

A large, stylized letter 'A' is formed using the characters 'S' and 'Y'. The left and right vertical strokes are composed of 'S' characters, while the central vertical stroke and the diagonal strokes are composed of 'Y' characters. The 'A' is symmetrical and has a bold, blocky appearance.

SYSSETIME
Table of contents

- SYSTEM SERVICE TO SET CURRENT SYSTEM T 16-SEP-1984 02:30:50 VAX/VMS Macro V04-00

Page 0

(1)	90	DECLARATIONS
(1)	153	SET TIME
(1)	233	COMPUTE TIME-OF-DAY CLOCK VALUE
(1)	263	CVT TIME
(1)	293	RECALIBRATE
(1)	412	SET TODR
(1)	497	SAVE TIME

SYS
V04


```
0000 1 .TITLE SYSSETIME - SYSTEM SERVICE TO SET CURRENT SYSTEM TIME
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27
0000 28 LEN KAWELL 14-FEB-79
0000 29
0000 30 MODIFICATION HISTORY:
0000 31
0000 32 V03-012 SRB0125 Steve Beckhardt 6-Jul-1984
0000 33 Zero distributed deadlock detection timestamp
0000 34 expirations when a SET TIME is performed. The approach
0000 35 taken in change SRB0117 may cause false deadlocks.
0000 36
0000 37 V03-011 SRB0117 Steve Beckhardt 9-Apr-1984
0000 38 Recompute distributed deadlock detection timestamp
0000 39 expirations when a SET TIME is performed.
0000 40
0000 41 V03-010 CWH3010 CW Hobbs 6-Dec-1983
0000 42 Change references to EXES$READ TODR and EXES$WRITE TODR
0000 43 to use the physical register routines EXES$READP TODR and
0000 44 EXES$WRITEP TODR. This is so that the non-p routines for
0000 45 Nautilus can fabricate the TODR from the quadword system
0000 46
0000 47 V03-009 KDM0086 Kathleen D. Morse 13-Oct-1983
0000 48 Rearrange order in which PR$ TODR and EXES$GQ SYSTIME
0000 49 are set so that the same code works for MicroVAX I and
0000 50 for all other VAX systems.
0000 51
0000 52 V03-008 CWH3008 CW Hobbs 10-Sep-1983
0000 53 If parameter map in BOOTCB is null, then return success
0000 54 without writing time out. If running standalone and
0000 55 system disk has been removed, then map will be null.
0000 56
0000 57 V03-007 DWT0125 David W. Thiel 23-Aug-1983
```



```
0000 58 : Add internal entry point EXESSETIME_INT that avoids writing
0000 59 : to the system disk. Restructure code to support this new
0000 60 : entry point.
0000 61 :
0000 62 : V03-006 KDM0056 Kathleen D. Morse 12-Jul-1983
0000 63 : Change references to IPR TODR to use cpu-dependent
0000 64 : routines: EXESREAD_TODR and EXESWRITE_TODR.
0000 65 :
0000 66 : V03-005 ROW0121 Ralph O. Weber 24-AUG-1982
0000 67 : Change JSB to EXESBOOTCB_CHK so that it is performed at
0000 68 : IPL$ SYNCH. A revision to EXESBOOTCB_CHK requires that the
0000 69 : routine be called at or above this IPL to prevent modification
0000 70 : of volatile portions of the SYS.EXE window control block which
0000 71 : is a part of the area checksummed.
0000 72 :
0000 73 : V03-004 KDM0002 Kathleen D. Morse 28-Jun-1982
0000 74 : Added $IODEF and $$SDEF.
0000 75 :
0000 76 : V03-003 PHL0101 Peter H. Lipman 20-Jun-1982
