

BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS



(1)	389	COMBODEV MACRO
(1)	442	CLASSDEV MACRO
(1)	478	LOCAL offsets
(1)	566	AUTOCONFIGURATION TABLES
(1)	882	AUTO CONFIGURATION OF DEVICE DATA BASE
(1)	922	ACFSMBA - MASSBUS ADAPTER AUTO CONFIGURATION
(1)	1139	ACFSDR - DR32 ADAPTER AUTO CONFIGURATION
(1)	1161	ACFSCI - CI ADAPTER AUTO CONFIGURATION
(1)	1201	ACFSUBA - UNIBUS ADAPTER AUTO CONFIGURATION
(1)	1366	FIX DEV NAME - Check for system device name match/conflict
(1)	1397	LOAD DRIVER - Co-routine callback to load driver
(1)	1424	ACFSADD UNITS - GENERIC ROUTINE FOR DEVICE GENERATION
(1)	1547	CLASS DRIVER DEVICE GENERATOR
(1)	1641	RL11 MULTIPLE UNIT GENERATOR
(1)	1689	MULTIPLE DEVICE GENERATOR
(1)	1785	AUTO CONFIGURATION DEVICE DATA BASE RESET
(1)	1827	ROUTINE INC_CHAR
(1)	1893	ROUTINE CLR_ACF

```
0000 1 .TITLE AUTOCONFG - AUTO CONFIGURATION OF DEVICE DATA BASE
0000 2 .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5 *
0000 6 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
0000 7 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
0000 8 * ALL RIGHTS RESERVED. *
0000 9 *
0000 10 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000 11 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
0000 12 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
0000 13 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000 14 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
0000 15 * TRANSFERRED. *
0000 16 *
0000 17 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
0000 18 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
0000 19 * CORPORATION. *
0000 20 *
0000 21 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
0000 22 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
0000 23 *
0000 24 *
0000 25 *****
0000 26
0000 27 :++
0000 28
0000 29 D. N. CUTLER 19-JUN-78
0000 30
0000 31 AUTO CONFIGURATION OF DEVICE DATA BASE
0000 32
0000 33 MODIFIED BY:
0000 34
0000 35 V03-018 WHM0011 Bill Matthews 01-Aug-1984
0000 36 Check that the CSR for the system device is actually in unibus
0000 37 address space. The RB730 changes it's CSR address and therefore
0000 38 appears to autoconfigure to be a different device which causes
0000 39 the logic in FIX_DEV_NAME to fail.
0000 40
0000 41 V03-017 WHM0010 Bill Matthews 19-Jun-1984
0000 42 Made XI nosupport. Remove restriction that you cannot boot
0000 43 from a unibus device other than the first unibus unless there
0000 44 is exactly one device of that type on each preceding unibus.
0000 45
0000 46 V03-016 WHM0009 Bill Matthews 11-Apr-1984
0000 47 Changed the name of the QNA back to XQ. The name was already
0000 48 used in code that was blasted into ROM.
0000 49
0000 50 V03-015 WHM0008 Bill Matthews 05-Mar-1984
0000 51 Changed the name of the QNA from XQ to XH.
0000 52
0000 53 V03-014 WHM0007 Bill Matthews 23-Feb-1984
0000 54 Changed autoconfigure tables to have only one copy of
0000 55 the ascii device name string and driver name string.
0000 56 Use driver name 'OSDRIVER' when no driver is to be loaded.
0000 57 (e.g.DMF32,DMZ32,CPI32)
```

```

0000 58 :
0000 59 : V03-013 WHM0006 Bill Matthews 16-Feb-1984
0000 60 : Changed the device name of the DQ11 from XQ to OR.
0000 61 : Added support for the QVSS. Changed the name of IX from
0000 62 : IEX11 to IEQ11.
0000 63 :
0000 64 : V03-012 WHM0005 Bill Matthews 04-Feb-1984
0000 65 : Added support for the new field ACF$B_COMBO_VECTOR_OFFSET.
0000 66 : Cleans up support of combo style devices.
0000 67 :
0000 68 : V03-011 WHM0004 Bill Matthews 03-Feb-1984
0000 69 : Changed the device name of the DMV11 from OO to XD.
0000 70 : Added support for multiple devices with the same name.
0000 71 :
0000 72 : V03-010 WHM0003 Bill Matthews 02-Feb-1984
0000 73 : checked in v03-009 source by mistake.
0000 74 :
0000 75 : V03-009 WHM0002 Bill Matthews 29-Dec-1983
0000 76 : Added autoconfigure support for TC11.
0000 77 :
0000 78 : V03-008 WHM0001 Bill Matthews 13-Dec-1983
0000 79 : Rewrote support for autoconfiguring combo devices.
0000 80 :
0000 81 : V03-007 JLV0313 Jake VanNoy 18-Nov-1983
0000 82 : Put in support for first QNA. Also add DHV/DHU.
0000 83 : Remove NOSUPPORT flag from UNA.
0000 84 :
0000 85 : V03-006 MSH0003 Maryann Hinden 09-Feb-1983
0000 86 : Add support for cluster device names. Change TU to MU.
0000 87 :
0000 88 : V03-005 MSH0002 Maryann Hinden 28-Dec-1982
0000 89 : Calculate floating vector correctly for nth (n>1) device.
0000 90 :
0000 91 : V03-004 TCM0002 Trudy C. Matthews 22-Oct-1982
0000 92 : Use new field in ADP, ADPSL_AVECTOR, to calculate ACF$W_AVECTOR,
0000 93 : instead of using adapter's TR number (ADPSW_TR).
0000 94 :
0000 95 : V03-003 MSH0001 Maryann Hinden 07-SEP-1982
0000 96 : Add TU81 and VS100; change UNA device name to XE, and
0000 97 : add CSR and vector to allow booting from it.
0000 98 :
0000 99 : V03-002 JLV0203 Jake VanNoy 29-MAR-1982
0000 100 : Change DMF in UBADEV to COMBO to prevent ACF$INC_CHAR
0000 101 : from matching it with RK611's (DM). The code in
0000 102 : CHECK_OTHERS is not robust enough, changes in the
0000 103 : UNIBUS table may cause other problems.
0000 104 : Change UNA driver name to XEDRIVER.
0000 105 :
0000 106 : V03-001 JLV0197 Jake VanNoy 17-MAR-1982
0000 107 : Change the vector modulus from 4 (octal) to 10 (octal)
0000 108 : for the DC11 and TU58, and from 10 to 4 for the DR11B.
0000 109 : Remove two of the four CSR's for the DX11. Change the
0000 110 : name of the DUP11 driver to 'NODRIVER' so that we won't
0000 111 : load any driver instead of loading the wrong driver.
0000 112 : Add two Q bus devices - DMV11 and DPV11. Correct offsets
0000 113 : for the KMS11 and add the PCL11 as XP. Add patch space
0000 114 : to end of UBATABLE. Change floating CSR algorithm bug.

```

```

0000 115 : Make ACF$INC_CHAR smarter so that it gets device names
0000 116 : correct for all subsequent UNIBUS's.
0000 117 :--
0000 118 :
0000 119 : MACRO LIBRARY CALLS
0000 120 :
0000 121 :
0000 122 : $ACFDEF ;DEFINE ACF OFFSETS
0000 123 : $ADPDEF ;DEFINE ADP OFFSETS
0000 124 : $DCDEF ;DEFINE DEVICE CLASSES AND TYPES
0000 125 : $DDBDEF ;DEFINE DDB OFFSETS
0000 126 : $IDBDEF ;DEFINE IDB OFFSETS
0000 127 : $MBADEF ;DEFINE MBA REGISTER OFFSETS
0000 128 : $PDTDEF ;DEFINE PDT OFFSETS
0000 129 : $RPBDEF ;DEFINE RPB OFFSETS
0000 130 : $UBADEF ;DEFINE UBA REGISTER OFFSETS
0000 131 : $UCBDEF ;DEFINE UCB OFFSETS
0000 132 : $DPTDEF ;DEFINE DPT REGISTER OFFSETS
0000 133 :
0000 134 : $CRBDEF
0000 135 : $VECDEF
0000 136 :
0000 137 :
0000 138 : LOCAL MACROS
0000 139 :
0000 140 : FULSH INSTRUCTION BUFFER AND DELAY
0000 141 :
0000 142 : .MACRO FREEIB
0000 143 : NOP
0000 144 : NOP
0000 145 : NOP
0000 146 : NOP
0000 147 : NOP
0000 148 : NOP
0000 149 : NOP
0000 150 : NOP
0000 151 : .ENDM FREEIB
0000 152 :
0000 153 :
0000 154 : GENERATE ADAPTER TYPE CONTROL TABLE ENTRY
0000 155 :
0000 156 : ADAPTER TYPE
0000 157 :
0000 158 :
0000 159 : .MACRO ADAPTER TYPE
0000 160 : .WORD 0
0000 161 : .IRP X,<UBA,MBA,DR,CI>
0000 162 : .IF IDN <X>,<TYPE>
0000 163 : .=-2
0000 164 : $$$=
0000 165 : .WORD AT$ 'TYPE
0000 166 : .LONG ACF$ 'TYPE
0000 167 : ADAPTERLEN=-$$$
0000 168 : .MEXIT
0000 169 : .ENDC
0000 170 : .ENDM
0000 171 : .ENDM ADAPTER

```

```

0000 172
0000 173 :
0000 174 : GENERATE MBA DEVICE DESCRIPTOR
0000 175 :
0000 176 :     MBADEV  DEVNAME,DRVNAME,TYPES
0000 177 :
0000 178
0000 179     .MACRO  MBADEV  DEVNAME,DRVNAME,TYPES
0000 180         .IF NOT_DEFINED $'DEVNAME'$
0000 181         .PSECT -ACF$DEVNAME
0000 182     $'DEVNAME'$=
0000 183         .ASCIC  \'DEVNAME'A\
0000 184
0000 185     $$$=.
0000 186         .PSECT  ACF$RESET
0000 187         .ADDRESS $$$
0000 188
0000 189         .ENDC
0000 190
0000 191         .IF NOT_DEFINED $'DRVNAME'$
0000 192         .PSECT -ACF$DRVNAME
0000 193     $'DRVNAME'$=
0000 194         .ASCIC  \'DRVNAME\
0000 195
0000 196         .ENDC
0000 197
0000 198         .PSECT  ACF$DEVDESC
0000 199     SDEVDESC$=
0000 200         .LONG  $'DEVNAME'$
0000 201         .LONG  $'DRVNAME'$
0000 202         .IRP   X,<TYPES>
0000 203         .WORD  X
0000 204         .ENDM
0000 205         .WORD  0
0000 206         .PSECT  NONPAGED_DATA    rd,wrt,noexe,quad
0000 207         .LONG  SDEVDESC$
0000 208     .ENDM  MBADEV
0000 209
0000 210 :
0000 211 : GENERATE DR DEVICE DESCRIPTOR
0000 212 :
0000 213 :     DRDEV  DEVNAME,DRVNAME
0000 214 :
0000 215
0000 216     .MACRO  DRDEV  DEVNAME,DRVNAME
0000 217
0000 218         .IF NOT_DEFINED $'DEVNAME'$
0000 219         .PSECT -ACF$DEVNAME
0000 220     $'DEVNAME'$=
0000 221         .ASCIC  \'DEVNAME'A\
0000 222     $$$=.
0000 223         .PSECT  ACF$RESET
0000 224         .ADDRESS $$$
0000 225         .ENDC
0000 226
0000 227         .IF NOT_DEFINED $'DRVNAME'$
0000 228         .PSECT -ACF$DRVNAME

```

```
0000 229      $'DRVNAME'S=  
0000 230      .ASCIC  \DRVNAME\  
0000 231      .ENDC  
0000 232  
0000 233      .PSECT  ACF$DEVDESC  
0000 234      $DEVDESC$=  
0000 235      .LONG  $'DEVNAME'S  
0000 236      .LONG  $'DRVNAME'S  
0000 237      .PSECT  NONPAGED DATA  rd,wrt,noexe,quad  
0000 238      .LONG  $DEVDESC$  
0000 239      .ENDM  DRDEV
```



```

0000 241 :
0000 242 : GENERATE UBA DEVICE DESCRIPTOR
0000 243 :
0000 244 :
0000 245 :
0000 246 :
0000 247 : The UBADEV macro generates the following data structures:
0000 248 :
0000 249 : The first part is the same for fixed, fixed / fixed, floating / or
0000 250 : floating, floating
0000 251 :
0000 252 : structure: for example:
0000 253 :
0000 254 : +-----+
0000 255 : ! address of devname ! 0 ---> '3LPA' (ascic string)
0000 256 : +-----+
0000 257 : ! address of drvname ! 4 ---> '8LPDRIVER'
0000 258 : +-----+
0000 259 : ! address of rtnname (device) ! 8 ---> '4LP11'
0000 260 : +-----+
0000 261 : ! address of action routine ! 12 ---> ACF$LP11:
0000 262 : +-----+
0000 263 : ! initial controller !letter ! 16 'A' as in lp'A' (For IOC$RESET)
0000 264 : +-----+
0000 265 : ! numvec ! # ! 17 #1
0000 266 : +-----+
0000 267 : ! unused ! 18
0000 268 : +-----+
0000 269 : ! flags ! 19 m = 001 for fixed, fixed, support
0000 270 : +-----+
0000 271 :
0000 272 : And now, the structure changes based on vector and CSR assignment:
0000 273 :
0000 274 : Fixed CSR, Fixed vector:
0000 275 :
0000 276 : +-----+
0000 277 : ! first CSR ! 20
0000 278 : +-----+
0000 279 : ! first vector !
0000 280 : +-----+
0000 281 : ! 2nd CSR !
0000 282 : +-----+
0000 283 : ! 2nd vector !
0000 284 : +-----+
0000 285 : :
0000 286 : :
0000 287 : :
0000 288 : +-----+
0000 289 : ! last CSR !
0000 290 : +-----+
0000 291 : ! last vector !
0000 292 : +-----+
0000 293 : ! 0 ! zero word marks end of list
0000 294 : +-----+
0000 295 :
0000 296 :
0000 297 : Fixed CSR, Floating vector:

```

```

0000 298 :
0000 299 :
0000 300 : -----+
0000 301 : ! vector mask ! 20
0000 302 : -----+
0000 303 : ! first CSR !
0000 304 : -----+
0000 305 : ! 2nd CSR !
0000 306 : -----+
0000 307 :
0000 308 :
0000 309 : -----+
0000 310 : ! last CSR !
0000 311 : -----+
0000 312 : ! 0 ! zero word marks end of list
0000 313 : -----+
0000 314 :
0000 315 :
0000 316 : Floating CSR, Floating vector:
0000 317 :
0000 318 : -----+
0000 319 : ! vector mask ! 20
0000 320 : -----+
0000 321 : ! CSR mask !
0000 322 : -----+
0000 323 : (no zero word)
0000 324 :
0000 325 :
0000 326 :
0000 327 : .MACRO UBADEV DEVNAME,DRVNAME,ROUTINE,NUMVECT,CSRTYPE,-
0000 328 : VECTYPE,REGISTER,VECTOR,SUPPORT=SUPPORTED
0000 329 :
0000 330 : .IF NOT_DEFINED $'DEVNAME'$
0000 331 : .PSECT -ACF$DEVNAME
0000 332 : $'DEVNAME'$=
0000 333 : .ASCIC \DEVNAME'A\
0000 334 : .ENDC
0000 335 :
0000 336 : .IF NOT_DEFINED $'DRVNAME'$
0000 337 : .PSECT -ACF$DRVNAME
0000 338 : $'DRVNAME'$=
0000 339 : .ASCIC \DRVNAME\
0000 340 : .ENDC
0000 341 :
0000 342 : .PSECT ACF$RTNNAME
0000 343 : $RTNNAME$=
0000 344 : .ASCIC \ROUTINE\
0000 345 : .PSECT ACF$DEVDESC
0000 346 : $DEVDESC$=
0000 347 : .LONG $'DEVNAME'$
0000 348 : .LONG $'DRVNAME'$
0000 349 : .LONG $RTNNAME$
0000 350 : UBA_M_SUPPORT=1
0000 351 : UBA_V_SUPPORT=0
0000 352 : $$$=UBA_M_SUPPORT
0000 353 : .IF DIF <SUPPORT>,<SUPPORTED>
0000 354 : $$$=0

```

```

0000 355 .ENDC
0000 356 .LONG ACF$ROUTINE
0000 357 .BYTE ^A/A/
0000 358 .BYTE NUMVECT
0000 359 .BYTE 0 ; Unused
0000 360 UBA_M_FLOATCSR=2
0000 361 UBA_V_FLOATCSR=1
0000 362 UBA_M_FLOATVEC=4
0000 363 UBA_V_FLOATVEC=2
0000 364 .IIF IDN <VECTYPE>,<FLOAT>, $$$=$$$!UBA_M_FLOATVEC
0000 365 .IF IDN <CSRTYPE>,<FLOAT>
0000 366 .BYTE $$$!UBA_M_FLOATCSR
0000 367 .IF IDN <VECTYPE>,<FLOAT>
0000 368 .WORD VECTOR-1
0000 369 .ENDC
0000 370 .WORD REGISTER-1
0000 371 .IFF
0000 372 .BYTE $$$
0000 373 .IF IDN <VECTYPE>,<FLOAT>
0000 374 .WORD VECTOR-1
0000 375 .ENDC
0000 376 .WORD 0
0000 377 .IRP X,<REGISTER>
0000 378 .=-2
0000 379 .WORD X,0
0000 380 .ENDM
0000 381 .ENDC
0000 382 .PSECT NONPAGED_DATA rd,wrt,noexe,quad
0000 383 .IF NOT DEFINED ACF$AL_'ROUTINE'_'DEVNAME'
0000 384 ACF$AL_'ROUTINE'_'DEVNAME':
0000 385 .ENDC
0000 386 .LONG $DEVDESC$
0000 387 .ENDM UBADEV

```

```

0000 389 .SBTTL COMBODEV MACRO
0000 390
0000 391
0000 392 :+
0000 393 :
0000 394 : The COMBODEV macro is similar to the UBADEV macro. It is used to define
0000 395 : Unibus devices when more than one device appears on a single board (like
0000 396 : the DMF32).
0000 397 :
0000 398 :-
0000 399
0000 400 .MACRO COMBODEV DEVNAME,DRVNAME,NUMVECT,VEC_OFFSET,CSR_OFFSET,-
0000 401 MASK,SUPPORT=SUPPORTED
0000 402
0000 403 .IF NOT_DEFINED $'DEVNAME'$
0000 404 .PSECT -ACF$DEVNAME
0000 405 $'DEVNAME'$=.
0000 406 .ASCII \'DEVNAME'\
0000 407
0000 408 $$$=.
0000 409 .PSECT ACF$RESET
0000 410 .ADDRESS $$$
0000 411
0000 412 .ENDC
0000 413
0000 414 .IF NOT_DEFINED $'DRVNAME'$
0000 415 .PSECT -ACF$DRVNAME
0000 416 $'DRVNAME'$=.
0000 417 .ASCII \'DRVNAME'\
0000 418
0000 419 .ENDC
0000 420
0000 421 .PSECT ACF$DEVDESC
0000 422 $DEVDESC$=.
0000 423 .WORD MASK
0000 424 .LONG $'DEVNAME'$
0000 425 .LONG $'DRVNAME'$
0000 426 .BYTE NUMVECT
0000 427 .WORD VEC_OFFSET
0000 428 .LONG CSR_OFFSET
0000 429
0000 430 $$$ = UBA M SUPPORT
0000 431 .IF DIF <SUPPORT>,<SUPPORTED>
0000 432 $$$ = 0
0000 433 .ENDC
0000 434 .BYTE $$$
0000 435
0000 436 .PSECT NONPAGED DATA rd,wrt,noexe,quad
0000 437 .LONG $DEVDESC$
0000 438
0000 439 .ENDM COMBODEV
0000 440

