


```

SSSSSSSS  AAAAAA  TTTTTTTTTT  SSSSSSSS  SSSSSSSS  SSSSSSSS  SSSSSSSSSS  666666
SSSSSSSS  AAAAAA  TTTTTTTTTT  SSSSSSSS  SSSSSSSS  SSSSSSSS  SSSSSSSSSS  666666
SS        AA      AA      TT        TT        TT        SS        SS        SS        55        66
SS        AA      AA      TT        TT        TT        SS        SS        SS        55        66
SS        AA      AA      TT        TT        TT        SS        SS        SS        55        66
SS        AA      AA      TT        TT        TT        SS        SS        SS        55        66
SSSSSSS   AA      AA      TT        TT        TT        SSSSSS   SSSSSS   SSSSSS   55        66666666
SSSSSSS   AA      AA      TT        TT        TT        SSSSSS   SSSSSS   SSSSSS   55        66666666
SS        AA      AA      TT        TT        TT        SS        SS        SS        55        66        66
SS        AA      AA      TT        TT        TT        SS        SS        SS        55        66        66
SS        AA      AA      TT        TT        TT        SS        SS        SS        55        66        66
SSSSSSSS  AA      AA      TT        TT        TT        SSSSSSSS  SSSSSSSS  SSSSSSSS  55        666666
SSSSSSSS  AA      AA      TT        TT        TT        SSSSSSSS  SSSSSSSS  SSSSSSSS  55        666666

```

```

LL        IIIIII  SSSSSSSS
LL        IIIIII  SSSSSSSS
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

```

(1)	56	DECLARATIONS
(1)	127	CONDITION TABLES
(1)	162	TM SETUP, TM CLEANUP
(1)	258	CONDITION SUBROUTINES - SETUP AND CLEANUP
(1)	344	FORM CONDS
(1)	437	VERIFY
(1)	576	VFY CLEANUP
(1)	631	BUILD CLUST SUBROUTINE
(1)	690	SETXPBIT SUBROUTINE
(1)	768	SETSTBIT SUBROUTINE
(1)	791	WAITAST ROUTINE

```

0000 1 .TITLE SATSSS56,SATS SYSTEM SERVICE TESTS WAITS (SUCC S.C.)
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5
0000 6
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 * ALL RIGHTS RESERVED.
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 * TRANSFERRED.
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 * CORPORATION.
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *
0000 27 *
0000 28 *
0000 29 *+
0000 30 * FACILITY: SYSTST (SATS SYSTEM SERVICE TESTS)
0000 31 *
0000 32 * ABSTRACT:
0000 33 *
0000 34 * THIS MODULE CONTAINS SUBROUTINES WHICH, WHEN LINKED
0000 35 * WITH SUCCCOMMON.OBJ, FORM TEST MODULE SATSSS56 TO TEST SUCCESSFUL
0000 36 * OPERATION OF THE 3 WAIT SYSTEM SERVICES. EACH SERVICE IS INVOKED
0000 37 * UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000 38 * SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE. CORRECT
0000 39 * OPERATION OF EACH SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000 40 * CHECKING FOR AN SSS NORMAL STATUS CODE, EXPECTED RETURN ARGUMENTS
0000 41 * AND EXPECTED FUNCTIONALITY PERFORMED.
0000 42 *
0000 43 * ENVIRONMENT: USER MODE IMAGE; NEEDS CMKRNL PRIVILEGE,
0000 44 * DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000 45 *
0000 46 * AUTHOR: THOMAS L. CAFARELLA, CREATION DATE: OCT, 1977
0000 47 *
0000 48 * MODIFIED BY:
0000 49 *
0000 50 * VERSION 1.5 : 25-MAY-79
0000 51 *
0000 52 * 01 LDJ 10/11/79 Fixed bug caused by DIB$K_LENGTH change ACG052.RNO mem
0000 53 *
0000 54 *--

```

```
0000 56 .SBTTL DECLARATIONS
0000 57 :
0000 58 : INCLUDE FILES:
0000 59 :
0000 60 $PRVDEF ; PRIVILEGE BIT DEFINITIONS
0000 61 $PHDDEF ; PROCESS HEADER OFFSETS
0000 62 $PQLDEF
0000 63 $DIBDEF ; DEVICE INFO BLOCK OFFSETS
0000 64 :
0000 65 : MACROS:
0000 66 :
0000 67 :
0000 68 : EQUATED SYMBOLS:
0000 69 :
FF000001 0000 70 EFGRO_MASK = ^XFF000001 ; FLAGS USED BY SYSTEM IN EVENT FLAG GROUP 0
12345678 0000 71 PROC_CONS = ^X12345678 ; PROCESSING CONSTANT -- IN RO UNTIL WAIT
0000 72 ; ... SERVICE REPLACES IT WITH A STATUS CODE
0000 73 :
0000 74 : BIT NUMBERS FOR FLAGS CONTAINED IN 'FLAGS' BYTE:
0000 75 :
00000000 0000 76 EXP_WAIT = 0 ; WAIT EXPECTED WHEN SUBJECT SERVICE ISSUED
00000001 0000 77 REC_WAIT = 1 ; WAIT RECEIVED AFTER SUBJECT SERVICE ISSUED
00000002 0000 78 PROCESSING = 2 ; MAIN ROUTINE PROCESSING WHEN AST DELIVERED
00000003 0000 79 WAITING = 3 ; MAIN ROUTINE IN E.F. WAIT WHEN AST DELIV'D
00000004 0000 80 FELLTHRU = 4 ; MAIN RYN HAD FALLEN THRU WAIT BY AST DEL'Y
00000005 0000 81 ASTLOOP = 5 ; AST RE-ENTERED TOO OFTEN WITHOUT WAITING
00000006 0000 82 CLUSCHG = 6 ; CLUSTER CHANGED ACROSS A WAIT
0000 83 :
0000 84 : OWN STORAGE:
0000 85 :
```

```
00000000 87 .PSECT RODATA, RD, NOWRT, NOEXE, LONG
0000 88 TEST_MOD_NAME:: STRING C, <SATSS56> : TEST MODULE NAME
0009 89 TEST_MOD_NAME_D: STRING I, <SATSS56> : TEST MODULE NAME DESCRIPTOR
0019 90 MSG1_INP_CTL: STRING I, <SSWAT!4ZW: CONDITIONS:>
0039 91 : FAO CIL STRING FOR MSG1 IN SUCCOMMON.MAR
0039 92 MSG3_ERR_CTL:: STRING I, <*SSWAT!4ZW: !AS>
0051 93 : FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
0051 94 CREPRN: STRING I, <SATSS56_CRE> ; CREATED PROCESS NAME
0065 95 CLUS_NAME: STRING I, <SATSS56-CLUS> ; SUBJECT CLUSTER NAME
007A 96 IMAGNAM: STRING I, <SYSTST$RES:SATSUT11.EXE> ; IMAGE NAME FOR CREATED PROC
FFFFFFFF FFFD8F0 0099 97 TIMEINCR: .LONG -10*1000, -1 ; DELTA TIME OF 1 MILLISECOND
00A1 98 CHMKADD: : THE FOLLOWING 3 LONGWORDS ARE THE
00000002: 00A1 99 .ADDRESS SYSS$WAITFR+2 : ... CHMK ADDRESSES IN THE SYSTEM SERVICE
00000002: 00A5 100 .ADDRESS SYSS$WFLAND+2 : ... VECTORS FOR SYSS$WAITFR, SYSS$WFLAND,
00000002: 00A9 101 .ADDRESS SYSS$WFLOR+2 : ... AND SYSS$WFLOR, RESPECTIVELY.
00AD 102 QUOTALIST: $QUOTA CPULM, 0 ; INFINITE CPU
00B2 103 $QUOTA BYTLM, 512 ; BYTE LIMIT FOR BUFFERED I/O
00B7 104 $QUOTA FILLM, 2 ; OPEN FILE COUNT LIMIT
00BC 105 $QUOTA PGFLQUOTA, 10 ; PAGING FILE QUOTA
00C1 106 $QUOTA PRCLM, 2 ; SUBPROCESS QUOTA
00C6 107 $QUOTA TQELM, 3 ; TIMER QUEUE ENTRY QUOTA
00CB 108 $QUOTA LISTEND ; DEFINES END OF LIST
```

