
0061 482 : R0 - ADDR IN EMB OF START OF CPU-SPECIFIC REGISTERS=
0061 483 : OFFSET EMB$$_CR_CPUREG
0061 484 :
0061 485 : OUTPUTS:
0061 486 :
0061 487 : R0,R1 DESTROYED
0061 488 : ALL OTHER REGISTERS PRESERVED
0061 489 : -
0061 490 :
0061 491 : .ENABL LSB
0061 492 :
0061 493 EXESDUMPCPUREG:: :SUBROUTINE ENTRY
0061 494 :
0061 495 :
0061 509 :
0061 510 :
0061 524 :
0061 525 :
0061 536 :
0061 537 :
0061 558 :
0061 559 :
0061 561 CLRL (R0)+ :NO ICR TO LOG.
0063 562 MOVL R0,R1 :SAVE R0.
0066 563 JSB EXESREAD_TODR :GET APPROXIMATE TODR VALUE.
006C 564 MOVL R0,(R1)+ :LOG THE APPROXIMATE TODR VALUE.
006F 565 MOVL R1,R0 :RESTORE R0.
0072 566 MOVL #<<EMB$$_CR_CODE - EMB$$_CR_CPUREG>/4>-2, R1 ; -2 FOR THE
0075 567 : UNUSED ICR AND TODR ALREADY LOGGED.
0075 568 : THERE ARE NO OTHER CPU-SPECIFIC
0075 569 10$: CLRL (R0)+ : REGISTERS TO LOG, SO ZERO THE
0077 570 SOBGTR R1, 10$ : SPACE IN THE ERROR MSG BUFFER
007A 572 90$:
007A 573 RSB
007B 574 .DISABLE LSB
007B 575
```

51 80 D4 0061 561
51 50 D0 0063 562
000007B'EF 16 0066 563
81 50 D0 006C 564
50 51 D0 006F 565
51 16 D0 0072 566
80 D4 0075 567
FB 51 F5 0075 568
05 0077 569
007A 570
007A 571
007B 572
007B 573
007B 574
007B 575

```
007B 577 .SBTTL EXES$READ_TODR (P) - READ TIME-OF-DAY CLOCK
007B 578 :+
007B 579 : READS THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
007B 580 : DIFFERENT WAYS: AS AN INTERNAL PROCESSOR REGISTER, AS PART
007B 581 : OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE. IT
007B 582 : MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
007B 583 :
007B 584 : INPUTS:
007B 585 :
007B 586 : NONE.
007B 587 :
007B 588 : OUTPUTS:
007B 589 :
007B 590 : R0 - TODR VALUE
007B 591 : ALL OTHER REGISTERS PRESERVED
007B 592 :-
007B 593 :
007B 594 EXES$READP_TODR:: ; SUBROUTINE ENTRY
007B 595 :
007B 596 : NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
007B 597 : REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS. FIRST, THE PHYSICAL
007B 598 : TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION. SECOND, A
007B 599 : REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
007B 600 : NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXES$READ_TODR ENTRY
007B 601 : WHICH WILL FABRICATE THE TIME FROM THE QUADWORD SYSTEM TIME.
007B 602 :
007B 603 : NOT NAUTILUS - FALL THROUGH TO READ_TODR
007B 604 :
007B 605 EXES$READ_TODR:: ; SUBROUTINE ENTRY
007B 606 :
007B 607 :
007B 611 :
007B 612 :
007B 616 :
007B 617 :
007B 621 :
007B 622 :
007B 626 :
007B 628 :
007B 629 : Simulated TODR value is the last value stored into "TODR"
007B 630 : (EXES$GL_TODR) plus the number of ticks that have occurred
007B 631 : since then ((EXES$GQ_SYSTIME-EXES$GQ_TODCBASE)/100000).
007B 632 : 100000 is the number of 100 ns. in a TODR tick (10 ms).
007B 633 :
006 06 BB 007B 634 PUSHR #^M<R1,R2> ;SAVE REGISTERS
007D 635 DSBINT #IPL$ POWER,R2 ;SYNCHRONIZE WITH SET TIME AND RECAL
50 00000000'GF 7D 0083 636 MOVQ G^EXES$GQ_SYSTIME,R0 ;GET CURRENT SYSTEM TIME
50 00000000'GF C2 008A 637 SUBL2 G^EXES$GQ_TODCBASE,R0 ;COMPUTE DELTA SINCE LAST
51 00000004'GF D9 0091 638 SBWC G^EXES$GQ_TODCBASE+4,R1 ; SET TIME
00000000'GF DD 0098 639 PUSHL G^EXES$GL_TODR ;ORIGINAL "TODR" VALUE CORRESPONDING
009E 640 ; TO EXES$GQ TODCBASE VALUE
009E 641 ENBINT R2 ;RESTORE IPC LEVEL
00A1 642 :
00A1 643 : HAVE 100 NS. TICKS SINCE EXES$GL_TODR IN R0,R1.
00A1 644 : SCALE 100 NS. TICKS TO 10 MS. TICKS AND ADD TO EXES$GL_TODR.
00A1 645 : THE NEED TO OBTAIN AN UNSIGNED LONGWORD RESULT MAKES THIS NON-TRIVIAL.
00A1 646 :
```


51	52	51	000186A0	8F	D4	00A1	647	:	FACT:		
51	50	50	00030D40	8F	7B	00A1	648	:		$((A * 2^{32} + B) / C) \bmod 2^{32}$	
					7B	00A1	649	:	=		
						00A1	650	:		$((A \bmod C) * 2^{32} + B) / C \bmod 2^{32}$	
						00A1	651	:			
						00A1	652	:	CLRL	R2	:MAKE R1/R2 QUADWORD
						00A3	653	:	EDIV	#100000,R1,R2,R1	:HIGH DIGIT MOD 100000
						00AC	654	:	EDIV	#100000*2,R0,R0,R1	:SCALE TO 20 MS UNITS
						00B5	655	:			: R1 IS 0 OR 100000
						00B5	656	:	MNEGL	R1,R1	:SET CARRY IFF R1 .NE. 0
						00B8	657	:	ADWC	R0,R0	:SCALE TO 10 MS. UNITS, ADD
						00BB	658	:			: OPTIONAL TICK
						00BB	659	:	ADDL2	(SP)+,R0	:ADD BASE TODR VALUE
						00BE	660	:	POPR	#^M<R1,R2>	:RESTORE REGISTERS
						00C0	662	:			
						00C0	663	:			
						00C1	664	:	RSB		