```

```
0000 442      .SBTTL CLASSDEV MACRO
0000 443      :
0000 444      : THIS MACRO IS USED TO GENERATE A TABLE ASSOCIATING PORT DEVICE
0000 445      : NAMES WITH CLASS DRIVER AND DEVICE NAMES
0000 446      :
0000 447      :
0000 448      .MACRO CLASSDEV      DEVNAME,DRVNAME,PORTDEV
0000 449
0000 450      .PSECT ACF$DEVNAME
0000 451 $DEVNAME$ =
0000 452      .ASCII  \'DEVNAME'A\
0000 453 $$$ = .
0000 454
0000 455      .PSECT ACF$RESET
0000 456      .ADDRESS $$$
0000 457
0000 458      .PSECT ACF$DRVNAME
0000 459 $DRVNAME$ =
0000 460      .ASCII  \DRVNAME\
0000 461
0000 462      .PSECT ACF$DEVDESC
0000 463 $DEVDESC$ =
0000 464      .LONG   $DEVNAME$
0000 465      .LONG   $DRVNAME$
0000 466      .ASCII  \PORTDEV\
0000 467
0000 468
0000 469      .PSECT NONPAGED DATA      rd,wrt,noexe,quad
0000 470      .LONG   $DEVDESC$
0000 471
0000 472      .ENDM
0000 473
```

```
0000 475 :  
0000 476 : GLOBAL SYMBOLS  
0000 477 :  
0000 478 :SBTTL LOCAL offsets  
0000 479 :  
0000 480 : UBATABLE definitions  
0000 481 :  
0000 482 :  
00000000 0000 483 UBT$$_DEVNAME == 0  
00000004 0000 484 UBT$$_DRVNAME == 4  
00000008 0000 485 UBT$$_RTNNAME == 8  
0000000C 0000 486 UBT$$_ROUTINE == 12  
00000010 0000 487 UBT$$_LETTER == 16  
00000011 0000 488 UBT$$_NUMVEC == 17  
00000012 0000 489 UBT$$_UNUSED == 18  
00000013 0000 490 UBT$$_FLAGS == 19  
00000014 0000 491 UBT$$_REMAINDER == 20  
0000 492 :  
0000 493 :  
0000 494 : LOCAL SYMBOLS  
0000 495 :  
0000 496 :  
0000 497 :  
0000 498 : CLASSDEV OFFSETS  
0000 499 :  
0000 500 :  
00000000 0000 501 CLS$$_CLASSDEV = 0  
00000004 0000 502 CLS$$_CLASSDRV = 4  
00000009 0000 503 CLS$$_PORTDEV = 9  
0000 504 :  
0000 505 :  
0000 506 : DMF32 OFFSETS  
0000 507 :  
0000 508 :  
0000000C 0000 509 DMF$$_IDENT = 12  
00000004 0000 510 DMF$$_IDENT = 4  
0000 511 :  
0000 512 :  
0000 513 : CPI32/DMZ32 OFFSETS  
0000 514 :  
00000008 0000 515 CPI$$_SUBCNTRL = 8  
00000004 0000 516 CPI$$_SUBCNTRL = 4  
0000 517 :  
0000 518 :  
0000 519 : MBA DRIVE REGISTER OFFSET DEFINITIONS  
0000 520 :  
0000 521 :  
00000004 0000 522 MBA_DS=4 ;DRIVE STATUS REGISTER  
00000018 0000 523 MBA_DT=24 ;DRIVE TYPE REGISTER  
0000 524 :  
0000 525 :  
0000 526 : TM03 TAPE CONTROLLER REGISTER DEFINITIONS  
0000 527 :  
0000 528 :  
00000024 0000 529 TM03_TC=36 ; TAPE CONTROL REGISTER  
0000 530 :  
0000 531 :
```

```

0000 532 ; TM78 TAPE CONTROLLER REGISTER AND COMMAND DEFINITIONS
0000 533 ;
0000 534 ;
00000010 0000 535 TM78_AB=16 ; ATTENTION BIT REGISTER
0000001C 0000 536 TM78_DS=28 ; DEVICE STATUS REGISTER
0000002C 0000 537 TM78_NDTA=44 ; NON-DATA TRANSFER INTERRUPT STATUS REG
00000030 0000 538 TM78_NDT0=48 ; NON-DATA TRANSFER COMMAND REG #0
00000044 0000 539 TM78_ID=68 ; INTERNAL DATA REGISTER
00004000 0000 540 TM78_M_TMCLR=^X4000 ; BIT MASK FOR TMCLR BIT IN ID REGISTER
00008000 0000 541 TM78_M_TMRDY=^X8000 ; BIT MASK FOR TMRDY BIT IN ID REGISTER
00000009 0000 542 TM78_SENSE_GO=9 ; SENSE COMMAND AND GO BIT
0000 543 ;
0000 544 ;
0000 545 ;
0000 546 ; RL11 CONTROLLER REGISTER DEFINITIONS
0000 547 ;
0000 548 ;
00000000 0000 549 RL_CS=0 ; CONTROL STATUS REGISTER
00000400 0000 550 RL_CS_M_OPI=^X400 ; OPERATION INCOMPLETE
00000004 0000 551 RL_DA=4 ; DISK ADDRESS REGISTER
00000001 0000 552 RL_DA_M_MRK=1 ; MARK
00000002 0000 553 RL_DA_M_STS=2 ; GET STATUS
00000008 0000 554 RL_DA_M_RST=8 ; RESET DRIVE
0000 555 ;
0000 556 ;
0000 557 ; UBA I/O PAGE OFFSET DEFINITION
0000 558 ;
0000 559 ;
00001000 0000 560 UBA_IOBASE=8*512 ;
0000 561 ;
0000 562 ;
0000 563 ; LOCAL DATA
0000 564 ;

```

```
0000 566 .SBTTL AUTOCONFIGURATION TABLES
0000 567 :
0000 568 : ADAPTER TYPE CONTROL TABLE
0000 569 :
0000 570 :
0000 571 .LIST MEB
00000000 572 .PSECT NONPAGED_DATA rd,wrt,noexe,quad
0000 573 ACF$AB_ADPTYPE:
0000 574 ADAPTER MBA ;MASSBUS ADAPTER
0000 0000 .WORD 0
00000000 0002 .=-2
0000 0000 .WORD AT$ MBA
00000031' 0002 .LONG ACF$MBA
0000 0006 575 ADAPTER UBA ;UNIBUS ADAPTER
00000006 0008 .WORD 0
0001 0006 .=-2
00000285' 0008 .WORD AT$ UBA
0000 000C 576 ADAPTER DR ; DR32 ADAPTER
0000000C 000E .WORD 0
0002 000C .=-2
00000246' 000E .WORD AT$ DR
0000 0012 577 ADAPTER CI ; CI ADAPTER
00000012 0014 .WORD 0
0004 0012 .=-2
0000024D' 0014 .WORD AT$ CI
0000 0018 578 ADAPTER NULL ;
0000 0018 .WORD 0
001A 579 :
001A 580 :
001A 581 : CONTROLLER DESIGNATOR RESET POINTER TABLE
001A 582 :
001A 583 :
00000000 584 .PSECT ACF$RESET
0000 585 ACF$AL_RESET: ;TABLE BEGINNING
0000 586 :
0000 587 :
0000 588 : MASSBUS ADAPTER CONFIGURATION CONTROL TABLE
0000 589 :
0000 590 :
0000001A 591 .PSECT NONPAGED_DATA rd,wrt,noexe,quad ;
001A 592 ACF$AB_MBATABLE:
001A 593 MBADEV DB,DBDRIVER,<<^X10>,<^X11>,<^X12>> ;RPOX
00000000 .PSECT ACF$DEVNAME
41 42 44 00' 0000 .ASCIC \DBA\
03 0000
00000000 .PSECT ACF$RESET
00000004' 0000 .ADDRESS $$$
00000000 .PSECT ACF$DRVNAME
52 45 56 49 52 44 42 44 00' 0000 .ASCIC \DBDRIVER\
08 0000
00000000 .PSECT ACF$DEVDESC
00000000' 0000 .LONG $DB$
00000000' 0004 .LONG $DBDRIVERS
0010 0008 .WORD ^X10
```



	0011 000A			.WORD	^X11	
	0012 000C			.WORD	^X12	
	0000 000E			.WORD	0	
	0000001A			.PSECT	NONPAGED DATA	rd,wrt,noexe,quad
	00000000' 001A			.LONG	\$DEVDESC\$	
	001E	594	MBADEV	DR,DRDRIVER, <<^X14>,<^X16>,<^X17>,<^X22>>		;RMOX
	00000004			.PSECT	ACF\$DEVNAME	
41 52 44	00' 0004			.ASCII	\DRA\	
	03 0004					
	00000004			.PSECT	ACF\$RESET	
	00000008' 0004			.ADDRESS	\$\$\$	
	00000009			.PSECT	ACF\$DRVNAME	
52 45 56 49 52 44 52 44	00' 0009			.ASCII	\DRDRIVER\	
	08 0009					
	00000010			.PSECT	ACF\$DEVDESC	
	00000004' 0010			.LONG	\$DR\$	
	00000009' 0014			.LONG	\$DRDRIVERS	
	0014 0018			.WORD	^X14	
	0016 001A			.WORD	^X16	
	0017 001C			.WORD	^X17	
	0022 001E			.WORD	^X22	
	0000 0020			.WORD	0	
	0000001E			.PSECT	NONPAGED DATA	rd,wrt,noexe,quad
	00000010' 001E			.LONG	\$DEVDESC\$	
	0022	595	MBADEV	MT,TMDRIVER, <<^XC028>>		; TM03 CONTOLLER SPECIFIC DATA
	00000008			.PSECT	ACF\$DEVNAME	
41 54 4D	00' 0008			.ASCII	\MTA\	
	03 0008					
	00000008			.PSECT	ACF\$RESET	
	0000000C' 0008			.ADDRESS	\$\$\$	
	00000012			.PSECT	ACF\$DRVNAME	
52 45 56 49 52 44 4D 54	00' 0012			.ASCII	\TMDRIVER\	
	08 0012					
	00000022			.PSECT	ACF\$DEVDESC	
	00000008' 0022			.LONG	\$MT\$	
	00000012' 0026			.LONG	\$TMDRIVERS	
	C028 002A			.WORD	^XC028	
	0000 002C			.WORD	0	
	00000022			.PSECT	NONPAGED DATA	rd,wrt,noexe,quad
	00000022' 0022			.LONG	\$DEVDESC\$	
	0026	596	MBADEV	MF,TFDRIVER, <<^XC040>>		; TM78
	0000000C			.PSECT	ACF\$DEVNAME	
41 46 4D	00' 000C			.ASCII	\MFA\	
	03 000C					
	0000000C			.PSECT	ACF\$RESET	
	00000010' 000C			.ADDRESS	\$\$\$	
	0000001B			.PSECT	ACF\$DRVNAME	
52 45 56 49 52 44 46 54	00' 001B			.ASCII	\TFDRIVER\	
	08 001B					
	0000002E			.PSECT	ACF\$DEVDESC	
	0000000C' 002E			.LONG	\$MFS	
	0000001B' 0032			.LONG	\$TFDRIVERS	
	C040 0036			.WORD	^XC040	
	0000 0038			.WORD	0	
	00000026			.PSECT	NONPAGED DATA	rd,wrt,noexe,quad
	0000002E' 0026			.LONG	\$DEVDESC\$	
	00000000 002A	597	.LONG	0		;

```

002E 598
002E 599
002E 600 : DR32 ADAPTER CONFIGURATION CONTROL TABLE
002E 601 :
002E 602
0000002E 603 .PSECT NONPAGED_DATA rd,wrt,noexe,quad
002E 604 ACF$AB_DRTABLE:
002E 605 DRDEV XF,XFDRIVER
41 46 58 00' 0010 .PSECT ACF$DEVNAME
03 0010 .ASCIC \XFA\
00000010 .PSECT ACF$RESET
00000014' 0010 .ADDRESS $$$
00000024 .PSECT ACF$DRVNAME
52 45 56 49 52 44 46 58 00' 0024 .ASCIC \XFDRIVER\
08 0024
0000003A .PSECT ACF$DEVDESC
00000010' 003A .LONG $XF$
00000024' 003E .LONG $XFDRIVER$
0000002E .PSECT NONPAGED_DATA rd,wrt,noexe,quad
0000003A' 002E .LONG $DEVDESC$
0032 606
0032 607 :
0032 608 : CI ADAPTER CONFIGURATION CONTROL TABLE (uses DRDEV macro)
0032 609 :
0032 610
0032 611 ACF$AB_CITABLE:
0032 612 DRDEV PA,PADRIVER
41 41 50 00' 0014 .PSECT ACF$DEVNAME
03 0014 .ASCIC \PAA\
00000014 .PSECT ACF$RESET
00000018' 0014 .ADDRESS $$$
0000002D .PSECT ACF$DRVNAME
52 45 56 49 52 44 41 50 00' 002D .ASCIC \PADRIVER\
08 002D
00000042 .PSECT ACF$DEVDESC
00000014' 0042 .LONG $PAS
0000002D' 0046 .LONG $PADRIVER$
00000032 .PSECT NONPAGED_DATA rd,wrt,noexe,quad
00000042' 0032 .LONG $DEVDESC$
0036 613
0036 614 :
0036 615 : ADDRESS OF UBA DEVICE GENERATION ROUTINE AND RETURN
0036 616 :
0036 617
0000003A 0036 618 ACF$SL_RETURN: :
0036 619 .BLKL 1 :
0000003E 003A 620 ACF$SL_RETURN2: : Alternate
003E 621 .BLKL 1 :
00000042 003E 622 ACF$SL_ROUTINE: :
0042 623 .BLKL 1 :
00000046 0042 624 ACF$SL_DELIVER_UNIT: :
0046 625 .BLKL 1 :
0000006E 0046 626 ACF$SL_ACF_SAVE: : Save area for ACF block
0046 627 .BLKB ACF$C_LENGTH :
006E 628 ACF$W_VECMOD: : Save for floating vector modulus
    
```

```

00000070 006E 629 .BLKW 1
          0070 630
          0070 631 : REGISTER SAVE AREA
          0070 632 :
          0070 633 :
          0070 634
          0070 635 ACF$$_R0SAVE:
00000074 0070 636 .BLKL 1
          0074 637 ACF$$_R1SAVE:
00000078 0074 638 .BLKL 1
          0078 639 ACF$$_R2SAVE:
0000007C 0078 640 .BLKL 1
          007C 641 ACF$$_R3SAVE:
00000080 007C 642 .BLKL 1
          0080 643 ACF$$_R4SAVE:
00000084 0080 644 .BLKL 1
          0084 645 ACF$$_R5SAVE:
00000088 0084 646 .BLKL 1
          0088 647 ACF$$_R2R3SAVE:
00000090 0088 648 .BLKL 2
          0090 649
          0090 650 :
          0090 651 : UNIBUS ADAPTER CONFIGURATION CONTROL TABLE
          0090 652 :
          0090 653 : NOTE: ALL FLOATING ADDRESS ENTRIES MUST BE IN PRIORITY ORDER.
          0090 654 :
          0090 655
          0090 656 .NLIST MEB ; Cut down listing size
          0090 657
          0090 658 ACF$$_UBATABLE::
          0090 659 UBADEV CR,CRDRIVER,CR11,1,FIXED,FIXED,<<^017160,^0230>> ;
          0094 660 UBADEV DM,DMDRIVER,RK611,1,FIXED,FIXED,<<^017440,^0210>> ;
          0098 661 UBADEV LP,LPDRIVER,LP11,1,FIXED,FIXED,<-
          0098 662 <^017514,^0200>,-
          0098 663 <^004004,^0170>,-
          0098 664 <^004014,^0174>,-
          0098 665 <^004024,^0270>,-
          0098 666 <^004034,^0274>>
          009C 667 UBADEV DL,DLDRIVER,RL11,1,FIXED,FIXED,<<^014400,^0160>> ;
          00A0 668 UBADEV MS,TSDRIVER,TS11,1,FIXED,FIXED,<<^012520,^0224>> ;
          00A4 669 :
          00A4 670 : NOTE: THE CSR AND VECTOR ASSIGNED TO THE RX02 COINCIDES WITH
          00A4 671 : THE RX01. ANY DEVICE AT THIS CSR IS ASSUMED TO BE AN RX02.
          00A4 672 :
          00A4 673 UBADEV DY,DYDRIVER,RX211,1,FIXED,FIXED,<<^017170,^0264>> ;
          00A8 674 UBADEV DQ,DQDRIVER,RB730,1,FIXED,FIXED,<<^015606,^0250>> ;
          00AC 675 UBADEV PU,PUDRIVER,UDA,1,FIXED,FIXED,<<^012150,^0154>>
          00B0 676 UBADEV PT,PUDRIVER,TU81,1,FIXED,FIXED,<<^014500,^0260>>
          00B4 677 UBADEV XE,XEDRIVER,UNA,1,FIXED,FIXED,<<^014510,^0120>>
          00B8 678 UBADEV XQ,XQDRIVER,QNA,1,FIXED,FIXED,<<^014440,^0120>>
          00BC 679 UBADEV OM,OMDRIVER,DC11,2,FIXED,FLOAT,<-
          00BC 680 <^014000>,<^014010>,<^014020>,<^014030>,-
          00BC 681 <^014040>,<^014050>,<^014060>,<^014070>,-
          00BC 682 <^014100>,<^014110>,<^014120>,<^014130>,-
          00BC 683 <^014140>,<^014150>,<^014160>,<^014170>,-
          00BC 684 <^014200>,<^014210>,<^014220>,<^014230>,-
          00BC 685 <^014240>,<^014250>,<^014260>,<^014270>,-
    
```