00000000	0000	110	.PSECT	RWDATA, RD, WRT, NOEXE, LONG	
00000008	0000	111	PRIVMASK:	.BLKQ 1	: ADDR OF PRIVILEGE MASK (IN PHD)
0000000C	0008	112	MBXCHAN:	.BLKL 1	: CHAN NO. FOR MAILBOX FOR CREATED PROCESS
	000C	113	MBXCHANINFO:		: CHANNEL INFO RETURNED BY GETCHN
00000074	000C	114		.LONG DIBSK_LENGTH	
00000014	0010	115		.ADDRESS +4	
00000088	0014	116		.BLKB DIBSK_LENGTH	
0000008C	0088	117	MBXUNIT:	.BLKL 1	: SAVE AREA FOR MAILBOX UNIT NUMBER
	008C	118	MBXBUFF:	STRING 0,120	: MAILBOX BUFFER FOR CREATED PROCESS
00000110	010C	119	SUBJ_MASK:	.BLKL 1	: MASK TO BE SPECIFIED ON WAIT SYST SERVICES
00000114	0110	120	TRIAL_MASK:	.BLKL 1	: CLUSTER MASK; USED TO SET SUBJECT CLUSTER
00000116	0114	121	TM WORK:	.BLKW 1	: WORK AREA FOR TRIAL MASK
0000011A	0116	122	CLOS_STATE:	.BLKL 1	: STATE OF SUBJECT CLOSTER
00	011A	123	CONSEC_P:	.BYTE 0	: COUNT OF CONSEC. 'PROCESSING' AST ENTRIES
00	011B	124	FLAGS:	.BYTE 0	: GEN. PURP. FLAGS; BIT DEFINITIONS ABOVE
00	011C	125	PROCFLAG:	.BYTE 0	: PROCESS FLAG; 0 = CREATED PROC NON-EXIST.

```

011D 127 .SBTTL CONDITION TABLES
011D 128 :
011D 129 ***** CONDITION TABLES FOR WAIT SYSTEM SERVICES *****
011D 130 :
011D 131 COND 1,NOTARG,<WHERE WAITS ARE CLEARED>,-
011D 132 <IN A DIFFERENT PROCESS>,-
011D 133 <IN THE SAME PROCESS>,-
011D 134
0169 135 COND 2,NOTARG,<TYPE OF WAIT SYSTEM SERVICE>,-
0169 136 <$WAITFR>,-
0169 137 <$WFLAND>,-
0169 138 <$WFLOR>,-
0169 139
01A9 140 COND 3,LONG,<EFN>,-
01A9 141 <EVENT FLAGS 0-31 (EV FLAG GROUP 0)>,-
01A9 142 <EVENT FLAGS 32-63 (EV FLAG GROUP 1)>,-
01A9 143 <EVENT FLAGS 64-95 (EV FLAG GROUP 2)>,-
01A9 144 <EVENT FLAGS 96-127 (EV FLAG GROUP 3)>,-
01A9 145
00000000 024E 146 .LONG 0 ; EVENT FLAG GROUP (CLUSTER) 0
00000020 0252 147 .LONG 32 ; EVENT FLAG GROUP (CLUSTER) 1
00000040 0256 148 .LONG 64 ; EVENT FLAG GROUP (CLUSTER) 2
00000060 025A 149 .LONG 96 ; EVENT FLAG GROUP (CLUSTER) 3
025E 150 :
025E 151 COND 4,LONG,<MASK>,-
025E 152 <ALL 1-BITS>,-
025E 153 <MIXED 0- AND 1-BITS>,-
025E 154
FFFFFFF 028B 155 .LONG ^XFFFFFFF ; ALL 1-BITS
53544942 028F 156 .LONG ^A/BITS/ ; MIXED BITS
0293 157 :
0293 158 COND 5,NULL
0294 159
00000000 160 .PSECT SATSS56,RD,WRT,EXE

```



```

0000 162 .SBTTL TM_SETUP, TM_CLEANUP
0000 163 :++
0000 164 : FUNCTIONAL DESCRIPTION:
0000 165 :
0000 166 : TM SETUP AND TM CLEANUP ARE CALLED TO PERFORM
0000 167 : REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
0000 168 : TEST MODULE EXECUTION.
0000 169 :
0000 170 : CALLING SEQUENCE:
0000 171 :
0000 172 : BSBW TM_SETUP BSBW TM_CLEANUP
0000 173 :
0000 174 : INPUT PARAMETERS:
0000 175 :
0000 176 : NONE
0000 177 :
0000 178 : IMPLICIT INPUTS:
0000 179 :
0000 180 : NONE
0000 181 :
0000 182 : OUTPUT PARAMETERS:
0000 183 :
0000 184 : NONE
0000 185 :
0000 186 : IMPLICIT OUTPUTS:
0000 187 :
0000 188 : TM_SETUP: COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
0000 189 : ALL PRIVILEGES ACQUIRED.
0000 190 :
0000 191 : COMPLETION CODES:
0000 192 :
0000 193 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0000 194 :
0000 195 : SIDE EFFECTS:
0000 196 :
0000 197 : SS CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0000 198 : (VIA RSB) IF ERROR ENCOUNTERED.
0000 199 :
0000 200 :--
0000 201 :
0000 202 :
0000 203 :
0000 204 TM_SETUP::
0000 205 CLRL R2 ; INITIALIZE
0000 206 CLRL R3 ; .. CONDITION
0000 207 CLRL R4 ; .... TABLE
0000 208 CLRL R5 ; ..... INDEX
0000 209 CLRL R6 ; ..... REGISTERS
0000 210 BSBW MOD MSG PRINT ; PRINT TEST MODULE BEGIN MSG
0000 211 MOVAL TEST_MOD_SUCC,TMD_ADDR ; ASSUME END MSG WILL SHOW SUCCESS
0000 212 INSV #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
0000 213 MODE TO,5$,KRNL ; KERNEL MODE TO ACCESS PHD
0000 214 MOVL @#CTL$GL PHD,R9 ; GET PROCESS HEADER ADDRESS
0000 215 MOVAL PHD$Q PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
0000 216 MODE FROM,5$ ; BACK TO USER MODE
0000 217 PRIV ADD,ALL ; GET ALL PRIVILEGES

```

```

52 D4 0000
53 D4 0002
54 D4 0004
55 D4 0006
56 D4 0008
FFF3' 30 000A
00000000'EF 00000000'EF DE 000D
03 00 00000000'8F FO 0018
00000000'EF 0020
59 00000000'9F D0 0048
00000000'EF 69 DE 004F
0056
0057