0000 77 : $QIOW now synchronizes the EFN and IOSB parameters
0000 78 : correctly. Eliminate the synchronization code here.
0000 79 :
0000 80 :
0000 81 : SYSTEM SERVICE TO SET THE CURRENT SYSTEM TIME
0000 82 :
0000 83 : DOES ANYBODY REALLY KNOW WHAT TIME IT IS?
0000 84 : DOES ANYBODY REALLY CARE?
0000 85 : IF SO I CAN'T IMAGINE WHY.
0000 86 : WE'VE ALL GOT TIME ENOUGH TO DIE.
0000 87 : CHICAGO 1971
0000 88 :
```



```
0000 90      .SBTTL  DECLARATIONS
0000 91      :
0000 92      : MACRO LIBRARY CALLS
0000 93      :
0000 94      :
0000 95      $BOODEF      ;DEFINE BOOT CONTROL BLOCK OFFSETS
0000 96      $IODEF      ;DEFINE I/O FUNCTION CODES
0000 97      $IPLDEF     ;DEFINE INTERRUPT PRIORITY LEVELS
0000 98      $PCBDEF     ;DEFINE PCB OFFSETS
0000 99      $PRDEF      ;DEFINE PROCESSOR REGISTERS
0000 100     $PRVDEF     ;DEFINE PRIVILEGES
0000 101     $SSDEF      ;DEFINE SYSTEM STATUS CODES
0000 102     $TQDEF     ;DEFINE TIMER QUEUE ENTRIES
0000 103     $VADEF      ;DEFINE VIRTUAL ADDRESSES
0000 104     :
0000 105     : LOCAL SYMBOLS
0000 106     :
0000 107     : ARGUMENT LIST OFFSET DEFINITIONS
0000 108     :
0000 109     :
0000 110     :
00000004 0000 111 TIMADR=4      ;NEW TIME
0000 112     :
0000 113     : OWN STORAGE
0000 114     :
0000 115     :
00000000 0000 116     .PSECT  YEXEPAGED      ;PAGED DATA AND CODE
0000 117     :
0000 118     :
0000 119     : BASE TIME FOR TIME-OF-DAY PROCESSOR REGISTER (TODR). THE TODR CAN'T
0000 120     : CONTAIN THE ENTIRE SYSTEM TIME, SO IT IS DEFINED TO CONTAIN A VALUE
0000 121     : RELATIVE TO THE BASE TIME. THE BASE TIME IS DEFINED AS:
0000 122     :
0000 123     : 01-JAN-CURRENT_YEAR 00:00:00.00
0000 124     :
0000 125     : TO ALLOW CONVERSION OF THE TODR VALUE TO SYSTEM TIME FORMAT,
0000 126     : THE TODR AND ITS CORRESPONDING SYSTEM TIME ARE STORED IN THE SYSTEM
0000 127     : IMAGE FILE (SYS$SYSTEM:SYS.EXE). BECAUSE ALL TODR'S HAVE THE SAME
0000 128     : BASE, A DISK THAT IS BOOTED ON ONE SYSTEM CAN BE TRANSPORTED TO
0000 129     : ANOTHER SYSTEM AND THE TIME WILL BE SET CORRECTLY.
0000 130     :
0000 131     BASETIME:      ;BASE TIME FOR TIME-OF-DAY REGISTER
0000 132     .ASCII  /01-JAN-/
0000 133     BASEYEAR = .-BASETIME      ;(OFFSET TO YEAR)
0000 134     .ASCII  /XXXX 00:00:00.00/
0000 135     BASETIMESZ = .-BASETIME      ;BASE TIME STRING SIZE
0000 136     :
0000 137     : TO DETERMINE WHETHER OR NOT THE PROCESSOR TIME-OF-YEAR CLOCK (PR$ TODR) HAS
0000 138     : ROLLED OVER THE YEAR, WE MUST KNOW HOW LONG THE BASE YEAR WAS. TO CALCULATE
0000 139     : THAT TIME, WE NEED A DEFINITION OF THE END OF THE YEAR. USING THE ABOVE
0000 140     : DEFINITION OF THE FIRST OF THE YEAR AND THE FOLLOWING END OF YEAR, WE CAN
0000 141     : USE THE APPROPRIATE SYSTEM SERVICES TO CALCULATE THAT SPAN OF TIME - ALWAYS
0000 142     : PROPERLY ALLOWING FOR LEAP YEARS.
0000 143     :
0000 144     ENDTIME:
0000 145     .ASCII  /31-DEC-XXXX 23:59:59.99/      ; END OF YEAR (- 10ms)
```

