

File created: 26-Feb-91 13:32:30 {DSK}<users>sybalsky>3-BYTE-ATOM-CHANGES>10MBDRIVER.;1

changes to: (FNS \HANDLE.RAW.3TO10)

previous date: 5-Sep-90 11:30:30 {PELE:MV:ENVOS}<LISPCORE>SOURCES>10MBDRIVER.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
:: Copyright (c) 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1990, 1991 by Venue & Xerox Corporation. All rights reserved.

### (RPAQQ 10MBDRIVERCOMS

```
((COMS ; raw packet interface
(FNS \10MBGETPACKET \10MSENDPACKET \10MBENCAPSULATE \10MB.BROADCASTP \10MBWATCHER))
(COMS ; Machine independent part
(FNS \10MB.STARTDRIVER \10MB.CREATENDB \10MB.INPUT.INTERRUPT \10MB.OUTPUT.INTERRUPT \10MB.NOTESTAT
)
(INITVARS (\10MB.RCLK.BOX (CREATECELL \FIXP))
(\10MB.EXPECTED.RECEIVE.INTERVAL 60)
(\10MB.INPUT.TIMEOUT (TIMES \RCLKSECOND \10MB.EXPECTED.RECEIVE.INTERVAL))
(\10MB.INPUT.TIMER (SETUPTIMER 0)))
(GLOBALVARS \10MB.RCLK.BOX \10MB.EXPECTED.RECEIVE.INTERVAL \10MB.INPUT.TIMEOUT \10MB.INPUT.TIMER))
(COMS ; Buffer management
(FNS \10MB.LOADINPUTQ \RELEASE.IOCB \GET.IOCB \INIT.ETHER.BUFFER.POOL)
(INITVARS (\10MBPACKETLENGTH 488)))
(COMS ; Dolphin/Dlion head
(FNS \10MB.GETPACKETLENGTH \10MB.GETPACKETSTATUS \QUEUE.INPUT.IOCB \QUEUE.OUTPUT.IOCB
\10MB.TURNOFFETHER \10MB.TURNONETHER \10MB.RESTART.ETHER))
(COMS ; Misc
(FNS \NOMACHINETYPE \10MB.PRINT.ENCAPSULATION)
(FNS IOCBLENGTH))
(INITVARS \10MB.GETGARBAGE \10MB.COLLECTSTATS)
[COMS ; PUP address resolution
(FNS \HANDLE.RAW.3TO10 \TRANSLATE.3TO10 PRINT3TO10 \NOTE.3TO10)
(INITVARS (\10MBTYPE.PUP 512)
(\10MBTYPE.3TO10 513)
(\10MBTYPE.TRANSLATIONS))
(GLOBALVARS \10MBTYPE.3TO10 \10MBTYPE.PUP \10MBTYPE.TRANSLATIONS)
(CONSTANTS \EPT.3TO10)
(ADDVARS (\PACKET.PRINTERS (513 . PRINT3TO10))
(DECLARE%: EVAL@COMPILE DONTCOPY (LOCALVARS . T)
(FILE (SOURCE)
10MBDECLS DOVEDECLS DOVEETHERDECLS LLNSDECLS)
(FILE (LOADCOMP)
LLETER)))
```

:: raw packet interface

(DEFINEQ

### (\10MBGETPACKET

```
[LAMBDA NIL ; Edited 31-Mar-87 18:11 by bvm:
(LET (PACKET TYPE)
(COND
((SETQ PACKET (\DEQUEUE \10MB.RAWPACKETQ))
(replace EPTYPE of PACKET with (for PAIR in \10MBTYPE.TRANSLATIONS
bind (TYPE _ (fetch 10MBTYPE of PACKET))
when (EQ TYPE (CAR PAIR)) do
;; TYPE is the raw type of the etherpacket. These do not always correspond one-to-one with the
;; EPTYPE constants we use (in particular, for pups), so translate if necessary.
(RETURN (CDR PAIR))
finally (RETURN TYPE))))
[COND
(\RAWTRACING (\MAYBEPRINTPACKET PACKET 'RAWGET]
PACKET])
```

### (\10MSENDPACKET

```
[LAMBDA (NDB PACKET) ; Edited 14-Apr-88 15:30 by bvm
(PROG ([DROPIT (AND \ETHERLIGHTNING (EQ 0 (RAND 0 \ETHERLIGHTNING)
IOCB BUFLLENGTH)
[COND
(\RAWTRACING (\MAYBEPRINTPACKET PACKET 'RAWPUT]
[COND
([AND (NOT (fetch NDBCANHEARSELF of NDB))
(OR (fetch 10MBMULTICASTP of PACKET)
(EQNSADDRESS.HOST \MY.NSADDRESS (fetch 10MBDESTNSADDRESSBASE of PACKET)
; We would hear this packet if our hardware let us, so fake
; receipt
(PROG ((COPYPACKET (\ALLOCATE.ETHERPACKET)))
(\BLT (LOCF (fetch 10MBLENGTH of COPYPACKET))
```

```

                (LOCF (fetch 10MBLENGTH of PACKET))
                (ADD1 (fetch 10MBLENGTH of PACKET)) ; Copy all data that would have been transmitted
                (replace EPNETWORK of COPYPACKET with NDB)
                (\ENQUEUE \10MB.RAWPACKETQ COPYPACKET]
(UNINTERRUPTABLY
  (replace EPTRANSMITTING of PACKET with T)
  (COND
    ([OR DROPT (NULL (SETQ IOCB (\GET.IOCB 'OUTPUT]
                                ; Fake transmission
                                (replace EPNETWORK of PACKET with NIL))
    (T (replace EPNETWORK of PACKET with IOCB)
        (SETQ BUFLNGTH (IMAX (fetch 10MBLENGTH of PACKET)
                             \10MB.MINPACKETLENGTH))
        (\TEMPLOCKPAGES PACKET (COND
          ((IGEQL BUFLNGTH \MIN2PAGEBUFLNGTH)
           2)
          (T 1))) ; Put on microcode queue
        (\QUEUE.OUTPUT.IOCB NDB IOCB (fetch 10MBPACKETBASE of PACKET)
         BUFLNGTH)
        T))
    (\ENQUEUE (fetch NDBTQ of NDB)
              PACKET) ; Put on driver's queue to pick up after microcode finishes with it
  )
  (RETURN (AND IOCB T])