```
00C1 666 .SBTTL EXES$WRITE_TODR (P) - WRITES TIME-OF-DAY CLOCK
00C1 667 :+
00C1 668 : WRITES THE TIME-OF-DAY CLOCK, SINCE IT MAY BE ACCESSED IN
00C1 669 : DIFFERENT WAYS: AS AN INTERNAL PROCESSOR REGISTER, AS PART
00C1 670 : OF THE CONSOLE, OR BY READING AN ADDRESS IN I/O SPACE. IT
00C1 671 : MAY ALSO BE IN DIFFERENT FORMATS AND HAVE TO BE CONVERTED.
00C1 672 :
00C1 673 : INPUTS:
00C1 674 :
00C1 675 : RO - CONTAINS VALUE TO BE WRITTEN INTO TODR
00C1 676 :
00C1 677 : OUTPUTS:
00C1 678 :
00C1 679 : NEW TIME VALUE WRITTEN INTO TODR.
00C1 680 : ALL REGISTERS PRESERVED.
00C1 681 :-
00C1 682
00C1 683 EXES$WRITEP_TODR:: ; SUBROUTINE ENTRY
00C1 684
00C1 685 : NAUTILUS PROCESSOR NEEDS TO USE A SEPARATE ROUTINE TO ACCESS PHYSICAL TODR
00C1 686 : REGISTER IN THE CONSOLE PROCESSOR FOR TWO REASONS. FIRST, THE PHYSICAL
00C1 687 : TODR HAS ONE SECOND RESOLUTION INSTEAD OF 10 MSEC RESOLUTION. SECOND, A
00C1 688 : REFERENCE TO THE PHYSICAL TODR IS A VERY SLOW, NON-INTERRUPTIBLE ACTION.
00C1 689 : NON-PHYSICAL NAUTILUS TODR REFERENCES WILL USE THE EXES$WRITE_TODR ENTRY
00C1 690 : WHICH WILL FABRICATE A NEW QUADWORD SYSTEM TIME.
00C1 691
00C1 692 : NOT NAUTILUS - FALL THROUGH TO WRITE_TODR
00C1 693
00C1 694 EXES$WRITE_TODR:: ; SUBROUTINE ENTRY
00C1 695
00C1 696
00C1 700
00C1 701
00C1 705
00C1 706
00C1 710
00C1 711
00C1 715
00C1 716
00C1 718 ;DO NOTHING TO SIMULATE WRITING A
00C1 719 ; NON-EXISTENT REGISTER
00C1 721
05 00C1 722 RSB
```