2D 4E 41 4A 2D 31 30
00000007
30 3A 30 30 3A 30 30 20 58 58 58 58
30 30 2E 30
00000017

20 58 58 58 58 2D 43 45 44 2D 31 33

SYSSETIME
V04-000

- SYSTEM SERVICE TO SET CURRENT SYSTEM T 16-SEP-1984 02:30:50 VAX/VMS Macro V04-00
DECLARATIONS 5-SEP-1984 03:56:59 [SYS.SRC]SYSSETIME.MAR;1

Page 4
(1)

```
39 39 2E 39 35 3A 39 35 3A 33 32 0023
                                002E 146
                                002E 147 :
                                002E 148 : SYSTEM DISK LOGICAL NAME
                                002E 149 :
                                002E 150 SYSDISK: ;SYSTEM DISK LOGICAL NAME
59 53 24 53 59 53 00000036'010E0000' 002E 151 .ASCID /SYS$SYSDEVICE/
                                002E
                                003C 45 43 49 56 45 44 53
```

SYS
Pse

PSE

\$AB
YEX

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
649
The
560
31

Mac

-\$2
-\$2
TOT
136
The
MAC


```
0043 153 .SBTTL SET TIME
0043 154 :+
0043 155 : EXESSETIME - SET CURRENT SYSTEM TIME
0043 156 :
0043 157 : THIS SERVICE PROVIDES THE CAPABILITY TO RESET THE CURRENT SYSTEM TIME.
0043 158 :
0043 159 : EXESSETIME_INT - INTERNAL ENTRY POINT THAT RECOMPUTES INTERNAL TIME
0043 160 : AND WRITES NOTING TO THE DISK.
0043 161 :
0043 162 : INPUTS:
0043 163 :
0043 164 : TIMADR(AP) = ADDRESS OF NEW TIME VALUE. ZERO FOR ADDRESS OR
0043 165 : TIME VALUE CAUSES THE PROCESSOR'S TIME-OF-DAY CLOCK
0043 166 : TO BE USED TO RECALIBRATE THE CURRENT SYSTEM TIME.
0043 167 :
0043 168 : R4 = ADDRESS OF CURRENT PROCESS PCB.
0043 169 :
0043 170 : OUTPUTS:
0043 171 :
0043 172 : R0 LOW BIT CLEAR INDICATES FAILURE TO SET SYSTEM TIME.
0043 173 :
0043 174 : R0 = SS$ ACCVIO - TIME VALUE CANNOT BE READ BY CALLING
0043 175 : ACCESS MODE.
0043 176 :
0043 177 : R0 = SS$ IVTIME - A NEGATIVE TIME VALUE WAS SPECIFIED,
0043 178 : OR AN INVALID PROCESSOR CLOCK VALUE WAS FOUND.
0043 179 :
0043 180 : R0 = SS$ NOIOCHAN - NO I/O CHANNEL IS AVAILABLE FOR
0043 181 : USE BY THE SERVICE.
0043 182 :
0043 183 : R0 = SS$ NOPRIV - THE CALLING PROCESS DOES NOT HAVE
0043 184 : OPER OR LOG_IO PRIVILEGE.
0043 185 :
0043 186 : R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
0043 187 :
0043 188 : R0 = SS$_NORMAL - NORMAL COMPLETION, SYSTEM TIME SET.
0043 189 :
0043 190 :
0043 191 : CURRENT SYSTEM TIME (EXESGO SYSTIME) IS SET TO SPECIFIED
0043 192 : TIME. PROCESSOR'S TIME-OF-DAY CLOCK IS UPDATED TO NEW TIME.
0043 193 :
0043 194 : EXESSETIME ONLY -- NEW TIME IN SYSTEM TIME FORMAT AND TIME-OF-DAY
0043 195 : CLOCK FORMAT IS WRITTEN TO SYSTEM IMAGE FILE
0043 196 : (SYS$SYSTEM:SYS.EXE).
0043 197 : -
0043 198 : .ENABLE LSB
0043 199 : EXESSETIME_INT::
0043 200 : .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; ENTRY MASK
51 04 AC DO 0045 201 : MOVL TIMADR(AP),R1 ;GET ADDR OF NEW TIME ARGUMENT
0049 202 : PUSHL #1 ;STORE FLAG
22 11 004B 203 : BRB 20$
004D 204 :
004D 205 : .ENTRY EXESSETIME,- ;SET CURRENT SYSTEM TIME
004F 206 : ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ;(ENTRY MASK)
004F 207 :
50 24 3C 004F 208 : MOVZWL S^#SS$ NOPRIV,R0 ;ASSUME NO PRIVILEGE
0052 209 : IFNPRIV OPER,90$ ;BR IF NO OPER PRIVILEGE
```



```

51 50 0C 3C 0058 210 IFNPRIV LOG IO,90$ ;BR IF NO LOG IO PRIVILEGE
    04 AC D0 005E 211 MOVZWL S^#SS$ ACCVIO,R0 ;ASSUME ACCESS VIOLATION
    06 13 0061 212 MOVL TIMADR(TAP),R1 ;GET ADDR OF NEW TIME ARG
    7E D4 0065 213 BEQL 10$ ;BR IF NONE
    56 7E 0067 214 IFNORD #8,(R1),90$ ;CAN TIME BE READ?
    58 7E 006D 215 10$: CLRL -(SP) ;STORE FLAG ON STACK
    5E 18 006F 216 20$: MOVAQ -(SP),R6 ;ALLOCATE BINARY TIME BUFFER
    57 5E 0072 217 MOVAQ -(SP),R8 ;ALLOCATE ASCII TIME BUFFER DESCRIPTOR
    04 A8 57 D0 0075 218 SUBL2 #<BAS$TIMESZ+3>8^C3,SP ;ALLOCATE ASCII TIME BUFFER
    68 17 D0 0078 219 MOVL SP,R7 ;HOLD POINTER TO IT
    51 D5 007B 220 MOVL R7,4(R8) ;CREATE DESCR
    5C 13 007F 221 MOVL #BAS$TIMESZ,(R8) ;LENGTH FIELD
    0184 8F 3C 0082 222 TSTL R1 ;IS A TIME SPECIFIED?
    5A 61 7D 0084 223 BEQL RECAL ;BR IF NO - RECALIBRATE
    03 14 0086 224 MOVZWL #SS$_IVTIME,R0 ;ASSUME INVALID TIME
    50 13 008B 225 MOVQ (R1),R10 ;GET TIME - R10,R11
    0090 226 BGTR COMPUTE_TODR ;BR IF VALID TIME
    0092 227 BEQL RECAL ;BR IF ZERO - RECALIBRATE
    04 0092 228 90$: RET ;RETURN WITH FAILURE
    0093 229
    0093 230
    0093 231 .DISABLE LSB
```