```

**(\10MBENCAPSULATE**

```

[LAMBDA (NDB PACKET PDH LENGTH TYPE) ; Edited 14-Jan-88 15:35 by bvm
;; Encapsulates PACKET for transmission on 10mb net NDB: Fill in destination host (48 bits), source host (me, 48 bits), LENGTH (in bytes) and
;; protocol TYPE. PDH can be an NSHOSTNUMBER or any piece of storage resembling an NSADDRESS.
(if (type? NSHOSTNUMBER PDH)
  then (replace 10MBDESTHOST of PACKET with PDH)
  else ; PDH is an NSADDRESS object or piece of storage
    (\BLT (fetch 10MBDESTHOSTBASE of PACKET)
          (LOCF (FFETCH NSHNM0 OF PDH))
          \10MBHOST.WORDS))
  (\BLT (fetch 10MBSOURCEHOSTBASE of PACKET)
        (LOCF (FFETCH NSHNM0 OF \MY.NSADDRESS))
        \10MBHOST.WORDS))
  (replace 10MBLENGTH of PACKET with (+ (FOLDHI LENGTH BYTESPERWORD)
                                         \10MBENCAPSULATION.WORDS))
  (replace 10MBTYPE of PACKET with TYPE)
  PACKET])

```

**(\10MB.BROADCASTP**

```

[LAMBDA (PACKET) ; (* bvm%: "23-Apr-84 14:34")
  (fetch 10MBMULTICASTP of PACKET])

```

**(\10MBWATCHER**

```

[LAMBDA (NDB) ; (* bvm%: "26-OCT-83 15:23")
;; Process that watches the 10mb net and pulls packets in. Decodes the type and passes packet to interested party
(PROG ((CNTR 0)
      PACKET)
  LP (UNINTERRUPTABLY
    (\10MB.INPUT.INTERRUPT NDB)
    (\10MB.OUTPUT.INTERRUPT NDB))
  [COND
    ((SETQ PACKET (\10MBGETPACKET)) ; Got something
     (\HANDLE.RAW.PACKET PACKET)
     (COND
       ((ILESSP (add CNTR 1)
                \MAXWATCHERGETS)
        (GO LP] ; Hack to get better ether service in lieu of preemption
    (BLOCK)
    (SETQ CNTR 0)
    (GO LP])
  )

```

;; Machine independent part

(DEFINEQ

**(\10MB.STARTDRIVER**

```

[LAMBDA (NDB RESTARTFLG MYNSNUMBER) ; (* ejs%: " 7-Sep-85 19:24")
  (\10MB.TURNOFFETHER NDB)
  (OR (INIT.ETHER.BUFFER.POOL)
      (ERROR "Unable to create buffer pool"))
  (replace NDBTQ of NDB with (create SYSQUEUE))
  (SETQ \10MB.RAWPACKETQ (create SYSQUEUE))
  (SETQ \10MB.INPUT.TIMEOUT (TIMES \RCLKSECOND \10MB.EXPECTED.RECEIVE.INTERVAL))
  (\10MB.TURNONETHER NDB NIL NIL (OR MYNSNUMBER T)

```

```

0 0)
(PROG ((LEN 0)
      (IQ (fetch NDBIQ of NDB)))
[COND
  (IQ (SETQ LEN (\10MB.LOADINPUTQ NDB (fetch SYSQUEUEHEAD of IQ)
    (T (replace NDBIQ of NDB with (SETQ IQ (create SYSQUEUE)
(bind IOCB PACKET to (IDIFFERENCE \10MB.IDEAL.INPUT.LENGTH LEN) while (SETQ IOCB (\GET.IOCB 'INPUT))
do (SETQ PACKET (\ALLOCATE.ETHERPACKET))
  (\TEMPLOCKPAGES PACKET 2)
  (replace EPNETWORK of PACKET with IOCB)
  (\QUEUE.INPUT.IOCB NDB IOCB (fetch 10MBPACKETBASE of PACKET)
    \10MBPACKETLENGTH)
  (\ENQUEUE IQ PACKET) ; Add IOCB to microcode's queue
  (add LEN 1)) ; and to driver's queue, so it can process it after arrival
(replace NDBIQLength of NDB with LEN)
(replace NDBWATCHER of NDB with (ADD.PROCESS (LIST '\10MBWATCHER (KWOTE NDB)
' RESTARTABLE
' SYSTEM
' AFTEREXIT
' DELETE))
(RETURN NDB])

```

(\10MB.CREATENDB

[LAMBDA (ETHERTASK#) ; Edited 31-Mar-87 19:33 by bvm:

```

;; If using new ethernet style pups, set up translation list
(if (NEQ \10MBTYPE.PUP \EPT.PUP)
  then (CL:PUSHNEW (CONS \10MBTYPE.PUP \EPT.PUP)
    \10MBTYPE.TRANSLATIONS :TEST 'EQUAL))
(if (NEQ \10MBTYPE.3TO10 \EPT.3TO10)
  then (CL:PUSHNEW (CONS \10MBTYPE.3TO10 \EPT.3TO10)
    \10MBTYPE.TRANSLATIONS :TEST 'EQUAL))
(\10MB.STARTDRIVER (create NDB
  NDBNSNET# _ 0
  NDBPUPNET# _ 0
  NETTYPE _ 10
  NDBPUPTYPE _ \10MBTYPE.PUP
  NDBTRANSMITTER _ (FUNCTION \10MBSENDPACKET)
  NDBENCAPSULATOR _ (FUNCTION \10MBENCAPSULATE)
  NDBBROADCASTP _ (FUNCTION \10MB.BROADCASTP)
  NDBTASK# _ ETHERTASK#
  NDBETHERFLUSHER _ (FUNCTION \10MB.TURNOFFETHER)
  NDBCANHEARSELF _ (EQ \MACHINETYPE \DOLPHIN])