00BC	686		<^014300>,<^014310>,<^014320>,<^014330>,-
00BC	687		<^014340>,<^014350>,<^014360>,<^014370>>,8,NOSUPPORT
00C0	688	UBADEV	DD,DDDRIVER,TU58,2,FIXED,FLOAT,-
00C0	689		<^016500>,<^016510>,<^016520>,<^016530>,-
00C0	690		<^016540>,<^016550>,<^016560>,<^016570>,-
00C0	691		<^016600>,<^016610>,<^016620>,<^016630>,-
00C0	692		<^016640>,<^016650>,<^016660>,<^016670>>,8
00C4	693	UBADEV	OB,OBDRIVER,DN11,1,FIXED,FLOAT,-
00C4	694		<^015200>,<^015210>,<^015220>,<^015230>,-
00C4	695		<^015240>,<^015250>,<^015260>,<^015270>,-
00C4	696		<^015300>,<^015310>,<^015320>,<^015330>,-
00C4	697		<^015340>,<^015350>,<^015360>,<^015370>>,4,NOSUPPORT
00C8	698	UBADEV	YM,YMDRIVER,DM18,1,FIXED,FLOAT,-
00C8	699		<^010500>,<^010510>,<^010520>,<^010530>,-
00C8	700		<^010540>,<^010550>,<^010560>,<^010570>,-
00C8	701		<^010600>,<^010610>,<^010620>,<^010630>,-
00C8	702		<^010640>,<^010650>,<^010660>,<^010670>>,4,NOSUPPORT
00CC	703	UBADEV	OA,OADRIVER,DR11C,2,FIXED,FLOAT,-
00CC	704		<^07600>,<^07570>,<^07560>,<^07550>,-
00CC	705		<^07540>,<^07530>,<^07520>,<^07510>,-
00CC	706		<^07500>,<^07470>,<^07460>,<^07450>,-
00CC	707		<^07440>,<^07430>,<^07420>,<^07410>>,8,NOSUPPORT
00D0	708	UBADEV	PR,PRDRIVER,PR611,1,FIXED,FLOAT,-
00D0	709		<^012600>,<^012604>,<^012610>,<^012614>,-
00D0	710		<^012620>,<^012624>,<^012630>,<^012634>>,8,NOSUPPORT
00D4	711	UBADEV	PP,PPDRIVER,PP611,1,FIXED,FLOAT,-
00D4	712		<^012700>,<^012704>,<^012710>,<^012714>,-
00D4	713		<^012720>,<^012724>,<^012730>,<^012734>>,8,NOSUPPORT
00D8	714	UBADEV	OC,OCDRIVER,DT11,2,FIXED,FLOAT,-
00D8	715		<^017420>,<^017422>,<^017424>,<^017426>,-
00D8	716		<^017430>,<^017432>,<^017434>,<^017436>>,8,NOSUPPORT
00DC	717	UBADEV	OD,ODDRIVER,DX11,2,FIXED,FLOAT,-
00DC	718		<^016200>,<^016240>>,8,NOSUPPORT
00E0	719	UBADEV	YL,YLDRIVER,DL11C,2,FIXED,FLOAT,-
00E0	720		<^015610>,<^015620>,<^015630>,<^015640>,-
00E0	721		<^015650>,<^015660>,<^015670>,<^015700>,-
00E0	722		<^015710>,<^015720>,<^015730>,<^015740>,-
00E0	723		<^015750>,<^015760>,<^015770>,<^016000>,-
00E0	724		<^016010>,<^016020>,<^016030>,<^016040>,-
00E0	725		<^016050>,<^016060>,<^016070>,<^016100>,-
00E0	726		<^016110>,<^016120>,<^016130>,<^016140>,-
00E0	727		<^016150>,<^016160>,<^016170>>,8,NOSUPPORT
00E4	728	UBADEV	YJ,YJDRIVER,DJ11,2,FLOAT,FLOAT,8,8,NOSUPPORT
00E8	729	UBADEV	YH,YHDRIVER,DH11,2,FLOAT,FLOAT,16,8,NOSUPPORT
00EC	730	UBADEV	OE,OEDRIVER,GT40,4,FIXED,FLOAT,<<^012000>,<^012010>>,8,NOSUPPORT
00F0	731	UBADEV	LS,LSDRIVER,LPS11,6,FIXED,FLOAT,<<^010400>>,8,NOSUPPORT
00F4	732	UBADEV	OR,ORDRIVER,DQ11,2,FLOAT,FLOAT,8,8,NOSUPPORT
00F8	733	UBADEV	OF,OFDRIVER,KW11W,2,FIXED,FLOAT,<<^012400>>,8,NOSUPPORT
00FC	734	UBADEV	XU,XUDRIVER,DU11,2,FLOAT,FLOAT,8,8,NOSUPPORT
0100	735	UBADEV	XW,OODRIVER,DUP11,2,FLOAT,FLOAT,8,8,NOSUPPORT ;OODRIVER means don'
0104	736	UBADEV	XV,XVDRIVER,DV11,3,FIXED,FLOAT,-
0104	737		<^015000>,<^015040>,<^015100>,<^015140>>,8,NOSUPPORT
0108	738	UBADEV	OG,OGDRIVER,LK11,2,FLOAT,FLOAT,8,8,NOSUPPORT
010C	739	UBADEV	XM,XMDRIVER,DMC11,2,FLOAT,FLOAT,8,8
0110	740	UBADEV	TTA,DZDRIVER,DZ11,2,FLOAT,FLOAT,8,8
0114	741	UBADEV	XK,XKDRIVER,KMC11,2,FLOAT,FLOAT,8,8,NOSUPPORT
0118	742	UBADEV	OH,OHDRIVER,LPP11,2,FLOAT,FLOAT,8,8,NOSUPPORT

```

011C 743 UBADEV OI,OIDRIVER,VMV21,2,FLOAT,FLOAT,8,8,NOSUPPORT
0120 744 UBADEV OJ,OJDRIVER,VMV31,2,FLOAT,FLOAT,16,8,NOSUPPORT
0124 745 UBADEV OK,OKDRIVER,DWR70,2,FLOAT,FLOAT,8,8,NOSUPPORT
0128 746 UBADEV DL,DLDRIVER,RL11,1,FLOAT,FLOAT,8,4
012C 747 UBADEV MS,TSRIVER,TS11,1,FIXED,FLOAT,<-
012C 748 <^012524>,<^012530>,<^012534>>,4
0130 749 UBADEV LA,LADRIVER,LPA11,2,FIXED,FLOAT,<-
0130 750 <^010460>>,8
0134 751 UBADEV LA,LADRIVER,LPA11,2,FLOAT,FLOAT,16,8
0138 752 UBADEV OL,OLDRIVER,KW11C,2,FLOAT,FLOAT,8,8,NOSUPPORT
013C 753 :
013C 754 : RESERVED FLOATING CSR: SHOULD NEVER BE OCCUPIED BY A
013C 755 : PHYSICAL DEVICE SINCE IT IS RESERVED.
013C 756 :
013C 757 : UBADEV RSV,RSVDRIVER,RSV,1,FLOAT,FLOAT,8,8,NOSUPPORT
0140 758 :
0140 759 : RX02'S AFTER THE FIRST: (NOTE: POSITION OF FIRST RX01 COINCIDES WITH
0140 760 : RX02 AND IS ALWAYS ASSUMED BY AUTOCONFIG TO BE RX02.)
0140 761 :
0140 762 : UBADEV DY,DYDRIVER,RX211,1,FLOAT,FLOAT,8,4
0144 763 UBADEV XA,XADRIVER,DR11W,1,FLOAT,FLOAT,8,4
0148 764 :
0148 765 : THE FIRST DR11B HAS FIXED CSR AND VECTOR AND NO SUPPORT; IT
0148 766 : THEREFORE HAS NO EFFECT ON AUTOCONFIGURATION.
0148 767 : THE SECOND DR11B HAS FIXED CSR AND FLOATING VECTOR.
0148 768 : DR11B'S AFTER THE SECOND HAVE BOTH FLOATING CSR AND VECTOR.
0148 769 :
0148 770 UBADEV XB,XBDRIVER,DR11B,1,FIXED,FIXED,<<^012410,^0124>>,-
0148 771 0,NOSUPPORT ; '0' is a placeholder here
014C 772 UBADEV XB,XBDRIVER,DR11B,1,FIXED,FLOAT,<<^J12430>>,4,NOSUPPORT
0150 773 UBADEV XB,XBDRIVER,DR11B,1,FLOAT,FLOAT,8,8,NOSUPPORT
0154 774 :
0154 775 UBADEV XD,XDDRIVER,DMP11,2,FLOAT,FLOAT,8,8
0158 776 UBADEV ON,ONDRIVER,DPV11,2,FLOAT,FLOAT,8,8,NOSUPPORT
015C 777 UBADEV IS,ISDRIVER,ISB11,2,FLOAT,FLOAT,8,8,NOSUPPORT
0160 778 UBADEV XD,XDDRIVER,DMV11,2,FLOAT,FLOAT,16,8
0164 779 UBADEV XE,XEDRIVER,UNA .1,FLOAT,FLOAT,8,4
0168 780 UBADEV PU,PUDRIVER,UDA .1,FLOAT,FLOAT,4,4
016C 781 UBADEV COMBO,OSDRIVER,DMF32,8,FLOAT,FLOAT,32,4 ;OSDRIVER means don't load a
0170 782 UBADEV XS,XSDRIVER,KMS11,3,FLOAT,FLOAT,16,8,NOSUPPORT
0174 783 UBADEV XP,XPDRIVER,PCL11,2,FIXED,FLOAT,<<^004200>,<^004240>,-
0174 784 <^004300>,<^004340>>,8,NOSUPPORT
0178 785 UBADEV VB,VBDRIVER,VS100,1,FLOAT,FLOAT,16,4,NOSUPPORT
017C 786 UBADEV PT,PUDRIVER,TU81 .1,FLOAT,FLOAT,4,4
0180 787 UBADEV OQ,OQDRIVER,KMV11,2,FLOAT,FLOAT,16,8,NOSUPPORT
0184 788 UBADEV UK,UKDRIVER,KCT32,2,FIXED,FLOAT,<<^004400>,<^004440>,-
0184 789 <^004500>,<^004540>>,8,NOSUPPORT
0188 790 UBADEV IX,IXDRIVER,IEQ11,2,FIXED,FLOAT,<<^004100>>,8,NOSUPPORT
018C 791 UBADEV TX,YFDRIVER,DHV11,2,FLOAT,FLOAT,16,8
0190 792 UBADEV COMBO,OSDRIVER,DM232,6,FLOAT,FLOAT,32,4 ;OSDRIVER means don't load a
0194 793 UBADEV COMBO,OSDRIVER,CPI32,6,FLOAT,FLOAT,32,4 ;OSDRIVER means don't load a
0198 794 UBADEV DT,DTDRIVER,TC11,1,FIXED,FIXED,<<^017340>,<^0214>>,-
0198 795 0,NOSUPPORT ; '0' here is a placeholder
019C 796 UBADEV VC,VCDRIVER,VC01B,2,FIXED,FIXED,<<^017200>,<^0060>>
00000000 01A0 797 .LONG 0 ; END OF LIST
000001F4 01A4 798
01A4 799 .BLKL 20 ; PATCH SPACE
    
```

```
01F4 800
01F4 801 .LIST MEB
01F4 802 ACF$AB_DMF32_TABLE:
01F4 803
01F4 804 : COMBODEV DEVNAME,DRVNAME,NUMVECT,VEC_OFFSET,CSR_OFFSET,
01F4 805 : MASK,SUPPORT=SUPPORTED
01F4 806
01F4 807 COMBODEV XG, XGDRIVER, 2, 0, 4, 4 : Sync
000000FD .PSECT ACF$DEVNAME
41 47 58 00' 00FD .ASCIC \XGA\
03 00FD
00000018 .PSECT ACF$RESET
00000101' 0018 .ADDRESS $$$
00000226 .PSECT ACF$DRVNAME
52 45 56 49 52 44 47 58 00' 0226 .ASCIC \XGDRIVER\
08 0226
00000834 .PSECT ACF$DEVDESC
0004 0834 .WORD 4
000000FD' 0836 .LONG $XGS
00000226' 083A .LONG $XGDRIVERS
02 083E .BYTE 2
0000 083F .WORD 0
00000004 0841 .LONG 4
01 0845 .BYTE $$$
000001F4 .PSECT NONPAGED DATA rd,wrt,noexe,quad
00000834' 01F4 .LONG $DEVDESC$
01F8 808 COMBODEV TX, YCDRIVER, 2, 16, 12, 8 : Async
0000022F .PSECT ACF$DRVNAME
52 45 56 49 52 44 43 59 00' 022F .ASCIC \YCDRIVER\
08 022F
00000846 .PSECT ACF$DEVDESC
0008 0846 .WORD 8
000000F1' 0848 .LONG $TXS
0000022F' 084C .LONG $YCDRIVERS
02 0850 .BYTE 2
0010 0851 .WORD 16
0000000C 0853 .LONG 12
01 0857 .BYTE $$$
000001F8 .PSECT NONPAGED DATA rd,wrt,noexe,quad
00000846' 01F8 .LONG $DEVDESC$
01FC 809 COMBODEV XI, XIDRIVER, 2, 8, 24, 1,NOSUPPORT; DR
00000101 .PSECT ACF$DEVNAME
41 49 58 00' 0101 .ASCIC \XIA\
03 0101
0000001C .PSECT ACF$RESET
00000105' 001C .ADDRESS $$$
00000238 .PSECT ACF$DRVNAME
52 45 56 49 52 44 49 58 00' 0238 .ASCIC \XIDRIVER\
08 0238
00000858 .PSECT ACF$DEVDESC
0001 0858 .WORD 1
00000101' 085A .LONG $XIS
00000238' 085E .LONG $XIDRIVERS
02 0862 .BYTE 2
0008 0863 .WORD 8
00000018 0865 .LONG 24
00 0869 .BYTE $$$
```

```
000001FC      .PSECT NONPAGED DATA   rd,wrt,noexe,quad
00000858' 01FC      .LONG   $DEVDESC$
      0200      810      COMBODEV LC, LCDRIVER,      1, 24, 20, 2      ; LP
41 43 4C 00' 0105      .PSECT ACF$DEVNAME
      03 0105      .ASCIC \LCA\
      00000020      .PSECT ACF$RESET
00000109' 0020      .ADDRESS $$$
      00000241      .PSECT ACF$DRVNAME
52 45 56 49 52 44 43 4C 00' 0241      .ASCIC \LCDRIVER\
      08 0241
      0000086A      .PSECT ACF$DEVDESC
      0002 086A      .WORD      2
00000105' 086C      .LONG   $LCS
00000241' 0870      .LONG   $LCDRIVERS
      01 0874      .BYTE      1
      0018 0875      .WORD      24
00000014 0877      .LONG      20
      01 087B      .BYTE      $$$
      00000200      .PSECT NONPAGED DATA   rd,wrt,noexe,quad
0000086A' 0200      .LONG   $DEVDESC$
      0204      811
00000000 0204      812      .LONG      0      ; END OF LIST
      0208      813
      0208      814 ACF$AB_DMZ32_TABLE:
      0208      815
      0208      816 :      COMBODEV DEVNAME,DRVNAME,NUMVECT,VEC_OFFSET,CSR_OFFSET,
      0208      817 :      MASK,SUPPORT=SUPPORTED
      0208      818
      0208      819      COMBODEV TX, YCDRIVER,      2, 0, 4, 1      ; Async
      0000087C      .PSECT ACF$DEVDESC
      0001 087C      .WORD      1
000000F1' 087E      .LONG   $TX$
0000022F' 0882      .LONG   $YCDRIVERS
      02 0886      .BYTE      2
      0000 0887      .WORD      0
00000004 0889      .LONG      4
      01 088D      .BYTE      $$$
      00000208      .PSECT NONPAGED DATA   rd,wrt,noexe,quad
0000087C' 0208      .LONG   $DEVDESC$
      020C      820      COMBODEV TX, YCDRIVER,      2, 8, 12, 2      ; Async
      0000088E      .PSECT ACF$DEVDESC
      0002 088E      .WORD      2
000000F1' 0890      .LONG   $TX$
0000022F' 0894      .LONG   $YCDRIVERS
      02 0898      .BYTE      2
      0008 0899      .WORD      8
0000000C 089B      .LONG      12
      01 089F      .BYTE      $$$
      0000020C      .PSECT NONPAGED DATA   rd,wrt,noexe,quad
0000088E' 020C      .LONG   $DEVDESC$
      0210      821      COMBODEV TX, YCDRIVER,      2, 16, 20, 4      ; Async
      000008A0      .PSECT ACF$DEVDESC
      0004 08A0      .WORD      4
000000F1' 08A2      .LONG   $TX$
0000022F' 08A6      .LONG   $YCDRIVERS
      02 08AA      .BYTE      2
```

```
0010 08AB .WORD 16
00000014 08AD .LONG 20
01 08B1 .BYTE $$$
00000210 .PSECT NONPAGED DATA rd,wrt,noexe,quad
000008A0' 0210 .LONG $DEVDESC$
00000000 0214 822
0214 823 .LONG 0 ; END OF LIST
0218 824
0218 825
0218 826 ACF$AB_CPI32_TABLE:
0218 827 :
0218 828 : COMBODEV DEVNAME,DRVNAME,NUMVECT,VEC_OFFSET,CSR_OFFSET,
0218 829 : MASK,SUPPORT=SUPPORTED
0218 830
0218 831 COMBODEV XG, XGDRIVER, 2, 0, 4, 1 ; Sync
000008B2 .PSECT ACF$DEVDESC
0001 08B2 .WORD 1
000000FD' 08B4 .LONG $XG$
00000226' 08B8 .LONG $XGDRIVERS$
02 08BC .BYTE 2
0000 08BD .WORD 0
00000004 08BF .LONG 4
01 08C3 .BYTE $$$
00000218 .PSECT NONPAGED DATA rd,wrt,noexe,quad
000008B2' 0218 832 .LONG $DEVDESC$
021C COMBODEV XG, XGDRIVER, 2, 8, 12, 2 ; Sync
000008C4 .PSECT ACF$DEVDESC
0002 08C4 .WORD 2
000000FD' 08C6 .LONG $XG$
00000226' 08CA .LONG $XGDRIVERS$
02 08CE .BYTE 2
0008 08CF .WORD 8
0000000C 08D1 .LONG 12
01 08D5 .BYTE $$$
0000021C .PSECT NONPAGED DATA rd,wrt,noexe,quad
000008C4' 021C 833 .LONG $DEVDESC$
0220 COMBODEV XG, XGDRIVER, 2, 16, 20, 4 ; Sync
000008D6 .PSECT ACF$DEVDESC
0004 08D6 .WORD 4
000000FD' 08D8 .LONG $XG$
00000226' 08DC .LONG $XGDRIVERS$
02 08E0 .BYTE 2
0010 08E1 .WORD 16
00000014 08E3 .LONG 20
01 08E7 .BYTE $$$
00000220 .PSECT NONPAGED DATA rd,wrt,noexe,quad
000008D6' 0220 834 .LONG $DEVDESC$
0224 835 .LONG 0 ; END OF LIST
0228 836
0228 837
0228 838 ACF$AB_CLASS_TABLE:
0228 839 :
0228 840 : CLASSDEV DEVNAME,DRVNAME,PORTDEV
0228 841 :
0228 842 CLASSDEV DU,DUDRIVER,PU
00000109 .PSECT ACF$DEVNAME
```



```

41 55 44 00' 0109      .ASCIC  \DUA\
      03 0109
      00000024      .PSECT  ACF$RESET
0000010D' 0024      .ADDRESS $$$
      0000024A      .PSECT  ACF$DRVNAME
52 45 56 49 52 44 55 44 00' 024A      .ASCIC  \DUDRIVER\
      08 024A
      000008E8      .PSECT  ACF$DEVDESC
00000109' 08E8      .LONG   $DEVNAME$
0000024A' 08EC      .LONG   $DRVNAME$
55 50 00' 08F0      .ASCIC  \PU\
      02 08F0
      00000228      .PSECT  NONPAGED DATA   rd,wrt,noexe,quad
000008E8' 0228      .LONG   $DEVDESC$
      022C      843 CLASSDEV MU,TUDRIVER,PT
      0000010D      .PSECT  ACF$DEVNAME
41 55 4D 00' 010D      .ASCIC  \MUA\
      03 010D
      00000028      .PSECT  ACF$RESET
00000111' 0028      .ADDRESS $$$
      00000253      .PSECT  ACF$DRVNAME
52 45 56 49 52 44 55 54 00' 0253      .ASCIC  \TUDRIVER\
      08 0253
      000008F3      .PSECT  ACF$DEVDESC
0000010D' 08F3      .LONG   $DEVNAME$
00000253' 08F7      .LONG   $DRVNAME$
54 50 00' 08FB      .ASCIC  \PT\
      02 08FB
      0000022C      .PSECT  NONPAGED DATA   rd,wrt,noexe,quad
000008F3' 022C      .LONG   $DEVDESC$
      0230      844
00000000 0230      845 .LONG   0
      0234      846
      0234      847 :
      0234      848 : CONTROLLER DESIGNATOR RESET POINTER TABLE END
      0234      849 :
      0234      850
      0000002C      851 .PSECT  ACF$RESET
      002C      852
00000000 002C      853 .LONG   0 ;
      0030      854
00000044 0030      855 .BLKB  20 ; PATCH SPACE
      0044      856
      0044      857 :
      0044      858 : UBA VECTOR AND CSR BASE OFFSETS
      0044      859 :
      0044      860
      00000234      861 .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
      0234      862
00000238 0234      863 ACF$AL_DEVNAME: .BLKL  1
0000023C 0238      864 ACF$AL_CSR:   .BLKL  1
0000023E 023C      865 ACF$AW_VECTOR: .BLKW  1
00000240 023E      866 ACF$AW_SAVEVEC: .BLKW  1
00000241 0240      867 ACF$AB_NUMVEC: .BLKB  1
      0241      868
      0241      869 ACF$W_CSRBASE:
00000243 0241      870 .BLKW  1
      870

```

```
00000245 0243 871 ACF$W_VECBASE:
00000249 0243 872 .BLKW 1
00000249 0245 873 ACF$S_SYS_CSR:
0000024A 0245 874 .BLKL 1
0000024A 0249 875 ACF$B_BOOT_TR:
0000024A 0249 876 .B[KB 1
0000025A 024A 877 ACF$T_SYS_DEVNAME:
0000025A 024A 878 .BLKB 16
0000025A 025A 879
0000025A 025A 880 .NLIST MEB
```