				0093	233	.SBTTL	COMPUTE TIME-OF-DAY CLOCK VALUE	
				0093	234	::++		
				0093	235			
				0093	236	COMPUT_TODR - COMPUTE NEW TIME-OF-DAY CLOCK VALUE.		
				0093	237			
				0093	238	COMPUTE NEW TIME-OF-DAY CLOCK VALUE BY COMPUTING THE NEW		
				0093	239	TIME AFTER THE BASE TIME AND SCALING TO 10 MSEC UNITS.		
				0093	240			
				0093	241	INPUTS:		
				0093	242			
				0093	243	R10,R11 = NEW SYSTEM TIME.		
				0093	244			
				0093	245	OUTPUTS:		
				0093	246			
				0093	247	R0 = NEW TIME-OF-DAY CLOCK VALUE.		
				0093	248			
				0093	249	--		
				0093	250	COMPUTE_TODR:		:COMPUTE TIME-OF-DAY CLOCK VALUE
	52	FF69	CF	9E	0093	MOVAB	BASETIME,R2	:GRAB CORRECT TIME STRING
			1E	10	0098	BSBB	CVT_TIME	:CONVERT THE TIME
		50	5A	7D	009A	MOVQ	R10,R0	:GET NEW TIME
		50	86	C2	009D	SUBL	(R6)+,R0	:COMPUTE YEAR'S DELTA
		51	66	D9	00A0	SBWC	(R6),R1	:IN 100 NANOSECOND UNITS
51	50	50	0185'CF	7B	00A3	EDIV	W^SCALE,R0,R0,R1	:SCALE TO 10 MS UNITS
		50	50	01	78	ASHL	#1,R0,R0	: (SINCE NO UNSIGNED EDIV)
	50	10000000	8F	C0	00AE	ADDL	#<1@28>,R0	:COMPUTE BIASED VALUE FOR TODR
					00B5			:VALUES LESS THAN BIAS ARE USED
					00B5			:TO DETECT CLOCK POWERFAIL
					00B5			:GO SET TIME-OF-DAY CLOCK
		00D5	31	00B5	261	BRW	SET_TODR	


```
00B8 263 .SBTTL CVT_TIME
00B8 264 :++
00B8 265 :
00B8 266 : CVT_TIME - CONVERT AN ASCII STRING TO A BINARY TIME
00B8 267 :
00B8 268 : INPUTS:
00B8 269 :
00B8 270 : R2 - POINTER TO TIME STRING
00B8 271 : R10, R11 - NEW TIME IN BINARY
00B8 272 :
00B8 273 : OUTPUTS:
00B8 274 :
00B8 275 : R6 - POINTER TO QUAD WORD TIME
00B8 276 : R7 - POINTER TO ASCII TIME
00B8 277 : R8 - POINTER TO DESCRIPTOR FOR ASCII TIME
00B8 278 : R9 - ASCII VALUE OF BASE YEAR
00B8 279 :
00B8 280 :--
00B8 281 :
00B8 282 CVT_TIME:
66 5A 7D 00B8 283 MOVQ R10,(R6) ;SET NEW TIME IN BUFFER
00B8 284 $ASCTIM_S TIMADR=(R6),- ;CONVERT NEW TIME TO ASCII
00B8 285 TIMBUF=(R8) ;
67 59 07 A7 D0 00CA 286 MOVL BASEYEAR(R7),R9 ;GET NEW TIME'S YEAR
07 62 17 28 00CE 287 MOVC #BASETIMESZ,(R2),(R7) ;MOVE TODR BASE TO ASCII BUFFER
07 A7 59 D0 00D2 288 MOVL R9,BASEYEAR(R7) ;MOVE NEW TIME'S YEAR TO BUFFER
00D6 289 $BINTIM_S TIMADR=(R6),- ;COMPUTE TODR BASE IN BINARY
00D6 290 TIMBUF=(R8) ;
05 00E1 291 RSB ;LEAVE
```



```
00E2 293 .SBTTL RECALIBRATE
00E2 294 :++
00E2 295
00E2 296 RECAL - RECALIBRATE USING TIME-OF-DAY CLOCK
00E2 297
00E2 298 THE TIME-OF-DAY CLOCK IS SPECIFIED TO BE MORE ACCURATE
00E2 299 THAN THE INTERVAL TIMER. IF THE SYSTEM HAS BEEN UP AWHILE, THIS
00E2 300 FUNCTION CAN BE USED TO RECALIBRATE THE SYSTEM TIME. THIS
00E2 301 FUNCTION IS ALSO USED BY SYSINIT TO SET THE SYSTEM TIME AT
00E2 302 EACH RE-BOOT.
00E2 303
00E2 304 IT IS IMPORTANT TO NOTE THAT THE HARDWARE CLOCK HAS A LIMIT
00E2 305 OF ABOUT 15 MONTHS. IT SHOULD NEVER BE ALLOWED TO REACH THIS
00E2 306 LIMIT; THUS ANY VMS SYSTEM SHOULD BE BOOTED, OR A SET TIME
00E2 307 REQUEST SHOULD BE DONE AT LEAST ONCE BETWEEN 1-JAN AND 1-APR
00E2 308 OF EACH YEAR.
00E2 309
00E2 310 DELTA-TOD = PR$ TODR - EX$GL TODR
00E2 311 DELTA-TIME = DELTA-TOD * SCALE
00E2 312 EX$GQ SYSTIME, EX$GQ TODCBASE = EX$GQ TODCBASE + DELTA-TIME
00E2 313 PR$_TODR = PR$_TODR - BASE-YEAR IF PR$_TODR > BASE_YEAR
00E2 314
00E2 315 INPUTS:
00E2 316
00E2 317 TIME-OF-DAY PROCESSOR CLOCK.
00E2 318
00E2 319 OUTPUTS:
00E2 320
00E2 321 R0 = NEW TIME-OF-DAY CLOCK VALUE
00E2 322 R10,R11 = NEW SYSTEM TIME COMPUTED FROM TIME-OF-DAY CLOCK
00E2 323
00E2 324 :--
00E2 325
00E2 326 RECAL: ;RECALIBRATE USING TOD CLOCK
00E2 327
00E2 328 CALCULATE HOW LONG THE BASE YEAR (DEFINED BY THE VALUE IN EX$GQ TODCBASE)
00E2 329 IS. THIS VALUE MAY BE NEEDED LATER TO CORRECT THE YEAR ROLLOVER IN THE
00E2 330 HARDWARE TIME OF YEAR CLOCK. THE CALCULATION IS DONE EARLY BECAUSE
00E2 331 1) IT DOES NOT DEPEND ON ANY CHANGING REGISTERS, AND 2) IT CAN BE DONE
00E2 332 AT ZERO IPL ($BINTIM IS PAGED).
