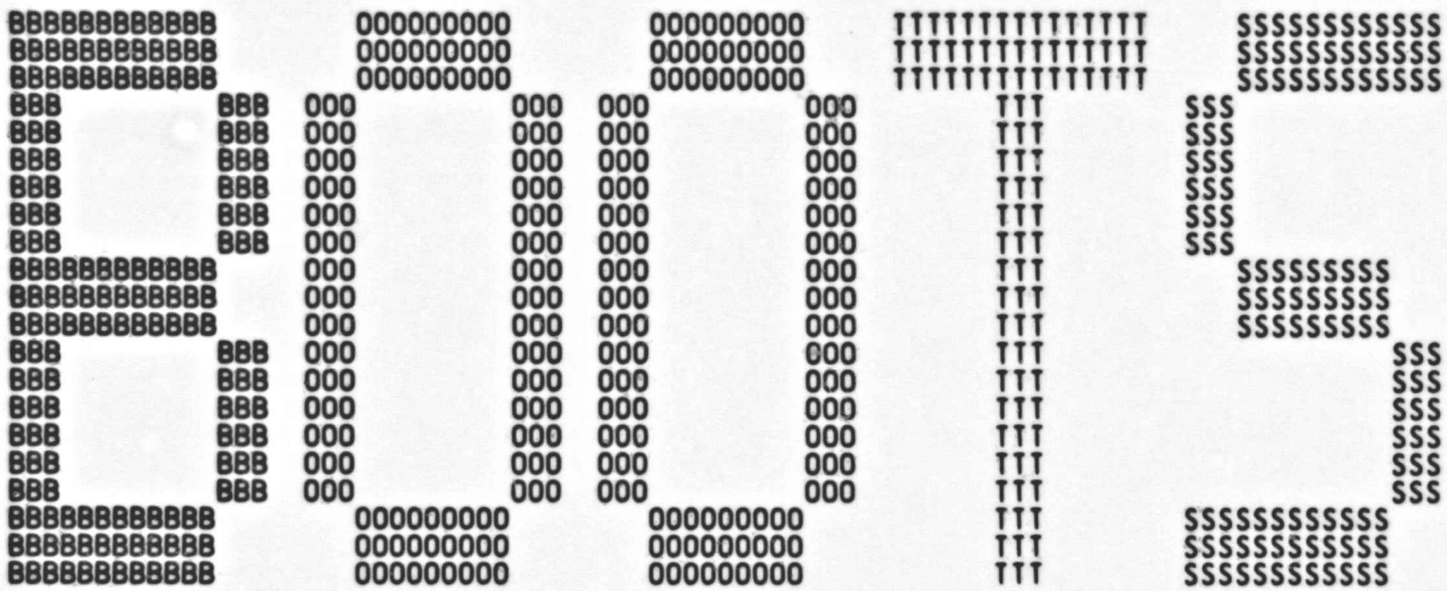
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTTTTTTTTTTTTTT SSSSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOO OOO OOO OOO TTT SSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTT SSSSSSSSSSS

```
   AAAAAA    UU      UU  TTTTTTTTTT    OOOOOO      CCCCCCCC    OOOOOO    NN      NN  FFFFFFFFFF    GGGGGGGG
   AAAAAA    UU      UU  TTTTTTTTTT    OOOOOO      CCCCCCCC    OOOOOO    NN      NN  FFFFFFFFFF    GGGGGGGG
 AA      AA  UU      UU      TT      OO      OO  CC          OO      OO  NN      NN  FF            GG
 AA      AA  UU      UU      TT      OO      OO  CC          OO      OO  NN      NN  FF            GG
 AA      AA  UU      UU      TT      OO      OO  CC          OO      OO  NNNN    NN  FF            GG
 AA      AA  UU      UU      TT      OO      OO  CC          OO      OO  NNNN    NN  FF            GG
 AA      AA  UU      UU      TT      OO      OO  CC          OO      OO  NN  NN  NN  FFFFFFFF      GG
 AA      AA  UU      UU      TT      OO      OO  CC          OO      OO  NN  NN  NN  FFFFFFFF      GG
 AAAAAAAAAA  UU      UU      TT      OO      OO  CC          OO      OO  NN    NNNN  FF            GG  GGGGGG
 AAAAAAAAAA  UU      UU      TT      OO      OO  CC          OO      OO  NN    NNNN  FF            GG  GGGGGG
 AA      AA  UU      UU      TT      OO      OO  CC          OO      OO  NN      NN  FF            GG      GG    ....
 AA      AA  UU      UU      TT      OO      OO  CC          OO      OO  NN      NN  FF            GG      GG    ....
 AA      AA  UUUUUUUUUU      TT        OOOOOO      CCCCCCCC    OOOOOO    NN      NN  FF              GGGGGG      ....
 AA      AA  UUUUUUUUUU      TT        OOOOOO      CCCCCCCC    OOOOOO    NN      NN  FF              GGGGGG      ....

 LL              IIIIII      SSSSSSSS
 LL              IIIIII      SSSSSSSS
 LL                II      SS
 LL                II      SS
 LL                II      SS
 LL                II      SS
 LL                II        SSSSSS
 LL                II        SSSSSS
 LL                II              SS
 LL                II              SS
 LL                II              SS
 LL                II              SS
 LLLLLLLLLL      IIIIII    SSSSSSSS
 LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
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 QR.
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$B_COMBO_VECTOR_OFFSET.
0000     66 ;               Cleans up support of combo style devices.
0000     67 ;
0000     68 ;       V03-011 WHM0004         Bill Matthews           03-Feb-1984
0000     69 ;               Changed the device name of the DMV11 from OO to XD.
0000     70 ;               Added support for multiple devices with the same name.
0000     71 ;
0000     72 ;       V03-010 WHM0003         Bill Matthews           02-Feb-1984
0000     73 ;               checked in v03-009 source by mistake.
0000     74 ;
0000     75 ;       V03-009 WHM0002         Bill Matthews           29-Dec-1983
0000     76 ;               Added autoconfigure support for TC11.
0000     77 ;
0000     78 ;       V03-008 WHM0001         Bill Matthews           13-Dec-1983
0000     79 ;               Rewrote support for autoconfiguring combo devices.
0000     80 ;
0000     81 ;       V03-007 JLV0313         Jake VanNoy             18-Nov-1983
0000     82 ;               Put in support for first QNA. Also add DHV/DHU.
0000     83 ;               Remove NOSUPPORT flag from UNA.
0000     84 ;
0000     85 ;       V03-006 MSH0003         Maryann Hinden          09-Feb-1983
0000     86 ;               Add support for cluster device names.  Change TU to MU.
0000     87 ;
0000     88 ;       V03-005 MSH0002         Maryann Hinden          28-Dec-1982
0000     89 ;               Calculate floating vector correctly for nth (n>1) device.
0000     90 ;
0000     91 ;       V03-004 TCM0002         Trudy C. Matthews       22-Oct-1982
0000     92 ;               Use new field in ADP, ADP$L_AVECTOR, to calculate ACF$W_AVECTOR,
0000     93 ;               instead of using adapter's TR number (ADP$W_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
```

```
                                                I 8
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         $DEVDESC$=.
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   $DEVDESC$
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
```

```
                                        J 8
0000    229          $'DRVNAME'$=.
0000    230                  .ASCIC  \DRVNAME\
0000    231                  .ENDC
0000    232
0000    233                  .PSECT  ACF$DEVDESC
0000    234          $DEVDESC$=.
0000    235                  .LONG   $'DEVNAME'$
0000    236                  .LONG   $'DRVNAME'$
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 ;       stucture:                                       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 ;       ! vector mask   ! 20
0000    301 ;       +---------------+
0000    302 ;       ! first CSR     !
0000    303 ;       +---------------+
0000    304 ;       ! 2nd CSR       !
0000    305 ;       +---------------+
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         .ASCIC  \'DEVNAME'A\
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         .ASCIC  \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          .ASCIC  \'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          .ASCIC  \DRVNAME\
0000    461
0000    462          .PSECT  ACF$DEVDESC
0000    463 $DEVDESC$ = .
0000    464          .LONG   $DEVNAME$
0000    465          .LONG   $DRVNAME$
0000    466          .ASCIC  \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$L_DEVNAME   ==      0
00000004             0000   484 UBT$L_DRVNAME   ==      4
00000008             0000   485 UBT$L_RTNNAME   ==      8
0000000C             0000   486 UBT$L_ROUTINE   ==      12
00000010             0000   487 UBT$B_LETTER    ==      16
00000011             0000   488 UBT$B_NUMVEC    ==      17
00000012             0000   489 UBT$B_UNUSED    ==      18
00000013             0000   490 UBT$B_FLAGS     ==      19
00000014             0000   491 UBT$W_REMAINDER ==      20
                     0000   492
                     0000   493 ;
                     0000   494 ; LOCAL SYMBOLS
                     0000   495 ;
                     0000   496
                     0000   497 ;
                     0000   498 ; CLASSDEV OFFSETS
                     0000   499 ;
                     0000   500
00000000             0000   501 CLS$L_CLASSDEV = 0
00000004             0000   502 CLS$L_CLASSDRV = 4
00000009             0000   503 CLS$T_PORTDEV  = 9
                     0000   504
                     0000   505 ;
                     0000   506 ; DMF32 OFFSETS
                     0000   507 ;
                     0000   508
0000000C             0000   509 DMF$V_IDENT = 12
00000004             0000   510 DMF$S_IDENT =  4
                     0000   511
                     0000   512 ;
                     0000   513 ; CPI32/DMZ32 OFFSETS
                     0000   514 ;
00000008             0000   515 CPI$V_SUBCNTRL = 8
00000004             0000   516 CPI$S_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_NDTO=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$MBA
                     0006   575          ADAPTER UBA                        ;UNIBUS ADAPTER
                0000 0006                .WORD   0
                00000006 0008            .=.-2
                0001 0006                .WORD   AT$_UBA
                00000285' 0008           .LONG   ACF$UBA
                     000C   576          ADAPTER DR                         ; DR32 ADAPTER
                0000 000C                .WORD   0
                0000000C 000E            .=.-2
                0002 000C                .WORD   AT$_DR
                00000246' 000E           .LONG   ACF$DR
                     0012   577          ADAPTER CI                         ; CI ADAPTER
                0000 0012                .WORD   0
                00000012 0014            .=.-2
                0004 0012                .WORD   AT$_CI
                0000024D' 0014           .LONG   ACF$CI
                     0018   578          ADAPTER NULL                       ;
                0000 0018                .WORD   0
                     001A   579
                     001A   580 ;
                     001A   581 ; CONTROLLER DESIGNATOR RESET POINTER TABLE
                     001A   582 ;
                     001A   583
                00000000    584          .PSECT  ACF$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>> ;RP0X
                00000000                 .PSECT  ACF$DEVNAME
          41 42 44 00' 0000              .ASCIC  \DBA\
                0003 0000
                00000000                 .PSECT  ACF$RESET
             00000004' 0000              .ADDRESS $$$
                00000000                 .PSECT  ACF$DRVNAME
    52 45 56 49 52 44 42 44 00' 0000     .ASCIC  \DBDRIVER\
                0008 0000
                00000000                 .PSECT  ACF$DEVDESC
             00000000' 0000              .LONG   $DB$
             00000000' 0004              .LONG   $DBDRIVER$
                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>>  ;RM0X
                          00000004                        .PSECT  ACF$DEVNAME
                 41 52 44 00' 0004                        .ASCIC  \DRA\
                          03  0004
                          00000004                        .PSECT  ACF$RESET
                    00000008' 0004                        .ADDRESS $$$
                          00000009                        .PSECT  ACF$DRVNAME
       52 45 56 49 52 44 52 44 00' 0009                   .ASCIC  \DRDRIVER\
                          08  0009
                          00000010                        .PSECT  ACF$DEVDESC
                    00000004' 0010                        .LONG   $DR$
                    00000009' 0014                        .LONG   $DRDRIVER$
                        0014  0018                        .WORD   ^X14
                        0016  001A                        .WORD   ^X16
                        0017  001C                        .WORD   ^X17
                        0022  001E                        .WORD   ^X22
                        0000  0020                        .WORD   0
                          0000001E                        .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
                    00000010' 001E                        .LONG   $DEVDESC$
                              0022  595         MBADEV    MT,TMDRIVER,<<^XC028>>  ; TM03 CONTOLLER SPECIFIC DATA
                          00000008                        .PSECT  ACF$DEVNAME
                 41 54 4D 00' 0008                        .ASCIC  \MTA\
                          03  0008
                          00000008                        .PSECT  ACF$RESET
                    0000000C' 0008                        .ADDRESS $$$
                          00000012                        .PSECT  ACF$DRVNAME
       52 45 56 49 52 44 4D 54 00' 0012                   .ASCIC  \TMDRIVER\
                          08  0012
                          00000022                        .PSECT  ACF$DEVDESC
                    00000008' 0022                        .LONG   $MT$
                    00000012' 0026                        .LONG   $TMDRIVER$
                        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$DEVNAME
                 41 46 4D 00' 000C                        .ASCIC  \MFA\
                          03  000C
                          0000000C                        .PSECT  ACF$RESET
                    00000010' 000C                        .ADDRESS $$$
                          0000001B                        .PSECT  ACF$DRVNAME
       52 45 56 49 52 44 46 54 00' 001B                   .ASCIC  \TFDRIVER\
                          08  001B
                          0000002E                        .PSECT  ACF$DEVDESC
                    0000000C' 002E                        .LONG   $MF$
                    0000001B' 0032                        .LONG   $TFDRIVER$
                        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$AB_DRTABLE:
                        002E    605                 DRDEV   XF,XFDRIVER
                    00000010                        .PSECT  ACF$DEVNAME
          41 46 58 00'  0010                        .ASCIC  \XFA\
                        03  0010
                    00000010                        .PSECT  ACF$RESET
                00000014'  0010                     .ADDRESS $$$
                    00000024                        .PSECT  ACF$DRVNAME
52 45 56 49 52 44 46 58 00'  0024                   .ASCIC  \XFDRIVER\
                        08  0024
                    0000003A                        .PSECT  ACF$DEVDESC
                00000010'  003A                     .LONG   $XF$
                00000024'  003E                     .LONG   $XFDRIVER$
                    0000002E                        .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
                0000003A'  002E                     .LONG   $DEVDESC$
                        0032    606
                        0032    607 ;