:  
:  
: SYSTEM DEVICE CSR ADDRESS  
:  
: SYSTEM DEVICE ON THIS ADAPTER  
:  
: SYSTEM DEVICE NAME

```

025A 882 .SBTTL AUTO CONFIGURATION OF DEVICE DATA BASE
025A 883 :+
025A 884 : IOC$AUTOCONFIG - AUTO CONFIGURATION OF DEVICE DATA BASE
025A 885 :
025A 886 : THIS ROUTINE IS CALLED TO AUTO CONFIGURE THE DEVICE DATA BASE FOR A SINGLE ADAPTER
025A 887 :
025A 888 : INPUTS:
025A 889 :
025A 890 : R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
025A 891 : R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
025A 892 : R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
025A 893 :
025A 894 : OUTPUTS:
025A 895 :
025A 896 :-
025A 897
00000000 898 .PSECT NONPAGED_CODE rd,nowrt,exe,long
0000 899
0000 900 IOC$AUTOCONFIG:: ;AUTO CONFIGURE DEVICE DATA BASE
00000249'EF 94 0000 901 CLR B ACF$B_BOOT TR ;INITIALIZE LOCAL FLAG BYTE
50 0000'CF 9E 0006 902 MOV AB W*ACF$AB_ADPTYPE,R0 ;GET ADDRESS OF ADAPTER TABLE
0E A8 80 B1 000B 903 10$: CMP W (R0)+,ADP$W_ADPTYPE(R8) ;ADAPTER TYPE MATCH?
50 0A 13 000F 904 BEQ 20$ ;IF EQL YES
04 C0 0011 905 ADDL #ADAPTERLEN-2,R0 ;ADVANCE TO NEXT ENTRY
60 B5 0014 906 TST W (R0) ;ANY MORE ENTRIES IN TABLE?
F3 12 0016 907 BNEQ 10$ ;IF NEQ YES
50 D4 0018 908 CLRL R0 ;INDICATE NO ADAPTER
05 001A 909 RSB ;
001B 910
001B 911 :
001B 912 : ADAPTER TYPE MATCH - INITIALIZE CONFIGURATION BLOCK AND DISPATCH TO ADAPTER CODE
001B 913 :
001B 914
001B 915 20$: MOVL R8,ACF$A_ADAPTER(R7) ;SET ADDRESS OF ADAPTER CONTROL BLOCK
04 A7 56 D0 C01B 916 MOVL R6,ACF$A_CONFIGREG(R7) ;SET ADDRESS OF CONFIGURATION STATUS REGISTE
51 1C A8 00000000'GF C3 0022 917 SUBL3 G^EXE$GL_SCB, - ;CALCULATE OFFSET INTO SCB OF
08 A7 51 B0 002B 918 ADP$A_AVECTOR(R8),R1 ;ADAPTER'S INTERRUPT VECTORS.
90 17 002F 919 MOVW R1,ACF$W_AVECTOR(R7) ;STORE IN ACF.
920 JMP @ (R0)+ ;DISPATCH TO PROPER ROUTINE

```

```

0031 922      .SBTTL  ACF$MBA - MASSBUS ADAPTER AUTO CONFIGURATION
0031 923      :
0031 924      : ACF$MBA - MASSBUS ADAPTER AUTO CONFIGURATION
0031 925      :
0031 926      : THIS ROUTINE IS CALLED TO AUTO CONFIGURE A MASSBUS ADAPTER.
0031 927      :
0031 928      : INPUTS:
0031 929      :
0031 930      : R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
0031 931      : R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
0031 932      : R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
0031 933      :
0031 934      : OUTPUTS:
0031 935      :
0031 936      : A SCAN IS MADE FOR EACH OF THE POSSIBLE DRIVE TYPES THAT CAN BE ATTACHED
0031 937      : TO THE ADAPTER. AS EACH DRIVE TYPE IS ENCOUNTERED, A CO-ROUTINE CALL IS
0031 938      : MADE TO THE CALLER DELIVERING THE DEVICE UNIT DESCRIPTOR.
0031 939      :
0031 940      :
0031 941      : .ENABL  LSB
0031 942      ACF$MBA: ;AUTO CONFIGURE MASSBUS ADAPTER
0031 943      MOVB  #1,ACF$B CNUMVEC(R7) ;SET NUMBER OF CONTROLLER VECTORS
10  A7 01 90 0035 944      MOVW  ACF$W AVECTOR(R7),ACF$W_CVECTOR(R7) ;SET CONTROLLER VECTOR OFFSET
55  001A'CF 9E 003A 945      MOVAB W^ACF$AB_MBATABLE,R5 ;GET ADDRESS OF MBA DRIVER DESCRIPTOR TABLE
54  85  D0 003F 946 10$:  MOVL  (R5)+,R4 ;GET ADDRESS OF NEXT DRIVER DESCRIPTOR
03  12  D0 0042 947      BNEQ  20$ ;IF NEQ ANOTHER ENTRY TO PROCESS
50  03  D4 0044 948      CLRL  R0 ;SET END OF SCAN INDICATOR
05  05  D0 0046 949      RSB   ;
0047 950      :
0047 951      :
0047 952      : SCAN MBA UNITS FOR DEVICE TYPE MATCH
0047 953      :
0047 954      :
14  A7 0745 30 0047 955 20$:  BSBW  ACF$CLR_ACF ;CLEAR DEVICE BLOCKS
18  A7 84  D0 004A 956      MOVL  (R4)+,ACF$S_DEVNAME(R7) ;SET ADDRESS OF DEVICE NAME STRING
52  53  D0 004E 957      MOVL  (R4)+,ACF$S_DRVNAME(R7) ;SET ADDRESS OF DRIVER NAME STRING
51  0418 C643 D4 0052 958      CLRL  R3 ;CLEAR DRIVE OFFSET CONSTANT
51  3E00 8F AA 0054 959 30$:  MOVAB (R4),R2 ;GET STARTING ADDRESS OF DRIVE TYPE TABLE
50  08 A6 50 D0 0057 960      MOVL  MBA$S_ERB+MBA_DT(R6)[R3],R1 ;READ DRIVE TYPE REGISTER
08  A6 50 D0 005D 961      BICW  #^C<^XC1FF>,R1 ;CLEAR EXTRANEIOUS BITS
15  50 12 E0 0062 962      MOVL  MBA$S_SR(R6),R0 ;READ MBA STATUS REGISTER
05  51 0F E0 0066 963      MOVL  R0,MBA$S_SR(R6) ;CLEAR MBA STATUS REGISTER
51  3E07 8F AA 006A 964      BBS  #MBA$V_SR_NED,R0,50$ ;IF SET, NONEXISTENT DRIVE
82  51 B1 006E 965      BBC  #15,R1,40$ ;IF CLR, NOT TAPE DEVICE
53  13 007A 966      BICW  #^C<^XC1F8>,R1 ; FOR TAPE, RETAIN CONTROLLER
62  B5 007C 967      : SPECIFIC INFORMATION ONLY.
F7  12 0077 968 40$:  CMPW  R1,(R2)+ ;DRIVE TYPE MATCH?
0041 31 007A 969      BEQL  60$ ;IF EQL YES
50  00000000'GF 000186A0 8F C5 007C 970      TSTW  (R2) ;ANY MORE ENTRIES IN TABLE?
0404 C643 D4 007E 971      BNEQ  40$ ;IF NEQ YES
51  0418 C643 D0 0080 972      BRW  56$ ;GO LOOK FOR MORE UNITS
0400 C643 0B 9A 0083 973      :
50  00000000'GF 000186A0 8F C5 0083 974 50$:  MULL3 #10000,G^EXE$GL_TENUSEC,R0 ;INIT LOOP COUNT;
0404 C643 D4 008F 975      CLRL  MBA$S_ERB+MBA_DS(R6)[R3]; ATTEM'T TO SEIZE THE PORT
51  0418 C643 D0 0094 976 52$:  MOVL  MBA$S_ERB+MBA_DT(R6)[R3],R1 ; READ DRIVE TYPE REGISTER
0400 C643 0B 9A 009A 977      BEQL  54$ ; IF EQL, THEN NO PORT
0400 C643 0B 9A 009C 978      MOVZBL #^X0B,MBA$S_ERB(R6)[R3] ; RELEASE THE PORT

```

```

50 08 A6 D0 00A2 979      MOVL  MBASL_SR(R6),R0      ;READ MBA STATUS REGISTER
08 A6 50 D0 00A6 980      MOVL  R0,MBASL_SR(R6)     ;CLEAR MBA STATUS REGISTER
51 3E00 8F AA 00AA 981      BICW  #^C<^XC1FF>,R1      ; CLEAR EXTRANEIOUS BITS
      C6 11 00AF 982      BRB   40$                ; GO LOOK FOR DRIVER TYPE MATCH
      00B1 983 54$:      FREEIB
      51 08 A6 D0 00B9 984      MOVL  MBASL_SR(R6),R1      ;READ MBA STATUS REGISTER
      08 A6 51 D0 00BD 985      MOVL  R1,MBASL_SR(R6)     ;CLEAR MBA STATUS REGISTER
      DO 50 F5 00C1 986      SOBGTR R0,52$                ;IF GTR, TRY TO READ DT REGISTER AGAIN
      00C4 987
FF88 53 20 00E0 8F 3D 00C4 988 56$:  ACBW  #^X80/4*7,#^X80/4,R3,30$ ;ANY MORE UNITS TO EXAMINE?
      FF70 31 00CC 989      BRW   10$                ;YES
      00CF 990
      00CF 991      ;
      00CF 992      ; MBA DRIVE TYPE/DEVICE TYPE MATCH - DELIVER ALL UNITS TO CALLER
      00CF 993      ;
      00CF 994
      39 51 0F E0 00CF 995 60$:  BBS   #15,R1,ACF$MBATAPE ;IF SET, TAPE DEVICE
      00D3 996
      00D3 997      ;
      00D3 998      ; DEVICE IS A DISK
      00D3 999      ;
      00D3 1000
      00D3 1001 ACF$MBADISK:
      0C A7 56 D0 00D3 1002      MOVL  R6,ACF$L_CONTRLREG(R7) ;SET ADDRESS OF CONTROL STATUS REGISTER
      51 14 A7 D0 00D7 1003      MOVL  ACF$L_DEVNAME(R7),R1 ;GET ADDRESS OF DEVICE NAME STRING
      52 61 9A 00DB 1004      MOVZBL (R1),R2 ;GET LENGTH OF STRING IN BYTES
6142 41 8F 0B AB 81 00DE 1005      ADDB3  ADP$B_NUMBER(R8),#^A/A/, (R1)[R2] ;CALCULATE LAST BYTE OF DEVICE NAME
      00E5 1006
      00E5 1007      ;
      00E5 1008      ; CONSTRUCT UNIT DATA BASE AND DELIVER TO CALLER
      00E5 1009      ;
      00E5 1010
      50 53 20 C7 00E5 1011 70$:  DIVL3 #^X80/4,R3,R0 ;CALCULATE ADAPTER DRIVE NUMBER
      0A A7 50 90 00E9 1012      MOVB  R0,ACF$B_AUNIT(R7) ;SET ADAPTER UNIT NUMBER
      12 A7 50 B0 00ED 1013      MOVW  R0,ACF$W_CUNIT(R7) ;SET CONTROLLER UNIT NUMBER
      007C'CF 53 7D 00F1 1014      MOVQ  R3,W^ACF$R3SAVE ;SAVE REGISTERS
      0084'CF 55 D0 00F6 1015      MOVL  R5,W^ACF$R5SAVE ;
      50 01 D0 00FB 1016      MOVL  #1,R0 ;SET SUCCESS INDICATOR
      53 007C'CF 9E 16 00FE 1017      JSB  @ (SP)+ ;DELIVER UNIT TO CALLER
      55 0084'CF 7D 0100 1018      MOVQ  W^ACF$R3SAVE,R3 ;RESTORE REGISTERS
      0084'CF D0 0105 1019      MOVL  W^ACF$R5SAVE,R5 ;
      B8 11 010A 1020      BRB   56$                ;BRANCH TO TEST FOR MORE DRIVES
      010C 1021
      010C 1022      ;
      010C 1023      ; DEVICE IS A TAPE
      010C 1024      ;
      010C 1025
      51 C040 8F B1 010C 1026 ACF$MBATAPE:
      50 53 20 C7 0113 1029      DIVL3 #^X80/4,R3,R0 ;CALCULATE ADAPTER UNIT NUMBER
      0A A7 50 90 0117 1030      MOVB  R0,ACF$B_AUNIT(R7) ;SET ADAPTER UNIT NUMBER
      007C'CF 53 7D 011B 1031      MOVQ  R3,W^ACF$R3SAVE ;SAVE REGISTERS
      0084'CF 55 D0 0120 1032      MOVL  R5,W^ACF$R5SAVE ;
      52 0400 C643 DE 0125 1033      MOVAL MBASL_ERB(R6)[R3],R2 ;GET ADDRESS OF DRIVE CONTROL REGISTER
      0C A7 52 D0 012B 1034      MOVL  R2,ACF$L_CONTRLREG(R7) ;SET ADDRESS OF DRIVE CONTROL REGISTER
      51 D4 012F 1035      CLRL  R1 ;CLEAR STARTING SLAVE DRIVE NUMBER

```

```

0131 1036
0131 1037 :
0131 1038 : CONSTRUCT UNIT DATA BASE AND DELIVER TO CALLER
0131 1039 :
0131 1040 :
50 18 A2 24 A2 51 D0 0131 1041 110$: MOVL R1, TM03 TC(R2) ; SELECT SLAVE DRIVE
      FFFF01FF 8F CB 0135 1042 BICL3 #^C<^XFE00>, MBA_DT(R2), R0 ; CLEAR EXTRANEIOUS BITS
50 C400 8F B1 013E 1043 CMPW #^XC400, R0 ; SLAVE TAPE DRIVE AND PRESENT?
      13 12 0143 1044 BNEQ 120$ ; IF NEQ NO
0074'CF 51 7D 0145 1045 MOVQ P1, W^ACF$R1SAVE ; SAVE REGISTERS
      12 A7 51 B0 014A 1046 MOVW R1, ACF$W_CONIT(R7) ; SET CONTROLLER UNIT NUMBER
      50 01 D0 014E 1047 MOVL #1, R0 ; SET SUCCESS INDICATOR
      9E 16 0151 1048 JSB @($P)+ ; DELIVER UNIT TO CALLER
51 0074'CF 7D 0153 1049 MOVQ W^ACF$R1SAVE, R1 ; RESTORE REGISTERS
      D5 51 07 F3 0158 1050 120$: AOBLEQ #7, R1, 1T0$ ; ANY MORE SLAVES TO PROCESS?
      52 14 A7 D0 015C 1051 MOVL ACF$R1_DEVNAME(R7), R2 ; GET ADDRESS OF DEVICE NAME STRING
      51 62 9A 0160 1052 MOVZBL (R2), R1 ; GET LENGHT OF STRING IN BYTES
      6241 96 0163 1053 INCB (R2)[R1] ; INCREMENT CONTROLLER DESIGNATION
53 007C'CF 7D 0166 1054 MOVQ W^ACF$R1_F3SAVE, R3 ; RESTORE REGISTERS
55 0084'CF D0 016B 1055 MOVL W^ACF$R1_R5SAVE, R5 ;
      FF51 31 0170 1056 BRW 56$ ;
0173 1057 :
0173 1058 :
0173 1059 : TM78 SPECIFIC CODE
0173 1060 :
0173 1061 :
50 53 20 C7 0173 1062 130$: DIVL3 #^X80/4, R3, R0 ; CALCULATE ADAPTER UNIT NUMBER
      0A A7 50 90 0177 1063 MOVW R0, ACF$B_AUNIT(R7) ; SET ADAPTER UNIT NUMBER
007C'CF 53 7D 017B 1064 MOVQ R3, W^ACF$R1_R3SAVE ; SAVE REGISTERS
0084'CF 55 D0 0180 1065 MOVL R5, W^ACF$R1_R5SAVE ;
52 0400 C643 DE 0185 1066 MOVAL MBA$R1_ERB(R6)[R3], R2 ; GET ADDRESS OF DRIVE CONTROL REGISTER
      0C A7 52 D0 018B 1067 MOVL R2, ACF$R1_CONTRLREG(R7) ; SET ADDRESS OF DRIVE CONTROL REGISTER
018F 1068 :
018F 1069 :
018F 1070 : Initialize TM78 controller.
018F 1071 :
018F 1072 :
44 A2 00004000 8F C8 018F 1073 BISL #TM78_M_TMCLR, TM78_ID(R2) ; Reset TM78 controller to initial state.
      51 FA 8F 9A 0197 1074 MOVZBL #250, R1 ; Set a counter.
51 001E8480 8F D0 019B 1075 140$: SOBGTR R1, 140$ ; Waste a little time while TMCLR starts.
      50 44 A2 D0 019E 1076 MOVL #2000000, R1 ; Set counter to wait for TMCLR to end.
50 00008000 8F D3 01A5 1077 150$: MOVL TM78_ID(R2), R0 ; Read TM78 register.
      06 12 01A9 1078 BITL #TM78_M_TMRDY, R0 ; See if TMCLR done yet.
      FO 51 F5 01B0 1079 BNEQ 160$ ; NEQ implies that TMCLR is done.
      0077 31 01B2 1080 SOBGTR R1, 150$ ; Else loop back and test again.
      51 D4 01B5 1081 BRW 220$ ; If we fall thru, TM78 did not come READY.
      51 D4 01B8 1082 160$: CLRL R1 ; CLEAR STARTING SLAVE DRIVE NUMBER
      01BA 1083 :
      01BA 1084 :
      01BA 1085 :
      01BA 1086 : CONSTRUCT UNIT DATA BASE AND DELIVER TO CALLER
      01BA 1087 :
      01BA 1088 :
50 0A A7 9A 01BA 1089 170$: MOVZBL ACF$B_AUNIT(R7), R0 ; Pickup which MASSBUS device we are.
50 01 50 78 01BE 1090 ASHL R0, #1, R0 ; Shift to appropriate attention bit.
      1C A2 D4 01C2 1091 CLRL TM78_DS(R2) ; Clear any old device info.
      10 A2 50 D0 01C5 1092 MOVL R0, TM78_AB(R2) ; Clear attention bit.