```

```

0077 218 $SETPRN S TEST MOD_NAME_D ; SET PROCESS NAME
0084 219 SS_CHECKR NORMAL ; CHECK STATUS CODE RETURNED FROM SETPRN
00B2 220 :
00B2 221 : GET 2 COMMON EVENT FLAG GROUPS ASSOCIATED FOR LATER USE
00B2 222 : ALSO, CREATE A PROCESS & ITS ASSOCIATED MAILBOX
00B2 223 :
00B2 224 $ASCEFC S EFN=#64, NAME=CLUS_NAME ; ASSOCIATE GROUP 2
00C9 225 SS_CHECKR NORMAL ; CHECK FOR NORMAL STATUS CODE
00F7 226 $ASCEFC S EFN=#96, NAME=CLUS_NAME ; ASSOCIATE GROUP 3
010E 227 SS_CHECKR NORMAL ; CHECK FOR NORMAL STATUS CODE
013C 228 $CREMBX_S CHAN=MBXCHAN, LOGNAM=CREPRN, -
013C 229 MAXMSG=#120, PROMSK=#0, BUFQUO=#240
0161 230 ; GET MAILBOX FOR PROCESS
0161 231 SS_CHECKR NORMAL ; CHECK NORMAL COMPLETION
018F 232 $GETCHN_S CHAN=MBXCHAN, PRIBUF=MBXCHANINFO
01A9 233 ; GET CHANNEL INFO (UNIT NUMBER)
01A9 234 SS_CHECKR NORMAL ; CHECK NORMAL COMPLETION
0000088'EF 0000020'EF 3C 01D7 235 MOVZWL MBXCHANINFO+8+DIB$W_UNIT, MBXUNIT
01E2 236 ; SAVE MAILBOX UNIT NUMBER
01E2 237 $CREPRC_S PRCNAM=CREPRN, IMAGE=IMAGNAM, -
01E2 238 MBXUNT=MBXUNIT, QUOTA=QUOTALIST
0214 239 ; CREATE PROCESS WHICH WILL CLEAR WAITS
0214 240 SS_CHECKR NORMAL ; CHECK ITS COMPLETION
0000011C'EF 01 90 0242 241 MOVB #1, PROCFLAG ; INDICATE CREATED PROCESS EXISTS
05 0249 242 RSB ; RETURN TO MAIN ROUTINE
024A 243 TM_CLEANUP::
0000011C'EF 95 024A 244 TSTB PROCFLAG ; DOES CREATED PROCESS EXIST ?
3E 13 0250 245 BEQL 10$ ; NO -- SKIP PROCESS-RELATED SERVICES
0252 246 $FORCEX_S PRCNAM=CREPRN, CODE=#SS$ NORMAL
0267 247 ; FORCE SUBPROCESS TO EXIT & BE DELETED
0267 248 $QIOW_S CHAN=MBXCHAN, FUNC=#IOS$ READVBLK, -
0267 249 P1=MBXBUFF+8, P2=MBXBUFF
0290 250 ; WAIT FOR IT TO SEND MAIL
0290 251 10$:
0290 252 $DELMBX_S MBXCHAN ; DELETE TERMINATION MAILBOX
029E 253 $DACEFC_S EFN=#64 ; DISASSOCIATE
02AB 254 $DACEFC_S EFN=#96 ; ... COMMON CLUSTERS
FD45' 30 0288 255 BSBW MOD_MSG_PRINT ; PRINT TEST MODULE END MSG
05 028B 256 RSB ; RETURN TO MAIN ROUTINE

```

```

02BC 258 .SBTTL CONDITION SUBROUTINES - SETUP AND CLEANUP
02BC 259 :++
02BC 260 : FUNCTIONAL DESCRIPTION:
02BC 261 :
02BC 262 :         CONDX AND CONDX CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
02BC 263 : BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
02BC 264 : CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
02BC 265 : ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
02BC 266 : CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
02BC 267 : UP, IF NECESSARY, IN THE CONDX CLEANUP SUBROUTINE. THIS INCLUDES,
02BC 268 : ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
02BC 269 : OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
02BC 270 : VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
02BC 271 : (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
02BC 272 :
02BC 273 : CALLING SEQUENCE:
02BC 274 :
02BC 275 :         BSBW CONDX   BSBW CONDX_CLEANUP
02BC 276 :         WHERE X = 1,2,3,4,5
02BC 277 :
02BC 278 : INPUT PARAMETERS:
02BC 279 :
02BC 280 :         CONFLICT = 0
02BC 281 :
02BC 282 : IMPLICIT INPUTS:
02BC 283 :
02BC 284 :         R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
02BC 285 :         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
02BC 286 :
02BC 287 : OUTPUT PARAMETERS:
02BC 288 :
02BC 289 :         CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
02BC 290 :
02BC 291 : IMPLICIT OUTPUTS:
02BC 292 :
02BC 293 :         R2,3,4,5,6 PRESERVED
02BC 294 :
02BC 295 : COMPLETION CODES:
02BC 296 :
02BC 297 :         NONE
02BC 298 :
02BC 299 : SIDE EFFECTS:
02BC 300 :
02BC 301 :         NONE
02BC 302 :
02BC 303 : --
02BC 304 :
02BC 305 :
02BC 306 :
05 02BC 307 COND1::                                ; RETURN TO MAIN ROUTINE
02BC 308 RSB
05 02BD 309 COND1_CLEANUP::                        ; RETURN TO MAIN ROUTINE
02BC 310 RSB
05 02BE 311 COND2::                                ; RETURN TO MAIN ROUTINE
02BC 312 RSB
05 02BF 313 COND2_CLEANUP::                          ; RETURN TO MAIN ROUTINE
02BC 314 RSB