00E2 333
5C 00000000'GF 7E 00E2 334 MOVAQ G*EX$GQ TODCBASE,AP ;GET A POINTER
52 5A 6C 7D 00E9 335 MOVQ (AP),R10 ;PICK UP CURRENT BASE TIME
FF27 CF 9E 00EC 336 MOVAB ENDTIME,R2 ;GRAB PROPER TIME STRING
C5 10 00F1 337 BSBB CVT TIME ;CONVERT THE TIME
5A 66 7D 00F3 338 MOVQ (R6),R10 ;HOLD YEAR-END TIME
67 FF05 CF 28 00F6 339 MOVQ #BASETIMESZ,BASETIME,(R7) ;MOVE BASE STRING TO BUFFER
07 A7 59 D0 00FC 340 MOVL R9,BASEYEAR(R7) ;MOVE NEW TIME'S YEAR TO BUFFER
0100 341 $BINTIM_S TIMADR=(R6),- ;COMPUTE TODR BASE IN BINARY
0100 342 TIMBUF=(R8)
5A 86 C2 010B 343 SUBL (R6)+,R10 ;COMPUTE BASE YEAR'S LENGTH(-10 ms)
5B 66 D9 010E 344 SBWC (R6),R11 ;IN 100 NANOSECOND UNITS
SA 89'AF C0 0111 345 ADDL B*SCALE2,R10 ;ADD IN THE MISSING 10 MS
5B 00 D8 0115 346 ADWC #0,R11
50 5A 7D 0118 347 MOVQ R10,R0 ;COPY
51 50 50 85'AF 7B 011B 348 EDIV B*SCALE,R0,R0,R1 ;SCALE TO 10 MS UNITS
52 50 01 78 0121 349 ASHL #1,R0,R2 ;(SINCE NO UNSIGNED EDIV)
```



```
0125 350 :
0125 351 : AT THIS POINT:
0125 352 : R10:R11 = THE NUMBER OF 10 NANOSECONDS IN THE BASE YEAR
0125 353 : R2 = THE NUMBER OF 10 MILLISECONDS IN THE BASE YEAR
0125 354 :
0125 355 : WE MUST CHECK THE PROCESSOR'S TIME OF YEAR CLOCK (PR$ TODR) TO SEE IF
0125 356 : IT HAS MORE THAN A YEAR'S WORTH OF TICKS IN IT. IF SO, WE SUBTRACT
0125 357 : OUT THE YEAR FROM THE CLOCK. IF NOT, WE DO NOT CHANGE THE CLOCK.
0125 358 : IN EITHER CASE, WE THEN UPDATE THE SYSTEM TIME FROM THE CORRECT
0125 359 : VALUE OF THE TODC.
0125 360 :
0125 361 LOCK_START: ;START OF LOCKED-DOWN CODE
0125 362 DSBINT W^LOCK_IPL ;DISABLE ALL INTERRUPTS AND
012D 363 ; LOCK CODE IN MEMORY
012D 364 JSB EXE$READP TODR ;GET TIME OF PHYSICAL DAY CLOCK VALUE
51 50 00000000'EF 16 012D 365 SUBL3 G^EXE$GL_TODR,R0,R1 ;GET TOD DELTA TIME (10 MS UNITS)
00000000'GF C3 0133 366 BGTRU 5$ ;ZERO OR POSITIVE IS OKAY
FF7C2A00 8F 11 1A 013B 367 CMPL R1,#-<24*60*60*100> ;IF NEGATIVE, THEN IF MORE THAN A DAY
5A 00 51 89'AF 51 D1 013D 368 BLSSU ERROR ;INVALID TIME IN PROCESSOR CLOCK
00 51 89'AF 7A 0146 369 EMUL B^SCALE2,R1,#0,R10 ;ELSE COVERT TO 100 NANOSECONDS
25 11 014C 370 BRB 20$ ;AND CONTINUE ON
53 52 10000000 8F C1 014E 371 5$: ADDL3 #<1028>,R2,R3 ;ADD IN THE BIAS TO THE YEAR
53 50 D1 0156 372 CMPL R0,R3 ;CORRECTION NEEDED FOR CLOCK?
03 1F 0159 373 BLSSU 10$ ;NO CORRECTION NEEDED
015B 374 :
015B 375 : THE CLOCK HAS ONE YEAR TOO MANY TICKS. WE MUST SUBTRACT OUT
015B 376 : THE BASE YEAR FROM THE PROCESSOR TODC. AFTER THE YEAR IS SUBTRACTED OUT,
015B 377 : THE DELTA WILL BE CORRECT.
015B 378 :
015B 379 :
50 52 C2 015B 380 SUBL R2,R0 ;SUBTRACT OUT A YEAR
015E 381 10$:
015E 382 :
015E 383 : AT THIS POINT:
015E 384 : R0 = CORRECT TODR + BIAS TO GO INTO PROCESSOR REGISTER
015E 385 : R1 = CORRECT TODR DELTA TO FIGURE TIME
015E 386 :
5A 51 1F 01 EF 015E 387 EXTZV #1,#31,R1,R10 ;DIVIDE BY 2
0163 388 ; (SINCE NO UNSIGNED EMUL)
5A 00 5A 85'AF 7A 0163 389 EMUL B^SCALE,R10,#0,R10 ;COMPUTE 100 NANOSECOND DELTA
07 51 E9 0169 390 BLBC R1,20$ ;BR IF NO CORRECTION NEEDED
5A 89'AF C0 016C 391 ADDL B^SCALE2,R10 ;ADD LAST 10MS UNIT
5B 00 D8 0170 392 ADWC #0,R11 ;TO QUADWORD
5A 6C C0 0173 393 20$: ADDL (AP),R10 ;FORM ABSOLUTE TIME
5B 04 AC D8 0176 394 ADWC 4(AP),R11 ;
19 11 017A 395 BRB SET_TODR1 ;SET THE TODR REGISTER
017C 396 :
017C 397 : THE VALUE IN THE PROCESSOR CLOCK WAS LESS THAN THE BIAS (2^28) OR
017C 398 : IT WAS GREATER THAN THE LAST VALUE STORED IN EXE$GL TODR - INDICATING
017C 399 : THAT THE SYSTEM DISK HASN'T BEEN BOOTED IN A VERY LONG TIME AND THUS
017C 400 : THE YEAR WITHIN DATE WILL NOT BE CORRECT.
017C 401 :
017C 402 ERROR: ENBINT
50 0184 8F 3C 017F 403 MOVZWL #SS$_IVTIME,R0 ;INVALID PROCESSOR CLOCK VALUE
04 0184 404 RET ;LEAVE AND REPORT IT
0185 405 :
0185 406 : SCALING VALUE FOR CALCULATIONS - MUST BE HERE SINCE IT IS REFERENCED
```