```

51	FA	51	DD	01C9	1093	PUSHL	R1	:	Save drive number.
	FD	51	F5	01CB	1094	MOVZBL	#250,R1	:	Set a counter.
		51	D0	01CF	1095	SOBGTR	R1,180\$	:	Waste time while ATTN bit clears
		30	D0	01D2	1096	MOVL	(SP),R1	:	Restore drive number.
51	A241	09	D0	01D5	1097	MOVL	#TM78_SENSE_GO,TM78_NDT0(R2)[R1]	:	Set sense command.
		001E8480	D0	01DA	1098	MOVL	#2000000,R1	:	Set count to prevent hang.
				01E1	1099				
		08	DD	01E1	1100	PUSHL	MBASL_SR(R6)	:	Copy MBA status register to stack.
8E	G0010000	8F	D3	01E4	1101	BITL	#MBASL_SR_ATTEN,(SP)+	:	Test for MBA attention bit on
				01EB	1102			:	in MBA status register.
		03	12	01EB	1103	BNEQ	200\$	:	NEQ implies MBA attention came on.
		F1	F5	01ED	1104	SOBGTR	R1,190\$	:	Loop waiting for MBA attention bit.
				01F0	1105			:	If here, either MBA attention bit came
				01F0	1106			:	on or the loop gave out.
		51	8ED0	01F0	1107	POPL	R1	:	Restore drive number.
10	A2	50	D3	01F3	1108	BITL	R0,TM78_AB(R2)	:	Assure TM78 attention bit set.
		32	13	01F7	1109	BEQL	210\$	:	EQL implies NO attention which
				01F9	1110			:	means the device doesn't respond so
				01F9	1111			:	let's ignore it.
		1C	A2	DD	01F9	PUSHL	TM78_DS(R2)	:	Copy device status register to stack.
		2C	A2	DD	01FC	PUSHL	TM78_NDTA(R2)	:	Copy interrupt status data to stack.
10	A2	50	D0	01FF	1114	MOVL	R0,TM78_AB(R2)	:	Clear attention bit.
		50	8ED0	0203	1115	POPL	R0	:	R0 = interrupt status.
02	AE	50	B0	0206	1116	MOVW	R0,2(SP)	:	Copy to high word of TOP OF STACK.
		50	8ED0	020A	1117	POPL	R0	:	R0 = drive status in low word,
				020D	1118			:	interrupt status in high word.
				020D	1119				
01	50	06	10	ED	020D	CMPZV	#16,#6,R0,#1	:	See if we got back DONE status.
			17	12	0212	BNEQ	210\$	:	If not, then device is NO GOOD.
					0214				
		13	50	OE	E1	BBC	#14,R0,210\$	:	Branch if drive not present.
		0074	'CF	51	7D	MOVQ	R1,W^ACFSL_R1SAVE	:	SAVE REGISTERS
		12	A7	51	B0	MOVW	R1,ACFSL_CONIT(R7)	:	SET CONTROLLER UNIT NUMBER
				50	D0	MOVL	#1,R0	:	SET SUCCESS INDICATOR
					9E	JSB	@(SP)+	:	DELIVER UNIT TO CALLER
51	0074	'CF	7D	0226	1128	MOVQ	W^ACFSL_R1SAVE,R1	:	RESTORE REGISTERS
	8B	51	03	F3	022B	AOBLEQ	#3,R1,170\$	:	ANY MORE SLAVES TO PROCESS?
					022F				
		52	14	A7	D0	MOVL	ACFSL_DEVNAME(R7),R2	:	GET ADDRESS OF DEVICE NAME STRING
			51	62	9A	MOVZBL	(R2),R1	:	GET LENGTH OF STRING IN BYTES
				6241	96	INCB	(R2)[R1]	:	INCREMENT CONTROLLER DESIGNATION
53	007C	'CF	7D	0239	1134	MOVQ	W^ACFSL_R3SAVE,R3	:	RESTORE REGISTERS
55	0084	'CF	D0	023E	1135	MOVL	W^ACFSL_R5SAVE,R5	:	
				FE7E	31	BRW	56\$	:	
					0243			:	
					0246	.DSABL	LSB	:	





```

024D 1161      .SBTTL  ACFSCI - CI ADAPTER AUTO CONFIGURATION
024D 1162      :
024D 1163      : ACFSCI - CI ADAPTER AUTO CONFIGURATION
024D 1164      :
024D 1165      : THIS ROUTINE IS CALLED TO AUTO CONFIGURE A CI ADAPTER.
024D 1166      :
024D 1167      : INPUTS:
024D 1168      :
024D 1169      :     R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
024D 1170      :     R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
024D 1171      :     R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
024D 1172      :
024D 1173      : OUTPUTS:
024D 1174      :
024D 1175      :     A CO-ROUTINE CALL IS MADE TO THE CALLER DELIVERING THE DEVICE
024D 1176      :     UNIT DESCRIPTOR.
024D 1177      :
024D 1178      :
024D 1179      : ACFSCI:
54  0032'CF  D0 024D 1180      MOVL    W*ACFSAB_CITABLE,R4      ;AUTO CONFIGURE CI ADAPTER
                                ;GET ADDRESS OF DRIVER DESCRIPTOR
0252 1181
0252 1182      ACFSSINGLE DEVICES:
                                ;COMMON ENTRY POINT FOR DR AND CI
1E  A7  01  90 0252 1183      MOVB    #1,ACFSB_CNUMVEC(R7)      ;SET NUMBER OF CONTROLLER VECTORS
10 A7  08  A7  B0 0256 1184      MOVW   ACFSW_AVECTOR(R7),ACFSW_CVECTOR(R7) ;SET CNTRLR VECTOR OFFSET
0C  A7  56  D0 025B 1185      MOVL   R6,ACFSL_CONTRLREG(R7) ;SET ADDRESS OF CONTROL STATUS REGISTER
                                ;(R4)+,R1
                                ;GET ADDRESS OF DEVICE NAME STRING
14  A7  51  D0 025F 1186      MOVL   (R4)+,R1
                                ;SET ADDRESS OF DEVICE NAME STRING
18  A7  64  D0 0262 1187      MOVL   R1,ACFSL_DEVNAME(R7)
                                ;SET ADDRESS OF DRIVER NAME STRING
                                ;(R4),ACFSL_DRVNAME(R7)
                                ;GET LENGTH OF DEV. NAME STRING IN BYTES
6142 41 8F  0B  A8 81 026A 1189      MOVZBL (R1),R2
                                ;CALCULATE AND STORE LAST
                                ;BYTE OF DEVICE NAME
                                ;SET ADAPTER UNIT NUMBER = 0
                                ;SET CONTROLLER UNIT NUMBER = 0
                                ;CLEAR DEVICE BLOCKS
                                ;SET SUCCESS INDICATOR
                                ;DELIVER UNIT TO CALLER
                                ;SET END OF SCAN INDICATOR
                                ;
0274 1191      CLR    ACFSB_AUNIT(R7)
0274 1192      CLR    ACFSW_CUNIT(R7)
0277 1193      BSBW   ACFSCR_ACF
027A 1194      MOVL   #1,R0
027D 1195      JSB    @($P)+
0280 1196      CLRL   R0
0282 1197      RSB
0284 1198
0285 1199

```

```

0285 1201      .SBTTL  ACF$UBA - UNIBUS ADAPTER AUTO CONFIGURATION
0285 1202      :
0285 1203      : ACF$UBA - UNIBUS ADAPTER AUTO CONFIGURATION
0285 1204      :
0285 1205      : THIS ROUTINE IS CALLED TO AUTO CONFIGURE A UNIBUS ADAPTER.
0285 1206      :
0285 1207      : INPUTS:
0285 1208      :
0285 1209      :     R6 = ADDRESS OF CONFIGURATION CONTROL REGISTER.
0285 1210      :     R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
0285 1211      :     R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
0285 1212      :
0285 1213      : OUTPUTS:
0285 1214      :
0285 1215      :     A SCAN IS MADE FOR EACH OF THE POSSIBLE DEVICES THAT CAN BE ATTACHED TO
0285 1216      :     THE ADAPTER. AS EACH DEVICE IS ENCOUNTERED, A CO-ROUTINE CALL IS MADE TO
0285 1217      :     THE CALLER DELIVERING THE DEVICE UNIT DESCRIPTOR.
0285 1218      :
0285 1219      :
0285 1220      : .ENABL  LSB
0285 1221      ACF$UBA:
0000024A'EF  94 0285 1222      CLRFB  ACF$T_SYS_DEVNAME      :AUTO CONFIGURE UNIBUS ADAPTER
00000245'EF  D4 028B 1223      CLRL  ACF$L_SYS_CSR        :SET NO SYSTEM DISK ON THIS ADAPTER
51 00000000'GF  D0 0291 1224      MOVL  G^EX$GL_RPB,R1      :GET ADDRESS OF THE RPB
20 A1 0C A8 B1 0298 1225      CMPW  ADP$W_TR(R8),RPB$L_BOOTR1(R1);ADAPTER CONTAIN SYSTEM DISK?
      4A 12 029D 1226      BNEQ  9$
00000249'EF  96 029F 1227      INCB  ACF$B_BOOT_TR      :SET FLAG FOR SYSTEM DISK ON THIS ADAPTER
50 00000000'GF  9E 02A5 1228      MOVAB G^SYS$GL_BOOTUCB,R0 :GET SYSTEM DEVICE UCB ADDRESS
51 0084 C0 D0 02AC 1229      MOVL  UCB$L_PDT(R0),R1   :GET PORT DRIVER DESCRIPTOR TABLE ADDRESS
      05 13 02B1 1230      BEQL  5$
50 00DC C1 D0 02B3 1231      MOVL  PDT$L_UCB0(R1),R0 :USE PORT DRIVER UCB ADDRESS
51 24 A0 D0 02B8 1232 5$:      MOVL  UCB$L_CRB(R0),R1   :GET SYSTEM DEVICE CRB ADDRESS
51 2C A1 D0 02BC 1233      MOVL  CRB$L_INTD+VEC$L_IDB(R1),R1;GET SYSTEM DEVICE IDB ADDRESS
52 1000 C6 9E 02C0 1234      MOVAB UBA_IDBASE(R6),R2  :GET UNIBUS ADDRESS SPACE VIRTUAL ADDRESS
      52 61 D1 02C5 1235      CML  IDB$L_CSR(R1),R2   :CSR IN UNIBUS ADDRESS SPACE?
      1F 1B 02C8 1236      BLEQU 9$
00000245'EF  61 D0 02CA 1237      MOVL  IDB$L_CSR(R1),ACF$L_SYS_CSR;SAVE SYSTEM DEVICE CSR ADDRESS
51 28 A0 D0 02D1 1238      MOVL  UCB$L_DDB(R0),R1   :GET DDB ADDRESS OF SYSTEM DEVICE
50 14 A1 9A 02D5 1239      MOVZBL DDB$T_NAME(R1),R0 :GET SYSTEM DEVICE NAME SIZE
0000024A'EF  50 90 02D9 1240      MOVB  R0,ACF$T_SYS_DEVNAME :SAVE SYSTEM DEVICE NAME SIZE
      15 A1 50 28 02E0 1241      MOVCB R0,DDB$T_NAME+1(R1),- :SAVE SYSTEM DEVICE NAME
00000248'EF  02E4 1242
      0A A7 94 02E9 1243 9$:      CLRFB ACF$B_AUN (R7)      :CLEAR ADAPTER UNIT NUMBER
0243'CF 00C0 8F B0 02EC 1244      MOVW  #^010,W^AL,W_CSRBASE :SET INITIAL CSR BASE OFFSET
55 0090'CF 9E 02F1 1245      MOVW  #^0300,W^ACF$Q_VECBASE :SET INITIAL VECTOR BASE OFFSET
      54 85 D0 02F8 1246      MOVAB W^ACF$AB_UBATABLE,R5 :GET ADDRESS OF UBA DRIVER DESCRIPTOR TABLE
      03 12 0300 1247 10$:      MOVL  (R5)+,R4          :GET ADDRESS OF NEXT DRIVER DESCRIPTOR
      50 D4 0302 1248      BNEQ  20$              :IF NEQ ANOTHER ENTRY TO PROCESS
      05 0304 1249      CLRL  R0                :SET END OF SCAN INDICATOR
      0305 1250      RSB
      0305 1251
      0305 1252      : SCAN UBA DEVICE
      0305 1253
      0305 1254
      0305 1255
00000000'EF  D4 0305 1256 20$:      CLRL  ACF$GL_DPT        : Clear formerly loaded driver
      0481 30 030B 1257      BSBW  ACF$CLR_ACF      : Clear device blocks

```

```

14 A7 64 DO 030E 1258      MOVL  UBT$L_DEVNAME(R4),ACF$L_DEVNAME(R7) ;SET ADDRESS OF DEVICE NAME STRI
18 A7 04 A4 DO 0312 1259      MOVL  UBT$L_DRVNAME(R4),ACF$L_DRVNAME(R7) ;SET ADDRESS OF DRIVER NAME STRI
003E'CF 0C A4 DO 0317 1260      MOVL  UBT$L_ROUTINE(R4),W^ACF$L_ROUTINE ;SET ADDRESS OF DEVICE GENERATION
1E A7 11 A4 90 031D 1261      MOVB  UBT$B_NUMVEC(R4),ACF$B_CNOMVEC(R7) ;SET NUMBER OF CONTROLLER INTERRU
53 13 A4 9E 0322 1262      MOVAB UBT$B_FLAGS(R4),R3 ;COPY ADDRESS OF FLAG BYTE
0326 1263
0326 1264 ; The supported characteristic remains only to suppress the error
0326 1265 ; message in IOGEN$LOADER that driver wasn't found.
0326 1266
08 A7 10 88 0326 1267      BISB2 #ACF$M_SUPPORT,ACF$B_AFLAG(R7) ; ASSUME NOSUPPORT
04 63 00 E1 032A 1268      BBC   #UBA_V_SUPPORT,(R3),23$ ; BRANCH IF NOSUPPORT
08 A7 10 8A 032E 1269      BICB2 #ACF$M_SUPPORT,ACF$B_AFLAG(R7) ; SET SUPPORTED
0332 1270
54 14 A4 9E 0332 1271 23$: MOVAB  UBT$W_REMAINDER(R4),R4 ;ADDRESS OF VARIABLE BLOCK
17 63 02 E1 0336 1272      BBC   #UBA_V_FLOATVEC,(R3),25$ ;IF CLR, NO FLOATING VECTOR ASSIGNMENT
51 64 3C 033A 1273      MOVZWL (R4),RT ;GET VECTOR MODULO MASK
0000006E'EF 51 B0 033D 1274      MOVW  R1,ACF$W_VECMOD ;SAVE FOR LATER USE
51 0243'CF A0 0344 1275      ADDW  W^ACF$W_VECBASE,R1 ;ROUND UP TO NEXT VECTOR
023E'CF 51 84 AA 0349 1276      BICW  (R4)+,RT ;TRUNCATE TO ACTUAL VECTOR OFFSET
03 63 01 B0 034C 1277      MOVW  R1,W^ACF$W_SAVEVEC ;SAVE FOR CHECK LATER
0086 01 E1 0351 1278 25$: BBC   #UBA_V_FLOAT(CSR),(R3),ACF$UBAFIXED ;IF CLR, FIXED CSR
31 0355 1279      BRW   ACF$OB$FLOATING ;
0358 1280
0358 1281 ;
0358 1282 ; FIXED CSR DEVICE
0358 1283 ;
0358 1284
0358 1285 ACF$UBAFIXED:
52 84 3C 0358 1286 30$: MOVZWL (R4)+,R2 ;GET CSR OFFSET
13 6E 13 035B 1287      BEQL  37$ ;IF EQL NO MORE TO PROCESS
52 1000 C642 9E 035D 1288      MOVAB  UBA_IOBASE(R6)[R2],R2 ;GET ADDRESS OF CSR
0C A7 52 DO 0363 1289      MOVL  R2,ACF$L_CONTRLREG(R7) ;SET ADDRESS OF CONTROL REGISTER
03 63 02 E0 0367 1290      BBS   #UBA_V_FLOATVEC,(R3),31$ ;IF SET, FLOATING VECTOR ASSIGNMENT
51 84 3C 036B 1291      MOVZWL (R4)+,R1 ;GET ACTUAL VECTOR ADDRESS
10 A7 51 B0 036E 1292 31$: MOVW  R1,ACF$W_CVECTOR(R7) ;SET ADDRESS OF INTERRUPT VECTOR
50 52 DO 0372 1293      MOVL  R2,R0 ;GET COPY OF CSR
00000000'GF 16 E9 0375 1294      JSB   G^EXESTEST_CSR ;CHECK DEVICE CSR
DA 50 E9 037B 1295      BLBC  R0,30$ ;BRANCH IF CSR NON-EXISTENT
0074'CF 51 7D 037E 1296      MOVQ  R1,W^ACF$R1SAVE ;SAVE REGISTERS
007C'CF 53 7D 0383 1297      MOVQ  R3,W^ACF$R3SAVE
0084'CF 55 DO 0388 1298      MOVL  R5,W^ACF$R5SAVE
00EE 30 038D 1299      BSBW  FIX_DEV_NAME ;FIX UP DEVICE NAME
000004C1'EF 16 0390 1300      JSB   LOAD_DRIVER ;LOAD DRIVER
003E'DF 16 0396 1301      JSB   @W^ACF$L_ROUTINE ;CALL GENERATION ROUTINE
51 14 A7 DO 039A 1302      MOVL  ACF$L_DEVNAME(R7),R1 ;GET ADDRESS OF DEVICE NAME STRING
03AA 30 039E 1303      BSBW  ACF$INC_CHAR ;INCREMENT CONTROLLER
51 0074'CF 7D 03A1 1304      MOVQ  W^ACF$R1SAVE,R1 ;RESTORE REGISTERS
53 007C'CF 7D 03A6 1305      MOVQ  W^ACF$R3SAVE,R3
55 0084'CF DO 03AB 1306      MOVL  W^ACF$R5SAVE,R5
A4 63 02 E1 03B0 1307 35$: BBC   #UBA_V_FLOATVEC,(R3),30$ ;IF CLR, FIXED VECTOR ASSIGNMENT
50 1E A7 9A 03B4 1308      MOVZBL ACF$B_CNOMVEC(R7),R0 ;GET NUMBER OF CONTROLLER INTERRUPT VECTORS
51 6140 DE 03B8 1309      MOVAL (R1)[R0],R1 ;CALCULATE ADDRESS OF NEXT VECTOR
50 0000006E'EF 3C 03BC 1310      MOVZWL ACF$W_VECMOD,R0 ;GET VECTOR MODULUS
51 50 A0 03C3 1311      ADDW  R0,R1 ;ROUND UP TO NEXT VECTOR
51 50 AA 03C6 1312      BICW  R0,R1 ;TRUNCATE TO ACTUAL VECTOR OFFSET
OC 83 02 E1 03C9 1313      BRB   30$
OC 83 02 E1 03CB 1314 37$: BBC   #UBA_V_FLOATVEC,(R3)+,39$ ;IF CLR, NO FLOATING VECTOR ASSIGNMENT