```

			02C0	315	COND3::				
			02C0	316	:				
			02C0	317	: CHECK FIRST FOR COND2/COND3 CONFLICT				
			02C0	318	:				
			02C0	319					
	53	D5	02C0	319	TSTL	R3	: \$WAITFR (SINGLE FLAG) ?		
	0A	12	02C2	320	BNEQU	10\$: NO -- GO CHECK COND1/COND3 CONFLICT		
20	0000024E'EF	44	D1	02C4	321	CMPL	EFNR[4],#32	: YES -- LOCAL EVENT FLAG GROUP 0 ?	
	12	19	02CC	322	BLSS	20\$: YES -- FLAGS USED BY SYST; GO SET CONFLICT		
			02CE	323	10\$:		: CHECK FOR COND1/COND3 CONFLICT		
	52	D5	02CE	324	TSTL	R2	: WAITS TO BE CLEARED IN DIFF PROCESS ?		
	19	12	02D0	325	BNEQU	COND3X	: NO -- ALL IS OK; JUST EXIT		
00000040	8F		D1	02D2	326	CMPL	EFNR[4],#64	: YES -- LOCAL EVENT FLAG GROUP ?	
			0B	18	02DE	327	BGEQ	COND3X	: NO -- ALL IS OK; JUST EXIT
			02E0	328				: YES -- CAN'T SET LOCAL FLAGS IN DIFF PROC	
			02E0	329	20\$:				
00000000'EF			02E0	330	MOVB	ONES,CONFLICT		: INDICATE CONFLICT	
			02EB	331	COND3X:				
		05	02EB	332	RSB			: RETURN TO MAIN ROUTINE	
			02EC	333	COND3_CLEANUP::				
		05	02EC	334	RSB			: RETURN TO MAIN ROUTINE	
			02ED	335	COND4::				
		05	02ED	336	RSB			: RETURN TO MAIN ROUTINE	
			02EE	337	COND4_CLEANUP::				
		05	02EE	338	RSB			: RETURN TO MAIN ROUTINE	
			02EF	339	COND5::				
		05	02EF	340	RSB			: RETURN TO MAIN ROUTINE	
			02F0	341	COND5_CLEANUP::				
		05	02F0	342	RSB			: RETURN TO MAIN ROUTINE	


```
03F3 437 .SBTTL VERIFY
03F3 438 :++
03F3 439 : FUNCTIONAL DESCRIPTION:
03F3 440 :
03F3 441 : VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
03F3 442 : TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
03F3 443 : COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
03F3 444 : SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
03F3 445 : ($WAITEF, $WFLAND, $WFLOR). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE
03F3 446 : IS VERIFIED BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN
03F3 447 : ARGUMENTS AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE
03F3 448 : FORM OF COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON
03F3 449 : CAUSES AN ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
03F3 450 : THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
03F3 451 : PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
03F3 452 : WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
03F3 453 : AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
03F3 454 :
03F3 455 : CALLING SEQUENCE:
03F3 456 :
03F3 457 : BSBW VERIFY
03F3 458 :
03F3 459 : INPUT PARAMETERS:
03F3 460 :
03F3 461 : NONE
03F3 462 :
03F3 463 : IMPLICIT INPUTS:
03F3 464 :
03F3 465 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
03F3 466 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
03F3 467 : FOR X = 1,2,3,4,5 :
03F3 468 : CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
03F3 469 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
03F3 470 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
03F3 471 : FOR CONDX_E.
03F3 472 :
03F3 473 : OUTPUT PARAMETERS:
03F3 474 :
03F3 475 : NONE
03F3 476 :
03F3 477 : IMPLICIT OUTPUTS:
03F3 478 :
03F3 479 : VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
03F3 480 : IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
03F3 481 : ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
03F3 482 : AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
03F3 483 : ERRORS.
03F3 484 :
03F3 485 : COMPLETION CODES:
03F3 486 :
03F3 487 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
03F3 488 :
03F3 489 : SIDE EFFECTS:
03F3 490 :
03F3 491 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
03F3 492 : (VIA RSB) IF ERROR ENCOUNTERED.
03F3 493 :
```

```

03F3 494 :--
03F3 495
03F3 496
03F3 497
03F3 498 VERIFY::
00000000'EF 95 03F3 499 TSTB CFLAG : SHOULD CONDITIONS BE PRINTED ?
03 13 03F9 500 BEQL 5$ : NO -- CONTINUE
FEF3 30 03FB 501 BSBW FORM_CONDS : YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
FBFF' 30 03FE 502 5$: BSBW SAVE REGS : SAVE REGS 2-6 FOR LATER USE IN AST ROUTINE
0000010C'EF 00000114'EF B4 0401 504 CLRW TM_WORK : INIT TRIAL_MASK WORK AREA
0000028B'EF45 D0 0407 505 MOVL MASK[R5],SUBJ_MASK : GET SUBJECT_MASK FOR THIS TEST CASE
20 0000024E'EF44 D1 0413 506 CML EFNR[4],#32 : EVENT FLAG GROUP 0 ?
0B 18 041B 507 BGEQ 10$ : NO -- CONTINUE
0000010C'EF FF000001 8F CA 041D 508 BICL2 #EFGRO_MASK,SUBJ_MASK : YES -- STEER CLEAR OF FLAGS USED BY SYS
0428 509 10$: MOVB TM_WORK,TRIAL_MASK : PROPAGATE TM WORK ....
00000110'EF 00000114'EF 90 0428 510 MOVB TM_WORK,TRIAL_MASK+1 : ... BYTE VALUE .....
00000111'EF 00000114'EF 90 0433 511 MOVW TRIAL_MASK,TRIAL_MASK+2 : ..... THRU TRIAL_MASK
00000112'EF 00000110'EF B0 043E 512 MOVL EFNR[4],R10 : TELL BUILD CLUST WHICH CLUSTER TO BUILD
SA 0000024E'EF44 D0 0449 513 BSBW BUILD_CLUST : ESTAB CLUSTER ACCORDING TO TRIAL_MASK
02E1 30 0451 514 TSTB EFLAG : IS AN ERROR BEING PROCESSED ?
00000000'EF 95 0454 515 BEQL 15$ : NO -- CONTINUE
03 13 045A 516 BRW VERIFYX : YES -- RETURN IMMEDIATELY
02D4 31 045C 517 045F 518 15$: CLRB FLAGS : CLEAR ALL FLAGS
0000011B'EF 94 045F 519 BSBW SETEXPBIT : SET EXPECTATION BIT
03FD 30 0465 520 $SETIMR_S DAYTIM=TIMEINCR, - : SCHEDULE AN AST IN 1 MILLISECOND
0468 521 ASTADR=WAITAST
0468 522 SS CHECK NORMAL : CHECK SETIMR COMPLETION
50 12345678 8F D0 047F 523 MOVL #PROC_CONS,R0 : CONSTANT IN R0 TO BE CHECKED IN AST
04AD 524 : ***** ONE OF FOLLOWING WAIT CALLS IS THE SUBJECT OF THIS TEST CASE *****
04B4 525 :
04B4 526 :
04B4 527 :
53 D5 04B4 528 TSTL R3 : WAITFR (SINGLE FLAG) ?
10 12 04B6 529 BNEQU 20$ : NO -- CONTINUE
04B8 530 $WAITFR_S EFN=EFN[R4] : YES -- ISSUE SUBJECT SERVICE
2F 11 04C6 531 BRB 40$ : GO CHECK ITS COMPLETION
04C8 532 20$: CML R3,#1 : WFLAND ?
01 53 D1 04C8 533 BNEQU 30$ : NO -- CONTINUE
16 12 04CB 534 $WFLAND_S EFN=EFN[R4], MASK=SUBJ_MASK : YES -- ISSUE SUBJECT SERVICE
04E1 535 BRB 40$ : GO CHECK ITS COMPLETION
14 11 04E1 536 30$: $WFLOR_S EFN=EFN[R4], MASK=SUBJ_MASK ; MUST BE WFLOR
04E3 537 40$: BBSS #FELLTHRU,FLAGS,..+1 : INDICATE THAT WAIT IS DONE
04F7 540 40$: CML RO,#SS$_NORMAL : CODE RECEIVED = CODE EXPECTED ?
00 0000011B'EF 04 E2 04F7 541 BEQLU 50$ : YES -- CONTINUE
00000000'8F 50 D1 04FF 542 MOVL #SS$_NORMAL,EXPV : LOAD UP EXPECTED AND
00000000'EF 00000000'8F D0 0508 543 MOVL RO,RECV : ... RECEIVED VALUES, THEN EXIT
00000000'EF 50 D0 0513 544 ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM WAIT SERVICE>
051A 545 50$: $HIBER_S : SLEEP UNTIL WAITAST CHECKS RESULTS OF SERV
056F 547 50$: BBC #ASTLOOP,FLAGS,55$ : CONTINUE IF NO AST LOOP
4B 0000011B'EF 05 E1 0576 548 CLRB EXPV : AST LOOP -- SET UP FOR ERR_EXIT
00000000'EF 94 057E 549
057E 550