SYSSETIME
V04-000

- SYSTEM SERVICE TO SET CURRENT SYSTEM T 16-SEP-1984 02:30:50 VAX/VMS Macro V04-00
RECALIBRATE 5-SEP-1984 03:56:59 [SYS.SRC]SYSSETIME.MAR;1

Page 11
(1)

```
0185 407 : FROM WITHIN A LOCKED-DOWN SECTION OF CODE.
0185 408 :
00030D40 0185 409 SCALE: .LONG <100*1000*2> ;NUMBER OF 100 NS IN 10 MS * 2
000186A0 0189 410 SCALE2: .LONG <100*1000> ;NUMBER OR 100 NS IN 10 MS
```

SYSS
Symb
ENBF
EXES
EXES
EXES
PCBS
PCBS
PCBS
PCBS
PCBS
PCBS
PRVS
PSLS
PSLS
SETM
SS\$
SS\$
SS\$
SUPP
WATF

PSEC

A
\$ABS
YSEX

Phas

Init
Comm
Pass
Symb
Pass
Symb
Psec
Cros
Asse

The
3249
Ther
162
12 p


```
018D 412 .SBTTL SET TODR
018D 413 :++
018D 414 :
018D 415 : SET_TODR - SET NEW TIME-OF-DAY CLOCK VALUE
018D 416 :
018D 417 : SET NEW TIME IN TIME-OF-DAY CLOCK, SYSTEM TIME, AND SAVE BOTH
018D 418 : IN SYSTEM PARAMETERS. ALSO RECOMPUTE REMAINING TIME INTERVAL
018D 419 : IN TIMER QUEUE ENTRIES.
018D 420 :
018D 421 : INPUTS:
018D 422 :
018D 423 : R0 = NEW TIME-OF-DAY CLOCK VALUE
018D 424 : R10,R11 = NEW SYSTEM TIME
018D 425 :--
018D 426 SET_TODR:
018D 427 DSBINT W^LOCK_IPL ;DISABLE ALL INTERRUPTS AND
0195 428 ;LOCK CODE IN MEMORY
0195 429 SET_TODR1:
0195 430 MOVQ G^EXE$GQ_SYSTIME,R4 ;GET CURRENT SYSTEM TIME
019C 431 MOVQ R10,G^EXE$GQ_SYSTIME ;SET NEW SYSTEM TIME
01A3 432 MOVQ R10,G^EXE$GQ_TODCBASE ;SAVE SYSTEM TIME OF TOD REG
01AA 433 :
01AA 434 : At this point we should write the new value. We use the physical routine
01AA 435 : to force this to written into any hardware TODR registers that may exist.
01AA 436 : On Nautilus, this forces the slow, non-interruptible write into the clock
01AA 437 : of the console processor. On other processors, the physical function is
01AA 438 : identical to the non-physical function.
01AA 439 :
00000000'EF 16 01AA 440 JSB EXE$WRITEP_TODR ;SET AS CURRENT TIME INTO PHYSICAL REG
01B0 441 :
01B0 442 : Resetting the processor TODR value must be done after a new time is
01B0 443 : written into EXE$GQ_SYSTIME in order for this routine to work on
01B0 444 : MicroVAX I. This is because the routine EXE$WRITE_TODR re-computes
01B0 445 : the value to write based on a base year determined from EXE$GQ_SYSTIME.
01B0 446 : We use the physical function to make sure that we pick up any translations
01B0 447 : which might have been performed. One example would be the conversion to
01B0 448 : one-second time resolution for the Nautilus physical clock register.
01B0 449 :
00000000'EF 16 01B0 450 JSB EXE$READP_TODR ;GET TIME (MAY BE DIFFERENT THAN
01B6 451 ; WHAT WAS WRITTEN)
01B6 452 MOVL R0,G^EXE$GL_TODR ;SAVE TOD REG OF BASE
01BD 453 :
01BD 454 : Recompute expiration times that were not specified as absolute so that the
01BD 455 : remaining interval is the same for the new system time. Absolute expirations
01BD 456 : will occur with respect to the absolute value of the new system time.
01BD 457 :
01BD 458 SUBL R4,R10 ;COMPUTE DELTA OF
01C0 459 SBWC R5,R11 ; OLD AND NEW TIMES
01C3 460 MOVAB G^EXE$GL_TQFL,R3 ;GET ADDRESS OF TIMER QUEUE HEAD
01CA 461 MOVQ (R3),R0 ; Get existing timer queue
01CD 462 MOVQ R0,-(SP) ; Save existing timer queue
01D0 463 MOVL SP,TQE$L_TQFL(R1) ; Readjust links of endmost entries
01D3 464 MOVL SP,TQE$L_TQBL(R0) ;
01D7 465 MOVL R3,(R3) ; Reinitialize timer queue to empty
01DA 466 MOVL R3,4(R3) ;
01DE 467 10$: REMQUE @0(SP),R5 ; Remove an entry from saved queue
01E2 468 BVS 30$ ; Br if no more to remove
```



```
1C A5 D5 01E4 469 TSTL TQESQ_TIME+4(R5) ;IS THIS A VALID TIME?
01E7 470 ;(PERMANENT ENTRY IS INVALID)
0D 19 01E7 471 BLSS 20$ ;BR IF NOT
08 0B A5 03 E0 01E9 472 BBS #TQESV_ABSOLUTE,TQESB_RQTYPE(R5),20$ ;Br if specified as absolute time