```

```
51 023E'CF B1 03CF 1315 CMPW W^ACFSW_SAVEVEC,R1 ;ANY VECTORS ASSIGNED?
      05 13 03D4 1316 BEQL 39$ ;IF EQL NO
0243'CF 51 B0 03D6 1317 MOVW R1,W^ACFSW_VECBASE ;SET NEW VECTOR BASE ADDRESS
      FF1F 31 03DB 1318 39$: BRW 10$
      03DE 1319
      03DE 1320 ;
      03DE 1321 ; FLOATING VECTOR/CSR DEVICE
      03DE 1322 ;
      03DE 1323
      03DE 1324 ACFSUBAFLOATING:
52 0241'CF 3C 03DE 1325 MOVZWL W^ACFSW_CSRBASE,R2 ;GET BASE CSR OFFSET
      52 64 A0 03E3 1326 ADDW (R4),R2 ;ROUND TO NEXT CSR
      52 64 AA 03E6 1327 BICW (R4),R2 ;TRUNCATE BACK TO CSR OFFSET
50 1000 C642 9E 03E9 1328 MOVAR UBA_IOBASE(R6)[R2],R0 ;GET ACTUAL CSR ADDRESS
      00000000'GF 16 03EF 1329 JSB G^EXESTEST_CSR ;CHECK FOR NON-EXISTENT CSR
      03 50 E8 03F5 1330 BLBS R0,38$ ; branch if csr there
      0071 31 03F8 1331 BRW 60$ ;BRANCH IF CSR NON-EXISTENT
      0241'CF 52 B0 03FB 1332 38$: MOVW R2,W^ACFSW_CSRBASE ;SET NEW BASE CSR OFFSET
50 1000 C642 9E 0400 1333 40$: MOVAB UBA_IOBASE(R6)[R2],R0 ;GET ACTUAL CSR ADDRESS
      00000000'GF 16 0406 1334 JSB G^EXESTEST_CSR ;CHECK FOR NON-EXISTENT CSR
      5D 50 E9 040C 1335 BLBC R0,60$ ;BRANCH IF CSR NON-EXISTENT
OC A7 1000 C642 9E 040F 1336 MOVAB UBA_IOBASE(R6)[R2], - ;SET ADDRESS OF
      0416 1337 ACFSL_CONTRLREG(R7) ; CONTROL REGISTER
      10 A7 51 B0 0416 1338 MOVW R1,ACFSW_CVECTOR(R7) ;SET ADDRESS OF INTERRUPT VECTOR
      0074'CF 51 7D 041A 1339 MOVQ P1,W^ACFSL_R1SAVE ;SAVE REGISTERS
      007C'CF 53 7D 041F 1340 MOVQ R3,W^ACFSL_R3SAVE
      0084'CF 55 D0 0424 1341 MOVL R5,W^ACFSL_R5SAVE
      0052 30 0429 1342 BSBW FIX_DEV_NAME ;FIX UP DEVICE NAME
      000004C1'EF 16 042C 1343 JSB LOAD_DRIVER ;LOAD DRIVER
      003E'DF 16 0432 1344 JSB @W^ACFSL_ROUTINE ;CALL DEVICE GENERATION ROUTINE
      51 14 A7 D0 0436 1345 MOVL ACFSL_DEVNAME(R7),R1 ;GET ADDRESS OF DEVICE NAME STRING
      030E 30 043A 1346 BSBW ACFSINC_CHAR ;INCREMENT CONTROLLER
      043D 1347
      51 0074'CF 7D 043D 1348 MOVQ W^ACFSL_R1SAVE,R1 ;RESTORE REGISTERS
      53 007C'CF 7D 0442 1349 MOVQ W^ACFSL_R3SAVE,R3
      55 0084'CF D0 0447 1350 MOVL W^ACFSL_R5SAVE,R5
      B0 63 02 E1 044C 1351 50$: BBC #UBA_V_FLOATVEC,(R3),40$ ;IF CLR, FIXED VECTOR ASSIGNMENT
      50 1E A7 9A 0450 1352 MOVZBL ACFSB_NUMVEC(R7),R0 ;GET NUMBER OF CONTROLLER INTERRUPT VECTORS
      51 6140 DE 0454 1353 MOVAL (R1)[R0],R1 ;CALCULATE ADDRESS OF NEXT VECTOR
50 0000006E'EF 3C 0458 1354 MOVZWL ACFSW_VECMOD,R0 ;GET VECTOR MODULUS
      51 50 A0 045F 1355 ADDW R0,R1 ;ROUND UP TO NEXT VECTOR
      51 50 AA 0462 1356 BICW R0,R1 ;TRUNCATE TO ACTUAL VECTOR OFFSET
      52 64 A0 0465 1357 ADDW (R4),R2 ;CALCULATE ADDRESS OF NEXT CSR
      52 D6 0468 1358 INCL R2
      94 11 046A 1359 BRB 40$
      0243'CF 51 B0 046C 1360 60$: MOVW R1,W^ACFSW_VECBASE ;SAVE NEW VECTOR OFFSET
      0241'CF 52 B0 0471 1361 MOVW R2,W^ACFSW_CSRBASE ;SAVE NEW CSR OFFSET
      0241'CF 02 A0 0476 1362 ADDW #2,W^ACFSW_CSRBASE ;ADVANCE PAST ONE REGISTER BLOCK
      FE7F 31 047B 1363 BRW 10$
      047E 1364 .DSABL LSB
```

```

047E 1366          .SBTTL  FIX_DEV_NAME - Check for system device name match/conflict
047E 1367
047E 1368 ;+
047E 1369 :-
047E 1370 ; If the adapter # and CSR match the system device then use the system device
047E 1371 ; name else if the device to be configured matches the system device name
047E 1372 ; increment the controller of the device to be configured.
047E 1373 :-
047E 1374          R7 - Address of ACF block
047E 1375 :-
047E 1376 :-
047E 1377
047E 1378 FIX_DEV_NAME:
047E 1379
51 000024A'EF 9E 047E 1380      MOVAB  ACF$T_SYS_DEVNAME,R1      ;GET SYSTEM DEVICE NAME ADDRESS
   52 81      9A 0485 1381      MOVZBL (R1)+,R2      ;R2 = SIZE, R1 = ADDRESS
   53 14 A7   D0 0488 1382      MOVL   ACF$L_DEVNAME(R7),R3     ;GET DEVICE NAME ADDRESS
   54 83      9A 048C 1383      MOVZBL (R3)+,R4      ;R4 = SIZE, R3 = ADDRESS
1B 0000249'EF E9 048F 1384      BLBC   ACF$B_BOOT_TR,10$      ;SYSTEM DEVICE ON THIS ADAPTER?
0000245'EF 0C A7 D1 0496 1385      CMPL  ACF$L_CONTRLREG(R7),ACF$L_SYS_CSR; SYSTEM DEVICE CSR?
   11      12 049E 1386      BNEQ  10$      ;IF NEQ NO
63 54 20 61 52 2D 04A0 1387      CMPCS  R2,(R1),#^A/ /,R4,(R3) ;DEVICE NAME MATCH?
   08      13 04A6 1388      BEQL  5$      ;IF EQL YES, NO WORK
14 A7 000024A'EF 9E 04A8 1389      MOVAB  ACF$T_SYS_DEVNAME,ACF$L_DEVNAME(R7);USE SYSTEM DEVICE NAME
   07      05 04B0 1390 5$:      RSB      ;RETURN
63 54 20 61 52 2D 04B1 1391 10$:   CMPCS  R2,(R1),#^A/ /,R4,(R3) ;DEVICE NAME MATCH?
   07      12 04B7 1392      BNEQ  20$      ;IF NEQ NO, NO WORK
   51 14 A7   D0 04B9 1393      MOVL  ACF$L_DEVNAME(R7),R1     ;GET DEVICE NAME ADDRESS
   0288     30 04BD 1394      BSBW  ACF$INC_CHAR      ;INCREMENT THE CONTROLLER
   05 04C0 1395 20$:   RSB
  
```

```

04C1 1397          .SBTTL LOAD_DRIVER - Co-routine callback to load driver
04C1 1398
04C1 1399 :+
04C1 1400 :
04C1 1401 : Driver is preloaded before action routine is called if action routine
04C1 1402 : is ACF$ADD_UNITS.
04C1 1403 :
04C1 1404 :
04C1 1405 : R7 - Address of ACF block
04C1 1406 :
04C1 1407 :-
04C1 1408
04C1 1409 LOAD_DRIVER:
04C1 1410
003E'CF 000004E5'8F D1 04C1 1411          CMPL #ACF$ADD_UNITS,W^ACF$R_ROUTINE ; Is this a generic routine device?
01 13 04CA 1412          BEQL 10$ ; No - Exit
05 05 04CC 1413          RSB ; Return
04CD 1414
00 0B A7 03 E2 04CD 1415 10$: BBSS #ACF$V_NOLOAD_DB,ACF$B_AFLAG(R7),20$ ; Don't load data base
50 01 D0 04D2 1416 20$: MOVL #1,R0 ; Set success
0036'CF 8ED0 04D5 1417          POPL W^ACF$R_RETURN ; Save return back to AUTOCONFIGURE
9E 16 04DA 1418          JSB @(SP)+ ; Co-routine back to sysgen
04DC 1419 ; to load driver only
00 0B A7 03 E5 04DC 1420          BBCC #ACF$V_NOLOAD_DB,ACF$B_AFLAG(R7),30$ ; Clear NOLOAD bit
0036'DF 17 04E1 1421 30$: JMP @W^ACF$R_RETURN ; Return to caller
04E5 1422

```

```

04E5 1424      .SBTTL ACF$ADD_UNITS - GENERIC ROUTINE FOR DEVICE GENERATION
04E5 1425      :
04E5 1426      : ACF$CR11 - CR11 CARD READER
04E5 1427      : ACF$LP11 - LP11 LINE PRINTER
04E5 1428      : ACF$LPA11 - LPA11 LABORATORY I/O SUBSYSTEM, CONTROLLER A
04E5 1429      : ACF$LPA11B - LPA11 LABORATORY I/O SUBSYSTEM, CONTROLLER B
04E5 1430      : ACF$DMC11 - DMC11 SYNCHRONOUS COMMUNICATIONS
04E5 1431      : ACF$TS11 - TS11 MAGTAPE, CONTROLLER A
04E5 1432      : ACF$TS11B - TS11 MAGTAPE, CONTROLLER B
04E5 1433      :
04E5 1434      : THIS ROUTINE IS CALLED AS AN ACTION ROUTINE FROM THE UBA DEVICE SCAN TO GENERATE
04E5 1435      : A SINGLE UNIT DATA BASE.
04E5 1436      :
04E5 1437      : IT IS ALSO CALLED AS AN ACTION ROUTINE FOR UNSUPPORTED DEVICES.
04E5 1438      :
04E5 1439      : INPUTS:
04E5 1440      :
04E5 1441      :         R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
04E5 1442      :         R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
04E5 1443      :         R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
04E5 1444      :
04E5 1445      : OUTPUTS:
04E5 1446      :
04E5 1447      :         A CO-ROUTINE CALL IS MADE TO THE ORIGINAL CALLER DELIVERING THE DEVICE UNIT
04E5 1448      :         DESCRIPTOR.
04E5 1449      :
04E5 1450      :
04E5 1451      : Unsupported:
04E5 1452      :
04E5 1453      ACF$DH11:
04E5 1454      ACF$DJ11:
04E5 1455      ACF$DC11:
04E5 1456      ACF$DM11B:
04E5 1457      ACF$DN11:
04E5 1458      ACF$DQ11:
04E5 1459      ACF$DR11B:
04E5 1460      ACF$DR11C:
04E5 1461      ACF$DT11:
04E5 1462      ACF$DU11:
04E5 1463      ACF$DUP11:
04E5 1464      ACF$DV11:
04E5 1465      ACF$DWR70:
04E5 1466      ACF$DX11:
04E5 1467      ACF$GT40:
04E5 1468      ACF$DL11C:
04E5 1469      ACF$KMC11:
04E5 1470      ACF$KW11C:
04E5 1471      ACF$KW11W:
04E5 1472      ACF$LK11:
04E5 1473      ACF$LPP11:
04E5 1474      ACF$LPS11:
04E5 1475      ACF$PP611:
04E5 1476      ACF$PR611:
04E5 1477      ACF$RSV:
04E5 1478      ACF$RX11:
04E5 1479      ACF$VMV21:
04E5 1480      ACF$VMV31:

```

```

04E5 1481 ACF$DPV11:
04E5 1482 ACF$ISB11:
04E5 1483 ACF$KMS11:
04E5 1484 ACF$PCL11:
04E5 1485 ACF$VS100:
04E5 1486 ACF$KMV11:
04E5 1487 ACF$IEQ11:
04E5 1488 ACF$KCT32:
04E5 1489 ACF$TC11:
04E5 1490
04E5 1491 ; Supported:
04E5 1492
04E5 1493 ACF$CR11:
04E5 1494 ACF$LP11:
04E5 1495 ACF$TU58:
04E5 1496 ACF$RB730:
04E5 1497 ACF$LPA11:
04E5 1498 ACF$DMC11:
04E5 1499 ACF$TS11:
04E5 1500 ACF$DR11W:
04E5 1501 ACF$DMP11:
04E5 1502 ACF$DMV11:
04E5 1503 ACF$DZ11:
04E5 1504 ACF$RK611:
04E5 1505 ACF$RX211:
04E5 1506 ACF$DHV11:
04E5 1507 ACF$UNA:
04E5 1508 ACF$QNA:
04E5 1509 ACF$VCO1B:
04E5 1510
04E5 1511 ACF$ADD_UNITS:
04E5 1512
04E5 1513 POPL W^ACF$L_RETURN ; Save return address
04EA 1514 CLRL W^ACF$L_DELIVER_UNIT ; Deliver unit routine in driver
51 0042'CF D4 04EE 1515 MOVL W^ACF$GL_DPT,R1 ; DPT of driver just loaded
0000'CF D0 04F3 1516 BEQL 5$ ; Branch if none
OD 13 04F5 1517
0042'CF 1C A1 B0 04F5 1518 MOVW DPT$W_DELIVER(R1),W^ACF$L_DELIVER_UNIT
05 13 04FB 1519 BEQL 5$ ; No driver-specified routine
0042'CF 51 C0 04FD 1520 ADDL2 R1,W^ACF$L_DELIVER_UNIT ; Set address of driver action routine
55 D4 0502 1521
0502 1522 5$: CLRL R5 ; Set starting unit number
0504 1523
12 A7 55 B0 0504 1524 10$: MOVW R5,ACF$W_CUNIT(R7) ; Next unit number
51 0042'CF D0 0508 1525 MOVL W^ACF$L_DELIVER_UNIT,R1 ; Is there a driver routine?
17 13 050D 1526 BEQL 20$ ; Branch if not
53 00000000'EF D0 050F 1527 MOVL ACF$GL_IDB,R3 ; SET ADDRESS
50 01 D0 0516 1528 MOVL #1,R0 ; Set success for driver
54 0C A7 D0 0519 1529 MOVL ACF$L_CONTRLREG(R7),R4 ; Set CSR address for driver
61 16 051D 1530 JSB (R1) ; Call driver to deliver unit #n
12 A7 55 B0 051F 1531 MOVW R5,ACF$W_CUNIT(R7) ; Save Unit # returned by driver
05 50 E9 0523 1532 BLBC R0,30$ ; Branch if driver returns no
50 01 D0 0526 1533
50 9E 16 0526 1534 20$: MOVL #1,R0 ; Set success indicator
0529 1535 JSB @($P)+ ; Deliver unit to caller
51 0000'CF D0 052B 1536
052B 1537 30$: MOVL W^ACF$GL_DPT,R1 ; Address of DPT

```



51	1A	A1	OC	13	0530	1538	BEQL	40\$		; Driver not found
				3C	0532	1539	MOVZWL	DPT\$W_DEFUNITS(R1),R1		; Number of driver specified units
					0536	1540				; Unit number (R5) can change in DELIVER
55	12	A7		3C	0536	1541	MOVZWL	ACF\$W_CUNIT(R7),R5		; Set unit number
C6	55	51		F2	053A	1542	AOBLSS	R1,R5,10\$		; Any more units to process?
					053E	1543				
				0036'DF	17	053E	1544	40\$: JMP	@W^ACF\$L_RETURN	; Return to caller
					0542	1545				

```

0542 1547          .SBTTL CLASS DRIVER DEVICE GENERATOR
0542 1548          :
0542 1549          : ACF$UDA - UDA DISK CONTROLLER
0542 1550          : ACF$TU81 - TU81 TAPE CONTROLLER
0542 1551          :
0542 1552          : THIS ROUTINE IS CALLED AS AN ACTION ROUTINE FROM THE UBA DEVICE SCAN TO
0542 1553          : GENERATE THE DATA BASE FOR A SINGLE PORT DEVICE AND THE ASSOCIATED
0542 1554          : CLASS DRIVER DATA BASE.
0542 1555          :
0542 1556          : INPUTS:
0542 1557          :
0542 1558          :     R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
0542 1559          :     R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
0542 1560          :     R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
0542 1561          :
0542 1562          : OUTPUTS:
0542 1563          :
0542 1564          :     A CO-ROUTINE CALL IS MADE TO THE ORIGINAL CALLER DELIVERING EACH DEVICE
0542 1565          :     UNIT DESCRIPTOR.
0542 1566          :
0542 1567          :
0542 1568          : ACF$TU81:
0542 1569          : ACF$UDA:
0542 1570          :
003A'CF 8ED0 0542 1571          POPL      W^ACF$L_RETURN2          ;SAVE RETURN ADDRESS
0547 1572          :
0547 1573          : First load port driver and create units normally.
0547 1574          :
0547 1575          :     BSBB      ACF$ADD_UNITS
0549 1576          :
0549 1577          :
0549 1578          : Now, load class driver and its associated database.
0549 1579          :
0549 1580          :
00000046'EF 28 28 0549 1581          MOVCL   #ACF$C_LENGTH,-
0548 1582          :     (R7),ACF$L_ACF_SAVE          ; Save ACF block
0551 1583          :
0551 1584          :
0551 1585          : Pick up name of class device and driver
0551 1586          :
0551 1587          :
0551 1588          :
52 51 14 A7 D0 0551 1588          MOVL     ACF$L_DEVNAME(R7),R1          ; Get address of port device name
00000228'EF 9E 0555 1589          MOVAB   ACF$AB_CLASS_TABLE,R2          ; Get address of association table
055C 1590          :
055C 1590 10$: 055C 1591          MOVL     (R2)+,R0          ; Get pointer to next structure
055F 1592          BEQL     90$          ; If EQL, end of table
01 A1 09 A0 B1 0561 1593          CMPW    CLS$T_PORTDEV(R0),1(R1) ; Do we have a match?
0566 1594          BNEQ    10$          ; If NEQ, no - try again
0568 1595          MOVL     CLS$L_CLASSDEV(R0),-
056A 1596          :     ACF$L_DEVNAME(R7)          ; Store address of class device name
056C 1597          MOVL     CLS$L_CLASSDRV(R0),-
056F 1598          :     ACF$L_DRVNAME(R7)          ; Store address of class driver name
0571 1599          :
0571 1600          :
0571 1601          : Create SYSID
0571 1602          :
50 0C A7 D0 0571 1603          MOVL     ACF$L_CONTRLREG(R7),R0 ; Create SYSID, first longword

```

```

51 51 0C A8 3C 0575 1604      MOVZWL  ADPSW_TR(R8),R1      ; second longword
00008000 8F C8 0579 1605      BISL    #^X8000,R1         ; Set high bit in first word
52 00000000'EF 7E 0580 1606      MOVQA   BOO$GQ_CONSYSID,R2  ; get address of SYSID quadword
62 50 7D 0587 1607      MOVQ    R0,(R2)            ; Set SYSID
0C A7 52 D0 058A 1608      MOVL    R2,ACF$$_CONTRLREG(R7) ; Set in CSR field
058E 1609
10 90 058E 1610      MOVB    #ACF$$_SUPPORT,-    ;
0B A7 0590 1611      ACF$$_XFLAG(R7)           ; Clear configuration flag, set supported
21 A7 94 0592 1612      CLRB    ACF$$_NUMUNIT(R7)  ; Clear number of units
12 A7 B4 0595 1613      CLRW    ACF$$_CUNIT(R7)    ; Clear unit number
1C A7 B4 0598 1614      CLRW    ACF$$_MAXUNITS(R7) ; Clear maxunits
1E A7 01 90 059B 1615      MOVB    #1,ACF$$_CNUMVEC(R7) ; Set cnumvec to 1
01ED 30 059F 1616      BSBW    ACF$$_CLR_ACF
05A2 1617      ;
05A2 1618      ; Deliver class driver
05A2 1619      ;
50 01 D0 05A2 1620      MOVL    #1,R0              ; Set success
9E 16 05A5 1621      JSB     @($P)+             ;
05A7 1622
51 00000000'EF D0 05A7 1623      MOVL    ACF$$_GL_CRB,R1    ; *** temp ?
04 13 05AE 1624      BEQL    20$
67 D0 05B0 1625      MOVL    ACF$$_ADAPTER(R7),- ; Fill in ADP address in CRB
38 A1 05B2 1626      CRB$$_INTD+VEC$$_ADP(R1)
05B4 1627
05B4 1628      ;
05B4 1629      ; Increment device name
05B4 1630      ;
05B4 1631      20$:
51 14 A7 D0 05B4 1632      MOVL    ACF$$_DEVNAME(R7),R1 ; Set address
0190 30 05B8 1633      BSBW    ACF$$_INC_CHAR     ; Do the increment
05BB 1634
05BB 1635      MOVCS   #ACF$$_LENGTH,-    ;
67 00000046'EF 28 05BD 1636      ACF$$_ACF_SAVE,(R7)       ; Restore ACF block
05C3 1637      90$:
003A'DF 17 05C3 1638      JMP     @W^ACF$$_RETURN2   ; Return
05C7 1639