                        0032    608 ; CI ADAPTER CONFIGURATION CONTROL TABLE (uses DRDEV macro)
                        0032    609 ;
                        0032    610
                        0032    611 ACF$AB_CITABLE:
                        0032    612                 DRDEV   PA,PADRIVER
                    00000014                        .PSECT  ACF$DEVNAME
          41 41 50 00'  0014                        .ASCIC  \PAA\
                        03  0014
                    00000014                        .PSECT  ACF$RESET
                00000018'  0014                     .ADDRESS $$$
                    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   $PA$
                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
                        0036    618 ACF$L_RETURN:                                   ;
                0000003A  0036  619                 .BLKL   1                       ;
                        003A    620 ACF$L_RETURN2:                                  ; Alternate
                0000003E  003A  621                 .BLKL   1                       ;
                        003E    622 ACF$L_ROUTINE:                                  ;
                00000042  003E  623                 .BLKL   1                       ;
                        0042    624 ACF$L_DELIVER_UNIT:                             ;
                00000046  0042  625                 .BLKL   1                       ;
                        0046    626 ACF$L_ACF_SAVE:                                 ; Save area for ACF block
                0000006E  0046  627                 .BLKB   ACF$C_LENGTH            ;
                        006E    628 ACF$W_VECMOD:                                   ; Save for floating vector modulus
```

```
00000070  006E   629          .BLKW   1
          0070   630
          0070   631 ;
          0070   632 ; REGISTER SAVE AREA
          0070   633 ;
          0070   634
          0070   635 ACF$L_R0SAVE:                                        ;
00000074  0070   636          .BLKL   1                                    ;
          0074   637 ACF$L_R1SAVE:                                        ;
00000078  0074   638          .BLKL   1                                    ;
          0078   639 ACF$L_R2SAVE:                                        ;
0000007C  0078   640          .BLKL   1                                    ;
          007C   641 ACF$L_R3SAVE:                                        ;
00000080  007C   642          .BLKL   1                                    ;
          0080   643 ACF$L_R4SAVE:                                        ;
00000084  0080   644          .BLKL   1                                    ;
          0084   645 ACF$L_R5SAVE:                                        ;
00000088  0084   646          .BLKL   1                                    ;
          0088   647 ACF$L_R2R3SAVE:
00000090  0088   648          .BLKL   2
          0090   649
          0090   650 ;
          0090   651 ; UNIBUS ADAPTER CONFIGURATION CONTROL TABLE
          0090   652 ;
          0090   653 ;       NOTE: ALL FLOATING ADDRESS ENTRIES MUST BE IN PRIORITY ORDER.
          0090   654 ;
          0090   655
          0090   656 .NLIST MEB      ; Cut down listing size
          0090   657
          0090   658 ACF$AB_UBATABLE::                                    ;
          0090   659          UBADEV  CR,CRDRIVER,CR11,1,FIXED,FIXED,<<^O17160,^O230>> ;
          0094   660          UBADEV  DM,DMDRIVER,RK611,1,FIXED,FIXED,<<^O17440,^O210>> ;
          0098   661          UBADEV  LP,LPDRIVER,LP11,1,FIXED,FIXED,<-
          0098   662                  <^O17514,^O200>,-
          0098   663                  <^O04004,^O170>,-
          0098   664                  <^O04014,^O174>,-
          0098   665                  <^O04024,^O270>,-
          0098   666                  <^O04034,^O274>>
          009C   667          UBADEV  DL,DLDRIVER,RL11,1,FIXED,FIXED,<<^O14400,^O160>> ;
          00A0   668          UBADEV  MS,TSDRIVER,TS11,1,FIXED,FIXED,<<^O12520,^O224>> ;
          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,<<^O17170,^O264>> ;
          00A8   674          UBADEV  DQ,DQDRIVER,RB730,1,FIXED,FIXED,<<^O15606,^O250>> ;
          00AC   675          UBADEV  PU,PUDRIVER,UDA  ,1,FIXED,FIXED,<<^O12150,^O154>>
          00B0   676          UBADEV  PT,PUDRIVER,TU81 ,1,FIXED,FIXED,<<^O14500,^O260>>
          00B4   677          UBADEV  XE,XEDRIVER,UNA  ,1,FIXED,FIXED,<<^O14510,^O120>>
          00B8   678          UBADEV  XQ,XQDRIVER,QNA  ,1,FIXED,FIXED,<<^O14440,^O120>>
          00BC   679          UBADEV  OM,OMDRIVER,DC11,2,FIXED,FLOAT,<-
          00BC   680                  <^O14000>,<^O14010>,<^O14020>,<^O14030>,-
          00BC   681                  <^O14040>,<^O14050>,<^O14060>,<^O14070>,-
          00BC   682                  <^O14100>,<^O14110>,<^O14120>,<^O14130>,-
          00BC   683                  <^O14140>,<^O14150>,<^O14160>,<^O14170>,-
          00BC   684                  <^O14200>,<^O14210>,<^O14220>,<^O14230>,-
          00BC   685                  <^O14240>,<^O14250>,<^O14260>,<^O14270>,-
```

```
                00BC    686                     <^O14300>,<^O14310>,<^O14320>,<^O14330>,-
                00BC    687                     <^O14340>,<^O14350>,<^O14360>,<^O14370>>,8,NOSUPPORT
                00C0    688             UBADEV  DD,DDDRIVER,TU58,2,FIXED,FLOAT,<-
                00C0    689                     <^O16500>,<^O16510>,<^O16520>,<^O16530>,-
                00C0    690                     <^O16540>,<^O16550>,<^O16560>,<^O16570>,-
                00C0    691                     <^O16600>,<^O16610>,<^O16620>,<^O16630>,-
                00C0    692                     <^O16640>,<^O16650>,<^O16660>,<^O16670>>,8
                00C4    693             UBADEV  OB,OBDRIVER,DN11,1,FIXED,FLOAT,<-
                00C4    694                     <^O15200>,<^O15210>,<^O15220>,<^O15230>,-
                00C4    695                     <^O15240>,<^O15250>,<^O15260>,<^O15270>,-
                00C4    696                     <^O15300>,<^O15310>,<^O15320>,<^O15330>,-
                00C4    697                     <^O15340>,<^O15350>,<^O15360>,<^O15370>>,4,NOSUPPORT
                00C8    698             UBADEV  YM,YMDRIVER,DM11B,1,FIXED,FLOAT,<-
                00C8    699                     <^O10500>,<^O10510>,<^O10520>,<^O10530>,-
                00C8    700                     <^O10540>,<^O10550>,<^O10560>,<^O10570>,-
                00C8    701                     <^O10600>,<^O10610>,<^O10620>,<^O10630>,-
                00C8    702                     <^O10640>,<^O10650>,<^O10660>,<^O10670>>,4,NOSUPPORT
                00CC    703             UBADEV  OA,OADRIVER,DR11C,2,FIXED,FLOAT,<-
                00CC    704                     <^O7600>,<^O7570>,<^O7560>,<^O7550>,-
                00CC    705                     <^O7540>,<^O7530>,<^O7520>,<^O7510>,-
                00CC    706                     <^O7500>,<^O7470>,<^O7460>,<^O7450>,-
                00CC    707                     <^O7440>,<^O7430>,<^O7420>,<^O7410>>,8,NOSUPPORT
                00D0    708             UBADEV  PR,PRDRIVER,PR611,1,FIXED,FLOAT,<-
                00D0    709                     <^O12600>,<^O12604>,<^O12610>,<^O12614>,-
                00D0    710                     <^O12620>,<^O12624>,<^O12630>,<^O12634>>,8,NOSUPPORT
                00D4    711             UBADEV  PP,PPDRIVER,PP611,1,FIXED,FLOAT,<-
                00D4    712                     <^O12700>,<^O12704>,<^O12710>,<^O12714>,-
                00D4    713                     <^O12720>,<^O12724>,<^O12730>,<^O12734>>,8,NOSUPPORT
                00D8    714             UBADEV  OC,OCDRIVER,DT11,2,FIXED,FLOAT,<-
                00D8    715                     <^O17420>,<^O17422>,<^O17424>,<^O17426>,-
                00D8    716                     <^O17430>,<^O17432>,<^O17434>,<^O17436>>,8,NOSUPPORT
                00DC    717             UBADEV  OD,ODDRIVER,DX11,2,FIXED,FLOAT,<-
                00DC    718                     <^O16200>,<^O16240>>,8,NOSUPPORT
                00E0    719             UBADEV  YL,YLDRIVER,DL11C,2,FIXED,FLOAT,<-
                00E0    720                     <^O15610>,<^O15620>,<^O15630>,<^O15640>,-
                00E0    721                     <^O15650>,<^O15660>,<^O15670>,<^O15700>,-
                00E0    722                     <^O15710>,<^O15720>,<^O15730>,<^O15740>,-
                00E0    723                     <^O15750>,<^O15760>,<^O15770>,<^O16000>,-
                00E0    724                     <^O16010>,<^O16020>,<^O16030>,<^O16040>,-
                00E0    725                     <^O16050>,<^O16060>,<^O16070>,<^O16100>,-
                00E0    726                     <^O16110>,<^O16120>,<^O16130>,<^O16140>,-
                00E0    727                     <^O16150>,<^O16160>,<^O16170>>,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,<<^O12000>,<^O12010>>,8,NOSUPPORT
                00F0    731             UBADEV  LS,LSDRIVER,LPS11,6,FIXED,FLOAT,<<^O10400>>,8,NOSUPPORT
                00F4    732             UBADEV  OR,ORDRIVER,DQ11,2,FLOAT,FLOAT,8,8,NOSUPPORT
                00F8    733             UBADEV  OF,OFDRIVER,KW11W,2,FIXED,FLOAT,<<^O12400>>,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                     <^O15000>,<^O15040>,<^O15100>,<^O15140>>,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,TSDRIVER,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,<<^012430>>,4,NOSUPPORT
                0150    773          UBADEV  XB,XBDRIVER,DR11B,1,FLOAT,FLOAT,8,4,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,DMZ32,6,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
                01A4    798
000001F4        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 $$$
                    00000226               .PSECT  ACF$DRVNAME
52 45 56 49 52 44 47 58 00' 0226           .ASCIC  \XGDRIVER\
                        08   0226
                    00000834               .PSECT  ACF$DEVDESC
                    0004 0834              .WORD   4
                000000FD' 0836             .LONG   $XG$
                00000226' 083A             .LONG   $XGDRIVER$
                        02   083E          .BYTE   2
                    0000 083F              .WORD   0
                00000004 0841              .LONG   4
                        01   0845          .BYTE   $$$
                    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   $TX$
                0000022F' 084C             .LONG   $YCDRIVER$
                        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   $XI$
                00000238' 085E             .LONG   $XIDRIVER$
                        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
                00000105                            .PSECT  ACF$DEVNAME
             41 43 4C 00' 0105                      .ASCIC  \LCA\
                          03  0105
                00000020                            .PSECT  ACF$RESET
                00000109' 0020                      .ADDRESS $$$
                00000241                            .PSECT  ACF$DRVNAME
    52 45 56 49 52 44 43 4C 00' 0241                .ASCIC  \LCDRIVER\
                          08  0241
                0000086A                            .PSECT  ACF$DEVDESC
                    0002  086A                      .WORD   2
                00000105' 086C                      .LONG   $LC$
                00000241' 0870                      .LONG   $LCDRIVER$
                      01  0874                      .BYTE   1
                    0018  0875                      .WORD   24
                00000014  0877                      .LONG   20
                      01  087B                      .BYTE   $$$
                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   $YCDRIVER$
                      02  0886                      .BYTE   2
                    0000  0887                      .WORD   0
                00000004  0889                      .LONG   4
                      01  088D                      .BYTE   $$$
                00000208                            .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
                0000087C' 0208                      .LONG   $DEVDESC$
                          020C  820                 COMBODEV TX, YCDRIVER,    2,  8, 12, 2          ; Async
                0000088E                            .PSECT  ACF$DEVDESC
                    0002  088E                      .WORD   2
                000000F1' 0890                      .LONG   $TX$
                0000022F' 0894                      .LONG   $YCDRIVER$
                      02  0898                      .BYTE   2
                    0008  0899                      .WORD   8
                0000000C  089B                      .LONG   12
                      01  089F                      .BYTE   $$$
                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   $YCDRIVER$
                      02  08AA                      .BYTE   2
```

```
                                                M 9
                     0010  08AB                       .WORD   16
00000014             0012  08AD                       .LONG   20
                       01  08B1                       .BYTE   $$$
                     0000  08B2                       .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