01EE 473 ;ADD SYSTEM TIME DELTA
18 A5 5A C0 01EE 474 ADDL R10,TQESQ_TIME(R5) ; TO REMAINING INTERVAL
1C A5 5B D8 01F2 475 ADWC R11,TQESQ_TIME+4(R5) ; Get time value
50 18 A5 7D 01F6 476 20$: MOVQ TQESQ_TIME(R5),R0 ; Reinsert in queue
00000000 GF 16 01FA 477 JSB G^EXESINSTIMQ ;
DC 11 0200 478 BRB 10$ ;
5E 08 C0 0202 479 30$: ADDL2 #8,SP ; Clean up temporary queue
50 00000000 GF 7E 0205 480 MOVAQ G^LCK$GQ_BITMAP_EXP,R0 ; Get address of deadlock timestamps
80 7C 020C 481 CLRQ (R0)+ ; Reset deadlock expiration timestamps
60 7C 020E 482 CLRQ (R0) ;
04 11 0210 483 ENBINT ;RE-ENABLE INTERRUPTS
0213 484 BRB SAVE_TIME ;SAVE TIME IN SYSTEM IMAGE
0215 485 ;
0215 486 ;
0215 487 ; THE PRECEDING CODE MUST DISABLE ALL INTERRUPTS, BUT SINCE IT IS
0215 488 ; PAGED IT MUST BE LOCKED IN MEMORY. THE USAGE OF THE DSBINT MACRO
0215 489 ; ABOVE, BOTH DISABLES INTERRUPTS AND FAULTS INTO MEMORY THE CODE
0215 490 ; BETWEEN THE DSBINT AND LOCK_IPL.
0215 491 ;
0000001F 0215 492 LOCK_IPL: ;IPL NEEDED TO BLOCK EVERYTHING
0215 493 .LONG IPL$_POWER ;
0219 494 LOCK_END: ;END OF LOCKED-DOWN CODE
0219 495 ASSUME <LOCK_END-LOCK_START> LE 512 ;ONE PAGE TO A CUSTOMER
```



```
0219 497 .SBTTL SAVE TIME
0219 498 :++
0219 499 :
0219 500 : SAVE_TIME - SAVE TIME-OF-DAY CLOCK VALUE AND SYSTEM TIME
0219 501 :
0219 502 : SAVE TIME OF DAY REGISTER AND CORRESPONDING SYSTEM TIME IN THE
0219 503 : SYSTEM IMAGE FILE. THESE ARE LOCATED IN THE SYSTEM PARAMETERS
0219 504 : SO WE WILL JUST WRITE THE SYSTEM PARAMETER PAGE(S).
0219 505 :
0219 506 : INPUTS:
0219 507 :
0219 508 : EXESGL TODR = TIME-OF-DAY CLOCK VALUE.
0219 509 : EXESGQ TODCBASE = SYSTEM CORRESPONDING TO TIME-OF-DAY CLOCK.
0219 510 :--
0219 511 :.ENABLE LSB
0219 512 SAVE_TIME:
0219 513 BLBS -4(FP),5$ ;SAVE TIME IN SYSTEM IMAGE FILE
021D 514 MOVAB G^EXESA_SYSPARAM,R2 ;BRANCH TO SKIP WRITING DISK
0224 515 BEGIN_NOPAGE: ;GET SYSTEM PARAMETER ADDRESS
0224 516 DSBINT IPL SYNCH ;PROTECT CHECKSUM OPER. FROM OTHERS
022E 517 JSB EXESBOOTCB_CHK ;AND MAKE THIS CODE NOT PAGE FAULT.
0234 518 BEQL 10$ ;IS THE BOOT CONTROL BLOCK OK?
0236 519 ENBINT ;BRANCH IF YES
0239 520 BUG CHECK BADBOOTCB ;RESTORE PREVIOUS IPL
023D 521 5$: MOV C S^SS$_NORMAL,R0 ;ISSUE NON-FATAL BUGCHECK
0240 522 BRB EXIT ;FAKE A SUCCESSFUL COMPLETION
0242 523 10$: ENBINT ;AND EXIT
0245 524 MOVL BOOSL_PARAM_MAP(R1),R3 ;RESTORE PREVIOUS IPL
0249 525 BEQL 5$ ;ADR OF VIRTUAL TO LOGICAL MAP
024B 526 ;NO MAP, RETURN SUCCESS. NO MAP
024B 527 ASHL #-3,(R3)+,R6 ; IF RUNNING WITHOUT SYS DISK
0250 528 MOVAL -(SP),R4 ;RETRIEVAL POINTER COUNT
0253 529 MOVAQ -(SP),R5 ;ALLOCATE BUFF FOR CHANNEL
0256 530 $ASSIGN S DEVNAM=SYSDISK,- ;ALLOCATE BUFF FOR IOSB
0256 531 CHAN=(R4) ;ASSIGN A CHANNEL TO SYSTEM DISK
0265 532 BLBC R0,EXIT ;BR IF ERROR
0268 533 20$: ASHL #9,(R3)+,R7 ;BYTE COUNT FOR THIS RTRV PTR
026C 534 MOVL (R3)+,R8 ;STARTING LBN FOR THIS PIECE
026F 535 $QIOW S CHAN=(R4),- ;SAVE NEW TOD WITH SYSTEM PARAMS
026F 536 FUNC=S^#10$ WRITELBLK,-
026F 537 EFN=S^#EXESC_SYSEFN,-
026F 538 P1=(R2),- ;EVENT FLAG
026F 539 P2=R7,- ;VIRTUAL ADDRESS OF BUFFER
026F 540 P3=R8,- ;BYTE COUNT TO TRANSFER
026F 541 IOSB=(R5) ;LBN
028C 542 BLBC R0,60$ ;BRANCH IF FAILED TO QUEUE REQUEST
028F 543 MOVZWL (R5),R0 ;GET STATUS FROM IOSB
0292 544 BLBC R0,60$ ;IF ERROR, THEN QUIT
0295 545 ADDL R7,R2 ;SET UP TO WRITE FROM NEXT VA
0298 546 SOBGTR R6,20$ ;LOOP THROUGH THE RETRIEVAL POINTERS
029B 547 60$: PUSHL R0 ;SAVE STATUS
029D 548 $DASSGN S CHAN=(R4) ;DEASSIGN CHANNEL
02A7 549 POPR #^M<R0> ;RESTORE STATUS
02A9 550 EXIT:
02A9 551 RET ;AND RETURN
02AA 552
02AA 553 IPL__SYNCH:
```