```

```

05C7 1641          .SBTTL  RL11 MULTIPLE UNIT GENERATOR
05C7 1642          :
05C7 1643          : ACF$RL11 - RL11 DISK CONTROLLER
05C7 1644          :
05C7 1645          : THIS ROUTINE IS CALLED AS AN ACTION ROUTINE FROM THE UBA DEVICE SCAN TO GENERATE
05C7 1646          : THE DATE BASE FOR A SINGLE RL11.
05C7 1647          :
05C7 1648          : INPUTS:
05C7 1649          :
05C7 1650          :         R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
05C7 1651          :         R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
05C7 1652          :         R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
05C7 1653          :
05C7 1654          : OUTPUTS:
05C7 1655          :
05C7 1656          :         A CO-ROUTINE CALL IS MADE TO THE ORIGINAL CALLER DELIVERING EACH DEVICE
05C7 1657          :         UNIT DESCRIPTOR.
05C7 1658          :
05C7 1659          :
05C7 1660          : ACF$RL11:
05C7 1661          : POPL      W^ACF$R_RETURN          : SAVE RETURN ADDRESS
05C7 1662          : CLRL      R0                      : SET STARTING UNIT NUMBER
12 A7 50 B0 05CE 1663 70$: MOVW      R0,ACF$W_CUNIT(R7)      : SET CONTROLLER UNIT NUMBER
51 OC A7 D0 05D2 1664      MOVL      ACF$R_CONTRLREG(R7),R1    : GET ADDRESS OF DEVICE CSR
04 A1 0B B0 05D6 1665      BSBB      130$                    : WAIT FOR CONTROLLER READY
05DC 1666      MOVW      #RL_DA_M_MRK!RL_DA_M_STS!-          : SET GET STATUS BITS
05DC 1667      RL_DA_M_RST,RL_DA(R1)
53 50 08 78 05DC 1668      ASHL      #8,R0,R3                : MOVE UNIT NUMBER TO POSITION
61 53 04 A9 05E0 1669      BISW3    #4,R3,RL_CS(R1)          : SELECT DRIVE AND GET STATUS
05E4 1670      BSBB      130$                    : WAIT FOR CONTROLLER READY
61 0400 8F B3 05E6 1671      BITW      #RL_CS_M_OPI,RL_CS(R1)  : OPERATION COMPLETE?
0070'CF 50 D0 05EB 1672      BNEQ     80$                    : IF NEQ NO - NONEXISTENT DRIVE
50 0070'CF 01 D0 05F2 1674      MOVL      R0,W^ACF$R_ROSAVE      : SAVE REGISTER
CE 50 03 F3 05F5 1675      JSB       @(SP)+                : SET SUCCESS INDICATOR
0036'DF 17 05F7 1676      MOVL      W^ACF$R_ROSAVE,R0          : DELIVER UNIT TO CALLER
52 01312D00 8F D0 05FC 1677 80$: AOBLEQ   #3,R0,70$          : RESTORE REGISTER
0604 1678      JMP        @W^ACF$R_RETURN                    : ANY MORE UNITS TO PROCESS?
0604 1679      :
0604 1680 130$: MOVL      #2*500*1000*20,R2                : * should be TIMEWAIT
060B 1681 140$: TSTB      RL_CS(R1)                        : Controller ready?
060D 1682      BGEQ     150$                    : branch if not
060F 1683      RSB       : Continue
0610 1684 150$: SOBGTR   R2,140$                        : Loop
0613 1685      TSTL     (SP)+                            : Get rid of return PC
0615 1686      JMP      @W^ACF$R_RETURN                    : error - exit
0619 1687

```

```

0619 1689      .SBTTL MULTIPLE DEVICE GENERATOR
0619 1690      :
0619 1691      : ACF$COMBO_DEVICES - MULTIPLE DEVICE COMBO BOARD
0619 1692      :
0619 1693      : THIS ROUTINE IS CALLED AS AN ACTION ROUTINE FROM THE UBA DEVICE SCAN TO GENERATE
0619 1694      : THE DATA BASE FOR THE DMF32, WHICH INCLUDES A SYNC LINE, ASYNC LINE (DZ TYPE),
0619 1695      : A LP LINE, AND A DR11C LINE. A MAXIMUM OF 3 OF THESE DEVICES CAN BE ON THE
0619 1696      : BOARD AT ONE TIME, BUT VARIOUS COMBINATIONS OF THE FOUR IS POSSIBLE.
0619 1697      : THIS ROUTINE IS ALSO CALLED FOR THE DMZ32 WHICH INCLUDES 3 ASYNC LINES, AND
0619 1698      : THE CPI32 WHICH INCLUDES 3 SYNC LINES.
0619 1699      :
0619 1700      : INPUTS:
0619 1701      :
0619 1702      : R2 = ADDRESS OF COMBO DEVICE DESCRIPTOR TABLE
0619 1703      : R3 = BITMASK OF DEVICES PRESENT
0619 1704      : R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
0619 1705      : R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
0619 1706      : R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
0619 1707      :
0619 1708      : OUTPUTS:
0619 1709      :
0619 1710      : A CO-ROUTINE CALL IS MADE TO THE ORIGINAL CALLER DELIVERING EACH DEVICE
0619 1711      : UNIT DESCRIPTOR.
0619 1712      :
0619 1713      :
0619 1714      : ACF$BITMASK:
7FFF 0619 1715      .WORD      ^X7FFF      ;BIT MASK FOR EXTZV INSTRUCTION TO
0619 1716      ;CONVERT A NUMBER TO A BIT MASK
0619 1717      : ACF$DMF32:
52 0000016C'EF  D0 0619 1718      MOVL      ACF$AL_DM32_COMBO,R2      ;ADDRESS OF UBADEV TABLE FOR DMF32
00000234'EF  62  D0 0622 1719      MOVL      UBT$L_DEVNAME(R2),ACF$AL_DEVNAME; SAVE COMBO DEVICE NAME
52 000001F4'EF  9E 0629 1720      MOVAB     ACF$AB_DM32_TABLE,R2      ;ADDRESS OF DESCRIPTOR TABLE
0630 1721      :
0630 1722      : READ THE DMF32 IDENT REGISTER TO GET BITMASK OF DEVICES PRESENT
0630 1723      :
0630 1724      : MOVW     @ACF$S_CONTRLREG(R7),R3
53 53 0C B7  B0 0630 1724      : EXTZV   #DMF$V_IDENT,#DMF$S_IDENT,R3,R3
0634 1725      : BRB     ACF$COMBO_DEVICES      ;CALL COMMON COMBO DEVICE ROUTINE
0639 1726      :
0638 1727      : ACF$CPI32:
52 00000194'EF  D0 0638 1729      MOVL      ACF$AL_CPI32_COMBO,R2      ;ADDRESS OF UBADEV TABLE FOR CPI32
00000234'EF  62  D0 0642 1730      MOVL      UBT$L_DEVNAME(R2),ACF$AL_DEVNAME; SAVE COMBO DEVICE NAME
52 00000218'EF  9E 0649 1731      MOVAB     ACF$AB_CPI32_TABLE,R2      ;ADDRESS OF DESCRIPTOR TABLE
0650 1732      : BRB     ACF$DMZCPI      ;CALL COMMON COMBO DEVICE ROUTINE
0652 1733      :
0652 1734      : ACF$DMZ32:
52 00000190'EF  D0 0652 1735      MOVL      ACF$AL_DMZ32_COMBO,R2      ;ADDRESS OF UBADEV TABLE FOR DMZ32
00000234'EF  62  D0 0659 1736      MOVL      UBT$L_DEVNAME(R2),ACF$AL_DEVNAME; SAVE COMBO DEVICE NAME
52 00000208'EF  9E 0660 1737      MOVAB     ACF$AB_DMZ32_TABLE,R2      ;ADDRESS OF DESCRIPTOR TABLE
0667 1738      : ACF$DMZCPI:
0667 1739      : MOVW     @ACF$S_CONTRLREG(R7),R3      ;GET CONTENTS OF THE MAIN CSR
0668 1740      : EXTZV   #CPI$V_SUBCNTRL,-      ;EXTRACT THE NUMBER OF SUBCONTROLLERS
066D 1741      : #CPI$S_SUBCNTRL,R3,R3      :
53 A4 AF 53 00  EF 0670 1742      : EXTZV   #0,R3,B^ACF$BITMASK,R3      ;CONVERT NUMBER TO BIT MASK
0676 1743      :
0676 1744      : ACF$COMBO_DEVICES:
003A'CF 8ED0 0676 1745      : POPL     W^ACF$S_RETURN2      ;SAVE RETURN ADDRESS
    
```

```

00000238'EF 0C A7 D0 067B 1746      MOVL  ACF$AL_CONTRLREG(R7),ACF$AL_CSR      ;SAVE CSR BASE
0000023C'EF 10 A7 B0 0683 1747      MOVW  ACF$W_CVECTOR(R7),ACF$W_VECTOR      ;SAVE VECTOR BASE
00000240'EF 1E A7 90 068B 1748      MOVB  ACF$B_CNUMVEC(R7),ACF$B_NUMVEC      ;SAVE NUMVEC
                                0693 1749
                                0693 1750
                                54 82 D0 0693 1751 10$: MOVL  (R2)+,R4                          ;NEXT DEVICE
                                1A 12 BNEQ 20$                          ;BRANCH IF NOT END OF LIST
14 A7 00000234'EF D0 0698 1753      MOVL  ACF$AL_DEVNAME,ACF$AL_DEVNAME(R7)  ;RESTORE DEVICE NAME
1E A7 00000240'EF 90 06A0 1754      MOVB  ACF$B_NUMVEC,ACF$B_CNUMVEC(R7)     ;RESTORE NUMVEC
                                1F A7 94 06A8 1755      CLRB  ACF$B_COMBO_VECTOR_OFFSET(R7)     ;RESET COMBO SPECIFIC ACF FIELDS
                                20 A7 94 06AB 1756      CLRB  ACF$B_COMBO_CSR_OFFSET(R7)       ;
                                06AE 1757
                                003A'DF 17 06AE 1758      JMP   @W^ACF$AL_RETURN2                ;RETURN
                                06B2 1759
                                53 84 B3 06B2 1760 20$: BITW  (R4)+,R3                          ;IS THIS DEVICE ON BOARD?
                                DC 13 06B5 1761      BEQL  10$                              ;BRANCH IF NOT
                                06B7 1762
                                14 A7 84 D0 06B7 1763      MOVL  (R4)+,ACF$AL_DEVNAME(R7)          ;DEVICE NAME
                                18 A7 84 D0 06BB 1764      MOVL  (R4)+,ACF$AL_DRVNAME(R7)          ;DRIVER NAME
                                1E A7 84 90 06BF 1765      MOVB  (R4)+,ACF$B_CNUMVEC(R7)          ;NUMBER OF VECTORS
51 64 08 02 EF 06C3 1766      EXTZV #2,#8,(R4),R1                    ;OFFSET IN LONGWORDS
                                1F A7 51 90 06C8 1767      MOVB  R1,ACF$B_COMBO_VECTOR_OFFSET(R7) ;OFFSET BACK TO START OF VECTOR
10 A7 0000023C'EF 84 A1 06CC 1768      ADDW3 (R4)+,ACF$AW_VECTOR,ACF$W_CVECTOR(R7) ;ADDRESS OF VECTOR
                                20 A7 64 8E 06D5 1769      MNEGB (R4),ACF$B_COMBO_CSR_OFFSET(R7) ;OFFSET BACK TO START OF CSR
0C A7 00000238'EF 84 C1 06D9 1770      ADDL3 (R4)+,ACF$AL_CSR,ACF$AL_CONTRLREG(R7) ;ADDRESS OF CSR
                                06E2 1771
                                0B A7 10 88 06E2 1772      BISB2 #ACF$M_SUPPORT,ACF$B_AFLAG(R7)   ;ASSUME NOSUPPORT
                                04 84 00 E1 06E6 1773      BBC   #UBA_V_SUPPORT,(R4)+,30$        ;BRANCH IF NOSUPPORT
                                0B A7 10 8A 06EA 1774      BICB2 #ACF$M_SUPPORT,ACF$B_AFLAG(R7)   ;SET SUPPORTED
                                06EE 1775
                                0088'CF 52 7D 06EE 1776 30$: MOVQ  R2,W^ACF$AL_R2R3SAVE      ;SAVE R2,R3
                                06F3 1777
                                FDEF 30 06F3 1778      BSBW  ACF$ADD_UNITS                    ;USE COMMON ROUTINE TO DELIVER UNITS
51 14 A7 D0 06F6 1779      MOVL  ACF$AL_DEVNAME(R7),R1            ;GET DEVICE NAME ADDRESS
                                004E 30 06FA 1780      BSBW  ACF$INC_CHAR                      ;INCREMENT CONTROLLER LETTER
52 0088'CF 7D 06FD 1781      MOVQ  W^ACF$AL_R2R3SAVE,R2            ;RESTORE REGISTERS
                                FF8E 31 0702 1782      BRW   10$                              ;LOOP
                                0705 1783

```

```

0705 1785      .SBTTL  AUTO CONFIGURATION DEVICE DATA BASE RESET
0705 1786      :+
0705 1787      : IOC$AUTORESET - AUTO CONFIGURATION DEVICE DATA BASE RESET
0705 1788      :
0705 1789      : THIS ROUTINE IS CALLED TO RESET THE AUTO CONFIGURATION DEVICE DATA BASE.
0705 1790      :
0705 1791      : INPUTS:
0705 1792      :
0705 1793      :     NONE.
0705 1794      :
0705 1795      : OUTPUTS:
0705 1796      :
0705 1797      :     THE CONTROLLER DESIGNATORS ARE ALL RESET TO 'A', WITH
0705 1798      :     THE EXCEPTION OF A FEW DEVICES IN UBATABLE THAT ARE RESET
0705 1799      :     TO THE LETTER STORED IN THAT TABLE.
0705 1800      :
0705 1801      :-
0705 1802
0705 1803 IOC$AUTORESET::
50  00000000'EF 9E 0705 1804 MOVAB  ACFSAL_RESET,R0      ;AUTO CONFIGURATION DEVICE DATA BASE RESET
      51  80  D0 070C 1805 10$:  MOVL  (R0)+,R1      ;GET ADDRESS OF RESET POINTER TABLE
      06  13 070F 1806      BEQL  20$      ;GET ADDRESS OF NEXT CONTROLLER DESIGNATOR
      71  41 8F 90 0711 1807      MOVB  #^A/A/,-(R1)  ;IF EQL END OF TABLE
      FS  11 0715 1808      BRB   10$      ;RESET CONTROLLER DESIGNATOR
      0717 1809      ;LOOP
50  00000090'EF 9E 0717 1810 20$:  MOVQ  R2,-(SP)      ;SAVE R2,R3
      51  80  D0 071A 1811      MOVAB  ACFSAB_UBATABLE,R0  ;BASE ADDRESS OF TABLE
      0D  13 0721 1812      ;
      52  61  D0 0721 1813 30$:  MOVL  (R0)+,R1      ;NEXT ELEMENT IN TABLE
      53  62  D0 0724 1814      BEQL  40$      ;END OF LIST
      6342 10 A1 90 0726 1815      MOVL  UBT$L_DEVNAME(R1),R2  ;ADDRESS OF NAME TO RESET
      EE  11 9A 0729 1816      MOVZBL (R2),R3      ;EXPAND LENGTH
      0731 1817      MOVB  UBT$B_LETTER(R1),(R3)[R2] ;RESET CONTROLLER LETTER
      0733 1818      BRB   30$      ;LOOP
52  00000110'EF D0 0733 1820 40$:  MOVL  ACFSAL_DZ11_TTA,R2  ;SPECIAL CASE TT
      52  62  D0 073A 1821      MOVL  UBT$L_DEVNAME(R2),R2  ;ADDRESS OF DEVICE NAME
      82  03 90 073D 1822      MOVB  #3,(R2)+      ;SET COUNT AT 3
62  20415454 8F D0 0740 1823      MOVL  #^A/TTA /,(R2)  ;MOVE IN CORRECT FIELD
      52  8E 7D 0747 1824      MOVQ  (SP)+,R2      ;RESTORE R2,R3
      05 074A 1825      RSB   ;RETURN

```

```

074B 1827 .SBILL ROUTINE INC_CHAR
074B 1828
074B 1829 :+
074B 1830 :
074B 1831 : Routine to increment device name last character
074B 1832 :
074B 1833 : CALLING SEQUENCE:
074B 1834 :
074B 1835 :     BSBx ACF$INC_CHAR (called from AUTOCONFIG and CONFIG)
074B 1836 :
074B 1837 : INPUT:
074B 1838 :
074B 1839 :     R1 - Address of device name ascic string
074B 1840 :
074B 1841 : OUTPUT:
074B 1842 :
074B 1843 :     Device name with last character incremented
074B 1844 :
074B 1845 :     For TTcn: (** NOT ACTIVATED YET **)
074B 1846 :         Controller is incremented until TTZ and then goes
074B 1847 :         to TTAA,TTAB,...TTAZ,TTBA,...
074B 1848 :
074B 1849 :     For all other:
074B 1850 :         DDA,...,DDP,DEA,...,DEP - wraps at P because its the
074B 1851 :         16th letter of the alphabet. (For RSX compatability).
074B 1852 :
074B 1853 :     TT devices (WILL BE) special cased in the AME.
074B 1854 :
074B 1855 :     R0 does not return status
074B 1856 :
074B 1857 :-
074B 1858
074B 1859 ACF$INC_CHAR::
074B 1860
50 61 9A 074B 1861     MOVZBL (R1),R0           ; Get length from first byte
074E 1862 :
074E 1863 : The following line of code can be nop'ed out to activate the TTAA, etc
074E 1864 : for terminals.
074E 1865 :
5454 8F 08 11 074E 1866     BRB 5$
01 A1 B1 0750 1867     CMPW 1(R1),#^A/TT/       ; Terminal ?
14 13 0756 1868     BEQL 20$              ; yes - special case code
0758 1869
51 8F 6041 96 0758 1870 5$:   INCB (R0)[R1]           ; Increment the last character
6041 09 91 075B 1871     CMPB (R0)[R1],#^A/Q/     ; Is this the 17th controller ?
FF A041 1F 0760 1872     BLSSU 10$              ; No, branch
6041 41 8F 96 0762 1873     INCB -1(R0)[R1]        ; Increment last char of device name
05 0766 1874     MOVB #^A/A/, (R0)[R1] ; Move /A/ to last
076B 1875 10$:   RSB           ; Return
076C 1876
5A 8F 6041 91 076C 1877 20$:   CMPB (R0)[R1],#^A/Z/     ; Ready to wrap ?
04 13 0771 1878     BEQL 30$              ; Branch if yes
6041 96 0773 1879     INCB (R0)[R1]        ; Increment last character
05 0776 1880     RSB           ; Return
0777 1881
61 03 91 0777 1882 30$:   CMPB #3, (R1)         ; Still TTx ?
0A 12 077A 1883     BNEQ 40$            ; Branch if not

```



03	A1	61	04	90	077C	1884		MOVB	#4,(R1)		; Set up longer name
		41	8F	80	077F	1885		MOVW	#^A/AA/,3(R1)		; Move in "AA"
				05	0785	1886		RSB			; Return
					0786	1887					
04	A1	03	A1	96	0786	1888	40\$:	INCB	3(R1)		; Increment first controller letter
		41	8F	90	0789	1889		MOVB	#^A/A/,4(R1)		; Reset last character
				05	078E	1890		RSB			
					078F	1891					

```

078F 1893 .SBTTL ROUTINE CLR_ACF
078F 1894
078F 1895 :+
078F 1896 :
078F 1897 : Routine to clear device data block portions of the autoconfigure
078F 1898 : context block (all those written by SGN$GET_DEVICE)
078F 1899 :
078F 1900 : INPUT
078F 1901 :     R7 - Address of ACF block
078F 1902 :
078F 1903 : OUTPUT
078F 1904 :     Cleared cells
078F 1905 :
078F 1906 : CALLING SEQUENCE
078F 1907 :     BSBx   ACF$CLR_ACF
078F 1908 :-
078F 1909
078F 1910 ACF$CLR_ACF::
078F 1911
00000000'EF D4 078F 1912     CLRL   ACF$GL_DDB           ; Clear cells
00000000'EF D4 0795 1913     CLRL   ACF$GL_UCB
00000000'EF D4 079B 1914     CLRL   ACF$GL_IDB
00000000'EF D4 07A1 1915     CLRL   ACF$GL_CRB
00000000'EF D4 07A7 1916     CLRL   ACF$GL_SB
00000000'EF D4 07AD 1917     CLRL   ACF$GL_LASTDDB
05 07B3 1918     RSB
07B4 1919
07B4 1920     .END
  