```


00000000'EF	94	0584	551	CLRB	RECV		
		058A	552	ERR_EXIT	BYTE,<UNSATISFIED WAIT CONDITION>		
54 0000011B'EF	06	05C9	553	55\$:			
		05C9	554	BBC	#CLUSCHG,FLAGS,58\$:	CONTINUE IF NO CLUSTER CHANGE ERROR
		05D1	555	ERR_EXIT	LONG,<CLUSTER EVENT FLAG SETTINGS CHANGED ACROSS WAIT>		
7D 0000011B'EF	00	0525	556	58\$:			
03 0000011B'EF	01	0625	557	BBC	#EXP_WAIT,FLAGS,70\$:	BRANCH IF FALL-THRU EXPECTED
	00EF	062D	558	BBC	#REC_WAIT,FLAGS,60\$:	WAIT EXP'D; BRANCH IF FALL-THRU RECEIVED
		0635	559	BRW	80\$:	WAIT EXPECTED & RECEIVED; THAT'S OK
00000000'EF	00000110'EF	0638	560	60\$:			
00000000'EF	00000110'EF	0638	561	MOVL	TRIAL_MASK,EXPV	:	SET UP TRIAL_MASK
		0643	562	MOVL	TRIAL_MASK,RECV	: FOR MSG, THEN EXIT
		064E	563	ERR_EXIT	LONG,<WAIT EXP'D; FALL-THRU REC'D; VALUE IS CLUST BEFORE WAIT>		
03 0000011B'EF	01	06AA	564	70\$:			
	0072	06AA	565	BBS	#REC_WAIT,FLAGS,75\$:	BRANCH IF WAIT REC'D; THAT'S AN ERROR
		06B2	566	BRW	80\$:	FALL-THRU EXPECTED & RECEIVED; THAT'S OK
00000000'EF	00000110'EF	06B5	567	75\$:			
00000000'EF	00000110'EF	06B5	568	MOVL	TRIAL_MASK,EXPV	:	SET UP TRIAL_MASK
		06C0	569	MOVL	TRIAL_MASK,RECV	: FOR MSG, THEN EXIT
		06CB	570	ERR_EXIT	LONG,<FALL-THRU EXP'D; WAIT REC'D; VALUE IS CLUST BEFORE WAIT>		
F5 00000114'EF	01	0727	571	80\$:			
	00FF	0727	572	ACBW	#255,#1,TM_WORK,10\$:	MAKE NEW TRIAL_MASK & LOOP
	8F	0731					
		0733	573	VERIFYX:			
		0733	574	RSB		:	RETURN TO CALLER

```
0734 576 .SBTTL VFY_CLEANUP
0734 577 :
0734 578 : ** FUNCTIONAL DESCRIPTION:
0734 579 :
0734 580 : VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
0734 581 : EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
0734 582 : ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
0734 583 : ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
0734 584 : ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
0734 585 : IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
0734 586 : WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
0734 587 : POSSIBLY DISCOVERING A SECOND ERROR.
0734 588 :
0734 589 : CALLING SEQUENCE:
0734 590 :
0734 591 : BSBW VFY_CLEANUP
0734 592 :
0734 593 : INPUT PARAMETERS:
0734 594 :
0734 595 : NONE
0734 596 :
0734 597 : IMPLICIT INPUTS:
0734 598 :
0734 599 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0734 600 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0734 601 : FOR X = 1,2,3,4,5 :
0734 602 : CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
0734 603 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0734 604 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0734 605 : FOR CONDX_E.
0734 606 :
0734 607 : OUTPUT PARAMETERS:
0734 608 :
0734 609 : NONE
0734 610 :
0734 611 : IMPLICIT OUTPUTS:
0734 612 :
0734 613 : NONE
0734 614 :
0734 615 : COMPLETION CODES:
0734 616 :
0734 617 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0734 618 :
0734 619 : SIDE EFFECTS:
0734 620 :
0734 621 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0734 622 : (VIA RSB) IF ERROR ENCOUNTERED.
0734 623 :
0734 624 : --
0734 625 :
0734 626 :
0734 627 :
05 0734 628 VFY_CLEANUP::
0734 629 RSB ; RETURN TO CALLER
```

```

0735 631 .SBTTL BUILD_CLUST SUBROUTINE
0735 632 :
0735 633 :*****
0735 634 :
0735 635 * BUILD_CLUST SUBROUTINE
0735 636 :
0735 637 * THIS SUBROUTINE SETS THE SUBJECT CLUSTER EQUAL TO
0735 638 * THE TRIAL_MASK BY ISSUING THE PROPER COMBINATION
0735 639 * OF 32 SETEF/CLREF'S. THEN A REDEF IS ISSUED AND
0735 640 * THE NEW CLUSTER SETTINGS ARE VERIFIED.
0735 641 :
0735 642 * INPUTS:
0735 643 :
0735 644 * R10 - THE LOW-ORDER EFN IN THE SUBJECT CLUSTER
0735 645 :
0735 646 * TRIAL_MASK - LONGWORD CONTAINING THE MASK TO
0735 647 * WHICH THE SUBJECT CLUSTER IS
0735 648 * TO BE SET.
0735 649 :
0735 650 * OUTPUTS:
0735 651 :
0735 652 * SUBJECT CLUSTER - UPDATED TO LOOK LIKE TRIAL_MASK.
0735 653 :
0735 654 * CLUS_STATE - LONGWORD CONTAINING THE IMAGE
0735 655 * OF THE NEWLY SET SUBJECT CLUSTER.
0735 656 :
0735 657 * VOLATILE REGISTERS:
0735 658 *
0735 659 * R0, R1, R8, R9
0735 660 :
0735 661 :*****
0735 662 :
0735 663 BUILD_CLUST:
58 SA D0 0735 664 MOVL R10,R8 ; ESTABLISH FIRST EFN (EVENT FLAG NO.)
59 D4 0738 665 CLRL R9 ; INIT OFFSET INTO TRIAL_MASK
3A 00000110'EF 59 E0 073A 666 20$: BBS R9,TRIAL_MASK,30$ ; ISSUE $SETEF IF BIT FOR THIS FLAG IS SET
0742 668 $CLREF_S EFN=R8 ; ... OTHERWISE, ISSUE $CLREF
68 50 E8 074B 669 BLBS RO,40$ ; IF NORMAL STATUS, PROCESS NEXT EVENT FLAG
074E 670 SS_CHECK NORMAL ; USE SS_CHECK TO TERMINATE TEST MODULE
077C 671 30$:
2E 50 E8 077C 672 $SETEF_S EFN=R8 ; SET CURRENT EVENT FLAG
0785 673 BLBS RO,40$ ; IF NORMAL STATUS, PROCESS NEXT EVENT FLAG
0788 674 SS_CHECK NORMAL ; USE SS_CHECK TO TERMINATE TEST MODULE
07B6 675 40$:
FF7C 59 01 1F 9D 07B8 677 INCW R8 ; GET NEXT EFN
07BE 678 ACBB #31,#1,R9,20$ ; GO DO NEXT EVENT FLAG
2E 50 E8 07CD 679 $REDEF_S EFN=R10, STATE=CLUS_STATE ; READ THE CLUSTER JUST BUILT
07D0 680 BLBS RO,50$ ; CONTINUE IF NORMAL COMPLETION
07FE 681 50$: SS_CHECK NORMAL ; USE SS_CHECK TO TERMINATE TEST MODULE
00000110'EF 00000116'EF D1 07FE 682 CMPL CLUS_STATE,TRIAL_MASK ; DID CLUSTER GET BUILT OK ?
59 13 0809 683 BEQLU BUILD_CLUSTX ; YES -- SIMPLY EXIT
00000000'EF 00000110'EF D0 080B 684 MOVL TRIAL_MASK,EXPV ; NO -- LOAD EXPECTED AND ...
00000000'EF 00000116'EF D0 0816 685 MOVL CLUS_STATE,RECV ; ... RECEIVED VALUES, THEN EXIT
0821 686 ERR EXIT LONG.<TRIAL_MASK NOT BUILT CORRECTLY>
0864 687 BUILD_CLUSTX:

```

SATSSSS6
V04-000

SATS SYSTEM SERVICE TESTS WAITS (SUCC S 16-SEP-1984 00:58:38 VAX/VMS Macro V04-00
BUILD_CLUST SUBROUTINE 5-SEP-1984 04:32:34 [UETPSY.SRC]SATSSSS6.MAR;1

Page 17
(1)

SA
VO

05 0864 688 RSB

; RETURN TO CALLER

20

20

34

3A

20

20

30

```
0865 690 .SBTTL SETEXPBIT SUBROUTINE
0865 691 :
0865 692 :*****
0865 693 :*
0865 694 :* SETEXPBIT SUBROUTINE
0865 695 :*
0865 696 :* THIS SUBROUTINE DETERMINES WHETHER TO EXPECT A WAIT
0865 697 :* OR A FALL-THRU FROM THE UPCOMING ISSUANCE OF THE
0865 698 :* SUBJECT WAIT SYSTEM SERVICE FOR THIS TEST CASE.
0865 699 :* IF A WAIT IS EXPECTED, THE EXP WAIT BIT IS SET; IF
0865 700 :* A FALL-THRU IS EXPECTED, THE BIT IS NOT SET. THE
0865 701 :* ALGORITHM FOR DETERMINING WHICH STATE TO EXPECT DEPENDS
0865 702 :* ON WHICH OF THE THREE TYPES OF WAIT SYSTEM SERVICE
0865 703 :* WILL BE ISSUED FOR THIS TEST CASE.
0865 704 :*
0865 705 :* $WAITFR (SINGLE EVENT FLAG):
0865 706 :* IF THE LOW-ORDER BIT OF THE TRIAL MASK
0865 707 :* IS SET, EXPECT A FALL-THRU; OTHERWISE,
0865 708 :* EXPECT A WAIT.
0865 709 :*
0865 710 :* $WFLAND:
0865 711 :* DO A LOGICAL 'AND' OF THE TRIAL_MASK WITH
0865 712 :* THE SUBJECT MASK; IF THE RESULT EQUALS THE
0865 713 :* SUBJECT MASK, EXPECT A FALL-THRU; OTHERWISE,
0865 714 :* EXPECT A WAIT.
0865 715 :*
0865 716 :* $WFLOR:
0865 717 :* DO A LOGICAL 'AND' OF THE TRIAL_MASK WITH
0865 718 :* THE SUBJECT MASK; IF THE RESULT IS ZERO,
0865 719 :* EXPECT A WAIT; OTHERWISE, EXPECT A FALL-THRU.
0865 720 :*
0865 721 :* INPUTS:
0865 722 :*
0865 723 :* R3 - CONDITION TABLE 2 INDEX REG,
0865 724 :* INDICATING TYPE OF WAIT SERVICE
0865 725 :*
0865 726 :* SUBJ_MASK - SUBJECT MASK. LONGWORD DATA ITEM,
0865 727 :* USED AS THE VALUE FOR THE MASK
0865 728 :* ARGUMENT FOR $WFLAND & $WFLOR.
0865 729 :*
0865 730 :* TRIAL_MASK - TRIAL MASK. LONGWORD DATA ITEM,
0865 731 :* WHOSE BIT PATTERN IS EQUAL TO THE
0865 732 :* CURRENT SETTING OF THE SUBJECT
0865 733 :* EVENT FLAG CLUSTER.
0865 734 :*
0865 735 :* OUTPUTS:
0865 736 :*
0865 737 :* EXP_WAIT - BIT IN FLAGS BYTE. SET TO 1 IF
0865 738 :* A WAIT IS EXPECTED; REMAINS AS 0
0865 739 :* IF FALL-THRU IS EXPECTED.
0865 740 :*
0865 741 :* VOLATILE REGISTERS:
0865 742 :*
0865 743 :* R7
0865 744 :*
0865 745 :*****
0865 746 :
```

0
0
0

```

    53 D5 0865 747 SETEXPBIT:
    OF 12 0865 748 TSTL R3 ; $WAITFR (SINGLE FLAG) ?
31 39 00000110'EF 00 E8 0867 749 BNEQU 10$ ; NO -- CONTINUE
    00 E3 0869 750 BLBS TRIAL_MASK,SETEXPBITX ; IF LOW BIT OF MASK SET, EXPECT A FALL-THRU
    00 E3 0870 751 BBCS #EXP_WAIT,FLAGS,SETEXPBITX ; SET EXPECT-WAIT BIT & EXIT
57 57 0000010C'EF 57 D2 0878 752 10$: MCOML SUBJ_MASK,R7 ; LOGICALLY "AND" SUB_MASK
00000110'EF 57 CB 087F 753 BICL3 R7,TRIAL_MASK,R7 ; WITH TRIAL_MASK, RESULT IN R7
    01 53 D1 0887 754 CMPL R3,#1 ; $WFLAND ?
    0C 13 088A 755 BEQLU 20$ ; YES -- GO TEST RESULT
    57 D5 088C 756 TSTL R7 ; NO -- ASSUME $WFLOR
    19 12 088C 757 BNEQU SETEXPBITX ; RESULT ZERO ?
11 0000011B'EF 00 E3 088E 758 BBCS #EXP_WAIT,FLAGS,SETEXPBITX ; NO -- EXPECT A FALL-THRU
    0000010C'EF 57 D1 0890 760 20$: CMPL R7,SUBJ_MASK ; RESULT = SUBJECT MASK ?
00 0000011B'EF 00 08 13 0898 761 BEQLU SETEXPBITX ; YES -- EXPECT A FALL-THRU
    00 E3 089F 762 BBCS #EXP_WAIT,FLAGS,SETEXPBITX ; NO -- SET BIT & EXIT
    00 E3 08A1 763 SETEXPBITX:
    00 05 08A9 764 RSB
    00 05 08A9 765
    00 05 08A9 766
  
```

```

08AA 768 .SBTTL SETSTBIT SUBROUTINE
08AA 769 :
08AA 770 : THE SETSTBIT SUBROUTINE SETS A STATE BIT INDICATING
08AA 771 : THE STATE OF THE MAIN (INTERRUPTED) ROUTINE WHEN
08AA 772 : THE WAITAST WAS DELIVERED.
08AA 773 :
08AA 774 SETSTBIT:
00 0000011B'EF 02 E4 08AA 775 BBSC #PROCESSING,FLAGS,..+1 ; CLEAR 'PROCESSING' ...
00 0000011B'EF 03 E4 08B2 776 BBSC #WAITING,FLAGS,..+1 ; ... AND 'WAITING' FLAGS
03 0000011B'EF 04 E1 08BA 777 BBC #FELLTHRU,FLAGS,10$ ; CONTINUE IF NOT A FALL-THRU
0027 31 08C2 778 BRW SETSTBITX ; STATE BIT ALREADY SET ; JUST EXIT
08C5 779 10$:
000000A1'EF43 10 AC D1 08C5 780 CMPL 16(AP),CHMKADD[R3] ; PC POINTING TO WAIT'S CHMK ?
14 12 08CE 781 BNEQU 20$ ; NO -- GO SET 'PROCESSING'
12345678 8F 08 AC D1 08D0 782 CMPL 8(AP),#PROC_CONS ; YES -- PROCESSING CONST STILL SET IN RO ?
0A 13 08D8 783 BEQLU 20$ ; YES -- WAIT SERVICE NOT EXECUTED YET
00 0000011B'EF 03 E2 08DA 784 BBSS #WAITING,FLAGS,..+1 ; NO -- PROCESS IS WAITING
08 11 08E2 785 BRB SETSTBITX ; EXIT
08E4 786 20$:
00 0000011B'EF 02 E2 08E4 787 BBSS #PROCESSING,FLAGS,..+1 ; INDICATE STILL PROCESSING
08EC 788 SETSTBITX:
05 08EC 789 RSB ; RETURN TO CALLER (WAITAST RTN)