000008A0'            0004  08B2                       .LONG   $DEVDESC$
                     0008  08B6  822
00000000             0008  08B6  823                  .LONG   0                       ; END OF LIST
                     000C  08BA  824
                     000C  08BA  825
                     000C  08BA  826 ACF$AB_CPI32_TABLE:
                     000C  08BA  827
                     000C  08BA  828 ;       COMBODEV DEVNAME,DRVNAME,NUMVECT,VEC_OFFSET,CSR_OFFSET,
                     000C  08BA  829 ;                MASK,SUPPORT=SUPPORTED
                     000C  08BA  830
                     000C  08BA  831         COMBODEV XG, XGDRIVER,    2,  0,  4, 1          ; Sync
                     0000  08BA                 .PSECT  ACF$DEVDESC
                     0000  08BA                 .WORD   1
000000FD'            0002  08BC                 .LONG   $XG$
00000226'            0006  08C0                 .LONG   $XGDRIVER$
                       02  08C4                 .BYTE   2
                     0000  08C5                 .WORD   0
00000004             0002  08C7                 .LONG   4
                       01  08CB                 .BYTE   $$$
                     0000  08CC                 .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
000008B2'            0004  08CC                 .LONG   $DEVDESC$
                     0008  08D0  832         COMBODEV XG, XGDRIVER,    2,  8, 12, 2          ; Sync
                     0000  08D0                 .PSECT  ACF$DEVDESC
                     0000  08D0                 .WORD   2
000000FD'            0002  08D2                 .LONG   $XG$
00000226'            0006  08D6                 .LONG   $XGDRIVER$
                       02  08DA                 .BYTE   2
                     0008  08DB                 .WORD   8
0000000C             000A  08DD                 .LONG   12
                       01  08E1                 .BYTE   $$$
                     0000  08E2                 .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
000008C4'            0004  08E2                 .LONG   $DEVDESC$
                     0008  08E6  833         COMBODEV XG, XGDRIVER,    2, 16, 20, 4          ; Sync
                     0000  08E6                 .PSECT  ACF$DEVDESC
                     0000  08E6                 .WORD   4
000000FD'            0002  08E8                 .LONG   $XG$
00000226'            0006  08EC                 .LONG   $XGDRIVER$
                       02  08F0                 .BYTE   2
                     0010  08F1                 .WORD   16
00000014             0012  08F3                 .LONG   20
                       01  08F7                 .BYTE   $$$
                     0000  08F8                 .PSECT  NONPAGED_DATA   rd,wrt,noexe,quad
000008D6'            0004  08F8                 .LONG   $DEVDESC$
                     0008  08FC  834
00000000             0008  08FC  835                  .LONG   0                       ; END OF LIST
                     000C  0900  836
                     000C  0900  837
                     000C  0900  838 ACF$AB_CLASS_TABLE:
                     000C  0900  839
                     000C  0900  840 ;       CLASSDEV DEVNAME,DRVNAME,PORTDEV
                     000C  0900  841
                     000C  0900  842         CLASSDEV DU,DUDRIVER,PU
                     0000  0900                 .PSECT  ACF$DEVNAME
```

```
                                                     N 9
                     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                      ;
```

```
                     0243   871 ACF$W_VECBASE:                              ;
                     0243   872         .BLKW   1                           ;
                     0245   873 ACF$L_SYS_CSR:                             ; SYSTEM DEVICE CSR ADDRESS
                     0245   874         .BLKL   1
                     0249   875 ACF$B_BOOT_TR:                             ; SYSTEM DEVICE ON THIS ADAPTER
                     0249   876         .BLKB   1
                     024A   877 ACF$T_SYS_DEVNAME:                         ; SYSTEM DEVICE NAME
                     024A   878         .BLKB   16
                     025A   879
                     025A   880         .NLIST  MEB
```

```
                                    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          CLRB    ACF$B_BOOT_TR  ;INITIALIZE LOCAL FLAG BYTE
50    0000'CF                9E     0006   902          MOVAB   W^ACF$AB_ADPTYPE,R0    ;GET ADDRESS OF ADAPTER TABLE
  0E A8                     B1     000B   903 10$:     CMPW    (R0)+,ADP$W_ADPTYPE(R8) ;ADAPTER TYPE MATCH?
                            13     000F   904          BEQL    20$            ;IF EQL YES
       50                   C0     0011   905          ADDL    #ADAPTERLEN-2,R0       ;ADVANCE TO NEXT ENTRY
                            B5     0014   906          TSTW    (R0)           ;ANY MORE ENTRIES IN TABLE?
                            12     0016   907          BNEQ    10$            ;IF NEQ YES
                            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
                   67               58   D0   001B   915 20$:     MOVL    R8,ACF$L_ADAPTER(R7)    ;SET ADDRESS OF ADAPTER CONTROL BLOCK
                04 A7               56   D0   001E   916          MOVL    R6,ACF$L_CONFIGREG(R7)  ;SET ADDRESS OF CONFIGURATION STATUS REGISTE
51    1C A8    00000000'GF          C3   0022   917          SUBL3   G^EXE$GL_SCB, -         ;CALCULATE OFFSET INTO SCB OF
                                   002B   918                  ADP$L_AVECTOR(R8),R1    ;ADAPTER'S INTERRUPT VECTORS.
                08 A7               51   B0   002B   919          MOVW    R1,ACF$W_AVECTOR(R7)    ;STORE IN ACF.
                                    90   17   002F   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$MBA:                                    ;AUTO CONFIGURE MASSBUS ADAPTER
      1E A7    01    90         0031   943          MOVB    #1,ACF$B_CNUMVEC(R7)     ;SET NUMBER OF CONTROLLER VECTORS
10 A7    08 A7    B0         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         0042   947          BNEQ    20$                      ;IF NEQ ANOTHER ENTRY TO PROCESS
             50   D4         0044   948          CLRL    R0                       ;SET END OF SCAN INDICATOR
                  05         0046   949          RSB                              ;
                             0047   950
                             0047   951 ;
                             0047   952 ; SCAN MBA UNITS FOR DEVICE TYPE MATCH
                             0047   953 ;
                             0047   954
           0745    30        0047   955 20$:     BSBW    ACF$CLR_ACF              ;CLEAR DEVICE BLOCKS
      14 A7    84   D0         004A   956          MOVL    (R4)+,ACF$L_DEVNAME(R7) ;SET ADDRESS OF DEVICE NAME STRING
      18 A7    84   D0         004E   957          MOVL    (R4)+,ACF$L_DRVNAME(R7) ;SET ADDRESS OF DRIVER NAME STRING
               53   D4         0052   958          CLRL    R3                       ;CLEAR DRIVE OFFSET CONSTANT
          52   64   9E         0054   959 30$:     MOVAB   (R4),R2                  ;GET STARTING ADDRESS OF DRIVE TYPE TABLE
    51  0418 C643   D0         0057   960          MOVL    MBA$L_ERB+MBA_DT(R6)[R3],R1 ;READ DRIVE TYPE REGISTER
      51  3E00 8F   AA         005D   961          BICW    #^C<^X1FF>,R1            ;CLEAR EXTRANEOUS BITS
        50   08 A6   D0         0062   962          MOVL    MBA$L_SR(R6),R0          ;READ MBA STATUS REGISTER
        08 A6   50   D0         0066   963          MOVL    R0,MBA$L_SR(R6)          ;CLEAR MBA STATUS REGISTER
        15 50   12   E0         006A   964          BBS     #MBA$V_SR_NED,R0,50$     ;IF SET, NONEXISTENT DRIVE
        05 51   0F   E1         006E   965          BBC     #15,R1,40$               ;IF CLR, NOT TAPE DEVICE
      51  3E07 8F   AA         0072   966          BICW    #^C<^X1F8>,R1            ; FOR TAPE, RETAIN CONTROLLER
                             0077   967                                           ; SPECIFIC INFORMATION ONLY.
          82   51   B1         0077   968 40$:     CMPW    R1,(R2)+                 ;DRIVE TYPE MATCH?
               53   13         007A   969          BEQL    60$                      ;IF EQL YES
               62   B5         007C   970          TSTW    (R2)                     ;ANY MORE ENTRIES IN TABLE?
               F7   12         007E   971          BNEQ    40$                      ;IF NEQ YES
             0041   31         0080   972          BRW     56$                      ;GO LOOK FOR MORE UNITS
                             0083   973
50  00000000'GF    000186A0 8F   C5   0083   974 50$:     MULL3   #100000,G^EXE$GL_TENUSEC,R0 ;INIT LOOP COUNT;
            0404 C643   D4         008F   975          CLRL    MBA$L_ERB+MBA_DS(R6)[R3]; ATTEMPT TO SEIZE THE PORT
      51  0418 C643   D0         0094   976 52$:     MOVL    MBA$L_ERB+MBA_DT(R6)[R3],R1 ; READ DRIVE TYPE REGISTER
               15   13         009A   977          BEQL    54$                      ; IF EQL, THEN NO PORT
    0400 C643   0B   9A         009C   978          MOVZBL  #^X0B,MBA$L_ERB(R6)[R3] ; RELEASE THE PORT
```

```
                50    08 A6    D0  00A2   979          MOVL    MBA$L_SR(R6),R0          ;READ MBA STATUS REGISTER
                08 A6    50    D0  00A6   980          MOVL    R0,MBA$L_SR(R6)          ;CLEAR MBA STATUS REGISTER
              51    3E00 8F    AA  00AA   981          BICW    #^C<^XC1FF>,R1           ; CLEAR EXTRANEOUS BITS
                       C6    11  00AF   982          BRB     40$                      ; GO LOOK FOR DRIVER TYPE MATCH
                             00B1   983 54$:     FREEIB
                51    08 A6    D0  00B9   984          MOVL    MBA$L_SR(R6),R1          ;READ MBA STATUS REGISTER
                08 A6    51    D0  00BD   985          MOVL    R1,MBA$L_SR(R6)          ;CLEAR MBA STATUS REGISTER
                   D0    50    F5  00C1   986          SOBGTR  R0,52$                   ;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$L_DEVNAME(R7),R1     ;GET ADDRESS OF DEVICE NAME STRING
                   52    61    9A  00DB  1004          MOVZBL  (R1),R2                  ;GET LENGTH OF STRING IN BYTES
          6142  41 8F    0B A8    81  00DE  1005          ADDB3   ADP$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$W_CUNIT(R7)       ;SET CONTROLLER UNIT NUMBER
          007C'CF    53    7D  00F1  1014          MOVQ    R3,W^ACF$L_R3SAVE        ;SAVE REGISTERS
          0084'CF    55    D0  00F6  1015          MOVL    R5,W^ACF$L_R5SAVE        ;
                50    01    D0  00FB  1016          MOVL    #1,R0                    ;SET SUCCESS INDICATOR
                      9E    16  00FE  1017          JSB     @(SP)+                   ;DELIVER UNIT TO CALLER
          53    007C'CF    7D  0100  1018          MOVQ    W^ACF$L_R3SAVE,R3        ;RESTORE REGISTERS
          55    0084'CF    D0  0105  1019          MOVL    W^ACF$L_R5SAVE,R5        ;
                      B8    11  010A  1020          BRB     56$                      ;BRANCH TO TEST FOR MORE DRIVES
                             010C  1021
                             010C  1022 ;
                             010C  1023 ; DEVICE IS A TAPE
                             010C  1024 ;
                             010C  1025
                             010C  1026 ACF$MBATAPE:                              ;
          51    C040 8F    B1  010C  1027          CMPW    #^XC040,R1               ; TEST CONTROLLER TYPE
                      60    13  0111  1028          BEQL    130$                     ; BRANCH IF TM78
          50    53    20    C7  0113  1029          DIVL3   #^X80/4,R3,R0            ;CALCULATE ADAPTER UNIT NUMBER
            0A A7    50    90  0117  1030          MOVB    R0,ACF$B_AUNIT(R7)       ;SET ADAPTER UNIT NUMBER
          007C'CF    53    7D  011B  1031          MOVQ    R3,W^ACF$L_R3SAVE        ;SAVE REGISTERS
          0084'CF    55    D0  0120  1032          MOVL    R5,W^ACF$L_R5SAVE        ;
        52    0400 C643    DE  0125  1033          MOVAL   MBA$L_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
          24 A2    51    D0   0131  1041 110$:   MOVL    R1,TM03_TC(R2)              ; SELECT SLAVE DRIVE
50  18 A2  FFFF01FF 8F    CB   0135  1042         BICL3   #^C<^XFE00>,MBA_DT(R2),R0 ;CLEAR EXTRANEOUS BITS
          50   C400 8F    B1   013E  1043         CMPW    #^XC400,R0                  ;SLAVE TAPE DRIVE AND PRESENT?
                    13    12   0143  1044         BNEQ    120$                        ;IF NEQ NO
        0074'CF    51    7D   0145  1045         MOVQ    R1,W^ACF$L_R1SAVE           ;SAVE REGISTERS
          12 A7    51    B0   014A  1046         MOVW    R1,ACF$W_CONIT(R7)          ;SET CONTROLLER UNIT NUMBER
             50    01    D0   014E  1047         MOVL    #1,R0                       ;SET SUCCESS INDICATOR
                   9E    16   0151  1048         JSB     @(SP)+                      ;DELIVER UNIT TO CALLER
51      0074'CF    7D    0153  1049         MOVQ    W^ACF$L_R1SAVE,R1           ;RESTORE REGISTERS
          D5 51    07    F3   0158  1050 120$:   AOBLEQ  #7,R1,1T0$                 ;ANY MORE SLAVES TO PROCESS?