```

(\10MB.INPUT.INTERRUPT

[LAMBDA (NDB) ; Edited 31-Mar-87 18:10 by bvm:

```

;; This routine gets called when 10MB input signals an interrupt. See if the head of the input queue has indeed been processed, and if so, take
;; care of it
(PROG ((PACKET (fetch SYSQUEUEHEAD of (fetch NDBIQ of NDB)))
      STATUS ACCEPTPACKET IOCB)
[COND
  ((AND PACKET (NEQ [SETQ STATUS (\10MB.GETPACKETSTATUS (SETQ IOCB (fetch EPNETWORK of PACKET)
    \ES.PENDING)) ; Yes, something is there, and microcode is finished with it
  (\DEQUEUE (fetch NDBIQ of NDB))
[COND
  (\10MB.COLLECTSTATS (\10MB.NOTESTAT STATUS PACKET 'INPUT)
[COND
  ((SETQ ACCEPTPACKET (OR (EQ STATUS \ES.GOOD.PACKET)
    \10MB.GETGARBAGE))
  (PROG ((LENGTH (\10MB.GETPACKETLENGTH IOCB))) ; Accept the packet
    (replace 10MBLENGTH of PACKET with LENGTH)
    (\RCLK (LOCF (fetch EPTIMESTAMP of PACKET)))
    (replace EPRECEIVING of PACKET with NIL)
    (replace EPNETWORK of PACKET with NDB)
  (COND
    ((AND (OR (EQ \MACHINETYPE \DANDELION)
      (EQ \MACHINETYPE \DAYBREAK))
      (IGREATERP LENGTH \MIN2PAGEBUFLLENGTH))
      ;; Dandelion ether uCode doesn't set the dirty bit on pages of PACKET, so make sure the second page gets
      ;; marked dirty if it needs to be. The first page has been implicitly marked dirty by the replaces above
      (\PUTBASE PACKET (SUB1 (ITIMES WORDSPERPAGE 2))
        0)))
    (\TEMPUNLOCKPAGES PACKET 2)
    (\ENQUEUE \10MB.RAWPACKETQ PACKET))
  (PROGN
    (SETQ PACKET (\ALLOCATE.ETHERPACKET)) ; Now stuff a new buffer on queue
    (\TEMPLOCKPAGES PACKET 2)
    (replace EPNETWORK of PACKET with IOCB) ; Now stuff a buffer back on the input
  (\QUEUE.INPUT.IOCB NDB IOCB (fetch 10MBPACKETBASE of PACKET)
    \10MBPACKETLENGTH)
  (\ENQUEUE (fetch NDBIQ of NDB)
    PACKET))

```

```

(PACKET                                     ; There is something there, and the microcode is NOT finished
                                           ; with it
(COND
  ((NOT (fetch EPRECEIVING of PACKET))
   ;; Furthermore, this is the first time we've seen this packet at the head of the receive queue. We timestamp it, and if it's
   ;; still here sometime later, we kick the Ethernet receiver microcode, just in case it's turned itself off
   (SETUPTIMER \10MB.INPUT.TIMEOUT \10MB.INPUT.TIMER 'TICKS)
   (replace EPRECEIVING of PACKET with T))
  (T ;; We've seen this packet before. Check for timeout, and kick the receiver microcode if necessary
   (COND
     ((TIMEREXPIRED? \10MB.INPUT.TIMER 'TICKS)
      (\10MB.RESTART.ETHER NDB) ; Update the timestamp
      (SETUPTIMER \10MB.INPUT.TIMEOUT \10MB.INPUT.TIMER 'TICKS))
     (T)))
(RETURN ACCEPTPACKET])

```

**(\10MB.OUTPUT.INTERRUPT**

```

[LAMBDA (NDB)                               (* bvm%: "10-JUN-83 14:56")
  ;; This routine gets called when 10MB output signals an interrupt. Remove the head of the output queue and put it on the done queue
  (PROG ((NEXTPACKET (fetch SYSQUEUEHEAD of (fetch NDBTQ of NDB)))
         (STATUS IOCB)
         (RETURN (COND
           ((AND NEXTPACKET (OR (NULL (SETQ IOCB (fetch EPNETWORK of NEXTPACKET)))
                                (NEQ (SETQ STATUS (\10MB.GETPACKETSTATUS IOCB))
                                     \ES.PENDING)))) ; Yes, microcode has finished processing this buffer
          (\DEQUEUE (fetch NDBTQ of NDB))
          (replace EPTRANSMITTING of NEXTPACKET with NIL)
          (replace EPNETWORK of NEXTPACKET with NIL)
          (\REQUEUE.ETHERPACKET NEXTPACKET)
          [COND
            (IOCB (\RELEASE.IOCB IOCB 'OUTPUT)
                  (\TEMPUNLOCKPAGES NEXTPACKET 2)
                  (COND
                    (\10MB.COLLECTSTATS (\10MB.NOTESTAT STATUS NEXTPACKET 'OUTPUT)
                                         T]))
            (T))
          ]))

```

**(\10MB.NOTESTAT**

```

[LAMBDA (STATUS BUF USE)                   (* bvm%: "15-JUL-82 14:43")
  ;; Increment counter associated with this status
  (NIL))

```

```

)
(RPAQ? \10MB.RCLK.BOX (CREATECELL \FIXP))
(RPAQ? \10MB.EXPECTED.RECEIVE.INTERVAL 60)
(RPAQ? \10MB.INPUT.TIMEOUT (TIMES \RCLKSECOND \10MB.EXPECTED.RECEIVE.INTERVAL))
(RPAQ? \10MB.INPUT.TIMER (SETUPTIMER 0))
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS \10MB.RCLK.BOX \10MB.EXPECTED.RECEIVE.INTERVAL \10MB.INPUT.TIMEOUT \10MB.INPUT.TIMER)
)

```

;; Buffer management

```
(DEFINEQ
```

**(\10MB.LOADINPUTQ**

```

[LAMBDA (NDB PACKETS)                       (* bvm%: "28-FEB-83 17:36")
  ;; PACKETS points at the first of several buffers of NDB's IQ. We load them into the microcode's chain. Value returned is the number of buffers
  (bind (CNT _ 0) while PACKETS do (\TEMPLOCKPAGES PACKETS 2)
        (\QUEUE.INPUT.IOCB NDB (fetch EPNETWORK of PACKETS)
         (fetch 10MBPACKETBASE of PACKETS)
         \10MBPACKETLENGTH)
        (SETQ PACKETS (fetch EPLINK of PACKETS))
        (add CNT 1))
  finally (RETURN CNT])

```

**(\RELEASE.IOCB**

```

[LAMBDA (IOCB USE)                          (* bvm%: " 3-MAR-83 16:17")
  ;; Returns an IOCB to the free pool. USE is INPUT or OUTPUT, according to which side should be credited. Must be called uninterruptably
  (COND
    ((NOT (AND IOCB (EMADDRESSP IOCB)))
     (ERROR "ARG NOT IOCB" IOCB))
    (T (SELECTQ USE
          (INPUT (add \IOCB.INPUT.ALLOC 1))

```

```

      (OUTPUT (add \IOCB.OUTPUT.ALLOC 1))
      (\ILLEGAL.ARG USE))
  (replace DONEXTIOCB of IOCB with (\LOLOC \IOCBFREELIST)) ;(\LOLOC NIL) = 0 works also
  (SETQ \IOCBFREELIST IOCB)
  NIL))

```

(\GET.IOCB

```

[LAMBDA (USE)
  (* edited%: "14-Aug-85 16:46")
  ;; returns a IOCB for INPUT or OUTPUT use, or NIL if none is available. This must be called uninterruptably, since we don't have any easy way of
  ;; GCing these guys
  (DECLARE (GLOBALVARS \10MBLOCALNDB))
  (COND
    ((AND \IOCBFREELIST (IGREATERP (SELECTQ USE
      (INPUT \IOCB.INPUT.ALLOC)
      (OUTPUT \IOCB.OUTPUT.ALLOC)
      (\ILLEGAL.ARG USE))
      0))
      (SELECTQ USE
        (INPUT (add \IOCB.INPUT.ALLOC -1))
        (add \IOCB.OUTPUT.ALLOC -1))
      ;; I removed the call to HELP that used to be in here. If the IOCB freelist goes NIL, the packet is dropped on the floor
      (PROG1 \IOCBFREELIST
        (SETQ \IOCBFREELIST (EMPOINTER (fetch DONEXTIOCB of \IOCBFREELIST)))
          ; Note that (EMPOINTER 0) = NIL, so this works even when free
          ; list runs out
        )
      )))

```

(\INIT.ETHER.BUFFER.POOL

```

[LAMBDA NIL
  (* ejs%: "26-Jul-85 23:14")
  ;; Divides up the zone bcpl reserved for us into IOCB's used for sending/receiving ether packets. The IOCB's must be quad-aligned. When an
  ;; ether packet is to be sent, or prepared for receiving, an IOCB is assigned to it. The IOCB contains length and status info and a pointer to the
  ;; ether packet buffer in Lisp space. The IOCB's are chained for the microcode, and the packets are chained independently in Lisp space so that
  ;; we can keep track of them after the microcode finishes. \IOCBFREELIST points at the first IOCB; there are a total of \IOCBTOTAL of them.
  (PROG (LASTBUF BUFFER ZONE ZONELENGTH)
    (COND
      ((EQ (SETQ ZONELENGTH (fetch (IFPAGE MDSZoneLength) of \InterfacePage))
        0)
        ; Bcpl was unable to allocate any space for us
        (SETQ \IOCBFREELIST NIL)
        (SETQ \IOCBTOTAL 0)
        (RETURN))
      (T (SETQ ZONE (fetch (IFPAGE MDSZone) of \InterfacePage))
        [SETQ ZONELENGTH (IDIFFERENCE ZONELENGTH (IDIFFERENCE ZONE (SETQ ZONE (CEIL ZONE
          (ITIMES 2
            WORDSPERQUAD]
          ; 8-align the zone, in case the microcode cares, and adjust the
          ; length downward if necessary
          ; Make an actual pointer
          (SETQ ZONE (EMPOINTER ZONE))
        ]
      )))
      (SETQ \IOCBTOTAL (IMIN (IQUOTIENT (IDIFFERENCE ZONELENGTH 3)
        (SELECTC \MACHINETYPE
          (\DAYBREAK \DoveEther.IOIOCBLength)
          \IOCB.LENGTH))
        \MAXIOCBS))
      (SETQ \IOCBFREELIST ZONE)
      (to \IOCBTOTAL do
        [replace DONEXTIOCB of ZONE with (\LOLOC (SETQ ZONE (\ADDBASE ZONE (SELECTC
          \MACHINETYPE
          (\DAYBREAK
          \DoveEther.IOIOCBLength
          \IOCB.LENGTH]
        ]
        ; Link the idle IOCB's together using short addresses
        ; Pop one off for TeleRaid
      ]
      (add \IOCBTOTAL -1)
      [SETQ \IOCB.INPUT.ALLOC (SETQ \IOCB.INPUT.TOTAL (SETQ \IOCB.OUTPUT.ALLOC (SETQ \IOCB.OUTPUT.TOTAL
        (IQUOTIENT (ITIMES
          \IOCBTOTAL
          2)
        3]
      (SETQ \10MB.IDEAL.INPUT.LENGTH (IMIN \10MB.MAX.INPUT.LENGTH (IQUOTIENT \IOCBTOTAL 2)))
      (RETURN \IOCBTOTAL])
    )
  )
  (RPAQ? \10MBPACKETLENGTH 488)

```

;; Dolphin/Dlion head

(DEFINEQ

(\10MB.GETPACKETLENGTH

(\* ejs%: "18-Sep-85 15:09")

```
[LAMBDA (IOCB)
  (SELECTC \MACHINETYPE
    (\DANDELION (fetch DLIOCBBYTESUSED of IOCB))
    (\DAYBREAK (fetch (Dove.EtherIOIOCB count) of IOCB))
    (\DOLPHIN (FOLDLO (fetch D0IOCBBYTESUSED of IOCB)
      BYTESPERWORD))
    (\NOMACHINETYPE])
```

(\10MB.GETPACKETSTATUS

(\* ejs%: "26-Jul-85 22:43")

:: Translate device bits to device-independent constants
:: Now allow odd-length packets to be received

```
(SELECTC \MACHINETYPE
  (\DANDELION [PROG ((STATUS (fetch DLIOCBSTATUS of IOCB)))
    (RETURN (COND
      ((EQ STATUS 0)
        \ES.PENDING)
      [(EQ (fetch DLFOROUTPUTUSE of IOCB)
        0)
        ; Input errors
      (COND
        ((EQ (fetch DLIOCBLENGTH of IOCB)
          65535)
          \ES.PACKET.TOO.LONG)
        (T (COND
          ((EQ (SETQ STATUS (LOGAND STATUS (LOGOR \DL.BAD.ALIGNMENT
            \DL.OVERRUN \DL.BAD.CRC)
              0)
            \ES.GOOD.PACKET)
            ((NEQ (LOGAND STATUS \DL.OVERRUN)
              0)
            \ES.OVERRUN)
            ((NEQ (LOGAND STATUS \DL.BAD.CRC)
              0)
            (COND
              ((EQ (LOGAND STATUS (LOGOR \DL.BAD.ALIGNMENT \DL.ODDLENGTH)
                0)
                \ES.BAD.CRC)
                (T \ES.BAD.CRC&ALIGNMENT)))
              ((NEQ (LOGAND STATUS \DL.BAD.ALIGNMENT)
                0)
                \ES.BAD.ALIGNMENT)
              (T \ES.OTHER.ERROR]
            ; Output errors
          (COND
            ((EQ (fetch DLRETRANSMISSIONMASK of IOCB)
              8191)
            \ES.TOO.MANY.COLLISIONS)
            (T (COND
              ((EQ (LOGAND STATUS (LOGOR \DL.COLLISION \DL.UNDERRUN)
                0)
                \ES.GOOD.PACKET)
                (T \ES.UNDERRUN])
            (\DAYBREAK (\DoveEther.GetPacketStatus IOCB))
            (\DOLPHIN [PROG ((STATUS (fetch D0IOCBSTATUS of IOCB)))
              (RETURN (SELECTC STATUS
                (0 \ES.PENDING)
                (\D0.GOOD.PACKET
                  \ES.GOOD.PACKET)
                (\D0.LATE.COLLISION
                  \ES.LATE.COLLISION)
                (\D0.TOO.MANY.COLLISIONS
                  \ES.TOO.MANY.COLLISIONS)
                (\D0.PACKET.TOO.LONG
                  \ES.PACKET.TOO.LONG)
                (\D0.BAD.ALIGNMENT
                  \ES.BAD.ALIGNMENT)
                (\D0.INPUT.BAD.CRC
                  \ES.BAD.CRC)
                ((LOGOR \D0.BAD.ALIGNMENT \D0.INPUT.BAD.CRC)
                  \ES.BAD.CRC&ALIGNMENT)
                (COND
                  ((NEQ (LOGAND STATUS \D0.INPUT.OVERRUN)
                    0)
                    \ES.OVERRUN)
                  ((NEQ (LOGAND STATUS \D0.OUTPUT.UNDERRUN)
                    0)
                    \ES.UNDERRUN)
                  (T \ES.OTHER.ERROR])
                (\NOMACHINETYPE])
```

(QUEUE.INPUT.IOCB

[LAMBDA (NDB IOCB BUFFER LENGTH) ; Edited 29-Jun-88 16:57 by bvm

:: Add IOCB to the end of the microcode input queue, with BUFFER of LENGTH words as its buffer. I.e., this is a buffer that packets will be read into

```
(PROG ((CSB (fetch NDBCSB of NDB)))
  (SELECTC \MACHINETYPE
    (\DANDELION (replace DLFOROUTPUTUSE of IOCB with 0) ; So that \10MB.GETPACKETSTATUS can tell which way packet
                ; is going
                (replace DLNEXTIOCB of IOCB with 0)
                (replace DLRETRANSMISSIONMASK of IOCB with 0)
                (replace DLIOCBSTATUS of IOCB with \ES.PENDING)
                (replace DLIOCBLENGTH of IOCB with (SUB1 LENGTH))
                ; The "Length" field on DLion is actually "offset to last word in
                ; buffer"
                (replace DLIOCBBUFFER of IOCB with BUFFER)
    [COND
      ((NEQ (fetch DLFIRSTIOCB of CSB)
            0)
        ; There are some packets there, so add this to end
        (replace DLNEXTIOCB of (OR (EMPOINTER (fetch DLLASTIOCB of CSB))
                                   (RAID "Garbage last IOCB"))
          with (\LOLOC IOCB]
    (COND
      ((AND (EQ (fetch DLFIRSTIOCB of CSB)
                0)
            (EQ (fetch DLIOCBSTATUS of IOCB)
                \ES.PENDING))
        ; No buffers left, so queue this as the only one. While we were in the last clause, microcode could have eaten
        ; up its last buffer, which is why we test twice
        (replace DLFIRSTIOCB of CSB with (\LOLOC IOCB))
        (\DEVICE.OUTPUT \DL.ENABLE.INPUT \DL.ETHERINPUTREG))
      (replace DLLASTIOCB of CSB with (\LOLOC IOCB)))
    (\DAYBREAK (\DoveEther.QueueInput IOCB BUFFER LENGTH))
    (\DOLPHIN (replace DONEXTIOCB of IOCB with 0)
              (replace DORETRANSMISSIONMASK of IOCB with 0)
              (replace DOIIOCBSTATUS of IOCB with \ES.PENDING)
              (replace DOIIOCBLENGTH of IOCB with (UNFOLD LENGTH BYTESPERWORD))
              (replace DOIIOCBBUFFER of IOCB with BUFFER)
    [COND
      ((NEQ (fetch DOIIOCBSTATUS of CSB)
            0)
        ; There are some packets there, so add this to end
        (replace DONEXTIOCB of (OR (EMPOINTER (fetch DOIIOCBSTATUS of CSB))
                                   (RAID "Garbage last IOCB"))
          with (\LOLOC IOCB]
    [COND
      ((AND (EQ (fetch DOIIOCBSTATUS of CSB)
                0)
            (EQ (fetch DOIIOCBSTATUS of IOCB)
                \ES.PENDING))
        ; No buffers left, so queue this as the only one. While we were in the last clause, microcode could have eaten up
        ; its last buffer, which is why we test twice
        (replace DOIIOCBSTATUS of CSB with (\LOLOC IOCB))
        (\DEVICE.OUTPUT \D0.ENABLE.10MBINPUT (\D0.CONTROLLERBITS NDB \D0.INPUTSTATE)
          (replace DOIIOCBSTATUS of CSB with (\LOLOC IOCB)))
    (\NOMACHINETYPE]))
```

(QUEUE.OUTPUT.IOCB

[LAMBDA (NDB IOCB BUFFER LENGTH) ; Edited 29-Jun-88 16:57 by bvm

:: Queue up IOCB for transmission. LENGTH is length of BUFFER in words

```
(PROG ((CSB (fetch NDBCSB of NDB)))
  (SELECTC \MACHINETYPE
    (\DANDELION (replace DLFOROUTPUTUSE of IOCB with 1) ; So that \10MB.GETPACKETSTATUS can tell which way packet
                ; is going
                (replace DLIOCBLENGTH of IOCB with (SUB1 LENGTH))
                ; The "Length" field on DLion is actually "offset to last word in
                ; buffer"
                (replace DLNEXTIOCB of IOCB with 0)
                (replace DLRETRANSMISSIONMASK of IOCB with 0)
                (replace DLIOCBSTATUS of IOCB with \ES.PENDING)
                (replace DLIOCBBUFFER of IOCB with BUFFER)
    [COND
      ((NEQ (fetch DLFIRSTIOCB of CSB)
            0)
        ; Hardware active, add to end of chain
        (replace DLNEXTIOCB of (OR (EMPOINTER (fetch DLLASTIOCB of CSB))
                                   (RAID "Garbage Last IOCB"))
          with (\LOLOC IOCB]
    (COND
      ((AND (EQ (fetch DLFIRSTIOCB of CSB)
                0)
            (EQ (fetch DLIOCBSTATUS of IOCB)
                \ES.PENDING))
        ; Separate check, as the hardware could have just gone idle
        ; since we last checked
        (replace DLFIRSTIOCB of CSB with (\LOLOC IOCB))
```

```

      (\DEVICE.OUTPUT \DL.ENABLE.OUTPUT \DL.ETHEROUTPUTREG))
      (replace DLLASTOCB of CSB with (\LOLOC IOCB))
(\DAYBREAK (\DoveEther.QueueOutput IOCB BUFFER LENGTH))
(\DOLPHIN (replace D0IOCBLENGTH of IOCB with (UNFOLD LENGTH BYTESPERWORD))
(replace DONEXTIOCB of IOCB with 0)
(replace DORETRANSMISSIONMASK of IOCB with 0)
(replace D0IOCBSTATUS of IOCB with \ES.PENDING)
(replace D0IOCBBUFFER of IOCB with BUFFER)
[COND
  ((NEQ (fetch D0FIRSTOCB of CSB)
    0)
    ; Hardware active, add to end of chain
    (replace DONEXTIOCB of (OR (EMPOINTER (fetch D0LASTOCB of CSB))
      (RAID "Garbage Last OCB"))
      with (\LOLOC IOCB)
    ]
[COND
  ((AND (EQ (fetch D0FIRSTOCB of CSB)
    0)
    (EQ (fetch D0IOCBSTATUS of IOCB)
      \ES.PENDING))
    ; Separate check, as the hardware could have just gone idle
    ; since we last checked
    (replace D0FIRSTOCB of CSB with (\LOLOC IOCB))
    (\DEVICE.OUTPUT \D0.ENABLE.10MBOUPTUT (\D0.CONTROLLERBITS NDB \D0.OUTPUTSTATE])
    (replace D0LASTOCB of CSB with (\LOLOC IOCB))
  (\NOMACHINETYPE])

```

(\10MB.TURNOFFETHER

```

[LAMBDA (NDB)
  (SELECTC \MACHINETYPE
    (\DANDELION (\DL.TURNOFFETHER))
    (\DAYBREAK (\DoveEther.MakeSureOff))
    (\DOLPHIN (\D0.TURNOFFETHER NDB))
  (\NOMACHINETYPE])
(* ejs%: "26-Jul-85 22:47")

```

(\10MB.TURNONETHER

```

[LAMBDA (NDB SMASHSTATE NEWSTATE NSHOSTNUMBER ININTERRUPT OUTINTERRUPT)
  (* ejs%: "30-Oct-85 17:41")

```

;;; Reset and activate ether associated with NDB. If SMASHSTATE is given, it is a CSB-length block into which state is saved for later restoration by  
 ;;; passing as the NEWSTATE arg. If NEWSTATE is NIL, then the remaining non-NIL args give parameters for this activation: the host number for  
 ;;; microcode to watch for, T meaning my own number; and interrupt masks for when a packet arrives or finishes transmitting

;; For Daybreak, SMASHSTATE and NEWSTATE must be NIL

```

(PROG ((CSB (fetch NDBCSB of NDB)))
  (SELECTC \MACHINETYPE
    (\DANDELION (\DL.TURNOFFETHER))
    (\DAYBREAK (\DoveEther.MakeSureOff SMASHSTATE))
    (\DOLPHIN (\D0.TURNOFFETHER NDB))
  (\NOMACHINETYPE))
[COND
  ((AND SMASHSTATE (NEQ \MACHINETYPE \DAYBREAK))
    (COND
      (CSB (\BLT SMASHSTATE CSB \CSB.LENGTH))
      (T
        ; Arcane way of indicating ether is off. May have to revisit
        (\PUTBASE CSB 4 65535)
      )
    ]
[COND
  ((AND NEWSTATE (NEQ \MACHINETYPE \DAYBREAK))
    ; Smash old state into CSB
    (COND
      ((EQ (\GETBASE NEWSTATE 4)
        65535)
        ; Leave ether off
        (RETURN)))
      (\BLT CSB NEWSTATE \CSB.LENGTH))
    (T (SELECTC \MACHINETYPE
      (\DANDELION [OR CSB (replace NDBCSB of NDB with (SETQ CSB (LOCF (fetch DLEETHERNET of \IOPAGE]
        (COND
          (NEWSTATE
            ; Smash old state into CSB
            ; I don't think you can get here! --ejs
            (COND
              ((EQ (\GETBASE NEWSTATE 4)
                65535)
                ; Leave ether off
                (RETURN)))
              (\BLT CSB NEWSTATE \CSB.LENGTH))
            (T
              ; Initialize the Ether CSB according to args. No buffers initially
              (replace DLFIRSTOCB of CSB with 0)
              (replace DLFIRSTICB of CSB with 0)
              [AND NSHOSTNUMBER (COND
                ((EQ NSHOSTNUMBER T)
                  (\BLT (LOCF (fetch DLLOCALHOST0 of CSB))
                    (LOCF (fetch (IFPAGE NSHost0) of \InterfacePage
                      ))
                    \#WDS.NSHOSTNUMBER))
                (T (\STORENSHOSTNUMBER (LOCF (fetch DLLOCALHOST0
                  of CSB))
                    NSHOSTNUMBER]
              (AND OUTINTERRUPT (replace DLOUTPUTMASK of CSB with OUTINTERRUPT))
              (AND ININTERRUPT (replace DLINPUTMASK of CSB with ININTERRUPT))