```
00C2 724 .SBTTL EXE$REGSAVE - SAVE CPU-SPECIFIC IPR'S
00C2 725 :+
00C2 726 : EXE$REGSAVE - CALLED BY POWERFAIL TO SAVE CPU-SPECIFIC IPR'S ON
00C2 727 : THE STACK
00C2 728 :
00C2 729 : INPUTS: NONE
00C2 730 :
00C2 731 : OUTPUTS:
00C2 732 :
00C2 733 : RO DESTROYED
00C2 734 : OTHER GENERAL REGISTERS PRESERVED
00C2 735 : IPR'S SAVED ON THE STACK AS FOLLOWS:
00C2 736 :
00C2 737 : 11/780: 11/750: 11/730: 11/790: uVAX I:
00C2 738 :
00C2 739 : 0(SP) PME PME PME ACCS (none)
00C2 740 : 4(SP) SBIMT TBDR CSWP
00C2 741 : 8(SP) CADR PME
00C2 742 :
00C2 743 :-
00C2 744 :
00C2 745 : .ENABL LSB
00C2 746 :
00C2 747 EXE$REGSAVE:: ;SUBROUTINE ENTRY
00C2 779 :
05 00C2 781 : RSB
00C3 783 :
00C3 784 : .DSABL LSB
```



```
00C3 786 .SBTTL EXE$REGRESTOR - RESTORE CPU-SPECIFIC IPR'S
00C3 787 :+
00C3 788 : EXE$REGRESTOR - CALLED BY POWERFAIL RECOVERY TO RESTORE CPU-SPECIFIC
00C3 789 : IPR'S FROM THE STACK.
00C3 790 :
00C3 791 : INPUTS:
00C3 792 :
00C3 793 : R6 - TOP OF STACK
00C3 794 : STACK SET UP AS DEFINED IN OUTPUTS OF EXE$REGSAVE.
00C3 795 :
00C3 796 : OUTPUTS:
00C3 797 :
00C3 798 : R0 DESTROYED
00C3 799 : OTHER GENERAL REGISTERS PRESERVED
00C3 800 : CPU-SPECIFIC IPR'S RESTORED FROM STACK
00C3 801 : R6 - ADDRESS OF 1ST CPU-INDEPENDENT SAVED IPR
00C3 802 :
00C3 803 :-
00C3 804 :
00C3 805 .ENABL LSB
00C3 806
05 00C3 807 EXE$REGRESTOR:: ;SUBROUTINE ENTRY
00C3 841 RSB ;NOTHING TO DO, RETURN
00C4 843
00C4 844 .DSABL LSB
```



```
00C4 846 .SBTTL EXESINIPROCREG - CPU-DEPENDENT INITIALIZATION OF IPR'S
00C4 847 :+
00C4 848 : EXESINIPROCREG - PERFORM INITIALIZATION OF INTERVAL TIMER AND
00C4 849 : CPU-DEPENDENT REGISTERS. CALLED FROM INIT AND POWERFAIL.
00C4 850 :
00C4 851 : INPUTS:
00C4 852 :
00C4 853 : NONE
00C4 854 :
00C4 855 : OUTPUTS:
00C4 856 :
00C4 857 : NONE
00C4 858 :-
00C4 859
00C4 860 EXESINIPROCREG::
00C4 861 ; INIT PROCESSOR REGISTERS
1A 00000000'00' E1 00C4 949 BBC S^#EXESV_CRDENABL,-
7E 50 7D 00C6 950 @#EXESGL_FLAGS,20$ ; If clear, ignore CRD errors.
51 00000000'GF DE 00CC 951 MOVQ R0,-(SP) ; Save working registers.
50 81 DO 00CF 952 MOVAL G^EXESAL_MEMCSRS,R1 ; Get address of memory CSR array.
00 B1 01 A8 00D6 953 MOVL (R1)+,R0 ; Get count of memory CSRs.
51 04 C0 00DD 954 10$: BISW #1,@(R1) ; Reenable parity errors.
F6 50 F5 00E0 955 ADDL #4,R1 ; Get VA of next memory controller CSR.
50 8E 7D 00E3 956 SOBGTR R0,10$ ; Loop through all CSRs.
18 00000040 8F DA 00E6 957 MOVQ (SP)+,R0 ; Restore working registers.
05 00ED 958 00E6 959 20$: MTPR #^X40,S^#PR$_ICCS ; Enable 10ms interrupts.
00EE 960 RSB ; Return.
00EE 962
```