          52   14 A7    D0   015C  1051         MOVL    ACF$L_DEVNAME(R7),R2        ;GET ADDRESS OF DEVICE NAME STRING
             51   62    9A   0160  1052         MOVZBL  (R2),R1                     ;GET LENGHT OF STRING IN BYTES
                6241    96   0163  1053         INCB    (R2)[R1]                    ;INCREMENT CONTROLLER DESIGNATION
53      007C'CF    7D    0166  1054         MOVQ    W^ACF$L_R3SAVE,R3           ;RESTORE REGISTERS
55      0084'CF    D0    016B  1055         MOVL    W^ACF$L_R5SAVE,R5           ;
              FF51   31    0170  1056         BRW     56$                         ;
                                0173  1057
                                0173  1058 ;
                                0173  1059 ; TM78 SPECIFIC CODE
                                0173  1060 ;
                                0173  1061
50    53    20    C7    0173  1062 130$:   DIVL3   #^X80/4,R3,R0                   ;CALCULATE ADAPTER UNIT NUMBER
  0A A7    50    90    0177  1063         MOVB    R0,ACF$B_AUNIT(R7)          ;SET ADAPTER UNIT NUMBER
007C'CF    53    7D    017B  1064         MOVQ    R3,W^ACF$L_R3SAVE           ;SAVE REGISTERS
0084'CF    55    D0    0180  1065         MOVL    R5,W^ACF$L_R5SAVE           ;
52    0400 C643    DE    0185  1066         MOVAL   MBA$L_ERB(R6)[R3],R2        ;GET ADDRESS OF DRIVE CONTROL REGISTER
  0C A7    52    D0    018B  1067         MOVL    R2,ACF$L_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.
           FD 51    F5   019B  1075 140$:   SOBGTR  R1,140$                     ; Waste a little time while TMCLR starts.
   51   001E8480 8F    D0   019E  1076         MOVL    #2000000,R1                 ; Set counter to wait for TMCLR to end.
     50    44 A2    D0   01A5  1077 150$:   MOVL    TM78_ID(R2),R0              ; Read TM78 register.
   50   00008000 8F    D3   01A9  1078         BITL    #TM78_M_TMRDY,R0            ; See if TMCLR done yet.
              06    12   01B0  1079         BNEQ    160$                        ; NEQ implies that TMCLR is done.
         F0 51    F5   01B2  1080         SOBGTR  R1,150$                     ; Else loop back and test again.
            0077   31    01B5  1081         BRW     220$                        ; If we fall thru, TM78 did not come READY.
                         01B8  1082 160$:
             51    D4    01B8  1083         CLRL    R1                          ;CLEAR STARTING SLAVE DRIVE NUMBER
                         01BA  1084
                         01BA  1085 ;
                         01BA  1086 ; CONSTRUCT UNIT DATA BASE AND DELIVER TO CALLER
                         01BA  1087 ;
                         01BA  1088
     50    0A A7    9A   01BA  1089 170$:   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    DD  01C9  1093              PUSHL   R1                                      ; Save drive number.
              51      FA 8F    9A  01CB  1094              MOVZBL  #250,R1                                 ; Set a counter.
                      FD 51    F5  01CF  1095  180$:       SOBGTR  R1,180$                                 ; Waste time while ATTN bit clears
                   51    6E    D0  01D2  1096              MOVL    (SP),R1                                 ; Restore drive number.
                30 A249  09    D0  01D5  1097              MOVL    #TM78_SENSE_GO,TM78_NDTO(R2)[R1]; Set sense command.
          51    001E8480 8F    D0  01DA  1098              MOVL    #2000000,R1                             ; Set count to prevent hang.
                                   01E1  1099  190$:
                      08 A6    DD  01E1  1100              PUSHL   MBA$L_SR(R6)                    ; Copy MBA status register to stack.
          8E    00010000 8F    D3  01E4  1101              BITL    #MBA$M_SR_ATTN,(SP)+            ; Test for MBA attention bit on
                                   01EB  1102                                                      ;  in MBA status register.
                         03    12  01EB  1103              BNEQ    200$                            ; NEQ implies MBA attention came on.
                      F1 51    F5  01ED  1104              SOBGTR  R1,190$                         ; Loop waiting for MBA attention bit.
                                   01F0  1105  200$:                                               ; If here, either MBA attention bit came
                                   01F0  1106                                                      ;  on or the loop gave out.
                   51 8ED0     01F0  1107              POPL    R1                                  ; Restore drive number.
                10 A2 50       D3  01F3  1108              BITL    R0,TM78_AB(R2)                  ; Assure TM78 attention bit set.
                      32       13  01F7  1109              BEQL    210$                            ; EQL implies NO attention which
                                   01F9  1110                                                      ;  means the device doesn't respond so
                                   01F9  1111                                                      ;  let's ignore it.
                   1C A2       DD  01F9  1112              PUSHL   TM78_DS(R2)                     ; Copy device status register to stack.
                   2C A2       DD  01FC  1113              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  1120              CMPZV   #16,#6,R0,#1                    ; See if we got back DONE status.
                         17    12  0212  1121              BNEQ    210$                            ; If not, then device is NO GOOD.
                                   0214  1122
                13 50    0E    E1  0214  1123              BBC     #14,R0,210$                     ; Branch if drive not present.
          0074'CF    51    7D  0218  1124              MOVQ    R1,W^ACF$L_R1SAVE               ;SAVE REGISTERS
             12 A7    51    B0  021D  1125              MOVW    R1,ACF$W_CONIT(R7)              ;SET CONTROLLER UNIT NUMBER
                50    01    D0  0221  1126              MOVL    #1,R0                           ;SET SUCCESS INDICATOR
                      9E    16  0224  1127              JSB     @(SP)+                          ;DELIVER UNIT TO CALLER
          51    0074'CF    7D  0226  1128              MOVQ    W^ACF$L_R1SAVE,R1               ;RESTORE REGISTERS
             8B 51    03    F3  022B  1129  210$:       AOBLEQ  #3,R1,170$                      ;ANY MORE SLAVES TO PROCESS?
                                   022F  1130  220$:
             52    14 A7    D0  022F  1131              MOVL    ACF$L_DEVNAME(R7),R2            ;GET ADDRESS OF DEVICE NAME STRING
                51    62    9A  0233  1132              MOVZBL  (R2),R1                         ;GET LENGTH OF STRING IN BYTES
                   6241    96  0236  1133              INCB    (R2)[R1]                        ;INCREMENT CONTROLLER DESIGNATION
          53    007C'CF    7D  0239  1134              MOVQ    W^ACF$L_R3SAVE,R3               ;RESTORE REGISTERS
          55    0084'CF    D0  023E  1135              MOVL    W^ACF$L_R5SAVE,R5               ;
                   FE7E    31  0243  1136              BRW     56$                             ;
                               0246  1137              .DSABL  LSB
```

```
                     0246  1139          .SBTTL  ACF$DR - DR32 ADAPTER AUTO CONFIGURATION
                     0246  1140 ;
                     0246  1141 ; ACF$DR - DR32 ADAPTER AUTO CONFIGURATION
                     0246  1142 ;
                     0246  1143 ; THIS ROUTINE IS CALLED TO AUTO CONFIGURE A DR32 ADAPTER.
                     0246  1144 ;
                     0246  1145 ; INPUTS:
                     0246  1146 ;
                     0246  1147 ;       R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
                     0246  1148 ;       R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
                     0246  1149 ;       R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
                     0246  1150 ;
                     0246  1151 ; OUTPUTS:
                     0246  1152 ;
                     0246  1153 ;       A CO-ROUTINE CALL IS MADE TO THE CALLER DELIVERING THE DEVICE
                     0246  1154 ;       UNIT DESCRIPTOR.
                     0246  1155 ;
                     0246  1156
                     0246  1157 ACF$DR:                                         ;AUTO CONFIGURE DR32 ADAPTER
 54  002E'CF   D0    0246  1158          MOVL    W^ACF$AB_DRTABLE,R4            ;GET ADDRESS OF DRIVER DESCRIPTOR
       05      11    024B  1159          BRB     ACF$SINGLE_DEVICES             ;BRANCH TO COMMON CODE
```

```
                                024D  1161          .SBTTL  ACF$CI - CI ADAPTER AUTO CONFIGURATION
                                024D  1162 ;
                                024D  1163 ; ACF$CI - 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 ACF$CI:                                      ;AUTO CONFIGURE CI ADAPTER
          54    0032'CF   DO    024D  1180          MOVL    W^ACF$AB_CITABLE,R4       ;GET ADDRESS OF DRIVER DESCRIPTOR
                                0252  1181
                                0252  1182 ACF$SINGLE_DEVICES:                         ;COMMON ENTRY POINT FOR DR AND CI
            1E A7    01   90    0252  1183          MOVB    #1,ACF$B_CNUMVEC(R7)      ;SET NUMBER OF CONTROLLER VECTORS
          10 A7    08 A7   B0    0256  1184          MOVW    ACF$W_AVECTOR(R7),ACF$W_CVECTOR(R7) ;SET CNTRLR VECTOR OFFSET
            0C A7    56   D0    025B  1185          MOVL    R6,ACF$L_CONTRLREG(R7)    ;SET ADDRESS OF CONTROL STATUS REGISTER
               51    84   D0    025F  1186          MOVL    (R4)+,R1                  ;GET ADDRESS OF DEVICE NAME STRING
            14 A7    51   D0    0262  1187          MOVL    R1,ACF$L_DEVNAME(R7)      ;SET ADDRESS OF DEVICE NAME STRING
            18 A7    64   D0    0266  1188          MOVL    (R4),ACF$L_DRVNAME(R7)    ;SET ADDRESS OF DRIVER NAME STRING
               52    61   9A    026A  1189          MOVZBL  (R1),R2                   ;GET LENGTH OF DEV. NAME STRING IN BYTES
      6142  41 8F    0B A8   81    026D  1190          ADDB3   ADP$B_NUMBER(R8),#^A/A/,(R1)[R2] ;CALCULATE AND STORE LAST
                                0274  1191                                            ;BYTE OF DEVICE NAME
            0A A7    94   0274  1192          CLRB    ACF$B_AUNIT(R7)           ;SET ADAPTER UNIT NUMBER = 0
            12 A7    B4   0277  1193          CLRW    ACF$W_CUNIT(R7)           ;SET CONTROLLER UNIT NUMBER = 0
             0512    30   027A  1194          BSBW    ACF$CLR_ACF               ;CLEAR DEVICE BLOCKS
          50   01    D0   027D  1195          MOVL    #1,R0                     ;SET SUCCESS INDICATOR
               9E    16   0280  1196          JSB     @(SP)+                    ;DELIVER UNIT TO CALLER
               50    D4   0282  1197          CLRL    R0                        ;SET END OF SCAN INDICATOR
                     05   0284  1198          RSB
                          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$UBA:                                     ;AUTO CONFIGURE UNIBUS ADAPTER
     0000024A'EF  94 0285  1222          CLRB    ACF$T_SYS_DEVNAME           ;SET NO SYSTEM DISK ON THIS ADAPTER
     00000245'EF  D4 028B  1223          CLRL    ACF$L_SYS_CSR               ;
51   00000000'GF  D0 0291  1224          MOVL    G^EXE$GL_RPB,R1             ;GET ADDRESS OF THE RPB
   20 A1   0C A8  B1 0298  1225          CMPW    ADP$W_TR(R8),RPB$L_BOOTTR(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$L_PDT(R0),R1            ;GET PORT DRIVER DESCRIPTOR TABLE ADDRESS
             05  13 02B1  1230          BEQL    5$                          ;IF EQL BO PORT DRIVER
   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$L_IDB(R1),R1;GET SYSTEM DEVICE IDB ADDRESS
   52   1000 C6  9E 02C0  1234          MOVAB   UBA_IOBASE(R6),R2           ;GET UNIBUS ADDRESS SPACE VIRTUAL ADDRESS
     52   61  D1 02C5  1235          CMPL    IDB$L_CSR(R1),R2            ;CSR IN UNIBUS ADDRESS SPACE?
             1F  1B 02C8  1236          BLEQU   9$                          ;IF LEQU NO
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$T_SYS_DEVNAME        ;SAVE SYSTEM DEVICE NAME SIZE
     15 A1  50  28 02E0  1241          MOVC3   R0,DDB$T_NAME+1(R1),-       ;SAVE SYSTEM DEVICE NAME
   0000024B'EF      02E4  1242                  ACF$T_SYS_DEVNAME+1         ;
         0A A7  94 02E9  1243 9$:      CLRB    ACF$B_AUNIT(R7)             ;CLEAR ADAPTER UNIT NUMBER
   0241'CF  08  B0 02EC  1244          MOVW    #^O10,W^ACF$W_CSRBASE       ;SET INITIAL CSR BASE OFFSET
0243'CF  00C0  8F B0 02F1  1245          MOVW    #^O300,W^ACF$W_VECBASE      ;SET INITIAL VECTOR BASE OFFSET
   55   0090'CF  9E 02F8  1246          MOVAB   W^ACF$AB_UBATABLE,R5        ;GET ADDRESS OF UBA DRIVER DESCRIPTOR TABLE
     54   85  D0 02FD  1247 10$:      MOVL    (R5)+,R4                    ;GET ADDRESS OF NEXT DRIVER DESCRIPTOR
             03  12 0300  1248          BNEQ    20$                         ;IF NEQ ANOTHER ENTRY TO PROCESS
             50  D4 0302  1249          CLRL    R0                          ;SET END OF SCAN INDICATOR
             05  05 0304  1250          RSB                                 ;
                     0305  1251
                     0305  1252 ;
                     0305  1253 ; SCAN UBA DEVICE
                     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    D0  030E  1258          MOVL    UBT$L_DEVNAME(R4),ACF$L_DEVNAME(R7) ;SET ADDRESS OF DEVICE NAME STRI
             18 A7    04 A4    D0  0312  1259          MOVL    UBT$L_DRVNAME(R4),ACF$L_DRVNAME(R7) ;SET ADDRESS OF DRIVER NAME STRI
          003E'CF    0C A4    D0  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_CNUMVEC(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
                0B 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
                0B 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),R1                  ;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
                   51    84    AA  0349  1276          BICW    (R4)+,R1                 ;TRUNCATE TO ACTUAL VECTOR OFFSET
             023E'CF    51    B0  034C  1277          MOVW    R1,W^ACF$AW_SAVEVEC      ;SAVE FOR CHECK LATER
                03 63    01    E1  0351  1278 25$:     BBC     #UBA_V_FLOATCSR,(R3),ACF$UBAFIXED ;IF CLR, FIXED CSR
                      0086   31  0355  1279          BRW     ACF$UBAFLOATING          ;
                                   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
                         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    D0  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    D0  0372  1293          MOVL    R2,R0                    ;GET COPY OF CSR
                00000000'GF    16  0375  1294          JSB     G^EXE$TEST_CSR           ;CHECK DEVICE CSR
                      DA    50  E9  037B  1295          BLBC    R0,30$                   ;BRANCH IF CSR NON-EXISTENT
             0074'CF    51    7D  037E  1296          MOVQ    R1,W^ACF$L_R1SAVE        ;SAVE REGISTERS
             007C'CF    53    7D  0383  1297          MOVQ    R3,W^ACF$L_R3SAVE        ;
             0084'CF    55    D0  0388  1298          MOVL    R5,W^ACF$L_R5SAVE        ;
                      00EE    30  038D  1299          BSBW    FIX_DEV_NAME             ;FIX UP DEVICE NAME
             000004C1'EF    16  0390  1300          JSB     LOAD_DRIVER              ;LOAD DRIVER
                003E'DF    16  0396  1301          JSB     @W^ACF$L_ROUTINE         ;CALL GENERATION ROUTINE
                51    14 A7    D0  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$L_R1SAVE,R1        ;RESTORE REGISTERS
             53    007C'CF    7D  03A6  1305          MOVQ    W^ACF$L_R3SAVE,R3        ;
             55    0084'CF    D0  03AB  1306          MOVL    W^ACF$L_R5SAVE,R5        ;
                A4 63    02    E1  03B0  1307 35$:     BBC     #UBA_V_FLOATVEC,(R3),30$ ;IF CLR, FIXED VECTOR ASSIGNMENT
                50    1E A7    9A  03B4  1308          MOVZBL  ACF$B_CNUMVEC(R7),R0     ;GET NUMBER OF CONTROLLER INTERRUPT VECTORS
                   51    6140  DE  03B8  1309          MOVAL   (R1)[R0],R1              ;CALCULATE ADDRESS OF NEXT VECTOR
          50    0000006E'EF    3C  03BC  1310          MOVZWL  ACF$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
                            8D  11  03C9  1313          BRB     30$                      ;
                0C 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^ACF$AW_SAVEVEC,R1     ;ANY VECTORS ASSIGNED?
                          05    13   03D4  1316          BEQL    39$                     ;IF EQL NO
                0243'CF   51    B0   03D6  1317          MOVW    R1,W^ACF$W_VECBASE      ;SET NEW VECTOR BASE ADDRESS
                      FF1F    31   03DB  1318 39$:       BRW     10$                     ;
                                   03DE  1319
                                   03DE  1320 ;
                                   03DE  1321 ; FLOATING VECTOR/CSR DEVICE
                                   03DE  1322 ;
                                   03DE  1323
                                   03DE  1324 ACF$UBAFLOATING:                           ;
                52   0241'CF    3C   03DE  1325          MOVZWL  W^ACF$W_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          MOVAB   UBA_IOBASE(R6)[R2],R0   ;GET ACTUAL CSR ADDRESS
               00000000'GF    16   03EF  1329          JSB     G^EXE$TEST_CSR          ;CHECK FOR NON-EXISTENT CSR
                       03 50  E8   03F5  1330          BLBS    R0,38$                  ; branch if csr there
                     0071     31   03F8  1331          BRW     60$                     ;BRANCH IF CSR NON-EXISTENT
               0241'CF   52    B0   03FB  1332 38$:     MOVW    R2,W^ACF$W_CSRBASE      ;SET NEW BASE CSR OFFSET
             50    1000 C642    9E   0400  1333 40$:     MOVAB   UBA_IOBASE(R6)[R2],R0   ;GET ACTUAL CSR ADDRESS
               00000000'GF    16   0406  1334          JSB     G^EXE$TEST_CSR          ;CHECK FOR NON-EXISTENT CSR
                     5D 50    E9   040C  1335          BLBC    R0,60$                  ;BRANCH IF CSR NON-EXISTENT
          0C A7    1000 C642   9E   040F  1336          MOVAB   UBA_IOBASE(R6)[R2],-    ;SET ADDRESS OF
                               0416  1337                  ACF$L_CNTRLREG(R7)          ; CONTROL REGISTER
                  10 A7   51    B0   0416  1338          MOVW    R1,ACF$W_CVECTOR(R7)    ;SET ADDRESS OF INTERRUPT VECTOR
                0074'CF   51    7D   041A  1339          MOVQ    R1,W^ACF$L_R1SAVE       ;SAVE REGISTERS
                007C'CF   53    7D   041F  1340          MOVQ    R3,W^ACF$L_R3SAVE       ;
                0084'CF   55    D0   0424  1341          MOVL    R5,W^ACF$L_R5SAVE       ;
                     0052       30   0429  1342          BSBW    FIX_DEV_NAME            ;FIX UP DEVICE NAME
                 000004C1'EF    16   042C  1343          JSB     LOAD_DRIVER             ;LOAD DRIVER
                   003E'DF      16   0432  1344          JSB     @W^ACF$L_ROUTINE        ;CALL DEVICE GENERATION ROUTINE
                    51   14 A7   D0   0436  1345          MOVL    ACF$L_DEVNAME(R7),R1    ;GET ADDRESS OF DEVICE NAME STRING
                     030E       30   043A  1346          BSBW    ACF$INC_CHAR            ;INCREMENT CONTROLLER
                                   043D  1347
                51   0074'CF    7D   043D  1348          MOVQ    W^ACF$L_R1SAVE,R1       ;RESTORE REGISTERS
                53   007C'CF    7D   0442  1349          MOVQ    W^ACF$L_R3SAVE,R3       ;
                55   0084'CF    D0   0447  1350          MOVL    W^ACF$L_R5SAVE,R5       ;
                  B0 63    02    E1   044C  1351 50$:     BBC     #UBA_V_FLOATVEC,(R3),40$ ;IF CLR, FIXED VECTOR ASSIGNMENT
                  50    1E A7    9A   0450  1352          MOVZBL  ACF$B_CNUMVEC(R7),R0    ;GET NUMBER OF CONTROLLER INTERRUPT VECTORS
                  51    6140     DE   0454  1353          MOVAL   (R1)[R0],R1             ;CALCULATE ADDRESS OF NEXT VECTOR
            50   0000006E'EF    3C   0458  1354          MOVZWL  ACF$W_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$:     MOVW    R1,W^ACF$W_VECBASE      ;SAVE NEW VECTOR OFFSET
                0241'CF   52    B0   0471  1361          MOVW    R2,W^ACF$W_CSRBASE      ;SAVE NEW CSR OFFSET
                0241'CF   02    A0   0476  1362          ADDW    #2,W^ACF$W_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   0000024A'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 00000249'EF    E9   048F  1384          BLBC    ACF$B_BOOT_TR,10$       ;SYSTEM DEVICE ON THIS ADAPTER?
  00000245'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          CMPC5   R2,(R1),#^A/ /,R4,(R3)  ;DEVICE NAME MATCH?
                 08    13   04A6  1388          BEQL    5$                      ;IF EQL YES, NO WORK
   14 A7   0000024A'EF    9E   04A8  1389          MOVAB   ACF$T_SYS_DEVNAME,ACF$L_DEVNAME(R7);USE SYSTEM DEVICE NAME
                 05    04B0  1390 5$:     RSB                                     ;RETURN
63   54   20   61    52    2D   04B1  1391 10$:    CMPC5   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$L_DEVNAME(R7),R1    ;GET DEVICE NAME ADDRESS
                028B    30   04BD  1394          BSBW    ACF$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$L_ROUTINE  ; Is this a generic routine device?
                   01    13     04CA  1412          BEQL    10$                             ; No - Exit
                         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$:     MOVL    #1,R0                           ; Set success
                0036'CF 8ED0    04D5  1417          POPL    W^ACF$L_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$:     JMP     @W^ACF$L_RETURN                 ; Return to caller
                                04E5  1422
```

```
                     1424          .SBTTL  ACF$ADD_UNITS - GENERIC ROUTINE FOR DEVICE GENERATION
                     1425 ;
                     1426 ; ACF$CR11 - CR11 CARD READER
                     1427 ; ACF$LP11 - LP11 LINE PRINTER
                     1428 ; ACF$LPA11 - LPA11 LABORATORY I/O SUBSYSTEM, CONTROLLER A
                     1429 ; ACF$LPA11B - LPA11 LABORATORY I/O SUBSYSTEM, CONTROLLER B
                     1430 ; ACF$DMC11 - DMC11 SYNCHRONOUS COMMUNICATIONS
                     1431 ; ACF$TS11 - TS11 MAGTAPE, CONTROLLER A
                     1432 ; ACF$TS11B - TS11 MAGTAPE, CONTROLLER B
                     1433 ;
                     1434 ; THIS ROUTINE IS CALLED AS AN ACTION ROUTINE FROM THE UBA DEVICE SCAN TO GENERATE
                     1435 ; A SINGLE UNIT DATA BASE.
                     1436 ;
                     1437 ; IT IS ALSO CALLED AS AN ACTION ROUTINE FOR UNSUPPORTED DEVICES.
                     1438 ;
                     1439 ; INPUTS:
                     1440 ;
                     1441 ;       R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
                     1442 ;       R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
                     1443 ;       R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
                     1444 ;
                     1445 ; OUTPUTS:
                     1446 ;
                     1447 ;       A CO-ROUTINE CALL IS MADE TO THE ORIGINAL CALLER DELIVERING THE DEVICE UNIT
                     1448 ;       DESCRIPTOR.