SYSSETIME
V04-000

- SYSTEM SERVICE TO SET CURRENT SYSTEM T 16-SEP-1984 02:30:50 VAX/VMS Macro V04-00
SAVE TIME 5-SEP-1984 03:56:59 [SYS.SRC]SYSSETIME.MAR;1

Page 15
(1)

00000008	02AA	554	.LONG	IPL\$_SYNCH
	02AE	555		
	02AE	556	ASSUME	<IPL\$_SYNCH-BEGIN_NOPAGE> LE 512
	02AE	557		
	02AE	558	.DISABLE	LSB
	02AE	559		
	02AE	560	.END	

SYS
V04-

SYSSETIME
Symbol table

- SYSTEM SERVICE TO SET CURRENT SYSTEM T 16-SEP-1984 02:30:50 VAX/VMS Macro V04-00
5-SEP-1984 03:56:59 [SYS.SRC]SYSSETIME.MAR;1

Page 16
(1)

\$ST1	= 00000001		
BASETIME	= 00000000	R	02
BASETIMESZ	= 00000017		
BASEYEAR	= 00000007		
BEGIN_NOPAGE	= 00000224	R	02
BOOSL_PARAM_MAP	= 00000004		
BUG\$ BADBOOTCB	*****	X	02
COMPUTE TODR	00000093	R	02
CVT TIME	000000B8	R	02
ENDTIME	00000017	R	02
ERROR	0000017C	R	02
EXESA_SYSPARAM	*****	X	02
EXESBOOTCB_CHK	*****	X	02
EXESC_SYSEFN	*****	X	02
EXESGL_TODR	*****	X	02
EXESGL_TQFL	*****	X	02
EXESGQ_SYSTIME	*****	X	02
EXESGQ_TODCBASE	*****	X	02
EXESINSTIMQ	*****	X	02
EXESREADP TODR	*****	X	02
EXESSETIME	0000004D	RG	02
EXESSETIME_INT	00000043	RG	02
EXESWRITEP_TODR	*****	X	02
EXIT	000002A9	R	02
IOS WRITELBLK	= 00000020		
IPL\$ POWER	= 0000001F		
IPL\$ SYNCH	= 00000008		
IPL SYNCH	000002AA	R	02
LCK\$GQ_BITMAP_EXP	*****	X	02
LOCK_END	00000219	R	02
LOCK_IPL	00000215	R	02
LOCK_START	00000125	R	02
PCBSQ PRIV	= 00000084		
PR\$ IPL	= 00000012		
PRV\$V_LOG IO	= 00000007		
PRV\$V_OPER	= 00000012		
RECAL	000000E2	R	02
SAVE TIME	00000219	R	02
SCALE	00000185	R	02
SCALE2	00000189	R	02
SET TODR	0000018D	R	02
SET TODR1	00000195	R	02
SS\$ ACCVIO	= 0000000C		
SS\$ IVTIME	= 00000184		
SS\$ NOPRIV	= 00000024		
SS\$ NORMAL	= 00000001		
SYSSASCTIM	*****	GX	02
SYSSASSIGN	*****	GX	02
SYSSBINTIM	*****	GX	02
SYSSDASSGN	*****	GX	02
SYSSQIOW	*****	GX	02
SYSDISK	0000002E	R	02
TIMADR	= 00000004		
TQESB_RQTYPE	= 0000000B		
TQESL_TQBL	= 00000004		
TQESL_TQFL	= 00000000		
TQESQ_TIME	= 00000018		

TQESV_ABSOLUTE

= 00000003

+-----+
! 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	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
YEXEPAGED	000002AE (686.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.27
Command processing	114	00:00:00.55	00:00:01.64
Pass 1	334	00:00:11.44	00:00:26.49
Symbol table sort	0	00:00:01.86	00:00:03.70
Pass 2	117	00:00:02.42	00:00:05.17
Symbol table output	8	00:00:00.09	00:00:00.22
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	606	00:00:16.46	00:00:37.52

The working set limit was 1500 pages.

64976 bytes (127 pages) of virtual memory were used to buffer the intermediate code.

There were 60 pages of symbol table space allocated to hold 1174 non-local and 13 local symbols.

560 source lines were read in Pass 1, producing 18 object records in Pass 2.

31 pages of virtual memory were used to define 30 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	10
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	17
TOTALS (all libraries)	27

1361 GETS were required to define 27 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SYSSETIME/OBJ=OBJ\$:SYSSETIME MSRC\$:SYSSETIME/UPDATE=(ENH\$:SYSSETIME)+EXECML\$/LIB

0388

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY