```
SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE DLL
OBJECT MODULE PLACED IN :F1:DLL.OBJ
INVOCATION LINE CONTROLS:  DEBUG


LOC  OBJ                  LINE     SOURCE

                             1 +1  $TITLE ('DLL - DATA LINK LAYER 04/20/82')
                             2           NAME    DLL
                             3
                             4     ;;;     Intel Corporation Proprietary Information. This
                             5     ;       listing is supplied under the terms of a license
                             6     ;       agreement with Intel Corporation and may not be copied
                             7     ;       nor disclosed except in accordance with the terms of
                             8     ;       the agreement.
                             9
                            10 +1  $NOGEN
                            11     $INCLUDE (:F1:PORTS.INC)
                     =1     12
                     =1     13     ;       CONTOL PORTS
                     =1     14
  0000               =1     15     SET_TXSRT         EQU     000H
  0001               =1     16     RESET_TXSRT       EQU     001H
  0002               =1     17     SET_RXAV1         EQU     002H
  0003               =1     18     SET_RXAV2         EQU     003H
  0004               =1     19     SET_RXAV3         EQU     004H
  0005               =1     20     RESET_ERROR       EQU     005H
  0006               =1     21     RESET_CHANNEL_COUNTER   EQU     006H
  00A0               =1     22     SET_SYS           EQU     0A0H
  00B0               =1     23     SET_LOC           EQU     0B0H
                     =1     24
                     =1     25
                     =1     26     ;       PIT VALUES
                     =1     27
  00D3               =1     28     PITCMD            EQU     0D3H            ;PIT COMMAND PORT
  00D2               =1     29     PIT_BCK           EQU     0D2H            ;PIT BACKOFF TIMER
  00D1               =1     30     PIT_RTC           EQU     0D1H            ;PIT REAL-TIME CLOCK PORT
  00D0               =1     31     PIT_ALC           EQU     0D0H            ;PIT ALARM-CLOCK PORT
                     =1     32
  0000               =1     33     LATCH_RTC         EQU     00H             ;LATCH REAL-TIME CLOCK VALUE
  0040               =1     34     LATCH_ALC         EQU     40H             ;LATCH ALARM-CLOCK VALUE
                     =1     35
                     =1     36
                     =1     37     ;       DMA VALUES
                     =1     38
  00C8               =1     39     DMACMD            EQU     0C8H            ;DMA COMMAND AND STATUS PORT
  00C9               =1     40     DMAREQ            EQU     0C9H            ;DMA REQUEST PORT
  00CA               =1     41     DMAMSKB           EQU     0CAH            ;DMA MASK BIT PORT
  00CB               =1     42     DMAMODE           EQU     0CBH            ;DMA MODE PORT
  00CC               =1     43     DMABPTR           EQU     0CCH            ;DMA BYTE POINTER PORT
  00CD               =1     44     DMACLR            EQU     0CDH            ;? (TMP & CLEAR??)
  00CF               =1     45     DMAMASK           EQU     0CFH            ;DMA MASK PORT
                     =1     46
  00C0               =1     47     CH0ADDR           EQU     0C0H            ;CHANNEL 0 ADDRESS PORT
  00C1               =1     48     CH0WC             EQU     0C1H            ;CHANNEL 0 WORD COUNT PORT
  00C2               =1     49     CH1ADDR           EQU     0C2H            ;CHANNEL 1 ADDRESS PORT
  00C3               =1     50     CH1WC             EQU     0C3H            ;CHANNEL 1 WORD COUNT PORT
```

```
LOC  OBJ                  LINE     SOURCE

  00C4                =1    51     CH2ADDR         EQU     0C4H            ;CHANNEL 2 ADDRESS PORT
  00C5                =1    52     CH2WC           EQU     0C5H            ;CHANNEL 2 WORD COUNT PORT
  00C6                =1    53     CH3ADDR         EQU     0C6H            ;CHANNEL 3 ADDRESS PORT
  00C7                =1    54     CH3WC           EQU     0C7H            ;CHANNEL 3 WORD COUNT PORT
                      =1    55
                      =1    56
                      =1    57     ;       PIO VALUES
                      =1    58
  00E3                =1    59     PIOCMD          EQU     0E3H            ;PIO COMMAND PORT (CHECK: F3?)
  00E0                =1    60     PIOA            EQU     0E0H            ;PIO PORT A
  00E1                =1    61     PIOB            EQU     0E1H            ;PIO PORT B
  00E2                =1    62     PIOC            EQU     0E2H            ;PIO PORT C
                      =1    63
  00F3                =1    64     PIOCLR          EQU     11110011B       ;CLEAR SERDES
  002A                =1    65     PIOSEN          EQU     00101010B       ;SERIAL ENABLE
  0022                =1    66     PIOSENP         EQU     00100010B       ;PROMISCUOUS SERIAL ENABLE
  00EC                =1    67     PIOREAD         EQU     11100000B       ;READ ADDRESS COMMAND
                      =1    68
                      =1    69
                      =1    70     ;       PIC VALUES
                      =1    71
  00F0                =1    72     PICCMD          EQU     0F0H            ;PIC COMMAND PORT
  00F0                =1    73     PICDATA         EQU     0F0H            ;PIC DATA PORT
  00F1                =1    74     PICMASK         EQU     0F1H            ;PIC MASK PORT
                      =1    75
  0020                =1    76     EOI_PIC         EQU     20H             ;PIC END-OF-INTERRUPT COMMAND
  0060                =1    77     SEOI_PIC        EQU     60H             ;PIC SELECTIVE EOI COMMAND
  00CC                =1    78     POLL_PIC        EQU     0CH             ;POLL PIC COMMAND
  000A                =1    79     READ_IRR        EQU     0AH             ;PIC READ-IRR COMMAND
  0040                =1    80     RTC_DONE        EQU     40H             ;MASK FOR REAL-TIME CLOCK INTERRUPT
  0066                =1    81     RTC_INT_SEOI    EQU     60H+6           ;COMMAND TO EOI RTC INTERRUPT
  0086                =1    82     RTC_INT         EQU     80H+6           ;POLL COMMAND RETURN IF INTERRUPT
                      =1    83
  0001                =1    84     CH1_DONE        EQU     01H             ;RX CHANNEL 1 DONE
  0002                =1    85     CH2_DONE        EQU     02H             ;RX CHANNEL 2 DONE
  0004                =1    86     CH3_DONE        EQU     04H             ;RX CHANNEL 3 DONE
                      =1    87
                      =1    88
                      =1    89     ;       MISC. DEFINITIONS
                      =1    90
  0001                =1    91     MCFLAG          EQU     01H             ;MULTICAST BIT
  0007                =1    92     JUNK_BYTES      EQU     7               ;NUMBER OF RECIEVE JUNK BYTES
                        93     $INCLUDE (:F1:KAOS.DCA)
                      =1    94             EXTRN   CQSTART:NEAR,CQSCHEDULE:NEAR,CQHALTANDCATCHFIRE:NEAR
                      =1    95             EXTRN   CQCREATEPROCESS:NEAR,CQCREATELIST:NEAR
                      =1    96             EXTRN   CQCREATESEMAPHORE:NEAR,CQSIGNAL:NEAR,CQWAITSEM:NEAR
                      =1    97             EXTRN   CQCWAIT:NEAR
                      =1    98             EXTRN   CQCREATEMAILBOX:NEAR,CQSEND:NEAR,CQRECEIVE:NEAR,CQCRECEIVE:NEAR
                      =1    99             EXTRN   CQISIGNAL:NEAR,CQICWAIT:NEAR,CQISEND:NEAR,CQICRECEIVE:NEAR
                      =1   100             EXTRN   CQREADCLOCK:NEAR
                      =1   101             EXTRN   CQCREATEALARM:NEAR,CQSETALARM:NEAR
                      =1   102             EXTRN   CQCHECKALARM:NEAR,CQCLEARALARM:NEAR
                       103     $NOLIST INCLUDE (:F1:LOCGEN.MAC)
                       202
                       203
```

DLL

```
LOC  OBJ                  LINE      SOURCE

                           204              EXTRN   RANDOM:NEAR     ;A KAOS PRIMITIVE FOR BACKOFF NUMBER
                           205
                           206      $EJECT TITLE ('DLL - DATA BASES')
```

```
LOC  OBJ                  LINE     SOURCE

                           207     DGROUP  GROUP   DATA,STACKS,BUFFERS
                           208
----                       209     DATA    SEGMENT BYTE PUBLIC 'DATA'
                           210
                           211     ;       CONSTANTS
                           212
  0040                     213     SLOT_TIME       EQU     64      ;Number of 800nS clock ticks per slot time.
  003C                     214     MIN_PACKET_LEN  EQU     60      ;60 bytes of header and data.
  05EA                     215     MAX_PACKET_LEN  EQU     14+1500 ;14 byte header + 1500 bytes data.
  05FA                     216     SEG_BUF_LEN     EQU     6+MAX_PACKET_LEN+JUNK_BYTES+3  ;OS header
                           217                                     ; + packet length + garbage + safety factor.
  0001                     218     NUM_TX_BUF      EQU     1       ;Number of transmit buffers.
  0004                     219     NUM_RX_BUF      EQU     4       ;Number of receive buffers.
  000F                     220     ATTEMPT_LIMIT   EQU     15      ;Maximum number of transmit retries on collision
  03FF                     221     BACKOFF_LIMIT   EQU     3FFH    ;Maximum 10 bits (1023) for random number
                           222                                     ;in backoff alogorithm.
                           223
                           224
                           225     ;       PACKET BUFFER FORMAT
                           226
----                       227     SEGMNT  STRUC
0000                       228     KAOSMSGHDR      DD      ?       ;Space for the KAOS header when linked to a
                           229                                     ;mailbox.
                           230
0004                       231     SEG_LENGTH      DW      ?       ;The number of bytes from DL_DEST to the end
                           232                                     ;of the data in the DL_DATA field.
                           233
0006                       234     DL_DEST         DW      3 DUP (?) ;The data link destination address field.
                           235
000C                       236     DL_SOURCE       DW      3 DUP (?) ;The data link source address field.
                           237
0012                       238     DL_TYPE         DW      ?       ;The data link type field.
                           239
0014                       240     DL_DATA         DB      ?       ;The data link data field;  this must be at
                           241                                     ;least 46 bytes and not more than 1500 bytes.
                           242
----                       243     SEGMNT  ENDS
                           244
  0006                     245     VAL_DL_DEST     EQU     06H     ;An alternate representation of DL_DEST because
                           246                                     ;ASM86 doesn't allow structure members in some
                           247                                     ;sorts of expressions where is does allow
                           248                                     ;EQU-type constants.
                           249
                           250
                           251
                           252     ;       KAOS OBJECTS
                           253
                           254     ;       This section uses a version of "SYSGEN.MAC" modified
                           255     ;       to not declare things external.  This was done so that we could put
                           256     ;       the entire data link in one module, rather than splitting the
                           257     ;       definition of the KAOS objects out into a separate module.
                           258
                           259
                           260             PUBLIC  DLLTxFreeMbx
                           261             %OBJECTLIST (KAOSOBJECTS)
```

```
LOC  OBJ                  LINE     SOURCE

----CO                     266
                           267          %MAILBOX (DLLTxFreeMbx) ;The Transmit Free buffer pool;
----                  R    275                                  ;this is used to store buffers which are
                           276                                  ;available to the user.
                           277
                           278          %MAILBOX (DLL_MBX1)     ;Packets are normally sent by CQDLL$TRANSMIT
----                  R    286                                  ;without involving the operating operating
                           287                                  ;system.  If there is already a packet being
                           288                                  ;sent, however, the new packet will be stored
                           289                                  ;here.  The transmit interrupt service routine,
                           290                                  ;TXISR,  does a CReceive at this mailbox
                           291                                  ;and reintializes the transmit hardware
                           292                                  ;if anything is here.
                           293
                           294          %MAILBOX (DLL_MBX2)     ;CQDLL$TRANSMIT does local loopback on
----                  R    302                                  ;self-self addressed packets by calling
                           303                                  ;LOOP_PACKET, which copys the packet to a
                           304                                  ;receive  buffer and forwards it to the normal
                           305                                  ;received packet processing.
                           306                                  ;If there is no free receive buffer,
                           307                                  ;LOOP_PACKET increments BUF_LOOP and blocks
                           308                                  ;at this mailbox waiting for a buffer.
                           309                                  ;Note that a free buffer is one which is
                           310                                  ;neither in the hands of a user, nor
                           311                                  ;initialized on a receive DMA channel.
                           312                                  ;Since there are three receive DMA channels,
                           313                                  ;and only four receive buffers at this time,
                           314                                  ;there is only one buffer available for
                           315                                  ;loopback
                           316
                           317          %SEMAPHORE (DLL_SEM1,0) ;This semaphore is signalled by the
----                  R    326                                  ;receive interrupt service routine to
                           327                                  ;start up the receive DLLRXF process.
                           328
                           329          %PROCESS (0,DLLRXF,84 ) ;The Receive Forward process.
----5E                R    344
                           345          %ENDLIST
----                       350
                           351     $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           352     ;       BEGINNING OF THE MAIN DATA BASE
                           353
  0046                     354     DB_START        EQU     $       ;All memory between DB_START and DB_END
                           355                                     ;will be initialized to zero by CQDLL$START.
                           356
                           357
                           358
                           359     ;       NETWORK MANAGEMENT COUNTERS AND OTHER OBJECTS
                           360
                           361
C046 (2                    362     TOTAL_SENT      DW      2 DUP (?)  ;The total number of packets which have
     ????
     )
                           363                                     ;been sent by this node.  This two-word
                           364                                     ;counter wraps around to zero after reaching
                           365                                     ;its maximum value.  At the maximum
                           366                                     ;transmission rate, this will take weeks
                           367                                     ;to wrap around.
                           368
004A ????                  369     PRIMARY_COLL    DW      ?       ;The number of packets which have encountered
                           370                                     ;at least one collision.  It sticks at OFFFFH.
                           371
004C ????                  372     SECONDARY_COLL  DW      ?       ;Then number of times there has been a second
                           373                                     ;or higher collision.  If a packet has six
                           374                                     ;collisions, PRIMARY_COLL will be incremented
                           375                                     ;by one, and SECONDARY_COLL will be incremented
                           376                                     ;by five.  It sticks at OFFFFH.
                           377
004E ????                  378     EXCEEDED_COLL   DW      ?       ;The number of packets which have been aborted
                           379                                     ;becuase they had mode than the max
                           380                                     ;number of collisions.  It sticks at OFFFFH.
                           381
0050 ????                  382     TX_PKT_TOO_LONG DW      ?       ;The number of times the transmit hardware
                           383                                     ;watchdog time has aborted a packet.
                           384                                     ;This should never occur unless there is
                           385                                     ;a serioussoftware or controller hardware
                           386                                     ; failure.  It sticks at OFFFFH.
                           387
                           388
0052 (2                    389     TOTAL_RECEIVED  DW      2 DUP (?)  ;The total number of error-free packets
     ????
     )
                           390                                     ;which were addressed to either this node's
                           391                                     ;hostID or some multicast address it has
                           392                                     ;enabled,  and for which some user had
                           393                                     ;done a CQDLL$CONNECT for its type code.
                           394
0056 ????                  395     CRCERRS         DW      ?       ;The number of received packets discarded
                           396                                     ;because they had a CRC error.  It sticks
                           397                                     ;at OFFFFH.
                           398
0058 ????                  399     FRMERRS         DW      ?       ;The number of received packets discarded
                           400                                     ;because they were longer than MAX_PACKET_LEN.
                           401                                     ;It sticks at OFFFFH.
                           402
```

```
LOC  OBJ                  LINE     SOURCE

005A ????                  403     RSCERRS         DW      ?       ;The number of times the receive interrupt
                           404                                     ;routine has tried to re-initialize a channel
                           405                                     ;but could not get a buffer.  This gives
                           406                                     ;some idea of how many packets have been
                           407                                     ;dropped becuase there was no buffer to
                           408                                     ;put them in.  It sticks at OFFFFH.
                           409
005C (3                    410     HOSTID          DW      3 DUP (?)  ;This node's HostID
     ????
     )
  005C                     411     CQ_DLL_HOSTID   EQU     HOSTID
                           412             PUBLIC  CQ_DLL_HOSTID
                           413
0062 (2                    414     LOADING         DW      2 DUP (?)  ;The Ether Loading statistic.  It is a
     ????
     )
                           415                                     ;binary fraction in the range 0 <= LOADING < 1
                           416                                     ;representing the fraction of the time
                           417                                     ;the Ethernet cable is busy.  It is a moving
                           418                                     ;average which is updated each time the
                           419                                     ;routine DLLETHERLOAD is called.  Since
                           420                                     ;each sample is a busy or not busy value,
                           421                                     ;it is necessary to average over many samples
                           422                                     ;to get a meaningful value.  The time constant
                           423                                     ;of this statistic is therefore quite long,
                           424                                     ;on the rough order of a minute.  Only the
                           425                                     ;high order word of this double precision
                           426                                     ;number is available to the user;  the lower
                           427                                     ;word is needed to keep round-off error under
                           428                                     ;control.
                           429
                           430
                           431     ;       TRANSMIT DATA BASE (current xmit)
                           432
0066 ????                  433     CURRENT_PACKET  DW      ?       ;The offset of the packet currently being sent.
                           434
0068 ??                    435     XMITBUSY        DB      ?       ;This is set to a nonzero value if a
                           436                                     ;transmission is pending and will be
                           437                                     ;reset by the TXISR interrupt service routine.
                           438
0069 ??                    439     BACKOFF_WAIT    DB      ?       ;This is set to a nonzero value if waiting
                           440                                     ;for a backoff timer interrupt.  This is
                           441                                     ;needed becuase the 8259A has a bug which
                           442                                     ;can sometimes cause a false interrupt 7.
                           443                                     ;If this flag is not set, BACKOFFISR will
                           444                                     ;ignore level 7 interrupts.
                           445
006A ??                    446     BACKOFF_COUNT   DB      ?       ;The number of collisions this packet has
                           447                                     ;encountered.
                           448
006B ????                  449     BACKOFF_MASK    DW      ?       ;This is a mask used by the backoff alogrithm
                           450                                     ;to select the bottom N bits of the value
                           451                                     ;returned by RANDOM, where N is the minimum
                           452                                     ;of BACKOFF_COUNT and 10.
                           453
```

```
LOC  OBJ                  LINE     SOURCE

                           454     ;       RECEIVE DATA BASE
                           455     ;
                           456     ;       The receive side is designed to minimize the probability
                           457     ;       of dropping packets.  The receive interrupt service routines
                           458     ;       reinitialize the DMA channels as quickly as possible
                           459     ;       so that a quick burst of incoming packets up to the size of the
                           460     ;       receive buffer pool can be received without losing packets.
                           461     ;       Since SERCES DMA locks up the buffer RAM bus, access to this memory
                           462     ;       is eliminated during receive bursts by storing all the information
                           463     ;       about a buffer in a Buffer Control Block (BCB) in dynamic RAM.
                           464     ;       Also, the amount of processing done in the interrupt routine
                           465     ;       is minimized to just saving information data and resetting the hardware
                           466     ;       so the DMA channels can be reinitialized as fast as possible;  the
                           467     ;       rest of the per packet processing, such as address filtering, type
                           468     ;       code processing and anything which accesses the packet buffer,
                           469     ;       is done in the background by the DLLRXF process.  This optimizes
                           470     ;       for minimun packet loss at the cost of additional total processing
                           471     ;       time.
                           472     ;
                           473     ;       NOTE:  This strategy made a lot of sense at design time, when packets
                           474     ;       buffers were 560 bytes long, giving us 14 receive buffers.  Now that
                           475     ;       the Ethernet maximum packet size has been increased to 1500 data
                           476     ;       bytes, we only have for room four receive buffers;  optimizing for
                           477     ;       fast reinitialization doesn't make much sense any more.
                           478     ;       There is also a possible glitch in the original scheme:  there
                           479     ;       is an unconfirmed report that accesses to the DMA controller can
                           480     ;       also lock up the processor,  so the careful avoidance of the DMA
                           481     ;       bus with the BCBs may be for naught.  Ripping all this out and
                           482     ;       integrating DLLRFX with the receive interrupt routines is a reasonable
                           483     ;       thing to do under these new conditions.
                           484
                           485
                           486
006D ????                  487     NEXT_CH         DW      ?       ;The address of the receive interrupt service
                           488                                     ;routine for the next channel to be serviced.
                           489                                     ;All three channels vector through the same
                           490                                     ;state save and restore routine, RXISR,
                           491                                     ;which vectors through NEXT_CH to the
                           492                                     ;correct routine
                           493
                           494
----                       495     BCB_TYPE        STRUC           ;This is the structure of the Buffer
                           496                                     ;Control Blocks (BCBs).
                           497
0000                       498     LINK            DW      ?       ;Link to the next element in whatever
                           499                                     ;list this BCB is linked into.
                           500
0002                       501     WC              DW      ?       ;The receive interrupt routines place the
                           502                                     ;word count read from the DMA controller
                           503                                     ;directly in this word, without processing it.
                           504                                     ;If this BCB were put into a KAOS mailbox,
                           505                                     ;it would contain part of the KAOS link.
                           506                                     ;I don't think the current version ever sends
                           507                                     ;a BCB to a mailbox, however.
                           508
```

```
LOC  OBJ                  LINE     SOURCE

C004                       509     BUF             DW      ?       ;The offset of the buffer into the static.
                           510                                     ;This number is needed because the DMA
                           511                                     ;controller thinks that the first byte
                           512                                     ;of static memory is address zero.
                           513
----                       514     BCB_TYPE        ENDS
  00C6                     515     BCB_TYPE_LEN    EQU     6       ;The length in bytes of this structure
                           516                                     ;for initializing the data base in CQDLL$START.
                           517
C06F (24                   518     BCB             DB      NUM_RX_BUF*BCB_TYPE_LEN DUP (?) ;Space for the BCBs.
     ??
     )
                           519
                           520
                           521
C087 ????                  522     BCB_HEAD        DW      ?       ;The head pointer for a stack containing
                           523                                     ;the free (BCBs).
                           524
C089 ????                  525     FREE_HEAD       DW      ?       ;The head pointer for a stack containing
                           526                                     ;BCBs which point to free receive buffers.
                           527
C08B ????                  528     RCVD_HEAD       DW      ?       ;The head and tail pointers to a FIFO queue of
C08D ????                  529     RCVD_TAIL       DW      ?       ;BCBs which point to receive buffers containing
                           530                                     ;packets which have just been received.  This
                           531                                     ;queue is filled by the receive interrupt
                           532                                     ;service routines, and emptied by the
                           533                                     ;receive forward process.
                           534                                     ;IMPORTANT:  RCVD_TAIL must be initialized
                           535                                     ;to point to RCVD_HEAD.
                           536
C08F ????                  537     RX1_BUF         DW      ?       ;The addresses of the buffers currently
C091 ????                  538     RX2_BUF         DW      ?       ;initialized on the receive DMA channels.
C093 ????                  539     RX3_BUF         DW      ?       ;WARNING: These variables MUST remain in
                           540                                     ;sequence because some set-up routines
                           541                                     ;index into this as an array.
                           542
C095 ??                    543     BUF_LOOP        DB      ?       ;The number of loopback buffers needed by the
                           544                                     ;transmit side.
                           545
0096 ??                    546     BUF_FIRST       DB      ?       ;The number of the first receive channel which
                           547                                     ;needs to be reinitialized with a buffer.
                           548
C097 ??                    549     BUF_COUNT       DB      ?       ;The number of receive channels which need
                           550                                     ;need to be reinitialized with a buffer.
                           551
                           552
                           553     ;       MULTICAST ID TABLE
                           554
C098 (1                    555     ALLMC   DB      1 DUP (?)       ;True if all multicast IDs are to be accepted.
     ??
     )
C099 (1                    556     MCIDL   DW      1 DUP (?)       ;The number of multicast IDs in the table.
     ????
     )
009B (24                   557     MCID    DW      8*3 DUP (?)     ;The table of multicast IDs to accept.
```

buffer ptrs pool

```
LOC  OBJ                  LINE     SOURCE

     ????
     )
                          558
                          559
                          560      ;       DATA LINK TYPE DEMUX TABLE
                          561
COCB (1                   562      TYPEL   DW      1 DUP (?)       ;The number of types in the table.
     ????
     )
COCD (8                   563      TYPET   DW      8 DUP (?)       ;The table of types.
     ????
     )
CODD (8                   564      TYPEM   DW      8 DUP (?)       ;The mailboxes to which to send packets of the
     ????
     )
                          565                                      ;corresponding type.
                          566
COED (1                   567      RCV_HANG        DB      1 DUP (?)  ; NUMBER OF TIMES TX DETECTS RCVER HUNG
     ??
     )
                          568
                          569      ;       END OF THE INITIALIZED DATA BASE
                          570
  00EE                    571      DB_END  EQU     $               ;End plus one of the initialized data base;
                          572                                      ;all memory between DB_START and here are
                          573                                      ;cleared to zero by CGDLL$START.
                          574
                          575
----                      576      DATA    ENDS
                          577
                          578
                          579      ;       DEFINITION OF BUFFER SEGMENT
                          580
----                      581      BUFFERS SEGMENT COMMON 'BUFFERS'
C000 (1                   582      BUFFER_START    DW      1 DUP (?) ;Definition of the buffer segment.
     ????
     )
----                      583      BUFFERS ENDS
                          584
                          585
                          586      ;       BEGINNING OF CODE SEGMENT
                          587
                          588      CGROUP  GROUP   CODE
----                      589      CODE    SEGMENT BYTE PUBLIC 'CODE'
                          590              ASSUME  DS:DGROUP,CS:CGROUP
                          591
0018 ----           R     592      DGRP    DW      DGROUP          ;OFFSET OF DGROUP FOR INITIALIZING DS
                          593
                          594      $EJECT TITLE ('DLL - NETWORK MANAGEMENT ROUTINES')
```

```
LOC  OBJ                  LINE     SOURCE

                          595      ;;;       The Network Management routines are driven off a table with
                          596      ;       the following structure:
                          597      ;
                          598      ;       Byte C:
                          599      ;          Bits 0-2: The length in bytes of the object
                          600      ;          Bit 3:    Set if the object is clearable
                          601      ;          Bit 4:    Set if the object is setable.
                          602      ;       Bytes 1,2:   The base address in the current data segment of the
                          603      ;                    first or only instance of the object.
                          604      ;
                          605      ;          In addition, the following constant defines the limits
                          606      ;       of the data structure:
                          607      ;
                          608      ;       LAST_OBJECT: The number of the last object in the table.
                          609
                          610
   #                      611      CTL     RECORD  CLR:1=0,SET:1=0,LEN:3=2
----                      612      OBJ     STRUC
0000                      613      CTL_FLD DB      ?
0001                      614      LOC     DW      ?
----                      615      OBJ     ENDS
                          616
001A 14           R       617      TABLE   OBJ     <CTL<1,,4>,OFFSET DGROUP:TOTAL_SENT>
001B 4600
001D 12           R       618              OBJ     <CTL<1>,OFFSET DGROUP:PRIMARY_COLL>
001E 4A00
0020 12           R       619              OBJ     <CTL<1>,OFFSET DGROUP:SECONDARY_COLL>
0021 4C00
0023 12           R       620              OBJ     <CTL<1>,OFFSET DGROUP:EXCEEDED_COLL>
0024 4E00
0026 12           R       621              OBJ     <CTL<1>,OFFSET DGROUP:TX_PKT_TOO_LONG>
0027 5000
0029 14           R       622              OBJ     <CTL<1,,4>,OFFSET DGROUP:TOTAL_RECEIVED>
002A 5200
002C 12           R       623              OBJ     <CTL<1>,OFFSET DGROUP:CRCERRS>
002D 5600
002F 12           R       624              OBJ     <CTL<1>,OFFSET DGROUP:FRMERRS>
0030 5800
0032 12           R       625              OBJ     <CTL<1>,OFFSET DGROUP:RSCERRS>
0033 5A00
0035 06           R       626              OBJ     <CTL<,,6>,OFFSET DGROUP:HOSTID>
0036 5C00
0038 02           R       627              OBJ     <CTL<,,2>,OFFSET DGROUP:LOADING+2>
0039 6400
                          628
  000A                    629      LAST_OBJECT     EQU     10
                          630
                          631
                          632
                          633
                          634      ;;;     DLLSREAD (OBJECT, MODIFIER, VALUE$P) - READ NETWORK MANAGEMENT OBJECT.
                          635      ;
                          636      ;       PARAMETERS:
                          637      ;          OBJECT = THE OBJECT NUMBER (WORD).
                          638      ;          MODIFIER = WHICH COPY THE OF THE OBJECT (WORD, INGORED).
```

```
LOC  OBJ                  LINE     SOURCE

                           639     ;          VALUESP = POINTER TO A BUFFER TO RETURN THE VALUE IN (WORD).
                           640     ;
                           641     ;       RETURNS:
                           642     ;          THE BUFFER POINTED TO BY VALUE$P CONTAINS THE CURRENT VALUE OF
                           643     ;             THE OBJECT.
                           644     ;          AX = THE LENGTH IN BYTES OF THE OBJECT.
                           645     ;          AX = 0 IF ILLEGAL PARAMETER.
                           646
                           647
                           648             PUBLIC  DLLREAD
0038                       649     DLLREAD:
0038 E81E00                650             CALL    SERVIC
003E F3                    651             REP     MOVSB
003F A4
0040 8BC3                  652             MOV     AX,BX
0042 C3                    653             RET
                           654
                           655
                           656
                           657
                           658     ;;;     DLL$READC (OBJECT, MODIFIER, VALUE$P) - READ AND CLEAR
                           659     ;          NETWORK MANAGEMENT OBJECT.
                           660     ;
                           661     ;       PARAMETERS:
                           662     ;          OBJECT = THE OBJECT NUMBER (WORD).
                           663     ;          MODIFIER = WHICH COPY THE OF THE OBJECT (WORD, IGNORED).
                           664     ;          VALUESP = POINTER TO A BUFFER TO RETURN THE VALUE IN (WORD).
                           665     ;
                           666     ;       RETURNS:
                           667     ;          THE BUFFER POINTED TO BY VALUE$P CONTAINS THE CURRENT VALUE OF
                           668     ;             THE OBJECT.
                           669     ;          THE OBJECT HAS BEEN CLEARED TO 0 IF A CLEARABLE OBJECT.
                           670     ;          AX = THE LENGTH IN BYTES OF THE OBJECT.
                           671     ;          AX = 0 IF ILLEGAL PARAMETER.
                           672
                           673
                           674             PUBLIC  DLLREADC
0043                       675     DLLREADC:
0043 E81600                676             CALL    SERVIC
0046 F6C210                677             TEST    DL,MASK CLR     ;TEST IF OBJECT IS CLEARABLE
0049 7407                  678             JZ      RDC2            ;IF NOT CLEARABLE
004B A4                    679     RDC1:   MOVSB
004C C644FF00              680             MOV     BYTE PTR DS:[SI-1],0
0050 E2F9                  681             LOOP    RDC1
0052 F3                    682     RDC2:   REP     MOVSB           ;THIS WILL EXECUTE ONLY IF FROM BRANCH
0053 A4
0054 8BC3                  683             MOV     AX,BX
0056 C3                    684             RET
                           685
                           686
                           687
                           688
                           689     ;;;     DLL$SET (OBJECT, MODIFIER, VALUE$P) - SET NETWORK MANAGEMENT OBJECT.
                           690     ;
                           691     ;       PARAMETERS:
```

```
LOC  OBJ                  LINE     SOURCE

                           692     ;           OBJECT = THE OBJECT NUMBER (WORD).
                           693     ;           MODIFIER = WHICH COPY THE OF THE OBJECT (WORD, IGNORED).
                           694     ;           VALUE$P = POINTER TO BUFFER CONTAINING NEW VALUE FOR OBJECT (WORD).
                           695     ;
                           696     ;       RETURNS:
                           697     ;           AX = THE LENGTH IN BYTES OF THE OBJECT.
                           698     ;           AX = 0 IF ILLEGAL PARAMETER.
                           699
                           700
                           701             PUBLIC  DLLSET
0057 33C0                  702     DLLSET: XOR     AX,AX           ;NO SETABLE PARAMETERS; RETURN ZERO
0059 C20800                703             RET     8
                           704
                           705
                           706
                           707     ;;      SERVIC - DATA INTERFACE SERVICE ROUTINE
                           708     ;
                           709     ;       RETURNS:
                           710     ;           (BX) = (CX) = OBJECT LENGTH.
                           711     ;           (SI) = POINTER TO OBJECT.
                           712     ;           (DI) = POINTER TO VALUE BUFFER.
                           713     ;           (DL) = CTL BYTE FOR OBJECT.
                           714
                           715
005C 58                    716     SERVIC: POP     AX              ;MY RETURN ADDRESS
005D 5E                    717             POP     SI              ;CALLER'S RETURN ADDRESS
005E 5F                    718             POP     DI              ;VALUE POINTER
005F 07                    719             POP     ES
0060 5A                    720             POP     DX              ;MODIFIER
0061 5B                    721             POP     BX              ;OBJECT
0062 56                    722             PUSH    SI              ;RESTORE RETURN ADDRESSES
0063 50                    723             PUSH    AX
                           724
0064 83FBCA                725             CMP     BX,LAST_OBJECT
0067 7F19                  726             JG      SER1            ;IF ILLEGAL OBJECT NUMBER
                           727
0069 8BC3                  728             MOV     AX,BX           ;MULTIPLY OBJECT NUMBER BY 3
006B 03DB                  729             ADD     BX,BX
006D 03D8                  730             ADD     BX,AX
006F 2E8BB71B00      R     731             MOV     SI,TABLE[BX].LOC ;READ OBJECT BASE ADDRESS
0074 2E8A971A00      R     732             MOV     DL,TABLE[BX].CTL_FLD ;READ OBJECT LENGTH
0079 8ACA                  733             MOV     CL,DL
007B 81E1C70C              734             AND     CX,MASK LEN     ;EXTRACT LENGTH FIELD OF "CTL"
007F 8BD9                  735             MOV     BX,CX           ;WANT TO RETURN LENGTH IN BX ALSO
                           736
0081 C3                    737             RET
                           738
0082 33C0                  739     SER1:   XOR     AX,AX           ;ILLEGAL ITEM; RETURN 0 TO ORIGINAL CALLER
0084 5B                    740             POP     BX              ;CLEAR LOCAL RETURN ADDRESS
0085 C3                    741             RET                     ;RETURN TO CALLER'S CALLER
                           742
                           743
                           744
                           745     ;;      BUMPCNT - USED TO BUMP STICKY 16 BIT COUNTERS
                           746     ;
```

```
LOC  OBJ                  LINE     SOURCE

                           747     ;       (DI) = ADDRESS OF 16 BIT COUNTER
                           748     ;
C086 FF05                  749     BUMPCNT:        INC     WORD PTR [DI]
0088 7502                  750             JNZ     BC1
C08A FF0D                  751             DEC     WORD PTR [DI]           ; FLOWED OVER TO 0, RESET TO 0FFFFH
008C C3                    752     BC1:    RET
                           753
                           754
                           755
                           756
                           757
                           758     ;;;     DLLETHERLOAD - ENTRY POINT FOR COMPUTING ETHER LOADING.
                           759     ;
                           760     ;          This routine is called about every 25 mS by the KAOS
                           761     ;       real-time clock interrupt service routine.  It computes a
                           762     ;       moving average of the amount of traffic on the Ethernet
                           763     ;       by sampling the Carrier Sense (CS) flag and computing a
                           764     ;       new average according to the formula:
                           765     ;
                           766     ;             LOADING(T) = (1-ALPHA/64K) * LOADING(T-1) + (ALPHA/64K)*CS(T)
                           767     ;
                           768     ;       where LOADING is interpreted as a binary fraction in the
                           769     ;       range 0 <= LOADING < 1.
                           770
                           771
  0028                     772     ALPHA   EQU     40
                           773             PUBLIC  DLLETHERLOAD
C08D                       774     DLLETHERLOAD:
008D 50                    775             PUSH    AX              ;SAVE REGS I USE
C08E 53                    776             PUSH    BX
C08F 52                    777             PUSH    DX
C090 A16200          R     778             MOV     AX,WORD PTR LOADING     ;MULTIPLY BOTTOM HALF
C093 BBD6FF                779             MOV     BX,0FFFFH-(ALPHA+1)
C096 F7E3                  780             MUL     BX
C098 D1C0                  781             ROL     AX,1            ;ROUND TOP PART, DISCARD LOW PART
C09A 83D200                782             ADC     DX,0
C09D 52                    783             PUSH    DX              ;SAVE TOP PART
C09E A16400          R     784             MOV     AX,WORD PTR LOADING+2   ;MULTIPLY TOP HALF
C0A1 F7E3                  785             MUL     BX
C0A3 5B                    786             POP     BX              ;ADD TWO PRODUCTS
C0A4 03C3                  787             ADD     AX,BX
C0A6 83D200                788             ADC     DX,0
C0A9 A36200          R     789             MOV     WORD PTR LOADING,AX
C0AC E4E0                  790             IN      AL,PIOA         ;READ CS FLAG
C0AE A804                  791             TEST    AL,04H          ;TEST FLAG
00B0 7403                  792             JZ      DEL1            ;IF NO CS
C0B2 83C228                793             ADD     DX,ALPHA        ;ADD TO PERCENTAGE
C0B5 8916640C        R     794     DEL1:   MOV     WORD PTR LOADING+2,DX
C0B9 5A                    795             POP     DX
C0BA 5B                    796             POP     BX
C0BB 58                    797             POP     AX
C0BC C3                    798             RET
                           799
                           800
                           801     $EJECT TITLE ('DLL - MULTICAST PROCESSING')
```

```
LOC  OBJ                  LINE     SOURCE

                          802      ;;;     DLL$SET$MODE (MODE) - SET ADDRESS RECOGNITION MODE.
                          803      ;
                          804      ;       PARAMETER:
                          805      ;          MODE: 0 = NORMAL MODE.
                          806      ;                1 = ALL MULTICAST.
                          807      ;                2 = PROMISCUOUS.
                          808      ;                ALL OTHERS = NORMAL MODE.
                          809
                          810
                          811              PUBLIC  DLLSETMODE
00BD                      812      DLLSETMODE:
00BD 5A                   813              POP     DX
00BE 58                   814              POP     AX
00BF 52                   815              PUSH    DX
00C0 9C                   816              PUSHF                   ;MAKE THIS A CRITICAL REGION
00C1 FA                   817              CLI
00C2 3D0100               818              CMP     AX,1            ;TEST FOR "ALL MULTICAST"
00C5 7507                 819              JNZ     DSM1            ;IF NOT
00C7 C606980001     R     820              MOV     ALLMC,1
00CC EB15                 821              JMP     SHORT DSM3
00CE 3D0200               822      DSM1:   CMP     AX,2            ;TEST FOR PROMISCUOUS MODE
00D1 7508                 823              JNZ     DSM2            ;IF NOT
00D3 C6069800FF     R     824              MOV     ALLMC,0FFH      ;SET ALL MULTICAST FLAG
00D8 B022                 825              MOV     AL,PIOSENP      ;TELL HARDWARE TO ACCEPT EVERYTHING
00DA E6E1                 826              OUT     PIOB,AL
00DC 9D                   827              POPF
00DD C3                   828              RET
00DE C606980000     R     829      DSM2:   MOV     ALLMC,0         ;NORMAL MODE.  CLEAR "ALL MULTICAST" FLAG
00E3 B02A                 830      DSM3:   MOV     AL,PIOSEN       ;CLEAR PROMISCUOUS MODE
00E5 E6E1                 831              OUT     PIOB,AL
00E7 9D                   832              POPF                    ;DONE - RETURN
00E8 C3                   833              RET
                          834
                          835
                          836
                          837
                          838      ;;;     DLL$ADDMCID (MCID$O) - ADD MULTICAST ID.
                          839      ;
                          840      ;       PARAMETER:
                          841      ;          MCID$O - OFFSET OF 6-BYTE MULTICAST ID.
                          842      ;
                          843      ;       RETURNS:
                          844      ;          AL = 0 - OPERATION COMPLETE.
                          845      ;          AL = 1 - LIST OVERFLOW.
                          846
                          847
                          848              PUBLIC  DLLADDMCID
00E9                      849      DLLADDMCID:
00E9 58                   850              POP     AX              ;RETURN ADDRESS
00EA 5B                   851              POP     BX
00EB 50                   852              PUSH    AX
00EC 9C                   853              PUSHF                   ;MAKE THIS A CRITICAL REGION
00ED FA                   854              CLI
00EE E85500               855              CALL    CHECKMCID       ;CHECK IF ALREADY IN LIST
00F1 D0C8                 856              RCR     AL,1
```

```
LOC  OBJ                  LINE     SOURCE

00F3 721F                  857              JC      ADM1
                           858
00F5 A19900          R     859              MOV     AX,MCIDL        ;LENGTH OF TABLE
00F8 3D0800                860              CMP     AX,8            ;MAX LEN
00FB 7D13                  861              JGE     ADM2            ;IF MAX LEN EXCEEDED
                           862
00FD B406                  863              MOV     AH,6            ;COMPUTE INDEX INTO TABLE
00FF F6E4                  864              MUL     AH
0101 BF9B00          R     865              MOV     DI,OFFSET DGROUP:MCID
0104 03F8                  866              ADD     DI,AX
                           867
0106 8BF3                  868              MOV     SI,BX           ;MOVE INTO TABLE
0108 1E                    869              PUSH    DS
0109 C7                    870              POP     ES
010A B90300                871              MOV     CX,3
010D FC                    872              CLD
010E F3                    873              REP     MOVSW
010F A5
0110 FF06990C        R     874              INC     MCIDL           ;ONE MORE IN TABLE
                           875
0114 32C0                  876      ADM1:   XOR     AL,AL           ;RETURN SUCCESS
0116 9D                    877              POPF
0117 C3                    878              RET
                           879
0118 B001                  880      ADM2:   MOV     AL,1            ;RETURN FAILURE
011A 9D                    881              POPF
011B C3                    882              RET
                           883
                           884
                           885
                           886
                           887      ;;;     DLL$DELMCID (MICD$O) - DELETE MULTICAST ID.
                           888      ;
                           889      ;       PARAMETER:
                           890      ;          MCID$O - OFFSET OF 6-BYTE MULTICAST ID.
                           891
                           892
                           893              PUBLIC  DLLDELMCID
011C                       894      DLLDELMCID:
011C 58                    895              POP     AX              ;RETURN ADDRESS
011D 5B                    896              POP     BX              ;ID
011E 50                    897              PUSH    AX
011F 9C                    898              PUSHF                   ;MAKE THIS A CRITICAL REGION
0120 FA                    899              CLI
0121 E82200                900              CALL    CHECKMCID
0124 D0C8                  901              ROR     AL,1            ;CHECK IF IN LIST
0126 7315                  902              JNC     DMC1            ;IF NOT IN LIST
                           903
0128 8BFE                  904              MOV     DI,SI           ;DESTINATION IS MATCHING ENTRY
012A 83C606                905              ADD     SI,6            ;SOURCE IS NEXT ENTRY
                           906
012D 49                    907              DEC     CX              ;COMPUTE NUMBER TO MOVE
012E 8BC1                  908              MOV     AX,CX
0130 03C9                  909              ADD     CX,CX
0132 03C8                  910              ADD     CX,AX
```

```
LOC  OBJ                  LINE     SOURCE

                          911
0134 1E                   912              PUSH    DS              ;MOVE DOWN
0135 07                   913              POP     ES
0136 FC                   914              CLD
0137 F3                   915              REP     MOVSW
0138 A5
0139 FF0E990C        R    916              DEC     MCIDL           ;ONE LESS IN TABLE
                          917
013D 9D                   918      DMC1:   POPF
013E C3                   919              RET
                          920
                          921
                          922
                          923
                          924      ;;;     DLL$CLEARMC - CLEAR MULTICAST LIST.
                          925      ;
                          926
                          927
                          928              PUBLIC  DLLCLEARMC
013F                      929      DLLCLEARMC:
013F C706990C0000    R    930              MOV     MCIDL,0         ;CLEAR LIST
0145 C3                   931              RET
                          932
                          933
                          934
                          935
                          936      ;;;     CHECKMCID - CHECK FOR MULTICAST ID MATCH.
                          937      ;
                          938      ;       PARAMETER:
                          939      ;          [BX] - POINTER TO 48-BIT ETHERNET ADDRESS TO CHECK.
                          940      ;
                          941      ;       RETURNS:
                          942      ;          AL = TRUE - ADDRESS WAS IN MCID TABLE.
                          943      ;          AL = FALSE - ADDRESS WAS NOT IN TABLE.
                          944      ;          IF TRUE, B-CX = INDEX INTO MCID TABLE
                          945      ;                   AND SI POINTS TO MATCHING ENTRY.
                          946      ;
                          947      ;       PRESERVES:
                          948      ;          BX
                          949
                          950
0146                      951      CHECKMCID:
0146 8B0E990C        R    952              MOV     CX,MCIDL        ;NUMBER OF MCID'S
014A E31E                 953              JCXZ    CMCID3          ;IF NOT MCID'S
014C 8B07                 954              MOV     AX,[BX]         ;LOAD ADDRESS
014E 8B5702               955              MOV     DX,[BX+2]
0151 8B7F04               956              MOV     DI,[BX+4]
0154 BE9B00          R    957              MOV     SI,OFFSET DGROUP:MCID
                          958
0157 3B04                 959      CMCID1: CMP     AX,[SI]
0159 750A                 960              JNZ     CMCID2
015B 3B5402               961              CMP     DX,[SI+2]
015E 7505                 962              JNZ     CMCID2
0160 3B7C04               963              CMP     DI,[SI+4]
0163 7408                 964              JZ      CMCID4          ;IF A MATCH
```

```
LOC  OBJ                  LINE     SOURCE

0165 83C606               965      CMCID2: ADD     SI,6
0168 E2ED                 966              LOOP    CMCID1
                          967
016A 32C0                 968      CMCID3: XOR     AL,AL           ;RETURN FALSE
016C C3                   969              RET
                          970
016D B0FF                 971      CMCID4: MOV     AL,0FFH ;RETURN TRUE
016F C3                   972              RET
                          973
                          974
                          975      $EJECT TITLE ('DLL - TYPE PROCESSING ROUTINES')
```

```
LOC  OBJ                  LINE     SOURCE

                           976     ;;;     DLLCONNECT (TYPE, MBX$O) - REGISTER USER OF A DLL TYPE.
                           977     ;
                           978     ;       PARAMETERS:
                           979     ;          TYPE - THE 16-BIT TYPE CODE
                           980     ;          MBX$O - OFFSET OF A MAILBOX HEADER TO SEND PACKETS OF THIS TYPE TO.
                           981     ;
                           982     ;       RETURNS:
                           983     ;          AL = 0 - OPERATION COMPLETE.
                           984     ;          AL = 1 - LIST OVERFLOW.
                           985     ;
                           986     ;       ACCESSES:
                           987     ;          THE TYPE TABLE.
                           988
                           989
                           990             PUBLIC  DLLCONNECT
0170                       991     DLLCONNECT:
0170 5A                    992             POP     DX
0171 5B                    993             POP     BX              ;MBX
0172 58                    994             POP     AX              ;TYPE
0173 86C4                  995             XCHG    AL,AH           ;PUT BYTES IN PROPER SEQUENCE
0175 8B0ECB0C        R    996             MOV     CX,TYPEL        ;NUMBER OF TYPES
0179 83F908                997             CMP     CX,8
017C 741E                  998             JZ      RTY2            ;NO MORE SLOTS LEFT
                           999
017E BFCD00          R    1000             MOV     DI,OFFSET DGROUP:TYPET
0181 FC                   1001             CLD                     ;set auto increment
0182 1E                   1002             PUSH    DS
0183 07                   1003             POP     ES
0184 F2                   1004             REPNZ   SCASW           ;SEARCH LIST for this type
0185 AF
0186 7507                 1005             JNZ     RTY1            ;IF NOT MATCH
0188 895D0E               1006             MOV     [DI]+(TYPEM-TYPET)-2,BX ;STORE NEW MBX FOR OLD TYPE  old type
018B 32C0                 1007             XOR     AL,AL           ;OPERATION COMPLETE
018D FFE2                 1008             JMP     DX              ;RETURN
                          1009
018F 8905                 1010     RTY1:   MOV     [DI],AX         ;STORE TYPE AT ONE PAST LAST IN LIST  new type
0191 895D10               1011             MOV     [DI]+(TYPEM-TYPET),BX ;STORE MBX
0194 FF06CB0C        R    1012             INC     TYPEL           ;ONE MORE IN LIST
0198 32C0                 1013             XOR     AL,AL           ;OPERATION COMPLETE
019A FFE2                 1014             JMP     DX              ;RETURN
                          1015
019C B001                 1016     RTY2:   MOV     AL,1            ;TABLE OVERFLOW
019E FFE2                 1017             JMP     DX              ;RETURN
                          1018
                          1019
                          1020
                          1021
                          1022     ;;;     DLLDISCONNECT (TYPE) - REMOVE USER OF A DLL TYPE.
                          1023     ;
                          1024     ;       PARAMETERS:
                          1025     ;          TYPE - THE 16-BIT TYPE CODE.
                          1026     ;
                          1027     ;       ACCESSES:
                          1028     ;          THE TYPE TABLE.
                          1029
```

```
LOC  OBJ                  LINE     SOURCE

                          1030
                          1031             PUBLIC  DLLDISCONNECT
01A0                      1032     DLLDISCONNECT:
C1A0 5A                   1033             POP     DX              ;RETURN ADDRESS
01A1 58                   1034             POP     AX              ;TYPE CODE
C1A2 86C4                 1035             XCHG    AL,AH           ;PUT BYTES IN RIGHT ORDER
                          1036
                          1037     ;       SEARCH TABLE FOR TYPE
                          1038
01A4 880ECB0C       R     1039             MOV     CX,TYPEL        ;NUMBER OF TYPES
C1A8 BFCD00         R     1040             MOV     DI,OFFSET DGROUP:TYPET
01AB FC                   1041             CLD
01AC 1E                   1042             PUSH    DS
C1AD 07                   1043             POP     ES
01AE F2                   1044             REPNZ   SCASW           ;SEARCH
01AF AF
01B0 751A                 1045             JNZ     DIS1            ;IF NO MATCH
                          1046
                          1047     ;       REMOVE FROM TABLE
                          1048
C1B2 9C                   1049             PUSHF                   ;CRITICAL REGION
01B3 FA                   1050             CLI
                          1051
01B4 8BF7                 1052             MOV     SI,DI
C1B6 83EF02               1053             SUB     DI,2            ;MOVE ENTRIES DOWN
01B9 8BDF                 1054             MOV     BX,DI           ;SAVE ADDRESS
01BB 51                   1055             PUSH    CX              ;SAVE COUNT
01BC F3                   1056             REP     MOVSW           ;MOVE TYPE TYBLE DOWN
01BD A5
                          1057
01BE 8D7F10               1058             LEA     DI,[BX]+(TYPEM-TYPET) ;REPEAT FOR MBX TABLE
C1C1 8D7502               1059             LEA     SI,[DI]+2
01C4 59                   1060             POP     CX
01C5 F3                   1061             REP     MOVSW
01C6 A5
                          1062
01C7 FF0ECB00       R     1063             DEC     TYPEL           ;ONE LESS IN TABLE
01CB 9D                   1064             POPF                    ;RESTORE FLAGS
                          1065
01CC FFE2                 1066     DIS1:   JMP     DX              ;RETURN
                          1067
                          1068
                          1069
                          1070     $EJECT TITLE ('DLL - INITIALIZATION ROUTINE')
```

```
LOC  OBJ                  LINE     SOURCE

                          1071     ;;;     DLLSTART
                          1072     ;
                          1073     ;          DLLSTART PERFORMS ALL INITIALIZATION NEEDED FOR DLL.
                          1074     ;
                          1075     ;       RETURNS:
                          1076     ;          AX = 0 -- INITIALIZATION SUCCESSFUL
                          1077     ;          AX = 08H -- HARDWARE FAILURE PREVENTED READING HOSTID.
                          1078     ;
                          1079     ;       NOTE:  The hardware is not supposed to fail while reading
                          1080     ;              the HOSTID,  but there is a bad hardware race condition
                          1081     ;              which can cause it to fail at temperture extremes.
                          1082
                          1083
                          1084             PUBLIC  DLLSTART
C1CE                      1085     DLLSTART:
                          1086
                          1087     ;       CREATE KAOS OBJECTS
                          1088
01CE 0E                   1089             PUSH    CS
01CF B80000         R     1090             MOV     AX,OFFSET CGROUP:KAOSOBJECTS
01D2 50                   1091             PUSH    AX
01D3 E80000         E     1092             CALL    CGCREATELIST      create MBX's, Sem, Rcv Process
                          1093
                          1094     ;       CLEAR DATA BASE
                          1095
01D6 1E                   1096             PUSH    DS
01D7 07                   1097             POP     ES
01D8 B000                 1098             MOV     AL,0
01DA BF4600         R     1099             MOV     DI,OFFSET DGROUP:DB_START
01DD B9A800               1100             MOV     CX,DB_END - DB_START
01E0 F3                   1101             REP     STOSB             ;CLEAR MEMORY
01E1 AA
                          1102
                          1103     ;       INITIALIZE VARIABLES WHICH NEED NONZERO INITIAL VALUES
                          1104
01E2 C7066D0CC604   R     1105             MOV     NEXT_CH,OFFSET CGROUP:RX1ISR
01E8 C7068D0C8B00   R     1106             MOV     RCVD_TAIL,OFFSET DGROUP:RCVD_HEAD
01EE C606960C01     R     1107             MOV     BUF_FIRST,1       ;MARK THAT ALL CHANNELS NEED INITIALIZATION
01F3 C606970C03     R     1108             MOV     BUF_COUNT,3       3, starting @ 1
                          1109
                          1110
                          1111     ;       READ HOST ID FROM SERDES
                          1112
01F8 B70A                 1113             MOV     BH,10             ;UP TO 10 RETRIES
01FA E4F1                 1114             IN      AL,PICMASK        ;SAVE PIC MASK IN BL
01FC 8AD8                 1115             MOV     BL,AL
                          1116
01FE B0F3                 1117     SRT1:   MOV     AL,PIOCLR         ;CLEAR SERDES
0200 E6E1                 1118             OUT     PIOB,AL
0202 E606                 1119             OUT     RESET_CHANNEL_COUNTER,AL ;SYNCH HARDWARE WITH SOFTWARE / clear counters; DMA
0204 E605                 1120             OUT     RESET_ERROR,AL    ;CLEAR ANY EXISTING ERROR FLAGS  length/CRC errors on SerDes
0206 32C0                 1121             XOR     AL,AL             ;WILL DMA INTO BEGINNING OF BUFFER RAM
0208 E6C2                 1122             OUT     CH1ADDR,AL
020A E6C2                 1123             OUT     CH1ADDR,AL
020C E6C3                 1124             OUT     CH1WC,AL          lower limit
```

```
LOC  OBJ                  LINE    SOURCE

020E FEC0                 1125            INC     AL                      ;ABORT AFTER 257 BYTES
0210 E6C3                 1126            OUT     CH1WC,AL
0212 B00D                 1127            MOV     AL,0DH                  ;SET DMA MASK
0214 E6CF                 1128            OUT     DMAMASK,AL
0216 E602                 1129            OUT     SET_RXAV1,AL            ;SIGNAL CHANNEL AVAILABLE
0218 B0E2                 1130            MOV     AL,0E2H                 ;EXTRA COMMAND BECAUSE SERDES IS OBSTINATE
021A E6E1                 1131            OUT     PIOB,AL
021C B90A00               1132            MOV     CX,10                   ; WAIT AT LEAST 9.6US
021F E0FE                 1133    SRT1A:  LOOPNZ  SRT1A
0221 B0E0                 1134            MOV     AL,PIOREAD              ;SET SERDES TO READ
0223 E6E1                 1135            OUT     PIOB,AL
0225 B92C01               1136            MOV     CX,300                  ;TIMEOUT
                          1137
0228 B0FE                 1138    SRT2:   MOV     AL,NOT CH1_DONE         ;OPEN MASK FOR CHANNEL 1 ONLY
022A E6F1                 1139            OUT     PICMASK,AL
022C B00C                 1140            MOV     AL,POLL_PIC             ;POLL TO SEE IF CHANNEL 1 DONE
022E E6F0                 1141            OUT     PICCMD,AL
0230 E4F0                 1142            IN      AL,PICDATA
0232 3480                 1143            XOR     AL,80H                  ;TEST IF INTERRUPT 0 ASSERTED
0234 E0F2                 1144            LOOPNZ  SRT2                    ;IF NOT DONE AND NOT TIMEOUT
                          1145
0236 740C                 1146            JZ      SRT3                    ;IF OK
0238 FECF                 1147            DEC     BH                      ;RETRY READ HOSTID FUNCTION
023A 75C2                 1148            JNZ     SRT1
023C 8AC3                 1149            MOV     AL,BL                   ; RESTORE THE OLD MASK
023E E6F1                 1150            OUT     PICMASK,AL
0240 B80800               1151            MOV     AX,0BH                  ;RETURN FAILURE
0243 C3                   1152            RET
                          1153
0244 B060                 1154    SRT3:   MOV     AL,SEOI_PIC+0           ;EOI THE PIC
0246 E6F0                 1155            OUT     PICCMD,AL
0248 8AC3                 1156            MOV     AL,BL                   ;RESTORE OLD MASK
024A E6F1                 1157            OUT     PICMASK,AL
024C B0F3                 1158            MOV     AL,PIOCLR               ;CLEAR THE SERDES
024E E6E1                 1159            OUT     PIOB,AL
                          1160
0250 A10200          R    1161            MOV     AX,BUFFER_START[2] ;MOVE ADDRESS TO HOSTID
0253 A35C00          R    1162            MOV     HOSTID,AX
0256 A10400          R    1163            MOV     AX,BUFFER_START[4]
0259 A35E00          R    1164            MOV     HOSTID[2],AX
025C A10600          R    1165            MOV     AX,BUFFER_START[6]
025F A36000          R    1166            MOV     HOSTID[4],AX
0262 E606                 1167            OUT     RESET_CHANNEL_COUNTER,AL ;RESET TO START WITH CHANNEL 1
                          1168
                          1169    ;       INITIALIZE BCB POOL
                          1170
0264 BB6F00          R    1171            MOV     BX,OFFSET DGROUP:BCB
0267 B90400               1172            MOV     CX,NUM_RX_BUF
026A A18700          R    1173    SRT4:   MOV     AX,BCB_HEAD             ;LINK ONTO STACK
026D 8907                 1174            MOV     [BX].LINK,AX
026F 891E8700        R    1175            MOV     BCB_HEAD,BX
0273 83C306               1176            ADD     BX,BCB_TYPE_LEN ;ADVANCE TO NEXT
0276 E2F2                 1177            LOOP    SRT4
                          1178
                          1179    ;       INITIALIZE TRANSMIT BUFFER POOL
```

```
LOC  OBJ                  LINE     SOURCE

                          1180
C278 B80000         R     1181             MOV     AX,OFFSET DGROUP:BUFFER_START
C27B B90100               1182             MOV     CX,NUM_TX_BUF
C27E 50                   1183     SRT5:   PUSH    AX                ;SAVE BUFFER POINTER
C27F 51                   1184             PUSH    CX                ;SAVE COUNT
                          1185
0280 BB0000         R     1186             MOV     BX,OFFSET DGROUP:DLLTXFREEMBX ;SEND TO POOL
0283 53                   1187             PUSH    BX
0284 1E                   1188             PUSH    DS
0285 50                   1189             PUSH    AX
0286 E80000         E     1190             CALL    CQSEND
                          1191
0289 59                   1192             POP     CX
028A 58                   1193             POP     AX
028B 05FA05               1194             ADD     AX,SEG_BUF_LEN
028E E2EE                 1195             LOOP    SRT5
                          1196
                          1197     ;       INITIALIZE RECEIVE BUFFER POOL
                          1198
0290 B90400               1199             MOV     CX,NUM_RX_BUF
0293 50                   1200     SRT6:   PUSH    AX                ;SAVE BUFFER POINTER
0294 51                   1201             PUSH    CX                ;SAVE COUNT
                          1202
0295 50                   1203             PUSH    AX                ;SEND TO RCVD POOL
0296 E89601               1204             CALL    DLLRXRETBUF
                          1205
0299 59                   1206             POP     CX
029A 58                   1207             POP     AX
029B 05FA05               1208             ADD     AX,SEG_BUF_LEN
029E E2F3                 1209             LOOP    SRT6
                          1210
                          1211     ;       SET UP PORTS
                          1212
02A0 32C0                 1213             XOR     AL,AL             ;ZERO
02A2 E605                 1214             OUT     RESET_ERROR,AL    ;CLEAR ANY EXISTING ERROR FLAGS
02A4 E6CF                 1215             OUT     DMAMASK,AL        ;ENABLE ALL DMA CHANNELS
02A6 B02A                 1216             MOV     AL,PIOSEN         ;ENABLE SERDES
02A8 E6E1                 1217             OUT     PIOB,AL
                          1218
02AA 33C0                 1219             XOR     AX,AX             ;RETURN SUCCESS
02AC C3                   1220             RET
                          1221
                          1222
                          1223
                          1224     $EJECT TITLE ('DLL - TRANSMIT MAIN ROUTINES')
```

```
LOC  OBJ                  LINE     SOURCE

                          1225     ;;;     DLLTXSEND (PACKET) - TRANSMIT PACKET
                          1226     ;
                          1227     ;       PARAMETER:
                          1228     ;          PACKET - OFFSET IN DGROUP OF THE SEGMENT BUFFER
                          1229
                          1230
                          1231             PUBLIC  DLLTXSEND
02AD                      1232     DLLTXSEND:
02AD 5A                   1233             POP     DX              ;RETURN ADDRESS
02AE 5B                   1234             POP     BX              ;PACKET ADDRESS
02AF 52                   1235             PUSH    DX
                          1236
                          1237     ;       MOVE HOSTID TO DL_SOURCE FIELD
                          1238
02B0 A15C00         R     1239             MOV     AX,HOSTID
02B3 89470C               1240             MOV     [BX].DL_SOURCE,AX
02B6 A15E00         R     1241             MOV     AX,HOSTID[2]
02B9 89470E               1242             MOV     [BX+2].DL_SOURCE,AX
02BC A16000         R     1243             MOV     AX,HOSTID[4]
02BF 894710               1244             MOV     [BX+4].DL_SOURCE,AX
                          1245
                          1246     ;       CHECK FOR AND PROCESS SELF-ADDRESSED PACKET
                          1247
02C2 F6470601             1248             TEST    BYTE PTR [BX].DL_DEST,MCFLAG ;TEST MULTICAST BIT
02C6 751A                 1249             JNZ     TXP1            ;IF MULTICAST
02C8 A15C00         R     1250             MOV     AX,HOSTID       ;CHECK IF THIS HOSTID IS SELF-ADDRESSED
02CB 394706               1251             CMP     [BX].DL_DEST,AX
02CE 7523                 1252             JNZ     TXP3            ;IF NOT
02D0 A15E00         R     1253             MOV     AX,HOSTID[2]
02D3 394708               1254             CMP     [BX+2].DL_DEST,AX
02D6 751B                 1255             JNZ     TXP3
02D8 A16000         R     1256             MOV     AX,HOSTID[4]
02DB 39470A               1257             CMP     [BX+4].DL_DEST,AX
02DE 7513                 1258             JNZ     TXP3
02E0 EB0C                 1259             JMP     SHORT TXP2
02E2 53                   1260     TXP1:   PUSH    BX              ;SAVE PACKET POINTER
02E3 8D5F06               1261             LEA     BX,[BX].DL_DEST ;CORRECT PARAMETER
02E6 E85DFE               1262             CALL    CHECKMCID       ;CHECK IF THIS IS MULTICAST SELF-ADDRESSED
02E9 5B                   1263             POP     BX              ;RESTORE PACKET POINTER
02EA D0C8                 1264             ROR     AL,1
02EC 7305                 1265             JNC     TXP3            ;IF NOT SELF-ADDRESSED
                          1266
02EE 53                   1267     TXP2:   PUSH    BX              ;SAVE PACKET POINTER
02EF E8FC03               1268             CALL    LOOP_PACKET     ;SEND TO RECEIVE SIDE
02F2 5B                   1269             POP     BX
                          1270
                          1271     ;       TRANSMIT IF NO OTHERS PENDING; ELSE, QUEUE IT UP
                          1272
02F3 803E680000     R     1273     TXP3:   CMP     XMITBUSY,0      ;CHECK IF TRANSMITTER IS NOW BUSY
02F8 7512                 1274             JNZ     TXP4            ;JUMP IF BUSY
02FA 9C                   1275             PUSHF                   ;SET UP CRITICAL REGION
02FB FA                   1276             CLI
02FC E80B01               1277             CALL    SETUPXMIT       ;START TRANSMISSION
02FF C6066A0000     R     1278             MOV     BACKOFF_COUNT,0 ;CLEAR BACKOFF INDICATORS
0304 C7066B0000000  R     1279             MOV     BACKOFF_MASK,0
```

```
LOC  OBJ                  LINE     SOURCE

030A 9D                   1280             POPF                    ;RETURN
030B C3                   1281             RET
030C B81000         R     1282     TXP4:   MOV     AX,OFFSET DGROUP:DLL_MBX1 ;SEND TO TRANSMIT MAILBOX
030F 50                   1283             PUSH    AX
0310 1E                   1284             PUSH    DS
0311 53                   1285             PUSH    BX
0312 E80000         E     1286             CALL    COSEND
0315 C3                   1287             RET
                          1288
                          1289
                          1290
                          1291
                          1292     $EJECT TITLE ('DLL - TRANSMIT INTERRUPT SERVICE ROUTINES')
```

```
LOC  OBJ                  LINE     SOURCE

                          1293     ;;      XMITISR - TRANSMIT INTERRUPT SERVICE ROUTINE.
                          1294     ;
                          1295     ;          THIS ROUTINE IS INVOKED BY THE TRANSMIT COMPLETE INTERRUPT.
                          1296     ;       IT CHECKS THE ERROR STATUS,  INITIATES BACKOFF IF THERE WAS A
                          1297     ;       COLLISION,  RETURNS THE PACKET BUFFER TO THE APPROPRIATE FREE
                          1298     ;       LIST, AND SETS UP FOR TRANSMISSION OF THE NEXT PACKET, IF ONE
                          1299     ;       HAS BEEN QUEUED.
                          1300
                          1301
                          1302             PUBLIC  XMITISR
0316                      1303     XMITISR:
0316 06                   1304             PUSH    ES              ;SAVE STATE OF INTERRUPTED ROUTINE
0317 1E                   1305             PUSH    DS
0318 2E8E1E1800     R     1306             MOV     DS,DGRP
031D 50                   1307             PUSH    AX
031E 51                   1308             PUSH    CX
031F 52                   1309             PUSH    DX
0320 53                   1310             PUSH    BX
0321 56                   1311             PUSH    SI
0322 57                   1312             PUSH    DI
                          1313
0323 B020                 1314             MOV     AL,EOI_PIC      ;EOI THE PIC
0325 E6F0                 1315             OUT     PICCMD,AL
0327 E4E0                 1316             IN      AL,PIOA         ;READ STATUS
0329 24C0                 1317             AND     AL,0C0H         ;CHECK FOR ERRORS
032B 745B                 1318             JZ      XIS5            ;IF NO ERRORS
                          1319
                          1320     ;       PROCESS  ERROR
                          1321
032D A880                 1322             TEST    AL,80H          ;CHECK FOR COLLISION
032F 7451                 1323             JZ      XIS4            ;IF NOT COLLISION
                          1324
0331 803E6A0C0F     R     1325             CMP     BACKOFF_COUNT,ATTEMPT_LIMIT ;CHECK IF AT MAX COLLISION COUNT
0336 7508                 1326             JNE     XIS1            ;IF MORE COLLISIONS ALLOWED
0338 BF4EC0         R     1327             MOV     DI,OFFSET DGROUP:EXCEEDED_COLL  ;INCREMENT ERROR COUNTER
033B E848FD               1328             CALL    BUMPCNT
033E EB48                 1329             JMP     SHORT XIS5
                          1330
                          1331     ;       COMPUTE BACKOFF AND START TIMER
                          1332
0340 E80000         E     1333     XIS1:   CALL    RANDOM          ;GENERATE A RANDOM NUMBER IN AX
0343 8B1E6B0C       R     1334             MOV     BX,BACKOFF_MASK
0347 F9                   1335             STC                     ;SHIFT IN A 1
0348 D1D3                 1336             RCL     BX,1
034A 81E3FF03             1337             AND     BX,BACKOFF_LIMIT ;BIT MASK FOR MAX LEN OF BACKOFF_MASK
034E 891E6B0C       R     1338             MOV     BACKOFF_MASK,BX
0352 23C3                 1339             AND     AX,BX
0354 7501                 1340             JNZ     XIS1A           ; IF COUNT IS ZERO, MAKE IT ONE
0356 40                   1341             INC     AX
0357 BB4000               1342     XIS1A:  MOV     BX,SLOT_TIME    ;COMPUTE RANDOM*SLOT_TIME
035A F7E3                 1343             MUL     BX
035C E6D2                 1344             OUT     PIT_BCK,AL      ;OUTPUT THIS TO THE BACKOFF TIMER
035E 8AC4                 1345             MOV     AL,AH
0360 E6D2                 1346             OUT     PIT_BCK,AL
0362 C606690C01     R     1347             MOV     BACKOFF_WAIT,1  ;WE ARE WAITING FOR A BACKOFF INTERRUPT
```

```
LOC  OBJ                  LINE     SOURCE

                          1348
                          1349     ;       UPDATE NETWORK MANAGEMENT COUNTERS
                          1350
0367 FE066A0C       R     1351             INC     BACKOFF_COUNT   ;ANOTHER COLLISION
036B 803E6A0C01     R     1352             CMP     BACKOFF_COUNT,1 ;CHECK IF THIS IS THE FIRST COLLISION
0370 7508                 1353             JNZ     XIS2            ;IF NOT FIRST COLLISION
0372 BF4A00         R     1354             MOV     DI,OFFSET DGROUP:PRIMARY_COLL
0375 E80EFD               1355             CALL    BUMPCNT
0378 EB71                 1356             JMP     SHORT TXEXIT
037A BF4C00         R     1357     XIS2:   MOV     DI,OFFSET DGROUP:SECONDARY_COLL
037D E806FD               1358             CALL    BUMPCNT
0380 EB69                 1359     XIS3:   JMP     SHORT TXEXIT    ;WAIT FOR BACKOFF INTERRUPT
                          1360
                          1361     ;       PROCESS TIME-OUT ERROR
                          1362
0382 BF5000         R     1363     XIS4:   MOV     DI,OFFSET DGROUP:TX_PKT_TOO_LONG        ;INCREMENT ERROR COUNTER
0385 E8FEFC               1364             CALL    BUMPCNT
                          1365
                          1366     ;       TRANSMISSION COMPLETE
                          1367
0388 B81000         R     1368     XIS5:   MOV     AX,OFFSET DGROUP:DLL_MBX1 ;GET NEXT FROM TRANSMIT MAILBOX
038B 50                   1369             PUSH    AX
038C E80000         E     1370             CALL    CQICRECEIVE
038F 8B0E660C       R     1371             MOV     CX,CURRENT_PACKET
0393 81FBFFFF             1372             CMP     BX,0FFFFH       ;CHECK FOR NULL
0397 7413                 1373             JZ      XIS6
0399 51                   1374             PUSH    CX              ;SAVE CURRENT_PACKET ON STACK
039A E86D00               1375             CALL    SETUPXMIT
039D 59                   1376             POP     CX
039E C6066A0C00     R     1377             MOV     BACKOFF_COUNT,0 ;CLEAR BACKOFF INDICATORS
03A3 C7066B0C0000   R     1378             MOV     BACKOFF_MASK,0
03A9 EB0690               1379             JMP     XIS7
03AC C606680C00     R     1380     XIS6:   MOV     XMITBUSY,0      ;MARK NOT BUSY
03B1 B80000         R     1381     XIS7:   MOV     AX,OFFSET DGROUP:DLLTXFREEMBX   ;RETURN BUFFER TO POOL
03B4 50                   1382             PUSH    AX
03B5 1E                   1383             PUSH    DS
03B6 51                   1384             PUSH    CX
03B7 E80000         E     1385             CALL    CQISEND
                          1386
03BA FF06460C       R     1387             INC     TOTAL_SENT      ;INCREMENT COUNTER
03BE 7528                 1388             JNZ     TXEXIT          ;IF NO OVERFLOW
03C0 FF064B0C       R     1389             INC     TOTAL_SENT+2    ;INCREMENT HIGH PART OF COUNTER
03C4 EB25                 1390             JMP     SHORT TXEXIT    ;RETURN FROM INTERRUPT
                          1391
                          1392
                          1393
                          1394     ;;;     BACKOFFISR - SERVICE INTERRUPT FROM BACKOFF TIMER.
                          1395     ;
                          1396     ;          This routine is started by the backoff-timer-expired
                          1397     ;       interrupt.  It reinitializes the transmit channel to
                          1398     ;       retransmit a packet after a collision.
                          1399     ;       NOTE:  There is a bug in the 8259A interrupt controller
                          1400     ;       which will cause spurious interrupt 7's to be generated
                          1401     ;       if some other interrupt is asserted and removed before it
                          1402     ;       is serviced.  The flag BACKOFF_WAIT is set iff a real
```

```
LOC  OBJ                  LINE     SOURCE

                          1403     ;       interrupt for this routine is expected;  if it is not set,
                          1404     ;       this routine ignores the interrupt.
                          1405
                          1406
                          1407             PUBLIC  BACKOFFISR
03C6                      1408     BACKOFFISR:
03C6 06                   1409             PUSH    ES              ;SAVE STATE OF INTERRUPTED ROUTINE
03C7 1E                   1410             PUSH    DS
03C8 2E8E1E1800      R    1411             MOV     DS,DGRP
03CD 50                   1412             PUSH    AX
03CE 51                   1413             PUSH    CX
03CF 52                   1414             PUSH    DX
03D0 53                   1415             PUSH    BX
03D1 56                   1416             PUSH    SI
03D2 57                   1417             PUSH    DI
                          1418
03D3 B020                 1419             MOV     AL,EOI_PIC      ;EOI THE PIC
03D5 E6F0                 1420             OUT     PICCMD,AL
03D7 32C0                 1421             XOR     AL,AL           ;TEST AND CLEAR BACKOFF_WAIT FLAG
03D9 86066900        R    1422             XCHG    AL,BACKOFF_WAIT
03DD 0AC0                 1423             OR      AL,AL
03DF 7503                 1424             JNZ     BOX1
03E1 EB0890               1425             JMP     TXEXIT          ;EXIT IF NO BACKOFF PENDING
03E4 8B1E6600        R    1426     BOX1:   MOV     BX,CURRENT_PACKET
03E8 E81F00               1427             CALL    SHORT SETUPXMIT
                          1428     ;       JMP     SHORT TXEXIT
                          1429
                          1430
                          1431
                          1432
                          1433     ;;;     TXEXIT - EXIT CODE FOR TX INTERRUPT SERVICE ROUTINES
                          1434     ;
                          1435     ;          This is a common exit routine for the transmit interrupt
                          1436     ;       service routines.  Its only excuse for existing is to save
                          1437     ;       code space.
                          1438
                          1439
03EB E4E0                 1440     TXEXIT: IN      AL,PIOA         ; SEE IF ANY LEFT OVER RCV ERRORS
03ED 2430                 1441             AND     AL,30H
03EF 7410                 1442             JZ      TXE1            ; JUMP IF ALL IS OKAY
03F1 B00A                 1443             MOV     AL,READ_IRR     ; HAVE RCV ERROR, IGNORE IF THERE IS A RCV INT
03F3 E6F0                 1444             OUT     PICCMD,AL       ; this is a fix to a possible hardware error
03F5 E4F0                 1445             IN      AL,PICDATA
03F7 2407                 1446             AND     AL,CH1_DONE OR CH2_DONE OR CH3_DONE ; SEE IF ANY RCV INTERRUPTS
03F9 7506                 1447             JNZ     TXE1            ; IF NOT ZERO, THEN THERE IS A WAITING RCV INT
03FB E605                 1448             OUT     RESET_ERROR,AL  ; NO RCV INTERRUPT, I WILL CLEAR IT
03FD FE06ED00        R    1449             INC     RCV_HANG        ; BUMP COUNTER OF THIS ODD EVENT
                          1450
0401 5F                   1451     TXE1:   POP     DI
0402 5E                   1452             POP     SI
0403 5B                   1453             POP     BX
0404 5A                   1454             POP     DX
0405 59                   1455             POP     CX
0406 58                   1456             POP     AX
0407 1F                   1457             POP     DS
```

```
LOC  OBJ                  LINE     SOURCE

0408 07                   1458             POP      ES
0409 CF                   1459             IRET
                          1460
                          1461
                          1462     $EJECT TITLE ('DLL - TRANSMIT SERVICE SUBROUTINES')
```

```
LOC  OBJ                  LINE     SOURCE

                          1463     ;;      SETUPXMIT - SET UP TRANSMITTER HARDWARE.
                          1464     ;
                          1465     ;          Given the address of a packet buffer, this routine initializes
                          1466     ;       the transmit DMA and issues a transmit command to the SERDES.
                          1467     ;
                          1468     ;       PARAMETER:
                          1469     ;          (BX) = OFFSET IN DGROUP OF THE SEGMENT BUFFER.
                          1470     ;
                          1471     ;       USES: AX, BX.
                          1472
                          1473
040A                      1474     SETUPXMIT:
040A 8B4704               1475             MOV     AX,[BX].SEG_LENGTH ;OUTPUT PACKET LENGTH TO DMA
040D 48                   1476             DEC     AX                 ;DMA NEEDS ONE LESS THAN REAL LENGTH
040E E6C1                 1477             OUT     CHCWC,AL
0410 8AC4                 1478             MOV     AL,AH
0412 E6C1                 1479             OUT     CHCWC,AL
                          1480
0414 8BC3                 1481             MOV     AX,BX              ;CALCULATE PACKET BEGINNING AS SEEN BY DMA
0416 2DFAFF         R     1482             SUB     AX,OFFSET DGROUP:BUFFER_START - VAL_DL_DEST
0419 E6C0                 1483             OUT     CHCADDR,AL         ;OUTPUT PACKET ADDRESS TO DMA
041B 8AC4                 1484             MOV     AL,AH
041D E6C0                 1485             OUT     CHCADDR,AL
                          1486
041F B000                 1487             MOV     AL,0               ;CLEAR TRANSMIT MASK BIT
0421 E6CA                 1488             OUT     DMAMSKB,AL
0423 E600                 1489             OUT     SET_TXSRT,AL
                          1490
0425 891E660C       R     1491             MOV     CURRENT_PACKET,BX ;SAVE PACKET BUFFER ADDRESS
0429 C606680OFF     R     1492             MOV     XMITBUSY,OFFH     ;MARK TRANSMITTER AS BUSY
042E C3                   1493             RET
                          1494
                          1495
                          1496
                          1497
                          1498     $EJECT TITLE ('DLL - RECEIVE MAIN ROUTINE')
```

```
LOC  OBJ                  LINE     SOURCE

                          1499     ;;;     DLLRXRETBUF (BUFFER$P) - RETURN BUFFER TO DLL.
                          1500     ;
                          1501     ;       PARAMETER:
                          1502     ;          BUFFER$P = OFFSET IN DGROUP OF THE BUFFER.
                          1503
                          1504
                          1505             PUBLIC  DLLRXRETBUF
042F                      1506     DLLRXRETBUF:
042F 5F                   1507             POP     DI              ;RETURN ADDRESS
0430 5E                   1508             POP     SI              ;BUFFER POINTER
                          1509
0431 9C                   1510             PUSHF
0432 FA                   1511             CLI                     ;PREVENT INTERRUPTS FROM GETTING US HERE
                          1512
0433 8B1E870C       R     1513             MOV     BX,BCB_HEAD     ;GET NEXT BUFFER CONTROL BLOCK (dequeue it)
0437 8B07                 1514             MOV     AX,[BX].LINK
0439 A38700         R     1515             MOV     BCB_HEAD,AX
                          1516
043C 81EEF3FF       R     1517             SUB     SI,OFFSET DGROUP:BUFFER_START-VAL_DL_DEST+1 ;SUBTRACT OFFSET
0440 897704               1518             MOV     [BX].BUF,SI     ;INITIALIZE CONTROL BLOCK
                          1519
0443 803E950C00     R     1520             CMP     BUF_LOOP,0      ;CHECK IF ANYONE NEEDS A LOOPBACK BUFFER
0448 7410                 1521             JZ      RBF1            ;IF NOT
044A FE0E9500       R     1522             DEC     BUF_LOOP        ;ONE LESS
044E 9D                   1523             POPF                    ;FORMAT FOR CALL
044F 57                   1524             PUSH    DI              ;RETURN ADDRESS
0450 B82000         R     1525             MOV     AX,OFFSET DGROUP:DLL_MBX2 ;SEND TO WAITING MBX
0453 50                   1526             PUSH    AX
0454 1E                   1527             PUSH    DS
0455 53                   1528             PUSH    BX
0456 E80000         E     1529             CALL    CQSEND
0459 C3                   1530             RET
                          1531
045A 803E970C00     R     1532     RBF1:   CMP     BUF_COUNT,0     ;CHECK IF ANYBODY NEEDS INITING
045F 741E                 1533             JZ      RBF3            ;IF NO CHANNELS NEED INITING
                          1534
                          1535     ;       INIT A CHANNEL WHICH IS SHORT A BUFFER
                          1536
0461 8A169600       R     1537             MOV     DL,BUF_FIRST
0465 E8D902               1538             CALL    SETUPCHANNEL    ;SET UP THE DMA AND DATA STRUCTURES
                          1539
0468 FE0E970C       R     1540             DEC     BUF_COUNT       ;ONE LESS CHANNEL TO INIT
046C FE069600       R     1541             INC     BUF_FIRST       ;STEP TO NEXT CHANNEL
0470 803E960C04     R     1542             CMP     BUF_FIRST,4     ;CHECK IF WRAP AROUND
0475 7505                 1543             JNE     RBF2            ;IF NOT WRAP AROUND
0477 C606960C01     R     1544             MOV     BUF_FIRST,1     ;WRAP AROUND
047C 9D                   1545     RBF2:   POPF
047D FFE7                 1546             JMP     DI              ;RETURN
                          1547
                          1548     ;       LINK INTO FREE BUFFER POOL
                          1549
047F A18900         R     1550     RBF3:   MOV     AX,FREE_HEAD    ;SET UP LINK
0482 8907                 1551             MOV     [BX].LINK,AX
0484 891E8900       R     1552             MOV     FREE_HEAD,BX    ;PUT AT HEAD
0488 9D                   1553             POPF
```

```
LOC  OBJ                  LINE      SOURCE

0489 FFE7                 1554              JMP     DI              ;RETURN
                          1555
                          1556
                          1557
                          1558
                          1559      $EJECT TITLE ('DLL - RECEIVE INTERRUPT SERVICE ROUTINES')
```

```
LOC  OBJ                  LINE     SOURCE

                          1560     ;;;     RX1ISR THROUGH RX3ISR - RECEIVE CHANNEL INTERRPUT SERVICE ROUTINES.
                          1561     ;
                          1562     ;          THESE THREE ROUTINES SERVICE THE RECEIVE COMPLETE INTERRUPTS
                          1563     ;       FOR THE THREE RECEIVE DMA CHANNELS.
                          1564     ;          SINCE THE INTERRUPTS FOR THE THREE CHANNELS MAY COME IN OUT OF
                          1565     ;       ORDER DUE TO THE WAY INTERRUPTS ARE PIORITIZED,  ALL THREE ARE
                          1566     ;       VECTORED TO RXISR,  WHICH VECTORS TO THE APPROPRIATE ROUTINE.
                          1567     ;          WHEN EACH ROUTINE COMPLETES ITS PROCESSING,  IT CHECKS IF THE
                          1568     ;       INTERRUPT FOR THE NEXT CHANNEL IS ASSERTED,  AND IF SO,  JUMPS
                          1569     ;       DIRECTLY TO THE NEXT ROUTINE.
                          1570     ;          AFTER COMPLETION OF ALL NEEDED CHANNEL SERVICING,  RXEXIT IS
                          1571     ;       JUMPED TO TO COMPLETE INTERRUPT PROCESSING.
                          1572
                          1573
                          1574             PUBLIC  RXISR
C48B 06                   1575     RXISR:  PUSH    ES
C48C 1E                   1576             PUSH    DS
C48D 2E8E1E1800      R    1577             MOV     DS,DGRP
0492 50                   1578             PUSH    AX
0493 51                   1579             PUSH    CX
0494 52                   1580             PUSH    DX
0495 53                   1581             PUSH    BX
0496 56                   1582             PUSH    SI
0497 57                   1583             PUSH    DI
0498 E4F1                 1584             IN      AL,PICMASK      ;SAVE MASK
049A 50                   1585             PUSH    AX
049B FF266D00        R    1586             JMP     NEXT_CH
                          1587
                          1588
                          1589
                          1590
                          1591     ;;      RXEXIT - EXIT CODE FOR RX SERVICE ROUTINES.
                          1592     ;
                          1593     ;       NOTE: THIS ROUTINE GETS CLEVER WITH THE PIC MASK.
                          1594     ;             DON'T TOUCH PIC CODE IN OTHER RX SERVICE ROUTINES
                          1595     ;             WITHOUT STUDYING THIS FIRST.
                          1596
                          1597
C49F 803E970C03      R    1598     RXEXIT: CMP     BUF_COUNT,3     ;CHECK IF ALL CHANNELS NEED RE-INITING
04A4 7506                 1599             JNZ     RXE1            ;IF NOT
C4A6 BF5A00          R    1600             MOV     DI,OFFSET DGROUP:RSCERRS                ;INCREMENT RESOURCE ERRORS CO
                          UNTER
C4A9 E8DAFB               1601             CALL    BUMPCNT
04AC 5B                   1602     RXE1:   POP     BX              ;SET NEW INTERRUPT MASK
04AD 80CB07               1603             OR      BL,07H          ;DISABLE ALL RX CHANNELS
C4B0 E4F1                 1604             IN      AL,PICMASK      ;NEXT CHANNEL IS CURRENT PIC MASK
04B2 22C3                 1605             AND     AL,BL           ;ENABLE THE NEXT CHANNEL
C4B4 E6F1                 1606             OUT     PICMASK,AL
04B6 B83000          R    1607             MOV     AX,OFFSET DGROUP:DLL_SEM1 ;SIGNAL SEMAPHORE
C4B9 50                   1608             PUSH    AX
04BA E80000          E    1609             CALL    CQISIGNAL       ;START EXIT PROCESSING
04BD 5F                   1610             POP     DI
04BE 5E                   1611             POP     SI
04BF 5B                   1612             POP     BX
04C0 5A                   1613             POP     DX
```

```
LOC  OBJ                  LINE      SOURCE

04C1 59                   1614              POP     CX
04C2 58                   1615              POP     AX
04C3 1F                   1616              POP     DS
04C4 07                   1617              POP     ES
04C5 CF                   1618              IRET
                          1619
                          1620      $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          1621     ;;      RX1ISR - RECEIVE CHANNEL 1 INTERRUPT SERVICE ROUTINE
                          1622     ;
                          1623     ;       PARAMETERS:
                          1624     ;          NONE.
                          1625
                          1626
04C6 B060                 1627     RX1ISR: MOV     AL,SEOI_PIC+0   ;SELECTIVE EOI FOR THIS CHANNEL
04C8 E6F0                 1628             OUT     PICCMD,AL       ;SEND TO PIC
04CA E4C3                 1629             IN      AL,CH1WC        ;READ WORD COUNT OF RECEIVED PACKET IN CX
04CC 8AC8                 1630             MOV     CL,AL
04CE E4C3                 1631             IN      AL,CH1WC
04D0 8AE8                 1632             MOV     CH,AL
                          1633
                          1634     ;       CHECK FOR ERROR
                          1635
04D2 E4E0                 1636             IN      AL,PIOA         ;READ STATUS
04D4 2430                 1637             AND     AL,30H          ;CHECK RECEIVE STATUS BITS
04D6 7409                 1638             JZ      RX10            ;IF STATUS OK
04D8 B402                 1639             MOV     AH,CH2_DONE     ;NEXT CHANNEL FOLLOWING THIS ONE
04DA E8A702               1640             CALL    CHECK_STATUS    ;KICK APPROPRIATE COUNTERS
04DD 0AC0                 1641             OR      AL,AL           ;CHECK RETURNED VALUE
04DF 7406                 1642             JZ      RX11            ;IF ERROR
                          1643
                          1644     ;       CHECK FOR RUNT PACKET
                          1645
04E1 81F9AE05             1646     RX10:   CMP     CX,MAX_PACKET_LEN-MIN_PACKET_LEN  ;CHECK FOR RUNT PACKET
04E5 7E0B                 1647             JLE     RX12            ;IF NOT A RUNT
                          1648
                          1649     ;       REINITIALIZE AFTER ERROR
                          1650
04E7 8B1E3F00        R    1651     RX11:   MOV     BX,RX1_BUF      ;LOAD BUFFER CONTROL BLOCK POINTER
04EB B201                 1652             MOV     DL,1            ;CHANNEL 1
04ED E87702               1653             CALL    RESET_CHANNEL   ;RESET DMA CONTROLLER
04F0 EB48                 1654             JMP     SHORT RX15      ;DO END PROCESSING
                          1655
                          1656     ;       PACKET OK
                          1657
04F2 8B1E8900        R    1658     RX12:   MOV     BX,FREE_HEAD    ;READ NEXT AVAILABLE BUFFER
04F6 0BDB                 1659             OR      BX,BX           ;CHECK THAT A BUFFER IS THERE
04F8 7512                 1660             JNZ     RX13            ;IF A BUFFER IS AVAILABLE
                          1661
                          1662     ;       NO BUFFER TO REINIT WITH
                          1663
04FA FE069700        R    1664             INC     BUF_COUNT       ;MARK THAT AN ADDITIONAL CHANNEL NEEDS INITING
04FE 803E970001      R    1665             CMP     BUF_COUNT,1     ;CHECK IF OTHER CHANNELS ALREADY NEED INITING
0503 7520                 1666             JNZ     RX14            ;IF OTHERS, SKIP SETTING BUF_FIRST
0505 C606960001      R    1667             MOV     BUF_FIRST,1     ;MARK CHANNEL 1 AS FIRST CHANNEL TO INIT
050A EB19                 1668             JMP     SHORT RX14
                          1669
                          1670     ;       SET UP WITH NEW BUFFER
                          1671
050C 8B4704               1672     RX13:   MOV     AX,[BX].BUF     ;OUTPUT STARTING ADDRESS
050F E6C2                 1673             OUT     CH1ADDR,AL
0511 8AC4                 1674             MOV     AL,AH
0513 E6C2                 1675             OUT     CH1ADDR,AL
```

```
LOC  OBJ                  LINE     SOURCE

0515 B8F105               1676             MOV     AX,MAX_PACKET_LEN+JUNK_BYTES ;OUTPUT LENGTH TO DMA
0518 E6C3                 1677             OUT     CH1WC,AL
051A 8AC4                 1678             MOV     AL,AH
051C E6C3                 1679             OUT     CH1WC,AL
051E E602                 1680             OUT     SET_RXAV1,AL    ;SIGNAL CHANNEL AVAILABLE
                          1681
                          1682     ;       DO LINKED LIST HOUSEKEEPING
                          1683
0520 8307                 1684             MOV     AX,[BX].LINK    ;UNLINK NEW ELEMENT FROM FREE LIST
0522 A38900         R     1685             MOV     FREE_HEAD,AX
0525 871E8F0C       R     1686     RX14:   XCHG    BX,RX1_BUF      ;SAVE NEW BUFFER CB POINTER AND READ OLD
0529 894F02               1687             MOV     [BX].WC,CX      ;SAVE WORD COUNT
052C 88368D0C       R     1688             MOV     SI,RCVD_TAIL    ;LINK OLD PACKET INTO RCVD QUEUE
0530 891C                 1689             MOV     [SI].LINK,BX
0532 C7070000             1690             MOV     [BX].LINK,0
0536 891E8D0C       R     1691             MOV     RCVD_TAIL,BX
                          1692
                          1693     ;       CHECK FOR NEXT CHANNEL COMPLETE
                          1694
053A B0FD                 1695     RX15:   MOV     AL,NOT CH2_DONE ;OPEN MASK FOR CHANNEL 2 ONLY
053C E6F1                 1696             OUT     PICMASK,AL
053E B00C                 1697             MOV     AL,POLL_PIC     ;POLL TO SEE IF CHANNEL 2 DONE
0540 E6F0                 1698             OUT     PICCMD,AL
0542 E4F0                 1699             IN      AL,PICDATA
0544 3481                 1700             XOR     AL,81H          ;TEST IF INTERRUPT 1 ASSERTED
0546 7409                 1701             JZ      RX2ISR          ;IF CHANNEL 2 COMPLETE
0548 C7066D0C5105   R     1702             MOV     NEXT_CH,OFFSET CGROUP:RX2ISR ;NEXT VECTOR
054E E94EFF               1703             JMP     RXEXIT          ;EXIT PROCESSING
                          1704
                          1705     $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          1706     ;;      RX2ISR - RECEIVE CHANNEL 2 INTERRUPT SERVICE ROUTINE
                          1707     ;
                          1708     ;       PARAMETERS:
                          1709     ;          NONE.
                          1710
                          1711
0551 B061                 1712     RX2ISR: MOV     AL,SEOI_PIC+1   ;SELECTIVE EOI FOR THIS CHANNEL
0553 E6F0                 1713             OUT     PICCMD,AL       ;SEND TO PIC
0555 E4C5                 1714             IN      AL,CH2WC        ;READ WORD COUNT OF RECEIVED PACKET IN CX
0557 8AC8                 1715             MOV     CL,AL
0559 E4C5                 1716             IN      AL,CH2WC
055B 8AE8                 1717             MOV     CH,AL
                          1718
                          1719     ;       CHECK FOR ERRORS
                          1720
055D E4E0                 1721             IN      AL,PIOA         ;READ STATUS
055F 2430                 1722             AND     AL,30H          ;CHECK RECEIVE STATUS BITS
0561 7409                 1723             JZ      RX20            ;IF STATUS OK
0563 B404                 1724             MOV     AH,CH3_DONE     ;NEXT CHANNEL FOLLOWING THIS ONE
0565 E81C02               1725             CALL    CHECK_STATUS    ;KICK APPROPRIATE COUNTERS
0568 0AC0                 1726             OR      AL,AL           ;CHECK RETURNED VALUE
056A 7406                 1727             JZ      RX21            ;ERROR ON THIS PACKET
                          1728
                          1729     ;       CHECK FOR RUNT PACKET
                          1730
056C 81F9AE05             1731     RX20:   CMP     CX,MAX_PACKET_LEN-MIN_PACKET_LEN  ;CHECK FOR RUNT PACKET
0570 7E0B                 1732             JLE     RX22            ;IF NOT A RUNT
                          1733
                          1734     ;       REINITIALIZE AFTER ERROR
                          1735
0572 8B1E910C        R    1736     RX21:   MOV     BX,RX2_BUF      ;LOAD BUFFER CONTROL BLOCK POINTER
0576 B202                 1737             MOV     DL,2            ;CHANNEL 2
0578 E8EC01               1738             CALL    RESET_CHANNEL   ;RESET DMA CONTROLLER
057B EB48                 1739             JMP     SHORT RX25      ;DO END PROCESSING
                          1740
                          1741     ;       PACKET OK
                          1742
057D 8B1E890C        R    1743     RX22:   MOV     BX,FREE_HEAD    ;READ NEXT AVAILABLE BUFFER
0581 0BDB                 1744             OR      BX,BX           ;CHECK THAT A BUFFER IS THERE
0583 7512                 1745             JNZ     RX23            ;IF A BUFFER IS AVAILABLE
                          1746
                          1747     ;       NC BUFFER TO REINIT WITH
                          1748
0585 FE06970C        R    1749             INC     BUF_COUNT       ;MARK THAT AN ADDITIONAL CHANNEL NEEDS INITING
0589 803E970001      R    1750             CMP     BUF_COUNT,1     ;CHECK IF OTHER CHANNELS ALREADY NEED INITING
058E 7520                 1751             JNZ     RX24            ;IF OTHERS, SKIP SETTING BUF_FIRST
0590 C606960C02      R    1752             MOV     BUF_FIRST,2     ;MARK CHANNEL 2 AS FIRST CHANNEL TO INIT
0595 EB19                 1753             JMP     SHORT RX24
                          1754
                          1755     ;       SET UP WITH NEW BUFFER
                          1756
0597 8B4704               1757     RX23:   MOV     AX,[BX].BUF     ;OUTPUT STARTING ADDRESS
059A E6C4                 1758             OUT     CH2ADDR,AL
059C 8AC4                 1759             MOV     AL,AH
059E E6C4                 1760             OUT     CH2ADDR,AL
```

```
LOC  OBJ                  LINE     SOURCE

05A0 B8F105               1761             MOV     AX,MAX_PACKET_LEN+JUNK_BYTES ;OUTPUT LENGTH TO DMA
05A3 E6C5                 1762             OUT     CH2WC,AL
05A5 8AC4                 1763             MOV     AL,AH
05A7 E6C5                 1764             OUT     CH2WC,AL
05A9 E603                 1765             OUT     SET_RXAV2,AL    ;SIGNAL CHANNEL AVAILABLE
                          1766
                          1767     ;       DO LINKED LIST HOUSEKEEPING
                          1768
05AB 8B07                 1769             MOV     AX,[BX].LINK    ;UNLINK NEW ELEMENT FROM FREE LIST
05AD A38900         R     1770             MOV     FREE_HEAD,AX
05B0 871E910C       R     1771     RX24:   XCHG    BX,RX2_BUF      ;SAVE NEW BUFFER CB POINTER AND READ OLD
05B4 894F02               1772             MOV     [BX].WC,CX      ;SAVE WORD COUNT
05B7 8B368D00       R     1773             MOV     SI,RCVD_TAIL    ;LINK OLD PACKET INTO RCVD QUEUE
05BB 891C                 1774             MOV     [SI].LINK,BX
05BD C7070000             1775             MOV     [BX].LINK,0
05C1 891E8D00       R     1776             MOV     RCVD_TAIL,BX
                          1777
                          1778     ;       CHECK FOR NEXT CHANNEL COMPLETE
                          1779
05C5 B0FB                 1780     RX25:   MOV     AL,NOT CH3_DONE ;OPEN MASK FOR CHANNEL 3 ONLY
05C7 E6F1                 1781             OUT     PICMASK,AL
05C9 B00C                 1782             MOV     AL,POLL_PIC     ;POLL TO SEE IF CHANNEL 3 DONE
05CB E6F0                 1783             OUT     PICCMD,AL
05CD E4F0                 1784             IN      AL,PICDATA
05CF 3482                 1785             XOR     AL,82H          ;TEST IF INTERRUPT 3 ASSERTED
05D1 7409                 1786             JZ      RX3ISR          ;IF CHANNEL 3 COMPLETE
05D3 C7066D00CD05   R     1787             MOV     NEXT_CH,OFFSET CGROUP:RX3ISR ;NEXT VECTOR
05D9 E9C3FE               1788             JMP     RXEXIT          ;EXIT PROCESSING
                          1789
                          1790     $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                          1791     ;;      RX3ISR - RECEIVE CHANNEL 3 INTERRUPT SERVICE ROUTINE
                          1792     ;
                          1793     ;       PARAMETERS:
                          1794     ;          NONE.
                          1795
                          1796
05DC B062                 1797     RX3ISR: MOV     AL,SEOI_PIC+2   ;SELECTIVE EOI FOR THIS CHANNEL
05DE E6F0                 1798             OUT     PICCMD,AL       ;SEND TO PIC
05E0 E4C7                 1799             IN      AL,CH3WC        ;READ WORD COUNT OF RECEIVED PACKET IN CX
05E2 8AC8                 1800             MOV     CL,AL
05E4 E4C7                 1801             IN      AL,CH3WC
05E6 8AE8                 1802             MOV     CH,AL
                          1803
                          1804     ;       CHECK FOR ERRORS
                          1805
05E8 E4E0                 1806             IN      AL,PIOA         ;READ STATUS
05EA 2430                 1807             AND     AL,30H          ;CHECK RECEIVE STATUS BITS
05EC 7409                 1808             JZ      RX30            ;IF STATUS OK
05EE B401                 1809             MOV     AH,CH1_DONE     ;NEXT CHANNEL FOLLOWING THIS ONE
05F0 E89101               1810             CALL    CHECK_STATUS    ;KICK APPROPRIATE COUNTERS
05F3 0AC0                 1811             OR      AL,AL           ;CHECK RETURNED VALUE
05F5 7406                 1812             JZ      RX31            ;IF ERROR ON THIS PACKET
                          1813
                          1814     ;       CHECK FOR RUNT PACKET
                          1815
05F7 81F9AE05             1816     RX30:   CMP     CX,MAX_PACKET_LEN-MIN_PACKET_LEN  ;CHECK FOR RUNT PACKET
05FB 7E0B                 1817             JLE     RX32            ;IF NOT A RUNT
                          1818
                          1819     ;       REINITIALIZE AFTER ERROR
                          1820
05FD 8B1E930C        R    1821     RX31:   MOV     BX,RX3_BUF      ;LOAD BUFFER CONTROL BLOCK POINTER
0601 B203                 1822             MOV     DL,3            ;CHANNEL 3
0603 E86101               1823             CALL    RESET_CHANNEL   ;RESET DMA CONTROLLER
0606 EB48                 1824             JMP     SHORT RX35      ;DO END PROCESSING
                          1825
                          1826     ;       PACKET OK
                          1827
0608 8B1E890C        R    1828     RX32:   MOV     BX,FREE_HEAD    ;READ NEXT AVAILABLE BUFFER
060C 0BDB                 1829             OR      BX,BX           ;CHECK THAT A BUFFER IS THERE
060E 7512                 1830             JNZ     RX33            ;IF A BUFFER IS AVAILABLE
                          1831
                          1832     ;       NO BUFFER TO REINIT WITH
                          1833
0610 FE069700        R    1834             INC     BUF_COUNT       ;MARK THAT AN ADDITIONAL CHANNEL NEEDS INITING
0614 803E970001      R    1835             CMP     BUF_COUNT,1     ;CHECK IF OTHER CHANNELS ALREADY NEED INITING
0619 7520                 1836             JNZ     RX34            ;IF OTHERS, SKIP SETTING BUF_FIRST
061B C606960003      R    1837             MOV     BUF_FIRST,3     ;MARK CHANNEL 3 AS FIRST CHANNEL TO INIT
0620 EB19                 1838             JMP     SHORT RX34
                          1839
                          1840     ;       SET UP WITH NEW BUFFER
                          1841
0622 8B4704               1842     RX33:   MOV     AX,[BX].BUF     ;OUTPUT STARTING ADDRESS
0625 E6C6                 1843             OUT     CH3ADDR,AL
0627 8AC4                 1844             MOV     AL,AH
0629 E6C6                 1845             OUT     CH3ADDR,AL
```

```
LOC  OBJ                  LINE     SOURCE

062B B8F105               1846             MOV     AX,MAX_PACKET_LEN+JUNK_BYTES ;OUTPUT LENGTH TO DMA
062E E6C7                 1847             OUT     CH3WC,AL
0630 8AC4                 1848             MOV     AL,AH
0632 E6C7                 1849             OUT     CH3WC,AL
0634 E604                 1850             OUT     SET_RXAV3,AL    ;SIGNAL CHANNEL AVAILABLE
                          1851
                          1852     ;       DO LINKED LIST HOUSEKEEPING
                          1853
0636 8807                 1854             MOV     AX,[BX].LINK    ;UNLINK NEW ELEMENT FROM FREE LIST
0638 A38900         R     1855             MOV     FREE_HEAD,AX
063B 871E930C       R     1856     RX34:   XCHG    BX,RX3_BUF      ;SAVE NEW BUFFER CB POINTER AND READ OLD
063F 894F02               1857             MOV     [BX].WC,CX      ;SAVE WORD COUNT
0642 8B368D0C       R     1858             MOV     SI,RCVD_TAIL    ;LINK OLD PACKET INTO RCVD QUEUE
0646 891C                 1859             MOV     [SI].LINK,BX
0648 C7070000             1860             MOV     [BX].LINK,0
064C 891E8D0C       R     1861             MOV     RCVD_TAIL,BX
                          1862
                          1863     ;       CHECK FOR NEXT CHANNEL COMPLETE
                          1864
0650 B0FE                 1865     RX35:   MOV     AL,NOT CH1_DONE ;OPEN MASK FOR CHANNEL 1 ONLY
0652 E6F1                 1866             OUT     PICMASK,AL
0654 B00C                 1867             MOV     AL,POLL_PIC     ;POLL TO SEE IF CHANNEL 1 DONE
0656 E6F0                 1868             OUT     PICCMD,AL
0658 E4F0                 1869             IN      AL,PICDATA
065A 3480                 1870             XOR     AL,80H          ;TEST IF INTERRUPT 0 ASSERTED
065C 7409                 1871             JZ      RX36            ;IF CHANNEL 1 COMPLETE
065E C7066D0CC604   R     1872             MOV     NEXT_CH,OFFSET CGROUP:RX1ISR ;NEXT VECTOR
0664 E938FE               1873             JMP     RXEXIT          ;EXIT PROCESSING
0667 E95CFE               1874     RX36:   JMP     RX1ISR          ;NEED LONG JUMP
                          1875
                          1876
                          1877
                          1878
                          1879     $EJECT TITLE ('DLL - RECEIVE PACKET FORWARD PROCESS')
```

```
LOC  OBJ                  LINE     SOURCE

                          1880     ;;      DLLRXF - PROCESS TO  FORWARD RECEIVED PACKETS TO NETWORK LAYER.
                          1881     ;
                          1882     ;         THIS IS THE PROCESS WHICH MOVES RECEIVED PACKETS IN THE "RCVD"
                          1883     ;       TO RCVDMBX,  AFTER DOING ADDRESS CHECKING ON MULTICAST PACKETS.
                          1884     ;
                          1885     ;       GLOBALS MODIFIED: RCVD_HEAD, RCVD_TAIL.
                          1886     ;       KADS OBJECTS USED: DLL_SEM1 (WAITSEM).
                          1887
                          1888
                          1889             PUBLIC  DLLRXF
066A FA                   1890     RXF1:   CLI                     ;THIS PART MUST BE INDIVISIBLE
066B 8B07                 1891             MOV     AX,[BX].LINK    ;UNLINK BCB FROM QUEUE
066D A33B00         R     1892             MOV     RCVD_HEAD,AX
0670 0BC0                 1893             OR      AX,AX           ;CHECK IF AT END OF QUEUE
0672 7506                 1894             JNZ     RXF2            ;IF NOT AT END
0674 C7068D0C8B00   R     1895             MOV     RCVD_TAIL,OFFSET DGROUP:RCVD_HEAD ;REINIT QUEUE
                          1896
067A A18700         R     1897     RXF2:   MOV     AX,BCB_HEAD     ;LINK BCB INTO FREE LIST
067D 8907                 1898             MOV     [BX].LINK,AX
067F 891E870C       R     1899             MOV     BCB_HEAD,BX
                          1900
0683 8B7704               1901             MOV     SI,[BX].BUF     ;GET ACTUAL BUFFER ADDRESS
0686 B8EA05               1902             MOV     AX,MAX_PACKET_LEN ;COMPUTE PACKET LENGTH
0689 2B4702               1903             SUB     AX,[BX].WC      ;SUBTRACT DMA FINAL WORD COUNT
068C FB                   1904             STI                     ;CAN TURN INTERRUPTS BACK ON HERE
                          1905
068D 81C6FBFF       R     1906             ADD     SI,OFFSET DGROUP:BUFFER_START-VAL_DL_DEST+1 ;WANT ADDRESS IN DGROUP
0691 894404               1907             MOV     [SI].SEG_LENGTH,AX ;STORE LENGTH IN SEGMENT
                          1908
0694 F6440601             1909             TEST    BYTE PTR [SI].DL_DEST,MCFLAG ;TEST MC FLAG
0698 7410                 1910             JZ      RXF3            ;IF SINGLE ADDRESS
069A 56                   1911             PUSH    SI              ;SAVE POINTER
069B 8D5C06               1912             LEA     BX,[SI].DL_DEST
069E E8A5FA               1913             CALL    CHECKMCID       ;CHECK FOR MULTICAST ID
06A1 0A069800       R     1914             OR      AL,ALLMC        ;CHECK "ALL MULTICAST" FLAG
06A5 5E                   1915             POP     SI
06A6 D0C8                 1916             ROR     AL,1            ;TEST RESULT
06A8 732F                 1917             JNC     RXF4            ;IF NOT VALID MCID
                          1918
06AA 8B4412               1919     RXF3:   MOV     AX,[SI].DL_TYPE ;LOAD DESTINATION TYPE
06AD BFCD00         R     1920             MOV     DI,OFFSET DGROUP:TYPET ;SEARCH TYPE LIST
06B0 8B0EC80C       R     1921             MOV     CX,TYPEL
06B4 833D00               1922     RXF3A:  CMP     WORD PTR [DI],0 ; SEE IF TYPE IS ZERO
06B7 740B                 1923             JE      RXF3B   ; IF IS ZERO, THEN TAKE IT
06B9 3905                 1924             CMP     WORD PTR [DI],AX
06BB 7407                 1925             JE      RXF3B   ; TYPE MATCHES
06BD 47                   1926             INC     DI
06BE 47                   1927             INC     DI
06BF 49                   1928             DEC     CX
06C0 75F2                 1929             JNZ     RXF3A
06C2 EB15                 1930             JMP     SHORT RXF4      ; NOT IN LISTT
                          1931
06C4 8B4510               1932     RXF3B:  MOV     AX,[DI]+(TYPEM-TYPET)   ;SEND TO OUTPUT MAILBOX
06C7 50                   1933             PUSH    AX
06C8 1E                   1934             PUSH    DS
```

```
LOC  OBJ                  LINE     SOURCE

06C9 56                   1935             PUSH    SI
06CA E80000         E     1936             CALL    CQSEND
                          1937
06CD FF06520C       R     1938             INC     TOTAL_RECEIVED  ;INCREMENT TOTAL COUNTER
06D1 750A                 1939             JNZ     RXF5            ;IF NO OVERFLOW
06D3 FF065400       R     1940             INC     TOTAL_RECEIVED+2 ;INCREMENT HIGH PART
06D7 EB04                 1941             JMP     SHORT RXF5
                          1942
06D9 56                   1943     RXF4:   PUSH    SI              ;RETURN BUFFER
06DA E852FD               1944             CALL    DLLRXRETBUF
                          1945
06DD 8B1E8B0C       R     1946     RXF5:   MOV     BX,RCVD_HEAD    ;READ NEXT PACKET TO FORWARD
06E1 0BDB                 1947             OR      BX,BX           ;TEST FOR NONE TO FORWARD
06E3 7585                 1948             JNZ     RXF1            ;IF SOME TO FORWARD
                          1949
06E5                      1950     DLLRXF:
06E5 B83000         R     1951             MOV     AX,OFFSET DGROUP:DLL_SEM1 ;WAIT FOR RECEIVE COMPLETE
06E8 50                   1952             PUSH    AX
06E9 E80000         E     1953             CALL    CQWAITSEM
06EC EBEF                 1954             JMP     SHORT RXF5      ;ENTER LOOP
                          1955
                          1956
                          1957
                          1958
                          1959     $EJECT TITLE ('DLL - RECEIVE SERVICE SUBROUTINES')
```

```
LOC  OBJ                  LINE     SOURCE

                          1960     ;;      LOOP_PACKET - LOOP PACKET TO RECEIVE SIDE.
                          1961     ;
                          1962     ;          THIS ROUTINE COPIES THE PACKET PASSED TO IT INTO
                          1963     ;       A RECEIVE PACKET BUFFER, THEN FORWARDS THE COPY
                          1964     ;       AS A NORMAL RECEIVED PACKET. THE ORIGINAL PACKET
                          1965     ;       REMAINS UNCHANGED.
                          1966     ;
                          1967     ;       PARAMETER:
                          1968     ;          BX - OFFSET OF PACKET BUFFER CONTAINING
                          1969     ;               THE PACKET TO BE LOOPED BACK.
                          1970
                          1971
06EE                      1972     LOOP_PACKET:
06EE 53                   1973             PUSH    BX              ;SAVE PACKET POINTER
06EF 9C                   1974             PUSHF                   ;CRITICAL REGION
06F0 FA                   1975             CLI
06F1 8B1E890C        R    1976             MOV     BX,FREE_HEAD    ;READ NEXT AVAILABLE BUFFER
06F5 0BDB                 1977             OR      BX,BX           ;CHECK THAT A BUFFER IS THERE
06F7 7408                 1978             JZ      LOP1            ;IF NO BUFFER
06F9 8B07                 1979             MOV     AX,[BX].LINK    ;UNLINK FROM FREE LIST
06FB A38900          R    1980             MOV     FREE_HEAD,AX
06FE 9D                   1981             POPF
06FF EB0C                 1982             JMP     SHORT LOP2
                          1983
0701 9D                   1984     LOP1:   POPF
0702 FE069500        R    1985             INC     BUF_LOOP        ;FLAG THAT ANOTHER LOOPBACK BUFFER IS NEEDED
0706 B82000          R    1986             MOV     AX,OFFSET DGROUP:DLL_MBX2 ;WAIT AT MAILBOX FOR BUFFER
0709 50                   1987             PUSH    AX
070A E80000          E    1988             CALL    CQRECEIVE
                          1989
070D FC                   1990     LOP2:   CLD                     ;SET UP FOR COPY
070E 1E                   1991             PUSH    DS
070F 07                   1992             POP     ES
0710 8B7F04               1993             MOV     DI,[BX].BUF
0713 81C70100        R    1994             ADD     DI,OFFSET DGROUP:BUFFER_START+1 ;ADDRESS IN DGROUP
0717 5E                   1995             POP     SI              ;SOURCE BUFFER ADDRESS
0718 8B4C04               1996             MOV     CX,[SI].SEG_LENGTH
071B B8EA05               1997             MOV     AX,MAX_PACKET_LEN ;COMPUTE VALUE FOR "WC"
071E 2BC1                 1998             SUB     AX,CX
0720 894702               1999             MOV     [BX].WC,AX      ;PUT LENGTH INTO BUFFER CONTROL BLOCK
0723 8D7406               2000             LEA     SI,[SI].DL_DEST
0726 F3                   2001             REP     MOVSB           ;DO THE COPY
0727 A4
                          2002
0728 9C                   2003             PUSHF                   ;CRITICAL REGION
0729 FA                   2004             CLI
072A 8B368D00        R    2005             MOV     SI,RCVD_TAIL    ;LINK INTO OUTPUT LIST
072E 891C                 2006             MOV     [SI].LINK,BX
0730 C7070000             2007             MOV     [BX].LINK,0
0734 891E8D00        R    2008             MOV     RCVD_TAIL,BX
0738 9D                   2009             POPF
                          2010
0739 B83000          R    2011             MOV     AX,OFFSET DGROUP:DLL_SEM1 ;SIGNAL THAT A PACKET IS AVAILABLE
073C 50                   2012             PUSH    AX
073D E80000          E    2013             CALL    CQSIGNAL
```

```
LOC  OBJ                  LINE     SOURCE

0740 C3                   2014             RET
                          2015
                          2016
                          2017
                          2018
                          2019     ;;       SETUPCHANNEL - SET UP DMA FOR CHANNEL AND SIGNAL AVAILABLE.
                          2020     ;
                          2021     ;        PARAMETERS:
                          2022     ;           (DL) = CHANNEL NUMBER.
                          2023     ;           (BX) = POINTER TO BUFFER CONTROL BLOCK.
                          2024     ;
                          2025     ;        USES: AX,DX,SI
                          2026
                          2027
0741                      2028     SETUPCHANNEL:
0741 32F6                 2029             XOR     DH,DH              ;FILL DH WITH ZEROS
0743 8BF2                 2030             MOV     SI,DX              ;SAVE CHANNEL NUMBER IN SI
0745 D1E2                 2031             SHL     DX,1               ;COMPUTE CHnADDR PORT NUMBER
0747 81C2C00C             2032             ADD     DX,CH1ADDR-2
074B 8B4704               2033             MOV     AX,[BX].BUF        ;LOAD STARTING ADDRESS
074E EE                   2034             OUT     DX,AL
074F 8AC4                 2035             MOV     AL,AH
0751 EE                   2036             OUT     DX,AL
                          2037
0752 42                   2038             INC     DX                 ;STEP TO CHnWC
0753 B8F105               2039             MOV     AX,MAX_PACKET_LEN+JUNK_BYTES ;LOAD LENGTH
0756 EE                   2040             OUT     DX,AL
0757 8AC4                 2041             MOV     AL,AH
0759 EE                   2042             OUT     DX,AL
                          2043
075A BA0100               2044             MOV     DX,SET_RXAV1-1     ;COMPUTE RX AVAIL PORT NUMBER
075D 03D6                 2045             ADD     DX,SI
075F EE                   2046             OUT     DX,AL              ;SIGNAL CHANNEL AVAILABLE
                          2047
0760 D1E6                 2048             SHL     SI,1               ;INITIALIZE RXn_BUF
0762 899C8D0C       R     2049             MOV     RX1_BUF[SI-2],BX
0766 C3                   2050             RET
                          2051
                          2052
                          2053
                          2054
                          2055     ;;       RESET_CHANNEL - RETURN BUFFER TO CHANNEL AFTER ERROR
                          2056     ;
                          2057
                          2058
0767                      2059     RESET_CHANNEL:
0767 803E970000     R     2060             CMP     BUF_COUNT,0        ;CHECK IF OTHER CHANNELS NEED INITING
076C 74D3                 2061             JZ      SETUPCHANNEL       ;SET UP CHANNEL IF ONLY ONE NEEDING INITING
076E 8A16960C       R     2062             MOV     DL,BUF_FIRST       ;OTHERWISE, INIT THE FIRST ONE WHICH NEEDS BUF
0772 FE06960C       R     2063             INC     BUF_FIRST          ;STEP TO NEXT
0776 803E960004     R     2064             CMP     BUF_FIRST,4        ;CHECK FOR WRAP-AROUND
077B 75C4                 2065             JNZ     SETUPCHANNEL       ;EXIT IF NOT WRAP-AROUND
077D C606960C01     R     2066             MOV     BUF_FIRST,1        ;WRAP AROUND
0782 EB3D                 2067             JMP     SHORT SETUPCHANNEL
                          2068
```

```
LOC  OBJ                  LINE     SOURCE

                          2069
                          2070
                          2071
                          2072     ;;      CHECK_STATUS - IF ERROR APPLIES TO THIS CHANNEL, INCREMENT APPROPRIATE
                          2073     ;                      ERROR COUNTERS AND CLEAR ERROR INDICATIONS.
                          2074     ;
                          2075     ;       PARAMETERS:
                          2076     ;          AL = STATUS BYTE.
                          2077     ;          AH = INTERRUPT CHECK MASK FOR THE NEXT CHANNEL.
                          2078     ;          CX = WORD COUNT FROM DMA.
                          2079     ;
                          2080     ;       RETURNS:
                          2081     ;          AL = TRUE => ERROR NOT FOR THIS PACKET.
                          2082     ;          AL = FALSE => DROP THIS PACKET.
                          2083     ;          CX = WORD COUNT FROM DMA.
                          2084
                          2085
0784                      2086     CHECK_STATUS:
0784 8AD8                 2087             MOV     BL,AL           ;SAVE FLAGS
0786 B00A                 2088             MOV     AL,READ_IRR     ;CHECK IF NEXT CHANNEL COMPLETE;
0788 E6F0                 2089             OUT     PICCMD,AL       ;  IF SO, ERROR FLAGS DO NOT APPLY TO THIS ONE
078A E4F0                 2090             IN      AL,PICDATA
078C 84C4                 2091             TEST    AL,AH           ;CHECK WITH THE MASK WHICH WAS PASSED
078E 7521                 2092             JNZ     CHK3            ;IF NEXT CHANNEL COMPLETE
                          2093
0790 F6C310               2094             TEST    BL,10H          ;CHECK LENGTH ERROR
0793 7406                 2095             JZ      CHK1            ;IF NO LENGTH ERROR
0795 BF5800         R     2096             MOV     DI,OFFSET DGROUP:FRMERRS                ;INCREMENT COUNTER
0798 E8EBF8               2097             CALL    BUMPCNT
079B F6C320               2098     CHK1:   TEST    BL,20H          ;CHECK CRC ERROR
079E 740C                 2099             JZ      CHK2            ;IF NO CRC ERROR
07A0 81F9AE05             2100             CMP     CX,MAX_PACKET_LEN-MIN_PACKET_LEN  ;CHECK FOR RUNT PACKET
07A4 7F06                 2101             JG      CHK2            ;DO NOT INCREMENT COUNTER FOR RUNT
07A6 BF5600         R     2102             MOV     DI,OFFSET DGROUP:CRCERRS                ;INCREMENT COUNTER
07A9 E8DAF8               2103             CALL    BUMPCNT
07AC E605                 2104     CHK2:   OUT     RESET_ERROR,AL  ;RESET ERROR FLAGS
07AE 32C0                 2105             XOR     AL,AL           ;RETURN FALSE
07B0 C3                   2106             RET
                          2107
07B1 B0FF                 2108     CHK3:   MOV     AL,0FFH         ;RETURN TRUE
07B3 C3                   2109             RET
                          2110     ;
                          2111     ;
                          2112     ; defs for puconf when real module is not present
                          2113     ;
                          2114     ;       public  cqdevicetest,cqramtest,ctjumptable
                          2115     ;cqdevicetest:
                          2116     ;       xor     ax,ax
                          2117     ;       ret
                          2118     ;cqramtest:
                          2119     ;       xor     ax,ax
                          2120     ;       ret
                          2121     ;ctjumptable    dw      0
                          2122
                          2123
```

```
LOC  OBJ                  LINE      SOURCE

----                      2124      CODE    ENDS
                          2125              END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```