                     1449 ;
                     1450
                     1451 ; Unsupported:
                     1452
                     1453 ACF$DH11:
                     1454 ACF$DJ11:
                     1455 ACF$DC11:
                     1456 ACF$DM11B:
                     1457 ACF$DN11:
                     1458 ACF$DQ11:
                     1459 ACF$DR11B:
                     1460 ACF$DR11C:
                     1461 ACF$DT11:
                     1462 ACF$DU11:
                     1463 ACF$DUP11:
                     1464 ACF$DV11:
                     1465 ACF$DWR70:
                     1466 ACF$DX11:
                     1467 ACF$GT40:
                     1468 ACF$DL11C:
                     1469 ACF$KMC11:
                     1470 ACF$KW11C:
                     1471 ACF$KW11W:
                     1472 ACF$LK11:
                     1473 ACF$LPP11:
                     1474 ACF$LPS11:
                     1475 ACF$PP611:
                     1476 ACF$PR611:
                     1477 ACF$RSV:
                     1478 ACF$RX11:
                     1479 ACF$VMV21:
                     1480 ACF$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$QNA:                                     ;
                     04E5  1509 ACF$VC01B:                                   ;
                     04E5  1510
                     04E5  1511 ACF$ADD_UNITS:
                     04E5  1512
          0036'CF 8ED0   04E5  1513          POPL    W^ACF$L_RETURN           ; Save return address
          0042'CF   D4   04EA  1514          CLRL    W^ACF$L_DELIVER_UNIT     ; Deliver unit routine in driver
    51    0000'CF   D0   04EE  1515          MOVL    W^ACF$GL_DPT,R1          ; DPT of driver just loaded
                0D  13   04F3  1516          BEQL    5$                       ; Branch if none
                         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$L_DELIVER_UNIT ; Set address of driver action routine
                         0502  1521
                55  D4   0502  1522 5$:      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
                         0526  1533
          50  01   D0   0526  1534 20$:      MOVL    #1,R0                    ; Set success indicator
                9E  16   0529  1535          JSB     @(SP)+                   ; Deliver unit to caller
                         052B  1536
    51    0000'CF   D0   052B  1537 30$:      MOVL    W^ACF$GL_DPT,R1 ; Address of DPT
```

```
                 0C   13  0530  1538          BEQL    40$                             ; Driver not found
51    1A A1      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$L_RETURN2                 ;SAVE RETURN ADDRESS
                     0547  1572 ;
                     0547  1573 ; First load port driver and create units normally.
                     0547  1574 ;
     9C   10         0547  1575          BSBB    ACF$ADD_UNITS
                     0549  1576
                     0549  1577 ;
                     0549  1578 ; Now, load class driver and its associated database.
                     0549  1579 ;
                     0549  1580
          28   28    0549  1581          MOVC3   #ACF$C_LENGTH,-
00000046'EF   67     054B  1582                  (R7),ACF$L_ACF_SAVE     ; Save ACF block
                     0551  1583
                     0551  1584 ;
                     0551  1585 ; Pick up name of class device and driver
                     0551  1586 ;
                     0551  1587
     51   14 A7  D0  0551  1588          MOVL    ACF$L_DEVNAME(R7),R1    ; Get address of port device name
52   00000228'EF  9E  0555  1589          MOVAB   ACF$AB_CLASS_TABLE,R2   ; Get address of association table
                     055C  1590 10$:
     50   82  D0  055C  1591          MOVL    (R2)+,R0                ; Get pointer to next structure
          62  13  055F  1592          BEQL    90$                     ; If EQL, end of table
  01 A1   09 A0  B1  0561  1593          CMPW    CLS$T_PORTDEV(R0),1(R1) ; Do we have a match?
          F4  12  0566  1594          BNEQ    10$                     ; If NEQ, no - try again
          60  D0  0568  1595          MOVL    CLS$L_CLASSDEV(R0),-
          14 A7       056A  1596                  ACF$L_DEVNAME(R7)       ; Store address of class device name
          04 A0  D0   056C  1597          MOVL    CLS$L_CLASSDRV(R0),-
          18 A7       056F  1598                  ACF$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    0C A8   3C  0575  1604          MOVZWL  ADP$W_TR(R8),R1          ; second longword
        51  00008000 8F   C8  0579  1605          BISL    #^X8000,R1              ; Set high bit in first word
        52  00000000'EF   7E  0580  1606          MOVAQ   BOO$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$L_CONTRLREG(R7)  ; Set in CSR field
                            058E  1609
                      10    90  058E  1610          MOVB    #ACF$M_SUPPORT,-
                    0B A7           0590  1611                  ACF$B_AFLAG(R7)         ; Clear configuration flag, set supported
                    21 A7   94  0592  1612          CLRB    ACF$B_NUMUNIT(R7)       ; Clear number of units
                    12 A7   B4  0595  1613          CLRW    ACF$W_CUNIT(R7)         ; Clear unit number
                    1C A7   B4  0598  1614          CLRW    ACF$W_MAXUNITS(R7)      ; Clear maxunits
                1E A7   01  90  059B  1615          MOVB    #1,ACF$B_CNUMVEC(R7)    ; Set cnumvec to 1
                     01ED   30  059F  1616          BSBW    ACF$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     @(SP)+                  ;
                            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$L_ADAPTER(R7),-     ; Fill in ADP address in CRB
                    38 A1           05B2  1626                  CRB$L_INTD+VEC$L_ADP(R1)
                            05B4  1627
                            05B4  1628 ;
                            05B4  1629 ; Increment device name
                            05B4  1630 ;
                            05B4  1631 20$:
                51  14 A7   D0  05B4  1632          MOVL    ACF$L_DEVNAME(R7),R1    ; Set address
                     0190   30  05B8  1633          BSBW    ACF$INC_CHAR            ; Do the increment
                            05BB  1634
                      28    28  05BB  1635          MOVC3   #ACF$C_LENGTH,-
        67  00000046'EF           05BD  1636                  ACF$L_ACF_SAVE,(R7)     ; Restore ACF block
                            05C3  1637 90$:
                 003A'DF    17  05C3  1638          JMP     @W^ACF$L_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:                                                    ;
    0036'CF 8ED0        05C7  1661          POPL    W^ACF$L_RETURN                              ;SAVE RETURN ADDRESS
            50   D4     05CC  1662          CLRL    R0                                          ;SET STARTING UNIT NUMBER
 12 A7      50   B0     05CE  1663 70$:     MOVW    R0,ACF$W_CUNIT(R7)                          ;SET CONTROLLER UNIT NUMBER
 51   0C    A7   D0     05D2  1664          MOVL    ACF$L_CONTRLREG(R7),R1                      ;GET ADDRESS OF DEVICE CSR
            2C   10     05D6  1665          BSBB    130$                                        ;WAIT FOR CONTROLLER READY
 04 A1      0B   B0     05D8  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
            1E   10     05E4  1670          BSBB    130$                                        ;WAIT FOR CONTROLLER READY
 61   0400  8F   B3     05E6  1671          BITW    #RL_CS_M_OPI,RL_CS(R1)                      ;OPERATION COMPLETE?
            0F   12     05EB  1672          BNEQ    80$                                         ;IF NEQ NO - NONEXISTENT DRIVE
 0070'CF    50   D0     05ED  1673          MOVL    R0,W^ACF$L_R0SAVE                           ;SAVE REGISTER
            50   01     05F2  1674          MOVL    #1,R0                                       ;SET SUCCESS INDICATOR
            9E   16     05F5  1675          JSB     @(SP)+                                      ;DELIVER UNIT TO CALLER
 50   0070'CF    D0     05F7  1676          MOVL    W^ACF$L_R0SAVE,R0                           ;RESTORE REGISTER
   CE 50    03   F3     05FC  1677 80$:     AOBLEQ  #3,R0,70$                                   ;ANY MORE UNITS TO PROCESS?
        0036'DF   17     0600  1678          JMP     @W^ACF$L_RETURN                             ;ALL DONE
                        0604  1679
 52  01312D00  8F   D0   0604  1680 130$:    MOVL    #2*500*1000*20,R2                           ; * should be TIMEWAIT
            61   95     060B  1681 140$:    TSTB    RL_CS(R1)                                   ; Controller ready?
            01   18     060D  1682          BGEQ    150$                                        ; branch if not
                 05     060F  1683          RSB                                                 ; Continue
          F8 52   F5     0610  1684 150$:    SOBGTR  R2,140$                                     ; Loop
             8E   D5     0613  1685          TSTL    (SP)+                                       ; Get rid of return PC
        0036'DF   17     0615  1686          JMP     @W^ACF$L_RETURN                             ; error - exit
                        0619  1687
```

```
                        0619  1689          .SBTTL  MULTIPLE DEVICE GENERATOR
                        0619  1690 ;
                        0619  1691 ; ACF$COMBO_DEVICES - MULTIPLE DEVICE COMBO BOARD
                        0619  1692 ;
                        0619  1693 ; THIS ROUTINE IS CALLED AS AN ACTION ROUTINE FROM THE UBA DEVICE SCAN TO GENERATE
                        0619  1694 ; THE DATA BASE FOR THE DMF32, WHICH INCLUDES A SYNC LINE, ASYNC LINE (DZ TYPE),
                        0619  1695 ; A LP LINE, AND A DR11C LINE. A MAXIMUM OF 3  OF THESE DEVICES CAN BE ON THE
                        0619  1696 ; BOARD AT ONE TIME, BUT VARIOUS COMBINATIONS OF THE FOUR IS POSSIBLE.
                        0619  1697 ; THIS ROUTINE IS ALSO CALLED FOR THE DMZ32 WHICH INCLUDES 3 ASYNC LINES, AND
                        0619  1698 ; THE CPI32 WHICH INCLUDES 3 SYNC LINES.
                        0619  1699 ;
                        0619  1700 ; INPUTS:
                        0619  1701 ;
                        0619  1702 ;       R2 = ADDRESS OF COMBO DEVICE DECRIPTOR TABLE
                        0619  1703 ;       R3 = BITMASK OF DEVICES PRESENT
                        0619  1704 ;       R6 = ADDRESS OF CONFIGURATION STATUS REGISTER.
                        0619  1705 ;       R7 = ADDRESS OF CONFIGURATION CONTROL BLOCK.
                        0619  1706 ;       R8 = ADDRESS OF ADAPTER CONTROL BLOCK.
                        0619  1707 ;
                        0619  1708 ; OUTPUTS:
                        0619  1709 ;
                        0619  1710 ;       A CO-ROUTINE CALL IS MADE TO THE ORIGINAL CALLER DELIVERING EACH DEVICE
                        0619  1711 ;       UNIT DESCRIPTOR.
                        0619  1712 ;
                        0619  1713
                        0619  1714 ACF$BITMASK:
                   7FFF 0619  1715          .WORD   ^X7FFF                          ;BIT MASK FOR EXTZV INSTRUCTION TO
                        061B  1716                                                  ;CONVERT A NUMBER TO A BIT MASK
                        061B  1717 ACF$DMF32:
    52  0000016C'EF  D0 061B  1718          MOVL    ACF$AL_DMF32_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$AB_DMF32_TABLE,R2   ;ADDRESS OF DESCRIPTOR TABLE
                        0630  1721
                        0630  1722 ; READ THE DMF32 IDENT REGISTER TO GET BITMASK OF DEVICES PRESENT
                        0630  1723
        53  0C B7  B0 0630  1724          MOVW    @ACF$L_CONTRLREG(R7),R3
53  53  04  0C  EF 0634  1725          EXTZV   #DMF$V_IDENT,#DMF$S_IDENT,R3,R3
                3B  11 0639  1726          BRB     ACF$COMBO_DEVICES       ;CALL COMMON COMBO DEVICE ROUTINE
                        063B  1727
                        063B  1728 ACF$CPI32:
    52  00000194'EF  D0 063B  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
                15  11 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$DMZCPI:
        53  0C B7  B0 0667  1739          MOVW    @ACF$L_CONTRLREG(R7),R3 ;GET CONTENTS OF THE MAIN CSR
                08  EF 066B  1740          EXTZV   #CPI$V_SUBCNTRL,-               ;EXTRACT THE NUMBER OF SUBCONTROLLERS
    53  53  04          066D  1741                  #CPI$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$L_RETURN2                 ;SAVE RETURN ADDRESS
```

```
00000238'EF   0C A7   D0   067B  1746          MOVL    ACF$L_CONTRLREG(R7),ACF$AL_CSR  ;SAVE CSR BASE
0000023C'EF   10 A7   B0   0683  1747          MOVW    ACF$W_CVECTOR(R7),ACF$AW_VECTOR ;SAVE VECTOR BASE
00000240'EF   1E A7   90   068B  1748          MOVB    ACF$B_CNUMVEC(R7),ACF$AB_NUMVEC ;SAVE NUMVEC
                      0693  1749
                      0693  1750
              54   82   D0   0693  1751 10$:    MOVL    (R2)+,R4                        ;NEXT DEVICE
                     1A   12   0696  1752          BNEQ    20$                             ;BRANCH IF NOT END OF LIST
14 A7   00000234'EF   D0   0698  1753          MOVL    ACF$AL_DEVNAME,ACF$L_DEVNAME(R7);RESTORE DEVICE NAME
1E A7   00000240'EF   90   06A0  1754          MOVB    ACF$AB_NUMVEC,ACF$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$L_RETURN2                ;RETURN
                      06B2  1759
              53   84   B3   06B2  1760 20$:    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$L_DEVNAME(R7)         ;DEVICE NAME
            18 A7   84   D0   06BB  1764          MOVL    (R4)+,ACF$L_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$L_CONTRLREG(R7) ;ADDRESS OF CSR
                      06E2  1771
            0B A7   10   88   06E2  1772          BISB2   #ACF$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$:    MOVQ    R2,W^ACF$L_R2R3SAVE             ;SAVE R2,R3
                      06F3  1777
                FDEF   30   06F3  1778          BSBW    ACF$ADD_UNITS                   ;USE COMMON ROUTINE TO DELIVER UNITS
  51    14 A7   D0   06F6  1779          MOVL    ACF$L_DEVNAME(R7),R1            ;GET DEVICE NAME ADDRESS
                 004E   30   06FA  1780          BSBW    ACF$INC_CHAR                    ;INCREMENT CONTROLLER LETTER
52    0088'CF   7D   06FD  1781          MOVQ    W^ACF$L_R2R3SAVE,R2             ;RESTORE REGISTERS
                FF8E   31   0702  1782          BRW     10$                             ;LOOP
                      0705  1783
```

```
                                                                 J 11
                     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::                          ;AUTO CONFIGURATION DEVICE DATA BASE RESET
 50  00000000'EF  9E 0705  1804          MOVAB   ACF$AL_RESET,R0         ;GET ADDRESS OF RESET POINTER TABLE
       51    80  D0 070C  1805 10$:      MOVL    (R0)+,R1                ;GET ADDRESS OF NEXT CONTROLLER DESIGNATOR
             06  13 070F  1806          BEQL    20$                     ;IF EQL END OF TABLE
        71  41 8F  90 0711  1807          MOVB    #^A/A/,-(R1)            ;RESET CONTROLLER DESIGNATOR
             F5  11 0715  1808          BRB     10$                     ;LOOP
                 0717  1809
       7E    52  7D 0717  1810 20$:      MOVQ    R2,-(SP)                ;SAVE R2,R3
 50  00000090'EF  9E 071A  1811          MOVAB   ACF$AB_UBATABLE,R0      ;BASE ADDRESS OF TABLE
                 0721  1812
       51    80  D0 0721  1813 30$:      MOVL    (R0)+,R1                ;NEXT ELEMENT IN TABLE
             0D  13 0724  1814          BEQL    40$                     ;END OF LIST
       52    61  D0 0726  1815          MOVL    UBT$L_DEVNAME(R1),R2    ;ADDRESS OF NAME TO RESET
       53    62  9A 0729  1816          MOVZBL  (R2),R3                 ;EXPAND LENGTH
   6342    10  A1  90 072C  1817          MOVB    UBT$B_LETTER(R1),(R3)[R2] ;RESET CONTROLLER LETTER
             EE  11 0731  1818          BRB     30$                     ;LOOP
                 0733  1819
 52  00000110'EF  D0 0733  1820 40$:      MOVL    ACF$AL_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  0A 074A  1825          RSB                             ;RETURN
```

```
                     074B  1827 .SBTTL 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$INC_CHAR (called from AUTOCONFG and CONFIG)
                     074B  1836 ;
                     074B  1837 ; INPUT:
                     074B  1838 ;
                     074B  1839 ;       R1 - Address of device name ascic string
                     074B  1840 ;
                     074B  1841 ; OUTPUT:
                     074B  1842 ;
                     074B  1843 ;       Device name with last character incremented
                     074B  1844 ;
                     074B  1845 ;       For TTcn: (*** NOT ACTIVATED YET ***)
                     074B  1846 ;               Controller is incremented until TTZ and then goes
                     074B  1847 ;               to TTAA,TTAB,...TTAZ,TTBA,...
                     074B  1848 ;
                     074B  1849 ;       For all other:
                     074B  1850 ;               DDA,...,DDP,DEA,...,DEP - wraps at P because its the
                     074B  1851 ;               16th letter of the alphabet. (For RSX compatability).
                     074B  1852 ;
                     074B  1853 ;       TT devices (WILL BE) special cased in the AME.
                     074B  1854 ;
                     074B  1855 ;       R0 does not return status
                     074B  1856 ;
                     074B  1857 ;-
                     074B  1858
                     074B  1859 ACF$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 ;
              08     11 074E  1866         BRB     5$
    5454 8F   01 A1    B1 0750  1867         CMPW    1(R1),#^A/TT/           ; Terminal ?
                 14    13 0756  1868         BEQL    20$                     ; yes - special case code
                     0758  1869
            6041     96 0758  1870 5$:      INCB    (R0)[R1]                ; Increment the last character
       51 8F 6041     91 075B  1871         CMPB    (R0)[R1],#^A/Q/         ; Is this the 17th controller ?
                 09    1F 0760  1872         BLSSU   10$                     ; No, branch
             FF A041    96 0762  1873         INCB    -1(R0)[R1]              ; Increment last char of device name
     6041    41 8F      90 0766  1874         MOVB    #^A/A/,(R0)[R1]         ; Move /A/ to last
                 05    05 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    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
```

```
                61    04    90  077C  1884          MOVB    #4,(R1)                 ; Set up longer name
      03 A1     4141  8F    B0  077F  1885          MOVW    #^A/AA/,3(R1)           ; Move in "AA"
                              05  0785  1886          RSB                             ; Return
                                  0786  1887
                  03  A1      96  0786  1888 40$:     INCB    3(R1)                   ; Increment first controller letter
        04 A1     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
```

```
$$$                     = 00000111 R     04
$COMBO$                 = 000000D2 R     04
$CR$                    = 00000018 R     04
$CRDRIVER$              = 00000036 R     05
$DB$                    = 00000000 R     04
$DBDRIVER$              = 00000000 R     05
$DD$                    = 00000048 R     04
$DDDRIVER$              = 00000099 R     05
$DEVDESC$               = 000008F3 R     06
$DEVNAME$               = 00000100 R     04
$DL$                    = 00000024 R     04
$DLDRIVER$              = 00000051 R     05
$DM$                    = 0000001C R     04
$DMDRIVER$              = 0000003F R     05
$DQ$                    = 00000030 R     04
$DQDRIVER$              = 0000006C R     05
$DR$                    = 00000004 R     04
$DRDRIVER$              = 00000009 R     05
$DRVNAME$               = 00000253 R     05
$DT$                    = 000000F5 R     04
$DTDRIVER$              = 00000214 R     05
$DY$                    = 0000002C R     04
$DYDRIVER$              = 00000063 R     05
$DZDRIVER$              = 0000014D R     05
$IS$                    = 000000CE R     04
$ISDRIVER$              = 000001C3 R     05
$IX$                    = 000000ED R     04
$IXDRIVER$              = 00000202 R     05
$LA$                    = 000000B1 R     04
$LADRIVER$              = 00000183 R     05
$LC$                    = 00000105 R     04
$LCDRIVER$              = 00000241 R     05
$LP$                    = 00000020 R     04
$LPDRIVER$              = 00000048 R     05
$LS$                    = 00000078 R     04
$LSDRIVER$              = 00000105 R     05
$MF$                    = 0000000C R     04
$MS$                    = 00000028 R     04
$MT$                    = 00000008 R     04
$OA$                    = 00000054 R     04
$OADRIVER$              = 000000B4 R     05
$OB$                    = 0000004C R     04
$OBDRIVER$              = 000000A2 R     05
$OC$                    = 00000060 R     04
$OCDRIVER$              = 000000CF R     05
$OD$                    = 00000064 R     04
$ODDRIVER$              = 000000D8 R     05
$OE$                    = 00000074 R     04
$OEDRIVER$              = 000000FC R     05
$OF$                    = 00000080 R     04
$OFDRIVER$              = 00000117 R     05
$OG$                    = 00000090 R     04
$OGDRIVER$              = 0000013B R     05
$OH$                    = 000000A1 R     04
$OHDRIVER$              = 0000015F R     05
$OI$                    = 000000A5 R     04
$OIDRIVER$              = 00000168 R     05
$OJ$                    = 000000A9 R     04
$OJDRIVER$              = 00000171 R     05
$OK$                    = 000000AD R     04
$OKDRIVER$              = 0000017A R     05
$OL$                    = 000000B5 R     04
$OLDRIVER$              = 0000018C R     05
$OM$                    = 00000044 R     04
$OMDRIVER$              = 00000090 R     05
$ON$                    = 000000CA R     04
$ONDRIVER$              = 000001BA R     05
$OODRIVER$              = 00000129 R     05
$OQ$                    = 000000E5 R     04
$OQDRIVER$              = 000001F0 R     05
$OR$                    = 0000007C R     04
$ORDRIVER$              = 0000010E R     05
$OSDRIVER$              = 000001CC R     05
$PA$                    = 00000014 R     04
$PADRIVER$              = 0000002D R     05
$PP$                    = 0000005C R     04
$PPDRIVER$              = 000000C6 R     05
$PR$                    = 00000058 R     04
$PRDRIVER$              = 000000BD R     05
$PT$                    = 00000038 R     04
$PU$                    = 00000034 R     04
$PUDRIVER$              = 00000075 R     05
$RSV$                   = 000000B9 R     04
$RSVDRIVER$             = 00000195 R     05
$RTNNAME$               = 00000170 R     07
$TFDRIVER$              = 0000001B R     05
$TMDRIVER$              = 00000012 R     05
$TSDRIVER$              = 0000005A R     05
$TTA$                   = 00000098 R     04
$TX$                    = 000000F1 R     04
$UK$                    = 000000E9 R     04
$UKDRIVER$              = 000001F9 R     05
$VB$                    = 000000E1 R     04
$VBDRIVER$              = 000001E7 R     05
$VC$                    = 000000F9 R     04
$VCDRIVER$              = 0000021D R     05
$XA$                    = 000000BE R     04
$XADRIVER$              = 0000019F R     05
$XB$                    = 000000C2 R     04
$XBDRIVER$              = 000001A8 R     05
$XD$                    = 000000C6 R     04
$XDDRIVER$              = 000001B1 R     05
$XE$                    = 0000003C R     04
$XEDRIVER$              = 0000007E R     05
$XF$                    = 00000010 R     04
$XFDRIVER$              = 00000024 R     05
$XG$                    = 000000FD R     04
$XGDRIVER$              = 00000226 R     05
$XI$                    = 00000101 R     04
$XIDRIVER$              = 00000238 R     05
$XK$                    = 0000009D R     04
$XKDRIVER$              = 00000156 R     05
$XM$                    = 00000094 R     04
$XMDRIVER$              = 00000144 R     05
```

```
$XP$                            = 000000DD R     04
$XPDRIVER$                      = 000001DE R     05
$XQ$                            = 00000040 R     04
$XQDRIVER$                      = 00000087 R     05
$XS$                            = 000000D9 R     04
$XSDRIVER$                      = 000001D5 R     05
$XU$                            = 00000084 R     04
$XUDRIVER$                      = 00000120 R     05
$XV$                            = 0000008C R     04
$XVDRIVER$                      = 00000132 R     05
$XW$                            = 00000088 R     04
$YCDRIVER$                      = 0000022F R     05
$YFDRIVER$                      = 0000020B R     05
$YH$                            = 00000070 R     04
$YHDRIVER$                      = 000000F3 R     05
$YJ$                            = 0000006C R     04
$YJDRIVER$                      = 000000EA R     05
$YL$                            = 00000068 R     04
$YLDRIVER$                      = 000000E1 R     05
$YM$                            = 00000050 R     04
$YMDRIVER$                      = 000000AB R     05
ACF$AB_ADPTYPE                    00000000 R     02
ACF$AB_CITABLE                    00000032 R     02
ACF$AB_CLASS_TABLE                00000228 R     02
ACF$AB_CPI32_TABLE                00000218 R     02
ACF$AB_DMF32_TABLE                000001F4 R     02
ACF$AB_DMZ32_TABLE                00000208 R     02
ACF$AB_DRTABLE                    0000002E R     02
ACF$AB_MBATABLE                   0000001A R     02
ACF$AB_NUMVEC                     00000240 R     02
ACF$AB_UBATABLE                   00000090 RG    02
ACF$ADD_UNITS                     000004E5 R     08
ACF$AL_CPI32_COMBO                00000194 R     02
ACF$AL_CR11_CR                    00000090 R     02
ACF$AL_CSR                        00000238 R     02
ACF$AL_DC11_DM                    000000BC R     02
ACF$AL_DEVNAME                    00000234 R     02
ACF$AL_DH11_YH                    000000E8 R     02
ACF$AL_DHV1T_TX                   0000018C R     02
ACF$AL_DJ11_YJ                    000000E4 R     02
ACF$AL_DL11C_YL                   000000E0 R     02
ACF$AL_DM11B_YM                   000000C8 R     02
ACF$AL_DMC11_XM                   0000010C R     02
ACF$AL_DMF32_COMBO                0000016C R     02
ACF$AL_DMP11_XD                   00000154 R     02
ACF$AL_DMV11_XD                   00000160 R     02
ACF$AL_DMZ32_COMBO                00000190 R     02
ACF$AL_DN11_DB                    000000C4 R     02
ACF$AL_DPV1T_ON                   00000158 R     02
ACF$AL_DQ11_DR                    000000F4 R     02
ACF$AL_DR11B_XB                   00000148 R     02
ACF$AL_DR11C_DA                   000000CC R     02
ACF$AL_DR11W_XA                   00000144 R     02
ACF$AL_DT11_DC                    000000D8 R     02
ACF$AL_DU11_XU                    000000FC R     02
ACF$AL_DUP1T_XW                   00000100 R     02
ACF$AL_DV11_XV                    00000104 R     02
ACF$AL_DWR70_OK                   00000124 R     02
ACF$AL_DX11_DD                    000000DC R     02
ACF$AL_DZ11_TTA                   00000110 R     02
ACF$AL_GT40_OE                    00C000EC R     02
ACF$AL_IEQ1T_IX                   00000188 R     02
ACF$AL_ISB11_IS                   0000015C R     02
ACF$AL_KCT32_UK                   00000184 R     02
ACF$AL_KMC11_XK                   00000114 R     02
ACF$AL_KMS11_XS                   00000170 R     02
ACF$AL_KMV11_QQ                   00000180 R     02
ACF$AL_KW11C_QL                   00000138 R     02
ACF$AL_KW11W_QF                   000000F8 R     02
ACF$AL_LK11_QG                    00000108 R     02
ACF$AL_LP11_LP                    00000098 R     02
ACF$AL_LPA1T_LA                   00000130 R     02
ACF$AL_LPP11_QH                   00000118 R     02
ACF$AL_LPS11_LS                   000000F0 R     02
ACF$AL_PCL11_XP                   00000174 R     02
ACF$AL_PP611_PP                   000000D4 R     02
ACF$AL_PR611_PR                   000000D0 R     02
ACF$AL_QNA_XQ                     000000B8 R     02
ACF$AL_RB730_DQ                   000000A8 R     02
ACF$AL_RESET                      00000000 R     03
ACF$AL_RK611_DM                   00000094 R     02
ACF$AL_RL11_DL                    0000009C R     02
ACF$AL_RSV_RSV                    0000013C R     02
ACF$AL_RX2T1_DY                   000000A4 R     02
ACF$AL_TC11_DT                    00000198 R     02
ACF$AL_TS11_MS                    000000A0 R     02
ACF$AL_TU58_DD                    000000C0 R     02
ACF$AL_TU81_PT                    000000B0 R     02
ACF$AL_UDA_PU                     000000AC R     02
ACF$AL_UNA_XE                     000000B4 R     02
ACF$AL_VCOTB_VC                   0000019C R     02
ACF$AL_VMV21_QI                   0000011C R     02
ACF$AL_VMV31_QJ                   00000120 R     02
ACF$AL_VS100_VB                   00000178 R     02
ACF$AW_SAVEVEC                    0000023E R     02
ACF$AW_VECTOR                     0000023C R     02
ACF$BITMASK                       00000619 R     08
ACF$B_AFLAG                     = 0000000B
ACF$B_AUNIT                     = 0000000A
ACF$B_BOOT_TR                     00000249 R     02
ACF$B_CNUMVEC                   = 0000001E
ACF$B_COMBO_CSR_OFFSET          = 00000020
ACF$B_COMBO_VECTOR_OFFSET       = 0000001F
ACF$B_NUMUNIT                   = 00000021
ACF$CI                            0000024D R     08
ACF$CLR_ACF                       0000078F RG    08
ACF$COMBO_DEVICES                 00000676 R     08
ACF$CPI32                         0000063B R     08
ACF$CR11                          000004E5 R     08
ACF$C_LENGTH                    = 00000028
ACF$DC11                          000004E5 R     08
ACF$DH11                          000004E5 R     08
ACF$DHV11                         000004E5 R     08
ACF$DJ11                          000004E5 R     08
```

```
ACF$DL11C                       000004E5 R      08
ACF$DM11B                       000004E5 R      08
ACF$DMC11                       000004E5 R      08
ACF$DMF32                       0000061B R      08
ACF$DMP11                       000004E5 R      08
ACF$DMV11                       000004E5 R      08
ACF$DMZ32                       00000652 R      08
ACF$DMZCPI                      00000667 R      08
ACF$DN11                        000004E5 R      08
ACF$DPV11                       000004E5 R      08
ACF$DQ11                        000004E5 R      08
ACF$DR                          00000246 R      08
ACF$DR11B                       000004E5 R      08
ACF$DR11C                       000004E5 R      08
ACF$DR11W                       000004E5 R      08
ACF$DT11                        000004E5 R      08
ACF$DU11                        000004E5 R      08
ACF$DUP11                       000004E5 R      08
ACF$DV11                        000004E5 R      08
ACF$DWR70                       000004E5 R      08
ACF$DX11                        000004E5 R      08
ACF$DZ11                        000004E5 R      08
ACF$GL_CRB                      ********    X   08
ACF$GL_DDB                      ********    X   08
ACF$GL_DPT                      ********    X   08
ACF$GL_IDB                      ********    X   08
ACF$GL_LASTDDB                  ********    X   08
ACF$GL_SB                       ********    X   08
ACF$GL_UCB                      ********    X   08
ACF$GT40                        000004E5 R      08
ACF$IEQ11                       000004E5 R      08
ACF$INC_CHAR                    0000074B RG     08
ACF$ISBT1                       000004E5 R      08
ACF$KCT32                       000004E5 R      08
ACF$KMC11                       000004E5 R      08
ACF$KMS11                       000004E5 R      08
ACF$KMV11                       000004E5 R      08
ACF$KW11C                       000004E5 R      08
ACF$KW11W                       000004E5 R      08
ACF$LK11                        000004E5 R      08
ACF$LP11                        000004E5 R      08
ACF$LPA11                       000004E5 R      08
ACF$LPP11                       000004E5 R      08
ACF$LPS11                       000004E5 R      08
ACF$L_ACF_SAVE                  00000046 R      02
ACF$L_ADAPTER                 = 00000000
ACF$L_CONFIGREG               = 00000004
ACF$L_CONTRLREG               = 0000000C
ACF$L_DELIVER_UNIT              00000042 R      02
ACF$L_DEVNAME                 = 00000014
ACF$L_DRVNAME                 = 00000018
ACF$L_ROSAVE                    00000070 R      02
ACF$L_R1SAVE                    00000074 R      02
ACF$L_R2R3SAVE                  00000088 R      02
ACF$L_R2SAVE                    00000078 R      02
ACF$L_R3SAVE                    0000007C R      02
ACF$L_R4SAVE                    00000080 R      02
ACF$L_R5SAVE                    00000084 R      02
ACF$L_RETURN                    00000036 R      02
ACF$L_RETURN2                   0000003A R      02
ACF$L_ROUTINE                   0000003E R      02
ACF$L_SYS_CSR                   00000245 R      02
ACF$MBA                         00000031 R      08
ACF$MBADISK                     000000D3 R      08
ACF$MBATAPE                     0000010C R      08
ACF$M_SUPPORT                 = 00000010
ACF$PCL11                       000004E5 R      08
ACF$PP611                       000004E5 R      08
ACF$PR611                       000004E5 R      08
ACF$QNA                         000004E5 R      08
ACF$RB730                       000004E5 R      08
ACF$RK611                       000004E5 R      08
ACF$RL11                        000005C7 R      08
ACF$RSV                         000004E5 R      08
ACF$RX11                        000004E5 R      08
ACF$RX211                       000004E5 R      08
ACF$SINGLE_DEVICES              00000252 R      08
ACF$TC11                        000004E5 R      08
ACF$TS11                        000004E5 R      08
ACF$TU58                        000004E5 R      08
ACF$TU81                        00000542 R      08
ACF$T_SYS_DEVNAME               0000024A R      02
ACF$UBA                         00000285 R      08
ACF$UBAFIXED                    00000358 R      08
ACF$UBAFLOATING                 000003DE R      08
ACF$UDA                         00000542 R      08
ACF$UNA                         000004E5 R      08
ACF$VC01B                       000004E5 R      08
ACF$VMV21                       000004E5 R      08
ACF$VMV31                       000004E5 R      08
ACF$VS100                       000004E5 R      08
ACF$V_NOLOAD_DB               = 00000003
ACF$W_AVECTOR                 = 00000008
ACF$W_CSRBASE                   00000241 R      02
ACF$W_CUNIT                   = 00000012
ACF$W_CVECTOR                 = 00000010
ACF$W_MAXUNITS                = 0000001C
ACF$W_VECBASE                   00000243 R      02
ACF$W_VECMOD                    0000006E R      02
ADAPTERLEN                    = 00000006
ADP$B_NUMBER                  = 0000000B
ADP$L_AVECTOR                 = 0000001C
ADP$W_ADPTYPE                 = 0000000E
ADP$W_TR                      = 0000000C
AT$_CI                        = 00000004
AT$_DR                        = 00000002
AT$_MBA                       = 00000000
AT$_UBA                       = 00000001
BOO$GQ_CONSYSID                 ********    X   08
CLS$L_CLASSDEV                = 00000000
CLS$L_CLASSDRV                = 00000004
CLS$T_PORTDEV                 = 00000009
CPI$S_SUBCNTRL                = 00000004
CPI$V_SUBCNTRL                = 00000008
```

```
CRB$L_INTD                      = 00000024                      UCB$L_PDT                       = 00000084
DDB$T_NAME                      = 00000014                      VEC$L_ADP                       = 00000014
DMF$S_IDENT                     = 00000004                      VEC$L_IDB                       = 00000008
DMF$V_IDENT                     = 0000000C
DPT$W_DEFUNITS                  = 0000001A
DPT$W_DELIVER                   = 0000001C
EXE$GL_RPB                        ********    X   08
EXE$GL_SCB                        ********    X   08
EXE$GL_TENUSEC                    ********    X   08
EXE$TEST_CSR                      ********    X   08
FIX_DEV_NAME                      0000047E  R     08
IDB$L_CSR                       = 00000000
IOC$AUTOCONFIG                    00000000  RG    08
IOC$AUTORESET                     00000705  RG    08
LOAD_DRIVER                       000004C1  R     08
MBA$L_ERB                       = 00000400
MBA$L_SR                        = 00000008
MBA$M_SR_ATTN                   = 00010000
MBA$V_SR_NED                    = 00000012
MBA_DS                          = 00000004
MBA_DT                          = 00000018
PDT$L_UCBO                      = 000000DC
RL_CS                           = 00000000
RL_CS_M_OPI                     = 00000400
RL_DA                           = 00000004
RL_DA_M_MRK                     = 00000001
RL_DA_M_RST                     = 00000008
RL_DA_M_STS                     = 00000002
RPB$L_BOOTR1                    = 00000020
SYS$GL_BOOTUCB                    ********    X   08
TM03_TC                         = 00000024
TM78_AB                         = 00000010
TM78_DS                         = 0000001C
TM78_ID                         = 00000044
TM78_M_TMCLR                    = 00004000
TM78_M_TMRDY                    = 00008000
TM78_NDTO                       = 00000030
TM78_NDTA                       = 0000002C
TM78_SENSE_GO                   = 00000009
UBA_IOBASE                      = 00001000
UBA_M_FLOATCSR                  = 00000002
UBA_M_FLOATVEC                  = 00000004
UBA_M_SUPPORT                   = 00000001
UBA_V_FLOATCSR                  = 00000001
UBA_V_FLOATVEC                  = 00000002
UBA_V_SUPPORT                   = 00000000
UBT$B_FLAGS                     = 00000013  G
UBT$B_LETTER                    = 00000010  G
UBT$B_NUMVEC                    = 00000011  G
UBT$B_UNUSED                    = 00000012  G
UBT$L_DEVNAME                   = 00000000  G
UBT$L_DRVNAME                   = 00000004  G
UBT$L_ROUTINE                   = 0000000C  G
UBT$L_RTNNAME                   = 00000008  G
UBT$W_REMAINDER                 = 00000014  G
UCB$L_CRB                       = 00000024
UCB$L_DDB                       = 00000028
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	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
_255DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	12
_255DUA28:[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
```

BOOTS
MAR

AUTOCONFG
LIS

WRITEBOOT
MAP

BOOCVTATB
LIS

VMB
MAP

SYSGEN
MAP

ACTIMAGE
LIS

BOOT58
LIS

BOOTDEF
SDL