```

AUTOCONFIG  
Symbol table

\$SS	= 00000111 R	04	\$OJS	= 000000A9 R	04
\$COMBOS	= 000000D2 R	04	\$OJDRIVERS	= 00000171 R	05
\$CRS	= 00000018 R	04	\$OKS	= 000000AD R	04
\$CRDRIVERS	= 00000036 R	05	\$OKDRIVERS	= 0000017A R	05
\$DBS	= 00000000 R	04	\$OLS	= 000000B5 R	04
\$DBDRIVERS	= 00000000 R	05	\$OLDRIVERS	= 0000018C R	05
\$DDS	= 00000048 R	04	\$OMS	= 00000044 R	04
\$DDDRIVERS	= 00000099 R	05	\$OMDRIVERS	= 00000090 R	05
\$DEVDESCS	= 000008F3 R	06	\$ONS	= 000000CA R	04
\$DEVNAMES	= 0000010D R	04	\$ONDRIVERS	= 000001BA R	05
\$DLS	= 00000024 R	04	\$OODRIVERS	= 00000129 R	05
\$DLDRIVERS	= 00000051 R	05	\$OQS	= 000000E5 R	04
\$DMS	= 0000001C R	04	\$OQDRIVERS	= 000001F0 R	05
\$DMDRIVERS	= 0000003F R	05	\$ORS	= 0000007C R	04
\$DQS	= 00000030 R	04	\$ORDRIVERS	= 0000010E R	05
\$DQDRIVERS	= 0000006C R	05	\$OSDRIVERS	= 000001CC R	05
\$DRS	= 00000004 R	04	\$PAS	= 00000014 R	04
\$DRDRIVERS	= 00000009 R	05	\$PADRIVERS	= 0000002D R	05
\$DRVNAMES	= 00000253 R	05	\$PPS	= 0000005C R	04
\$DTS	= 000000F5 R	04	\$PPDRIVERS	= 000000C6 R	05
\$DTRIVERS	= 00000214 R	05	\$PRS	= 00000058 R	04
\$DYS	= 0000002C R	04	\$PRDRIVERS	= 000000BD R	05
\$DYDRIVERS	= 00000063 R	05	\$PTS	= 00000038 R	04
\$DZDRIVERS	= 0000014D R	05	\$PUS	= 00000034 R	04
\$ISS	= 000000CE R	04	\$PUDRIVERS	= 00000075 R	05
\$ISDRIVERS	= 000001C3 R	05	\$RSVS	= 000000B9 R	04
\$IXS	= 000000ED R	04	\$RSVDRIVERS	= 00000195 R	05
\$IXDRIVERS	= 00000202 R	05	\$RTNAMES	= 00000170 R	07
\$LAS	= 000000B1 R	04	\$TFDRIVERS	= 0000001B R	05
\$LADRIVERS	= 00000183 R	05	\$TMDRIVERS	= 00000012 R	05
\$LCS	= 00000105 R	04	\$TSDRIVERS	= 0000005A R	05
\$LCDRIVERS	= 00000241 R	05	\$TTAS	= 00000098 R	04
\$LPS	= 00000020 R	04	\$TXS	= 000000F1 R	04
\$LPDRIVERS	= 00000048 R	05	\$UKS	= 000000E9 R	04
\$LSS	= 00000078 R	04	\$UKDRIVERS	= 000001F9 R	05
\$LSDRIVERS	= 00000105 R	05	\$VBS	= 000000E1 R	04
\$MFS	= 0000000C R	04	\$VBDRIVERS	= 000001E7 R	05
\$MSS	= 00000028 R	04	\$VCS	= 000000F9 R	04
\$MTS	= 00000008 R	04	\$VCDRIVERS	= 0000021D R	05
\$OAS	= 00000054 R	04	\$XAS	= 000000BE R	04
\$OADRIVERS	= 000000B4 R	05	\$XADRIVERS	= 0000019F R	05
\$OBS	= 0000004C R	04	\$XBS	= 000000C2 R	04
\$OBDRIVERS	= 000000A2 R	05	\$XBDRIVERS	= 000001A8 R	05
\$OCS	= 00000060 R	04	\$XDS	= 000000C6 R	04
\$OCDRIVERS	= 000000CF R	05	\$XDDRIVERS	= 000001B1 R	05
\$ODS	= 00000064 R	04	\$XES	= 0000003C R	04
\$ODDRIVERS	= 000000D8 R	05	\$XEDRIVERS	= 0000007E R	05
\$OES	= 00000074 R	04	\$XFS	= 00000010 R	04
\$OEDRIVERS	= 000000FC R	05	\$XFDRIVERS	= 00000024 R	05
\$OFS	= 00000080 R	04	\$XGS	= 000000FD R	04
\$OFDRIVERS	= 00000117 R	05	\$XGDRIVERS	= 00000226 R	05
\$OGS	= 00000090 R	04	\$XIS	= 00000101 R	04
\$OGDRIVERS	= 0000013B R	05	\$XIDRIVERS	= 00000238 R	05
\$OHS	= 000000A1 R	04	\$XKS	= 0000009D R	04
\$OHDRIVERS	= 0000015F R	05	\$XKDRIVERS	= 00000156 R	05
\$OIS	= 000000A5 R	04	\$XMS	= 00000094 R	04
\$OIDRIVERS	= 00000168 R	05	\$XMDRIVERS	= 00000144 R	05

\$XPS	= 000000DD	R	04	ACFSAL_DWR70_OK	00000124	R	02
\$XPDRIVERS	= 000001DE	R	05	ACFSAL_DX11_OD	000000DC	R	02
\$XQS	= 00000040	R	04	ACFSAL_DZ11_TTA	00000110	R	02
\$XQDRIVERS	= 00000087	R	05	ACFSAL_GT40_OE	00C000EC	R	02
\$XSS	= 000000D9	R	04	ACFSAL_IEQ1T_IX	00000188	R	02
\$XSDRIVERS	= 000001D5	R	05	ACFSAL_ISB11-IS	0000015C	R	02
\$XUS	= 00C00084	R	04	ACFSAL_KCT32-UK	00000184	R	02
\$XUDRIVERS	= 00000120	R	05	ACFSAL_KMC11-XK	00000114	R	02
\$XVS	= 0000008C	R	04	ACFSAL_KMS11-XS	00000170	R	02
\$XVDRIVERS	= 00000132	R	05	ACFSAL_KMV11-OQ	00000180	R	02
\$XWS	= 00000088	R	04	ACFSAL_KW11C-OL	00000138	R	02
\$YCDRIVERS	= 0000022F	R	05	ACFSAL_KW11W-OF	000000F8	R	02
\$YFDRIVERS	= 0000020B	R	05	ACFSAL_LK11-OG	00000108	R	02
\$YHS	= 00000070	R	04	ACFSAL_LP11-LP	00000098	R	02
\$YHDRIVERS	= 000000F3	R	05	ACFSAL_LPA1T-LA	00000130	R	02
\$YJS	= 0000006C	R	04	ACFSAL_LPP11-OH	00000118	R	02
\$YJDRIVERS	= 000000EA	R	05	ACFSAL_LPS11-LS	000000F0	R	02
\$YLS	= 00000068	R	04	ACFSAL_PCL11-XP	00000174	R	02
\$YLDRIVERS	= 000000E1	R	05	ACFSAL_PP611-PP	000000C4	R	02
\$YMS	= 00000050	R	04	ACFSAL_PR611-PR	000000D0	R	02
\$YMDRIVERS	= 000000AB	R	05	ACFSAL_QNA_XQ	000000B8	R	02
ACFSAB_ADPTYPE	00000000	R	02	ACFSAL_RB730-DQ	00000148	R	02
ACFSAB_CITABLE	00000032	R	02	ACFSAL_RESET	00000000	R	03
ACFSAB_CLASS_TABLE	00000228	R	02	ACFSAL_RK611-DM	00000094	R	02
ACFSAB_CPI32_TABLE	00000218	R	02	ACFSAL_RL11-DL	0000009C	R	02
ACFSAB_DMF32_TABLE	000001F4	R	02	ACFSAL_RSV_RSV	0000013C	R	02
ACFSAB_DMZ32_TABLE	00000208	R	02	ACFSAL_RX2T1-DY	000000A4	R	02
ACFSAB_DRTABLE	0000002E	R	02	ACFSAL_TC11-DT	00000198	R	02
ACFSAB_MBATABLE	0000001A	R	02	ACFSAL_TS11-MS	000000A0	R	02
ACFSAB_NUMVEC	00000240	R	02	ACFSAL_TU58-DD	000000C0	R	02
ACFSAB_UBATABLE	00000090	RG	02	ACFSAL_TU81-PT	000000B0	R	02
ACFSADD_UNITS	000004E5	R	08	ACFSAL_UDA-PU	000000AC	R	02
ACFSAL_CPI32_COMBO	00000194	R	02	ACFSAL_UNA-XE	000000B4	R	02
ACFSAL_CR11_CR	00000090	R	02	ACFSAL_VCOTB-VC	0000019C	R	02
ACFSAL_CSR	00000238	R	02	ACFSAL_VMV21-OI	0000011C	R	02
ACFSAL_DC11_DM	000000BC	R	02	ACFSAL_VMV31-OJ	00000120	R	02
ACFSAL_DEVNAME	00000234	R	02	ACFSAL_VS100-VB	00000178	R	02
ACFSAL_DH11_YH	000000E8	R	02	ACFSAW_SAVEVEC	0000023E	R	02
ACFSAL_DHV1T_TX	0000018C	R	02	ACFSAW_VECTOR	0000023C	R	02
ACFSAL_DJ11_VJ	000000E4	R	02	ACFSBITMASK	00000619	R	08
ACFSAL_DL11C_YL	000000E0	R	02	ACFSB_AFLAG	= 0000000B		
ACFSAL_DM11B_YM	000000C8	R	02	ACFSB_AUNIT	= 0000000A		
ACFSAL_DMC11_XM	0000010C	R	02	ACFSB_BOOT_TR	00000249	R	02
ACFSAL_DMF32_COMBO	0000016C	R	02	ACFSB_CNUMVEC	= 0000001E		
ACFSAL_DMP11_XD	00000154	R	02	ACFSB_COMBO_CSR_OFFSET	= 00000020		
ACFSAL_DMV11_XD	00000160	R	02	ACFSB_COMBO_VECTOR_OFFSET	= 0000001F		
ACFSAL_DMZ32_COMBO	00000190	R	02	ACFSB_NUMUNIT	= 00000021		
ACFSAL_DN11_OB	000000C4	R	02	ACFSCT	0000024D	R	08
ACFSAL_DPV1T_ON	00000158	R	02	ACFSCLR_ACF	0000078F	RG	08
ACFSAL_DQ11_OR	000000F4	R	02	ACFSCOMBO_DEVICES	00000676	R	08
ACFSAL_DR11B_XB	00000148	R	02	ACFSCPI32	0000063B	R	08
ACFSAL_DR11C_OA	000000CC	R	02	ACFSCR11	000004E5	R	08
ACFSAL_DR11W_XA	00000144	R	02	ACFSC_LENGTH	= 00000028		
ACFSAL_DT11_OC	000000D8	R	02	ACFSDC11	000004E5	R	08
ACFSAL_DU11_XU	000000FC	R	02	ACFSDH11	000004E5	R	08
ACFSAL_DUP1T_XW	00000100	R	02	ACFSDHV11	000004E5	R	08
ACFSAL_DV11_XV	00000104	R	02	ACFSDJ11	000004E5	R	08

AUTOCONFIG  
Symbol table

ACFSDL11C	000004E5	R	08	ACFSL_RSSAVE	00000084	R	02
ACFSDM11B	000004E5	R	08	ACFSL_RETURN	00000036	R	02
ACFSDMC11	000004E5	R	08	ACFSL_RETURN2	0000003A	R	02
ACFSDMF32	0000061B	R	08	ACFSL_ROUTINE	0000003E	R	02
ACFSDMP11	000004E5	R	08	ACFSL_SYS_CSR	00000245	R	02
ACFSDMV11	000004E5	R	08	ACFSMBA	00000031	R	08
ACFSDMZ32	00000652	R	08	ACFSMBADISK	000000D3	R	08
ACFSDMZCP1	00000667	R	08	ACFSMBATAPE	0000010C	R	08
ACFSDN11	000004E5	R	08	ACFSM_SUPPORT	= 00000010		
ACFSDPV11	000004E5	R	08	ACFSPL11	000004E5	R	08
ACFSDQ11	000004E5	R	08	ACFSPP611	000004E5	R	08
ACFSDR	00000246	R	08	ACFSPR611	000004E5	R	08
ACFSDR11B	000004E5	R	08	ACFSQNA	000004E5	R	08
ACFSDR11C	000004E5	R	08	ACFSRB730	000004E5	R	08
ACFSDR11W	000004E5	R	08	ACFSRK611	000004E5	R	08
ACFSDT11	000004E5	R	08	ACFSRL11	000005C7	R	08
ACFSDU11	000004E5	R	08	ACFSRSV	0C0004E5	R	08
ACFS DUP11	000004E5	R	08	ACFSRX11	000004E5	R	08
ACFS DV11	000004E5	R	08	ACFSRX211	000004E5	R	08
ACFS DWR70	000004E5	R	08	ACFSSINGLE_DEVICES	00000252	R	08
ACFS DX11	000004E5	R	08	ACFSTC11	000004E5	R	08
ACFS DZ11	000004E5	R	08	ACFSTS11	000004E5	R	08
ACFSGL_CRB	*****	X	08	ACFSTU58	000004E5	R	08
ACFSGL_DDB	*****	X	08	ACFSTU81	00000542	R	08
ACFSGL_DPT	*****	X	08	ACFST_SYS_DEVNAME	0000024A	R	02
ACFSGL_IDB	*****	X	08	ACFSUBA	00000285	R	08
ACFSGL_LASTDDB	*****	X	08	ACFSUBAFIXED	00000358	R	08
ACFSGL_SB	*****	X	08	ACFSUBAFLOATING	000003DE	R	08
ACFSGL_UCB	*****	X	08	ACFSUDA	00000542	R	08
ACFSGT40	000004E5	R	08	ACFSUNA	000004E5	R	08
ACFSIEQ11	000004E5	R	08	ACFSVC01B	000004E5	R	08
ACFSINC_CHAR	0000074B	RG	08	ACFSVMV21	000004E5	R	08
ACFSISBT1	000004E5	R	08	ACFSVMV31	000004E5	R	08
ACFSKCT32	000004E5	R	08	ACFSVS100	000004E5	R	08
ACFSKMC11	000004E5	R	08	ACFSV_NOLOAD_DB	= 00000003		
ACFSKMS11	000004E5	R	08	ACFSW_AVECTOR	= 00000008		
ACFSKMV11	000004E5	R	08	ACFSW_CSRBASE	00000241	R	02
ACFSKW11C	000004E5	R	08	ACFSW_CUNIT	= 00000012		
ACFSKW11W	000004E5	R	08	ACFSW_CVECTOR	= 00000010		
ACFSLK11	000004E5	R	08	ACFSW_MAXUNITS	= 0000001C		
ACFSLP11	000004E5	R	08	ACFSW_VECBASE	00000243	R	02
ACFSLPA11	000004E5	R	08	ACFSW_VECMOD	0000006E	R	02
ACFSLPP11	000004E5	R	08	ADAPTERLEN	= 00000006		
ACFSLPS11	000004E5	R	08	ADPSB_NUMBER	= 0000000B		
ACFSL_ACF_SAVE	00000046	R	02	ADPSL_AVECTOR	= 0000001C		
ACFSL_ADAPTER	= 00000000			ADPSW_ADPTYPE	= 0000000E		
ACFSL_CONFIGREG	= 00000004			ADPSW_TR	= 0000000C		
ACFSL_CONTRLREG	= 0000000C			ATS_C1	= 00000004		
ACFSL_DELIVER_UNIT	00000042	R	02	ATS_DR	= 00000002		
ACFSL_DEVNAME	= 00000014			ATS_MBA	= 00000000		
ACFSL_DRVNAME	= 00000018			ATS_UBA	= 00000001		
ACFSL_ROSAVE	00000070	R	02	BOOSGQ_CONSYSID	*****	X	08
ACFSL_R1SAVE	00000074	R	02	CLSSL_CLASSDEV	= 00000000		
ACFSL_R2R3SAVE	00000088	R	02	CLSSL_CLASSDRV	= 00000C04		
ACFSL_R2SAVE	00000078	R	02	CLSST_PORTDEV	= 00000009		
ACFSL_R3SAVE	0000007C	R	02	CPISS_SUBCNTRL	= 00000004		
ACFSL_R4SAVE	00000080	R	02	CPIV_SUBCNTRL	= 00000008		

AUTOCONFIG  
Symbol table

CRBSL_INTD	=	00000024		
DDBST_NAME	=	00000014		
DMFSS_IDENT	=	00000004		
DMFSV_IDENT	=	0000000C		
DPTSW_DEFUNITS	=	0000001A		
DPTSW_DELIVER	=	0000001C		
EXESGC_RPB	*****		X	08
EXESGL_SCB	*****		X	08
EXESGL_TENUSEC	*****		X	08
EXESTEST_CSR	*****		X	08
FIX_DEV_NAME	0000047E		R	08
IDBSL_CSR	=	00000000		
IOCSAUTOCONFIG	00000000		RG	08
IOCSAUTORESET	00000705		RG	08
LOAD_DRIVER	000004C1		R	08
MBASC_ERB	=	00000400		
MBASL_SR	=	00000008		
MBASM_SR_ATTN	=	00010000		
MBASV_SR_NED	=	00000012		
MBA_DS	=	00000004		
MBA_DT	=	00000018		
PDTSL_UCBO	=	000000DC		
RL_CS	=	00000000		
RL_CS_M_OPI	=	00000400		
RL_DA	=	00000004		
RL_DA_M_MRK	=	00000001		
RL_DA_M_RST	=	00000008		
RL_DA_M_STS	=	00000002		
RPBSL_BOOTR1	=	00000020		
SYSSGC_BOOTUCB	*****		X	08
TM03_TC	=	00000024		
TM78_AB	=	00000010		
TM78_DS	=	0000001C		
TM78_ID	=	00000044		
TM78_M_TMCLR	=	00004000		
TM78_M_TMRDY	=	00008000		
TM78_NDT0	=	00000030		
TM78_NDTA	=	0000002C		
TM78_SENSE_GO	=	00000009		
UBA_TOBASE	=	00001000		
UBA_M_FLOATCSR	=	00000002		
UBA_M_FLOATVEC	=	00000004		
UBA_M_SUPPORT	=	00000001		
UBA_V_FLOATCSR	=	00000001		
UBA_V_FLOATVEC	=	00000002		
UBA_V_SUPPORT	=	00000000		
UBTSB_FLAGS	=	00000013	G	
UBTSB_LETTER	=	00000010	G	
UBTSB_NUMVEC	=	00000011	G	
UBTSB_UNUSED	=	00000012	G	
UBTSL_DEVNAME	=	00000000	G	
UBTSL_DRVNAME	=	00000004	G	
UBTSL_ROUTINE	=	0000000C	G	
UBTSL_RTNAME	=	00000008	G	
UBTSW_REMAINDER	=	00000014	G	
UCBSL_CRB	=	00000024		
UCBSL_DDB	=	00000028		

UCBSL_PDT	=	00000084
VECSL_ADP	=	00000014
VECSL_IDB	=	00000008

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NCJRT NOVEC BYTE
\$AB\$\$	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
NONPAGED_DATA	0000025A ( 602.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
ACF\$RESET	00000044 ( 68.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ACF\$DEVNAME	00000111 ( 273.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ACF\$DRVNAME	0000025C ( 604.)	05 ( 5.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ACF\$DEVDESC	000008FE ( 2302.)	06 ( 6.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
ACF\$RTNNAME	00000176 ( 374.)	07 ( 7.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
NONPAGED_CODE	000007B4 ( 1972.)	08 ( 8.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.09	00:00:00.95
Command processing	112	00:00:00.65	00:00:02.70
Pass 1	566	00:00:25.21	00:00:31.52
Symbol table sort	0	00:00:02.27	00:00:04.57
Pass 2	391	00:00:07.16	00:00:14.08
Symbol table output	46	00:00:00.35	00:00:00.44
Psect synopsis output	1	00:00:00.06	00:00:00.27
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1149	00:00:35.79	00:00:54.53

The working set limit was 2000 pages.  
163279 bytes (319 pages) of virtual memory were used to buffer the intermediate code.  
There were 80 pages of symbol table space allocated to hold 1463 non-local and 70 local symbols.  
1920 source lines were read in Pass 1, producing 54 object records in Pass 2.  
30 pages of virtual memory were used to define 26 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	12
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	16

1210 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:AUTOCONFG/OBJ=OBJ\$:AUTOCONFG MSRC\$:AUTOCONFG/UPDATE=(ENH\$:AUTOCONFG)+EXECML\$/LIB+LIB\$:BOOTS.MLB/LIB

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

BOOTS  
MAR

AUTOCONFIG  
LIS

WRITEBOOT  
MAP

BOOTTAB  
LIS

UMB  
MAP

SYSGEN  
MAP

ACTIMAGE  
LIS

BOOT58  
LIS

BOOTDEF  
SOL