```

```

      (replace DLMISSEDPACKETS of CSB with 0)
      (replace DLLASTICB of CSB with 0)
      (replace DLLASTOCB of CSB with 0)))
      (\DEVICE.OUTPUT \DL.ENABLE.INPUT \DL.ETHERINPUTREG))
      (\DAYBREAK (\DoveEther.TurnOn NSHOSTNUMBER NEWSTATE))
      (\DOLPHIN [OR CSB (replace NDBCBSB of NDB with (SETQ CSB
      (EMPOINTER (IPLUS 65280
      (LLSH (fetch NDBTASK#
      of NDB)
      4]
      ; Initialize the Ether CSB according to args. No buffers initially

      (replace D0FIRSTOCB of CSB with 0)
      (replace D0FIRSTICB of CSB with 0)
      [AND NSHOSTNUMBER (COND
      (EQ NSHOSTNUMBER T)
      (\BLT (LOCF (fetch D0LOCALHOST0 of CSB))
      (LOCF (fetch (IFPAGE NSHost0) of \InterfacePage))
      \#WDS.NSHOSTNUMBER))
      (T (\STORENSHOSTNUMBER (LOCF (fetch D0LOCALHOST0 of CSB))
      NSHOSTNUMBER]
      (AND OUTINTERRUPT (replace D0OUTPUTMASK of CSB with OUTINTERRUPT))
      (AND ININTERRUPT (replace D0INPUTMASK of CSB with ININTERRUPT))
      (replace D0LASTICB of CSB with 0)
      (replace D0LASTOCB of CSB with 0)
      (\DEVICE.OUTPUT \D0.ENABLE.10MBINPUT (\D0.CONTROLLERBITS NDB \D0.INPUTSTATE)))
      (\NOMACHINETYPE]
      (SELECTC \MACHINETYPE
      (\DANDELION (\DEVICE.OUTPUT \DL.ENABLE.INPUT \DL.ETHERINPUTREG))
      (\DAYBREAK)
      (\DOLPHIN (\DEVICE.OUTPUT \D0.ENABLE.10MBINPUT (\D0.CONTROLLERBITS NDB \D0.INPUTSTATE)))
      (\NOMACHINETYPE))
      (RETURN NDB])

```

**(\10MB.RESTART.ETHER**

[LAMBDA (NDB)

(\* ejs%: "13-Nov-85 13:36")

;;; Kick the Ethernet receiver microcode (or Daybreak IOP) to restart the Ethernet receiver task. This function gets called when the 10MBDRIVER thinks  
;;; the Ethernet has been accidentally disabled

```

      (SELECTC \MACHINETYPE
      (\DANDELION (\DEVICE.OUTPUT \DL.ENABLE.INPUT \DL.ETHERINPUTREG))
      (\DOLPHIN (\DEVICE.OUTPUT \D0.ENABLE.10MBINPUT (\D0.CONTROLLERBITS NDB \D0.INPUTSTATE)))
      (\DAYBREAK (LET [(EtherQueues (\ALLOCBLOCK (CONSTANT (FOLDHI (ITIMES 2 (MESASIZE Dove.QueueBlock))
      WORDSPERCELL]
      [\BLT EtherQueues (fetch (Dove.EtherFCB mesaOutQueue) of \DoveEther.FCBPointer)
      (CONSTANT (ITIMES 2 (MESASIZE Dove.QueueBlock)
      (\DoveEther.TurnOn \MY.NSHOSTNUMBER EtherQueues))
      NIL])

```

;; Misc

(DEFINEQ

**(\NOMACHINETYPE**

[LAMBDA NIL

(RAID "Operation not implemented on this machine")]

(\* bvm%: "24-JUL-82 17:47")

**(\10MB.PRINT.ENCAPSULATION**

[LAMBDA (PACKET CALLER STREAM)

; Edited 17-Dec-86 19:07 by bvm:

```

      (FRESHLINE STREAM)
      (AND CALLER (printout STREAM CALLER ": "))
      (PRIN1 "From host " STREAM)
      (PRINTNSHOSTNUMBER (fetch 10MBSOURCEHOST of PACKET)
      STREAM)
      (PRIN1 " to " STREAM)
      (PRINTNSHOSTNUMBER (fetch 10MBDESTHOST of PACKET)
      STREAM)
      (TERPRI STREAM)
      PACKET])

```

(DEFINEQ

**(\IOCBQLENGTH**

[LAMBDA (FIRSTIOCB)

(\* bvm%: " 2-MAR-83 17:52")

```

      (OR FIRSTIOCB (SETQ FIRSTIOCB \IOCBFREELIST))
      (while FIRSTIOCB sum (PROGN (SETQ FIRSTIOCB (EMPOINTER (fetch DONEXTIOCB of FIRSTIOCB)))
      1])

```

(RPAQ? \10MB.GETGARBAGE NIL)

(RPAQ? \10MB.COLLECTSTATS NIL)

:: PUP address resolution

(DEFINEQ

(\HANDLE.RAW.3TO10

[LAMBDA (PACKET TYPE)

; Edited 26-Feb-91 11:48 by jds

:: Called when a TRANSLATION packet is received. This is either a packet requesting a 10-to-3 translation, in which case we respond if it is asking about us; or it is a response to a request of ours, in which case we store the info in the cache

(COND

((EQ TYPE \EPT.3TO10)

(PROG ((NDB (fetch EPNETWORK of PACKET))

ADDR)

(AND XIPTRACEFLG (\MAYBEPRINTPACKET PACKET 'GET))

(AND NDB (SELECTC (fetch TRANSOPERATION of PACKET)

(\TRANS.OP.REQUEST

(COND

([AND (EQ (fetch TRANSPUPHOST of PACKET)

(fetch NDBPUPHOST# of NDB))

(>= (fetch 10MBLENGTH of PACKET)

(+ \10MBENCAPSULATION.WORDS (FOLDHI \TRANS.DATALLENGTH BYTESPERWORD) ; It's for us, and it's big enough

(\BLT [LOCF (fetch NSHNM0 of (SETQ ADDR (create NSADDRESS)

(LOCF (fetch BASETRANSSENDERNSHOST of PACKET))

3)

(\NOTE.3TO10 ADDR (fetch TRANSSENDERPUPHOST of PACKET)

NDB) ; Add sender's address to cache

(\BLT [LOCF (fetch BASETRANSNSHOST of PACKET))

(LOCF (fetch NSHNM0 of \MY.NSADDRESS))

3)

; Add in the information he wants

(replace TRANSOPERATION of PACKET with \TRANS.OP.RESPONSE)

(ENCAPSULATE.ETHERPACKET NDB PACKET ADDR \TRANS.DATALLENGTH \10MBTYPE.3TO10)

; Send back the response

(AND XIPTRACEFLG (NOT (MEMB 'TRANS XIPIGNORETYPES))

(PRINT3TO10 PACKET 'PUT XIPTRACEFILE))

(replace EPREQUEUE of PACKET with 'FREE)

(TRANSMIT.ETHERPACKET NDB PACKET)

(RETURN)))

(\TRANS.OP.RESPONSE ; Add the information to the cache

(\BLT [LOCF (fetch NSHNM0 of (SETQ ADDR (create NSADDRESS)

(LOCF (fetch BASETRANSNSHOST of PACKET))