```

```

08ED 791 .SBTTL WAITAST ROUTINE
08ED 792 :
08ED 793 : THIS AST ROUTINE IS ENTERED AFTER A 1-MILLISECOND TIMER EXPIRES.
08ED 794 : THE TIMER IS SET JUST BEFORE ISSUING A WAIT SYSTEM SERVICE. SINCE
08ED 795 : THE WAIT SERVICE IS FOLLOWED BY A $HIBER, THE MAIN ROUTINE HAS
08ED 796 : TO HAVE BEEN IN ONE OF THE THREE FOLLOWING STATES AT TIME OF AST
08ED 797 : INTERRUPT:
08ED 798 :
08ED 799 : WAITING -- THE WAIT SYSTEM SERVICE CAUSED AN EVENT FLAG WAIT.
08ED 800 : THE WAIT WILL BE CLEARED BY ISSUING SETEF'S AND
08ED 801 : THE REC_WAIT BIT WILL BE SET.
08ED 802 :
08ED 803 : FELLTHRU -- THE WAIT SYSTEM SERVICE DID NOT CAUSE AN
08ED 804 : EVENT FLAG WAIT (THIS IS A FALL-THRU).
08ED 805 : A $HIBER FOLLOWING THE WAIT WILL BE
08ED 806 : CLEARED WITH A $WAKE AND THE REC_WAIT
08ED 807 : BIT WILL BE CLEARED.
08ED 808 :
08ED 809 : PROCESSING -- NEITHER OF THE ABOVE TWO STATES. IN THIS CASE,
08ED 810 : THE 1-MILLISECOND TIMER IS REPEATED, EXPECTING
08ED 811 : ONE OF THE OTHER TWO STATES TO BE REACHED EVENT-
08ED 812 : UALLY.
08ED 813 :
08ED 814 WAITAST:
08ED 815 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9> ; ENTRY MASK
08EF 816 BSBW REST_REGS ; RESTORE REGS 2-6 FOR USE IN AST
08F2 817 BSBW SETSTBIT ; DETERMINE STATE OF EXECUTION & SET BIT
36 0000011B'EF 02 E1 08F5 818 BBC #PROCESSING,FLAGS,10$ ; BRANCH IF NOT PROCESSING
OA 0000011A'EF 91 08FD 819 CMPB CONSEC_P,#10 ; 10 CONSECUTIVE 'PROCESSING' AST ENTRIES ?
0000011A'EF 1D 0904 820 BGEQ 5$ ; YES -- DO NOT RE-SCHEDULE THIS AST
0000011A'EF 96 0906 821 INCB CONSEC_P ; NO -- LOG THIS OCCURRENCE
090C 822 $SETIMR_S DAYTIM=TIMEINCR, - ; STILL PROCESSING; RE-SCHEDULE THIS AST
090C 823 ASTADR=WAITAST
00E6 31 0920 824 BRW WAITASTX ; .... EXIT BACK TO INTERRUPTED CODE
0923 825 5$:
00 0000011B'EF 05 E2 0923 826 BBSS #ASTLOOP,FLAGS,..+1 ; INDICATE ERROR FOR LATER PROCESSING
00 0000011B'EF 03 E2 092B 827 BBSS #WAITING,FLAGS,..+1 ; FAKE WAITING STATE TO CLEAR WAIT
0933 828 10$:
0000011A'EF 94 0933 829 CLRB CONSEC_P ; RE-INIT CNT OF CONSEC 'PROCESSING' ENTRIES
0939 830 :
0939 831 : THE FOLLOWING CODE READS THE CLUSTER & COMPARES IT
0939 832 : AGAINST TRIAL MASK FOR EQUALITY. IF EVENT FLAG GROUP 0,
0939 833 : COMPARE ONLY FLAGS 1-23, SINCE OTHERS MAY BE USED BY
0939 834 : SYSTEM OR TEST MODULE.
0939 835 :
0939 836 $REDEF_S EFN=EFN[R4], STATE=CLUS_STATE
094D 837 : GET STATE OF CLUSTER
20 0000024E'EF44 D1 094D 838 Cmpl EFN[R4],#32 ; EVENT FLAG GROUP 0 ?
07 19 0955 839 BLSS 14$ ; YES -- GO SPECIFY POS & SIZE FOR COMPARE
58 D4 0957 840 CLRL R8 ; NO -- SPECIFY POSITION FOR COMPARE
59 20 90 0959 841 MOVB #32,R9 ; SPECIFY SIZE OF TRIAL_MASK FOR COMPARE
06 11 095C 842 BRB 17$ ; GO DO COMPARE
095E 843 14$:
58 01 9A 095E 844 MOVZBL #1,R8 ; SPEC POSITION FOR E.F. GROUP 0
59 17 90 0961 845 MOVB #23,R9 ; .... AND SIZE
0964 846 17$:
57 00000116'EF 00000110'EF CD 0964 847 XORL3 TRIAL_MASK,CLUS_STATE,R7 ; GET XOR OF 2 COMPARANDS

```



```

00 57 59 58 EC 0970 848 CMPV R8,R9,R7,#0 ; COMPARE FOR BIT MATCHES USING POS & SIZE
1E 13 0975 849 BEQLU 20$ ; BITS MATCH -- CONTINUE
00000000'EF 00000110'EF D0 0977 850 MOVL TRIAL_MASK,EXPV ; MISMATCH -- LOAD EXPECTED ....
00000000'EF 00000116'EF DC 0982 851 MOVL CLUS_STATE,RECV ; ... AND RECEIVED VALUES
00 0000011B'EF 06 E2 098D 852 BBSS #CLUSCHG,FLAGS,..+1 ; INDICATE ERROR FOR LATER PROCESSING
0995 853 20$:
0995 854
03 0000011B'EF 03 E0 09A0 855 $WAKE_S ; WAKE SELF TO SATISFY $HIBER
005E 31 09A8 856 BBS #WAITING,FLAGS,22$ ; BRANCH IF WAITING
09AB 857 22$: BRW WAITASTX ; MUST HAVE FALLEN THRU WAIT; JUST EXIT
00 0000011B'EF 01 E2 09AB 858 BBSS #REC_WAIT,FLAGS,..+1 ; SET 'WAIT RECEIVED' BIT & KEEP GOING
52 D5 09B3 859 TSTL R2 ; FIRST CONDITION 1 ELEMENT ?
1E 13 09B5 860 BEQLU 30$ ; YES -- CLEAR WAIT IN CREATED PROCESS
58 0000024E'EF44 D0 09B7 861 MOVL EFN[R4],R8 ; NO -- CLEAR WAIT IN THIS PROCESS
59 D4 09BF 862 CLRL R9 ; ESTABLISH FIRST EFN IN CLUSTER
09C1 864 25$: ; INIT OFFSET OF EFN IN CLUSTER
09C1 865 $SETEF_S EFN=R8 ; SET THIS EVENT FLAG
FFEF 59 01 58 B6 09CA 866 INCW R8 ; POINT TO NEXT ONE
1F 9D 09CC 867 ACBB #31,#1,R9,25$ ; LOOP BACK TO SET NEXT ONE
0034 31 09D2 868 BRW WAITASTX ; ALL DONE -- WAIT IS CLEARED
09D5 869 30$:
09D5 870 $WAKE_S PRCNAM=CREPRN ; HAVE CREATED PROCESS ISSUE SETEF'S
09E4 871 $QIOW_S CHAN=MBXCHAN, FUNC=#IOS_READVBLK, -
09E4 872 P1=MBXBUF+8, P2=#1 ; ... WAIT UNTIL HE SIGNALS 'DONE'
0A09 873
0A09 874 WAITASTX:
04 0A09 875 RET ; TERMINATE THIS AST
0A0A 876 .END