```

00EE 985      .SBTTL  SYSLSCLR SBIA
00EE 986      :++
00EE 987      : SYSLSCLR SBIA  - ON 11/790, CLEAR SBIA ERROR REGISTERS
00EE 988      :               - ON 11/780, 11/750, 11/730, AND MICRO-VAX I, THIS IS A NOP
00EE 989      :
00EE 990      : THIS ROUTINE IS CALLED TO CLEAR OUT SBIA ERROR BITS AFTER A MACHINE CHECK
00EE 991      : OCCURS (WHEN MACHINE CHECK IS HANDLED LOCALLY).
00EE 992      :
00EE 993      : THIS ROUTINE SHOULD BE CALLED AT IPL 31.
00EE 994      :
00EE 995      : INPUTS:
00EE 996      :     ABUS_TYPE      - AN ARRAY TYPE CODES; IDENTIFIES EACH ADAPTER ON THE
00EE 997      :                   ABUS.
00EE 998      :     ABUS_VA       - AN ARRAY OF ADAPTER SPACE VA'S FOR EACH ADAPTER
00EE 999      :                   ON THE ABUS.
00EE 1000     :
00EE 1001     : OUTPUTS:
00EE 1002     :     SBI ERROR BITS ARE CLEARED FOR EACH SBIA ON THE ABUS.
00EE 1003     :     ALL REGISTERS PRESERVED.
00EE 1004     :++
00EE 1005     SYSLSCLR SBIA::
05 00EE 1023     RSB                                ; AND RETURN

```



```
00EF 1025 .SBTTL EXESTEST_CSR
00EF 1026
00EF 1028 :+ EXESTEST_CSR - TEST A QBUS CONTROLLER CSR FOR EXISTENCE
00EF 1033
00EF 1034 : THIS TEST IS CPU-DEPENDENT. THE FOLLOWING CPU'S ARE SUPPORTED:
00EF 1035
00EF 1036 11/780 -TEST CSR AND CHECK RESULT IN THE UBA STATUS REGISTER.
00EF 1037 11/750 -NON-EXISTENT CSR IS REPORTED VIA MACHINE CHECK AS A
00EF 1038 NON-EXISTENT MEMORY REFERENCE. CONNECT A TEMPORARY
00EF 1039 MACHINE CHECK HANDLER, TEST THE CSR, AND RESTORE THE
00EF 1040 ORIGINAL MACHINE CHECK HANDLER.
00EF 1041 11/730 -ACTION IS THE SAME AS FOR THE 11/750.
00EF 1042 11/790 -ACTION IS THE SAME AS FOR THE 11/780.
00EF 1043 MICRO-VAX I -ACTION IS SAME AS FOR THE 11/750.
00EF 1044
00EF 1045 : THIS SUBROUTINE SHOULD BE CALLED VIA BRANCH OR JUMP TO SUBROUTINE AT IPL 31.
00EF 1046
00EF 1047 : INPUTS:
00EF 1048
00EF 1049 R0 = CSR ADDRESS
00EF 1050 R6 = ADAPTER CONFIGURATION REGISTER ADDRESS
00EF 1051
00EF 1052 : OUTPUTS:
00EF 1053
00EF 1054 R0 LOW BIT SET/CLEAR FOR EXISTENT/NONEX CSR
00EF 1055 OTHER REGISTERS PRESERVED.
00EF 1056 :-
00EF 1057
00EF 1058 .ENABL LSB
00EF 1059
00EF 1060 EXESTEST_CSR:: :SUBROUTINE ENTRY
00EF 1061
00EF 1062 06 BB PUSHR #^M<R1,R2> :SAVE REGISTERS
00F1 1063
00F1 1128 00000002 NEX = 2 :MACHINE-CHECK CODE FOR NON-EXIST MEM
00F1 1130
00F1 1141
00F1 1142 51 00000000'GF D0 10$: MOVL G^EXE$GL_SCB,R1 :GET SCB ADDRESS
00F8 1143 04 A1 DD PUSHL 4(R1) :SAVE CURRENT MCHECK HANDLER ADDR
00FB 1144 52 SE DD MOVL SP,R2 :MARK CURRENT STACK POSITION
00FE 1145 04 A1 0C'AF DE MOVAL B^MCHK_HANDLER,4(R1) :CONNECT TEMP MCHECK HANDLER
0103 1146 60 B5 TSTW (R0) :ATTEMPT TO READ CSR
0105 1147 50 01 9A OK: MOVZBL #SS$ _NORMAL,R0 :IF NO MCHECK, SET STATUS TO
0108 1148 : SUCCESS
0108 1149 13 11 BRB TEST_DONE :JOIN COMMON EXIT
010A 1150
010A 1151 :
010A 1152 : TEMPORARY CSR TEST MACHINE CHECK HANDLER
010A 1153 :
010A 1154
010A 1155 .ALIGN LONG :REQ'D MACHINE CHECK ALIGNMENT
010C 1156 MCHK_HANDLER:
010C 1157
010C 1161
010C 1165
010C 1167 26 OF DA MTPR #^XF,#PRUV1$_MCSR :CLEAR MACHINE-CHECK STATE
010F 1169
```