3)

(\NOTE.3TO10 ADDR (fetch TRANSPUPHOST of PACKET)

NDB))

NIL))

(\RELEASE.ETHERPACKET PACKET))

T])

(\TRANSLATE.3TO10

[LAMBDA (PUPHOSTNUMBER NDB)

; Edited 15-Jan-88 00:49 by bvm

:: Translate from an PUPHOSTNUMBER to a NSHOSTNUMBER for the indicated network. If we don't have the translation, we initiate a probe for it and return NIL

(OR [CADR (ASSOC PUPHOSTNUMBER (ffetch NDBTRANSLATIONS of (\DTEST NDB 'NDB)

(PROG ((MYPUPHOSTNUMBER (ffetch NDBPUPHOST# of NDB)

PACKET)

(COND

((EQ MYPUPHOSTNUMBER 0)

; We don't know who we are yet

(RETURN)))

(SETQ PACKET (\ALLOCATE.ETHERPACKET))

(replace EPTYPE of PACKET with \EPT.3TO10)

(replace TRANSOPERATION of PACKET with \TRANS.OP.REQUEST)

(replace TRANSPUPHOST of PACKET with PUPHOSTNUMBER)

(\BLT [LOCF (FETCH BASETRANSSENDERNSHOST of PACKET))

(LOCF (FETCH NSHNM0 OF \MY.NSADDRESS))

3)

(replace TRANSSENDERPUPHOST of PACKET with MYPUPHOSTNUMBER)

(ENCAPSULATE.ETHERPACKET NDB PACKET BROADCASTNSHOSTNUMBER \TRANS.DATALLENGTH \10MBTYPE.3TO10)

(AND XIPTRACEFLG (\MAYBEPRINTPACKET PACKET 'PUT))

(replace EPREQUEUE of PACKET with 'FREE)

(TRANSMIT.ETHERPACKET NDB PACKET)

; We didn't find out this time, but we will later on

(RETURN)]

(PRINT3TO10

[LAMBDA (EPKT CALLER FILE PRE.NOTE DOFILTER)

(\* bvm%: "14-Feb-85 21:21")

(COND

((OR (NOT DOFILTER)

(NOT (MEMB 'TRANS XIPIGNORETYPES)))

(OR FILE (SETQ FILE XIPTRACEFILE))

(FRESHLINE FILE)

(COND

(PRE.NOTE (PRIN1 PRE.NOTE FILE)))

```
(SELECTC (fetch TRANSOPERATION of EPKT)
(\TRANS.OP.REQUEST
(printout FILE CALLER " 3:10 trans request for " (fetch TRANSPUPHOST of EPKT)
" from "
(fetch TRANSENDEPUPHOST of EPKT)
" = ")
(PRINTNSHOSTNUMBER (fetch TRANSENDEPUPHOST of EPKT)
FILE))
(\TRANS.OP.RESPONSE
(printout FILE CALLER " 3:10 trans response: " (fetch TRANSPUPHOST of EPKT)
" = ")
(PRINTNSHOSTNUMBER (fetch TRANSENDEPUPHOST of EPKT)
FILE))
(printout FILE CALLER " unknown 10 to 3 translation operation " (fetch TRANSOPERATION of EPKT))
(TERPRI FILE])
```

(NOTE.3TO10

[LAMBDA (NSADDR PUPHOST NDB) ; Edited 15-Jan-88 00:47 by bvm

;; Update cache to include this pairing

```
(PROG [(A (ASSOC PUPHOST (fetch NDBTRANSLATIONS of (\DTEST NDB 'NDB)
(COND
(A (RPLACA (CDR A)
NSADDR))
(T (push (fetch NDBTRANSLATIONS of NDB)
(LIST PUPHOST NSADDR (CLOCK 0))
```

)

(RPAQ? \10MBTYPE.PUP 512)

(RPAQ? \10MBTYPE.3TO10 513)

(RPAQ? \10MBTYPE.TRANSLATIONS )

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \10MBTYPE.3TO10 \10MBTYPE.PUP \10MBTYPE.TRANSLATIONS)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \EPT.3TO10 513)

(CONSTANTS \EPT.3TO10)
)

(ADDTOVAR \PACKET.PRINTERS (513 . PRINT3TO10))

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(FILESLOAD (SOURCE)
10MBDECLS DOVEDECLS DOVEETHERDECLS LLNSDECLS)

(FILESLOAD (LOADCOMP)
LLEATHER)
)

(PUTPROPS 10MBDRIVER COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1988 1990 1991))

---

**FUNCTION INDEX**

IOCBQLength .....	9	\10MB.PRINT.ENCAPSULATION .....	9	\HANDLE.RAW.3TO10 .....	10
PRINT3TO10 .....	10	\10MB.RESTART.ETHER .....	9	\INIT.ETHER.BUFFER.POOL .....	5
\10MB.BROADCASTP .....	2	\10MB.STARTDRIVER .....	2	\NOMACHINETYPE .....	9
\10MB.CREATENDB .....	3	\10MB.TURNOFFETHER .....	8	\NOTE.3TO10 .....	11
\10MB.GETPACKETLENGTH .....	6	\10MB.TURNONETHER .....	8	\QUEUE.INPUT.IOCB .....	7
\10MB.GETPACKETSTATUS .....	6	\10MBENCAPSULATE .....	2	\QUEUE.OUTPUT.IOCB .....	7
\10MB.INPUT.INTERRUPT .....	3	\10MBGETPACKET .....	1	\RELEASE.IOCB .....	4
\10MB.LOADINPUTQ .....	4	\10MBSENDPACKET .....	1	\TRANSLATE.3TO10 .....	10
\10MB.NOTESTAT .....	4	\10MBWATCHER .....	2		
\10MB.OUTPUT.INTERRUPT .....	4	\GET.IOCB .....	5		

---

**VARIABLE INDEX**

\10MB.COLLECTSTATS .....	10	\10MB.INPUT.TIMER .....	4	\10MBTYPE.PUP .....	11
\10MB.EXPECTED.RECEIVE.INTERVAL ..	4	\10MB.RCLK.BOX .....	4	\10MBTYPE.TRANSLATIONS .....	11
\10MB.GETGARBAGE .....	9	\10MBPACKETLENGTH .....	5	\PACKET.PRINTERS .....	11
\10MB.INPUT.TIMEOUT .....	4	\10MBTYPE.3TO10 .....	11		

---

**CONSTANT INDEX**

\EPT.3TO10 .....	11
------------------	----

---