```

\$\$\$\$	= 0000082B	R	04	CONDS_T	00000293	R	03
\$\$\$CHARS	= 0000001E			CONDS_TAB	00000293	R	03
\$\$\$CHARS1	= 0000000A			CONFLICT	*****	X	04
\$\$\$CHARS2	= 00000013			CONSEC_P	0000011A	R	03
\$\$\$CHARS3	= 00000000			CREPRN	00000051	R	02
\$\$\$CHARS4	= 00000000			CTL\$GL_PHD	*****	X	04
\$\$\$CHARS5	= 00000000			DESC	= 00000010	G	
\$\$\$COND_A	= 00000001			DIB\$K_LENGTH	= 00000074		
\$\$\$STRINGS	= 00000001			DIB\$W_UNIT	= 0000000C		
\$\$\$STRINGS2	= 00000005			EFGRO_MASK	= FF000001		
\$ST1	= 00000001			EFLAG	*****	X	04
\$ST2	= 00000004			EFN	0000024E	R	03
ASTLOOP	= 00000005			EXPV	*****	X	04
BUILD_CLUST	00000735	R	04	EXP_WAIT	= 00000000		
BUILD_CLUSTX	00000864	R	04	FAO_DESC	*****	X	04
BYTE	= 00000001	G		FAO_LEN	*****	X	04
CFLAG	*****	X	04	FELTHRU	= 00000004		
CHKADD	000000A1	R	02	FLAGS	0000011B	R	03
CHMRTN	*****	X	04	FORM_CONDS	000002F1	RG	04
CHM_CONT	*****	X	04	FORM_CONDSX	000003F2	R	04
CLUSCHG	= 00000006			IMAGNAM	0000007A	R	02
CLUS_NAME	00000065	R	02	IOS_READVBLK	*****	X	04
CLUS_STATE	00000116	R	03	LONG	= 00000004	G	
COMP_SC	*****	X	04	MASK	0000028B	R	03
COND	000002BC	RG	04	MBXBUF	0000008C	R	03
COND1_C	= 00000000			MBX^HAN	00000008	R	03
COND1_CLEANUP	000002BD	RG	04	MBXCHANINFO	0000000C	R	03
COND1_E	00000169	R	03	MBXUNIT	00000088	R	03
COND1_H	00000135	RG	03	MOD_MSG_CODE	*****	X	04
COND1_T	0000011D	R	03	MOD_MSG_PRINT	*****	X	04
COND1_TAB	00000136	R	03	MSGT_INP_CTL	00000019	R	02
COND2	000002BE	RG	04	MSG3_ERR_CTL	00000039	RG	02
COND2_C	= 00000000			MSG_A	*****	X	04
COND2_CLEANUP	000002BF	RG	04	MSG_B	*****	X	04
COND2_E	000001A9	R	03	MSG_CTXT	*****	X	04
COND2_H	00000185	RG	03	MSG_DATA1	*****	X	04
COND2_T	00000169	R	03	NOTARG	= 00000000	G	
COND2_TAB	00000186	R	03	NULL	= 00000014	G	
COND3	000002C0	RG	04	ONES	*****	X	04
COND3X	000002EB	R	04	OUTPUT_MSG	*****	X	04
COND3_C	= 00000004			PCV	*****	X	04
COND3_CLEANUP	000002EC	RG	04	PHD\$Q_PRIVMSK	= 00000000		
COND3_E	0000024E	R	03	PQL\$_BYTLM	= 00000003		
COND3_H	000001AD	RG	03	PQL\$_CPULM	= 00000004		
COND3_T	000001A9	R	03	PQL\$_FILLM	= 00000006		
COND3_TAB	000001AE	R	03	PQL\$_LISTEND	= 00000000		
COND4	000002ED	RG	04	PQL\$_PGFLQUOTA	= 00000007		
COND4_C	= 00000004			PQL\$_PRCLM	= 00000008		
COND4_CLEANUP	000002EE	RG	04	PQL\$_TQELM	= 00000009		
COND4_E	0000028B	R	03	PRIVMASK	= 00000000	R	03
COND4_H	00000263	RG	03	PRIV_ARGS	= 00000002		
COND4_T	0000025E	R	03	PROCESSING	= 00000002		
COND4_TAB	00000264	R	03	PROCESS_ERR	*****	X	04
COND5	000002EF	RG	04	PROCFLAG	0000011C	R	03
COND5_C	= 00000014			PROC_CONS	= 12345678		
COND5_CLEANUP	000002F0	RG	04	QUAD	= 00000008	G	
COND5_H	00000293	RG	03	QUOTALIST	000000AD	R	02

RCV	*****	X	04
REC_WAIT	= 00000001		
REST_REGS	*****	X	04
SAVE_REGS	*****	X	04
SETXPBIT	00000865	R	04
SETXPBITX	000008A9	R	04
SETSTBIT	000008AA	R	04
SETSTBITX	000008EC	R	04
SSS_NORMAL	*****	X	04
SUBJ_MASK	0000010C	R	03
SUCCESS	*****	X	04
SYSSASCEFC	*****	GX	04
SYSSCLREF	*****	GX	04
SYSSCMKRNL	*****	GX	04
SYSSCREMBX	*****	GX	04
SYSSCREPRC	*****	GX	04
SYSSDACEFC	*****	GX	04
SYSSDELMBX	*****	GX	04
SYSSFAO	*****	X	04
SYSSFORCEX	*****	GX	04
SYSSGETCHN	*****	GX	04
SYSSHIBER	*****	GX	04
SYSSQIOW	*****	GX	04
SYSSREADEF	*****	GX	04
SYSSSETEF	*****	GX	04
SYSSSETIMR	*****	GX	04
SYSSSETPRN	*****	GX	04
SYSSSETPRV	*****	GX	04
SYSSWAITFR	*****	GX	02
SYSSWAKE	*****	GX	04
SYSSWFLAND	*****	GX	02
SYSSWFLOR	*****	GX	02
TESTNUM	*****	X	04
TEST_MOD_NAME	00000000	RG	02
TEST_MOD_NAME_D	00000009	R	02
TEST_MOD_SUCC	*****	X	04
TIMEINCR	00000099	R	02
TMD_ADDR	*****	X	04
TM_CLEANUP	0000024A	RG	04
TM_SETUP	00000000	RG	04
TM_WORK	00000114	R	03
TRIAL_MASK	00000110	R	03
VERIFY	000003F3	RG	04
VERIFYX	00000733	R	04
VFY_CLEANUP	00000734	RG	04
WAITAST	000008ED	R	04
WAITASTX	00000A09	R	04
WAITING	= 00000003		
WORD	= 00000002	G	
WRITE_MSG2	*****	X	04

! 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
RODATA	000000D0 (208.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	00000294 (660.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSS56	00000A0A (2570.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.46
Command processing	107	00:00:00.63	00:00:03.37
Pass 1	325	00:00:10.04	00:00:22.96
Symbol table sort	0	00:00:00.69	00:00:00.92
Pass 2	176	00:00:02.84	00:00:03.95
Symbol table output	21	00:00:00.15	00:00:00.31
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	662	00:00:14.45	00:00:32.01

The working set limit was 1500 pages.
54362 bytes (107 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 388 non-local and 74 local symbols.
876 source lines were read in Pass 1, producing 29 object records in Pass 2.
55 pages of virtual memory were used to define 45 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[SHRLIB]UETP.MLB;1	10
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	31
TOTALS (all libraries)	42

801 GETS were required to define 42 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSS56/OBJ=OBJ\$:SATSS56 MSRC\$:SATSS56/UPDATE=(ENH\$:SATSS56)+EXECML\$/LIB+SHRLIB\$:UETP/LIB