```
50 04 AE D0 010F 1170
SE 52 D0 010F 1179
02 50 D1 0113 1181      MOVL 4(SP),R0
EA 12 0116 1182      MOVL R2,SP
                                CMPL R0,#NEX
                                BNEQ OK
                                ;GET MACHINE-CHECK CODE
                                ;CLEAN MCHECK FRAME FROM STACK
                                ;IS MCHK CODE = NON-EXISTENT MEM?
                                ;BR IF NO, MUST HAVE BEEN PARITY ERR

                                ;
50 D4 011B 1186      NONEX_DEV:
                                CLRL R0
                                ;SET STATUS TO FAILURE
04 A1 8ED0 011B 1188 TEST_DONE:
                                POPL 4(R1)
                                ;RESTORE SYSTEM MCHECK HANDLER
06 BA 011D 1190 TEST_DONE_2:
05 05 0121 1191      POPR #^M<R1,R2>
                                RSB
                                ;RESTORE REGISTERS
                                ;RETURN RESULT TO CALLER
0124 1194      .DISABLE LSB
                                0124 1195
```



```

0124 1197      .SBTTL ADPLINK - LINK ADAPTER CONTROL BLOCK INTO ADP LIST
0124 1198      ;+
0124 1199      ; ADPLINK LINKS THE ADAPTER CONTROL BLOCK TO THE END OF THE ADP LIST
0124 1200      ;
0124 1201      ; INPUT:
0124 1202      ; R2 - ADDRESS OF NEW ADP
0124 1203      ; OUTPUTS:
0124 1204      ; ADP IS LINK TO THE END OF THE ADPLIST LOCATED BY IOC$GL_ADPLIST.
0124 1205      ; R0,R1 destroyed.
0124 1206      ; -
0124 1207      ;
0124 1208      ADPLINK::
50  FFFFFFFC'9F  9E 0124 1209      MOVAB  @#<IOC$GL_ADPLIST-ADP$L_LINK>,R0
      51  04 A0  D0 012B 1210      ; START OF LIST
      50  51  D0 012B 1211 10$:      MOVL  ADP$L_LINK(R0),R1      ; FLINK TO FIRST ENTRY
      04 A0  52  D0 012F 1212      BEQL  20$      ; AT END
      05  11  D0 0131 1213      MOVL  R1,R0      ; TRY AGAIN
      05  11  D0 0134 1214      BRB   10$      ;
      05  11  D0 0136 1215 20$:      MOVL  R2,ADP$L_LINK(R0)      ; CHAIN NEW ADP TO END OF LIST
      05  11  D0 013A 1216      RSB      ; AND RETURN
013B 1217
013B 1218      .END

```

ERRSUBV1
Symbol table

- ERROR SUBROUTINES FOR MICRO-VAX I J 14

16-SEP-1984 01:07:13 VAX/VMS Macro V04-00
13-SEP-1984 15:49:22 [SYSLOA.SRC]ERRSUB.MAR;5

Page 19
(15)

ADPSL_CSR	= 00000000		
ADPSL_LINK	= 00000004		
ADPSW-ADPTYPE	= 0000000E		
ADPLINK	00000124	RG	03
ADP_TBL_DWN	00000031	R	03
ADP_TBL_UP	00000049	R	03
C750_LIKE	= 00000001		
C780_LIKE	= 00000000		
CISSHUTDOWN	*****	X	03
CPU TYPE	= 00000007		
EMBSL_CR_CODE	= 000000F4		
EMBSL_CR_CPUREG	= 00000094		
EXESAC MEMCSRS	*****	X	03
EXESDUMPCPUREG	00000061	RG	03
EXESEXTRA1	00000000	RG	01
EXESEXTRA10	00000000	RG	01
EXESEXTRA2	00000000	RG	01
EXESEXTRA3	00000000	RG	01
EXESEXTRA4	00000000	RG	01
EXESEXTRA5	00000000	RG	01
EXESEXTRA6	00000000	RG	01
EXESEXTRA7	00000000	RG	01
EXESEXTRA8	00000000	RG	01
EXESEXTRA9	00000000	RG	01
EXESGL_FLAGS	*****	X	03
EXESGL_SCB	*****	X	03
EXESGL_TODR	*****	X	03
EXESGQ_SYSTIME	*****	X	03
EXESGQ_TODCBASE	*****	X	03
EXESINIBOOTADP	00000000	RG	03
EXESINIPROCREG	000000C4	RG	03
EXESREADP TODR	0000007B	RG	03
EXESREAD TODR	0000007B	RG	03
EXESREGRESTOR	000000C3	RG	03
EXESREGSAVE	000000C2	RG	03
EXESSHUTDOWNADP	0000000A	RG	03
EXESSTARTUPADP	00000002	RG	03
EXESTEST_CSR	000000EF	RG	03
EXESV CRDENABL	*****	X	03
EXESWRITEP TODR	000000C1	RG	03
EXESWRITE TODR	000000C1	RG	03
INI UBADP	00000001	R	03
IOCSGL_ADPLIST	*****	X	03
IPLS POWER	= 0000001F		
MASINITIAL	*****	X	03
MBASINITIAL	*****	X	03
MCHK_HANDLER	0000010C	R	03
NEX	= 00000002		
NONEX_DEV	0000011B	R	03
OK	00000105	R	03
PRS_ICCS	= 00000018		
PRS-IPL	= 00000012		
PRS_SID_TYP730	= 00000003		
PRS_SID_TYP750	= 00000002		
PRS_SID_TYP780	= 00000001		
PRS_SID_TYP790	= 00000004		
PRS_SID_TYPUV1	= 00000007		

PRUV1\$ MCESR
SS\$ NORMAL
SYS\$CLRSBIA
TEST_DONE
TEST_DONE 2
UBASINITIAL

= 00000026		
= 00000001		
000000EE	RG	03
0000011D	R	03
00000121	R	03
*****	X	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
. BLANK .	00000001 (1.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	0000013B (315.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.06	00:00:01.27
Command processing	141	00:00:00.44	00:00:04.50
Pass 1	348	00:00:07.27	00:00:27.11
Symbol table sort	0	00:00:01.07	00:00:04.34
Pass 2	127	00:00:02.00	00:00:08.03
Symbol table output	9	00:00:00.05	00:00:00.05
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	663	00:00:10.91	00:00:45.32

The working set limit was 1650 pages.
70296 bytes (138 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 1049 non-local and 12 local symbols.
1222 source lines were read in Pass 1, producing 16 object records in Pass 2.
22 pages of virtual memory were used to define 21 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	12
-\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	6
TOTALS (all libraries)	18

1130 GETs were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:ERRSUBUV1/OBJ=OBJ\$:ERRSUBUV1 MSRC\$:CPUSWUV1/UPDATE=(ENH\$:CPUSWUV1)+MSRC\$:ERRSUB/UPDATE=(ENH\$:ERRSUB)+EXECMLS/LIB

0395

AH - BT13A - SE

DIGITAL EQUIPMENT CORPORATION

ERRSUB790
LISERRSUBUV1
LIS

ERRSUB750
LIS