

# Typical registers

*control and status register (CSR)*

Write to initiate I/O operation

Read to determine device status

For character at a time I/O

*read data buffer register (RDB)*

Read to get next character

*transmit data buffer register (TDB)*

Write with character to transmit

## Direct Memory Access (DMA)

Devices read and write blocks of memory

*buffer descriptor list (BDL)*

*descriptor ring pointer (DRP)*

Gives buffers used in block transfer

# PMAD-AA registers

**PMAD-AA** TURBOchannel Ethernet Module

Assuming Module is in slot 0 of 5000/200

Some I/O registers

0x1E000000	Network buffer
0x1E100000	Register Address Port (RAP)
0x1E100004	Register Data Port (RDP)
0x1E1C0000	Ethernet address and other stuff

# RAP and RDP

PMAD-AA has four CSRs

Use Register Address Port  
to select the CSR

Use Register Data Port  
to read/write the selected CSR

CSR0      the *usual* CSR

CSR1      used only in  
CSR2      initialization

CSR3      controls byte swapping, *etc.*

# PMAD-AA CSR0

*bits 15-8*

- 15    **Error summary (ERR)**  
Set if any of BABL, CERR, MISS, or MESS are set. Cleared by clearing those four bits
- 14    **Transmitter timeout error (BABL)**  
Set if transmitter used channel too long.
- 13    **Collision error (CERR)**  
Set if heartbeat (SQE) test failed.
- 12    **Missed packet (MISS)**  
Set if packet lost due to lack of buffer.
- 11    **Memory error (MERR)**  
Set if DMA transfer failed.
- 10    **Receive interrupt (RINT)**  
Set if packet reception failed.
- 9     **Transmit interrupt (TINT)**  
Set if packet transmission failed.
- 8     **Initialization (IDON)**  
Set if initialization is complete.

Except for ERR, all bits are cleared by writing 1.

# PMAD-AA CSR0

*bits 7-0*

- 7    **Interrupt request (INTR)**  
Set if any of BABL, MISS, MERR, RINT, TINT, or IDON are set.
- 6    **Interrupt enable (INEA)**  
Set to request the PMAD-AA to generate an interrupt whenever INTR is set.
- 5    **Receiver on (RXON)**  
Set if the receiver is enabled.
- 4    **Transmitter on (TXON)**  
Set if the transmitter is enabled.
- 3    **Transmit demand (TDMD)**  
Set to demand immediate transmission (without waiting for polltime interval to elapse).
- 2    **Stop external action (STOP)**  
Set to force a reset.
- 1    **Start operation (STRT)**  
Set to enable chip to send and receive packets.
- 0    **Initialize (INT)**  
Set to start initialization process.

# Device-OS interaction

*receiving a packet*

Buffers synchronized with **OWN** flags

if 0, *owned* by host (OS)

if 1, *owned* by device

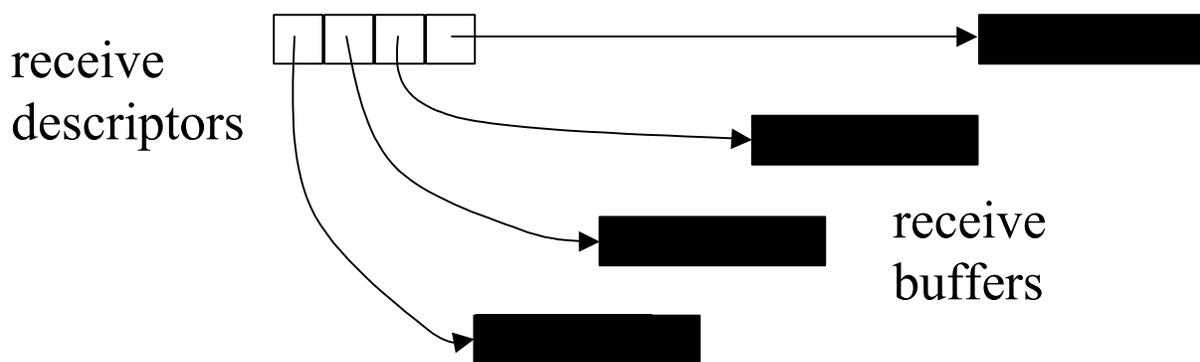
## OS

Allocates  $n$  buffers for received packets

All buffers are owned by device.

Constructs Receive Descriptor Ring that  
references the receive buffers

*Informs* device of buffers at initialization



# Device-OS interaction

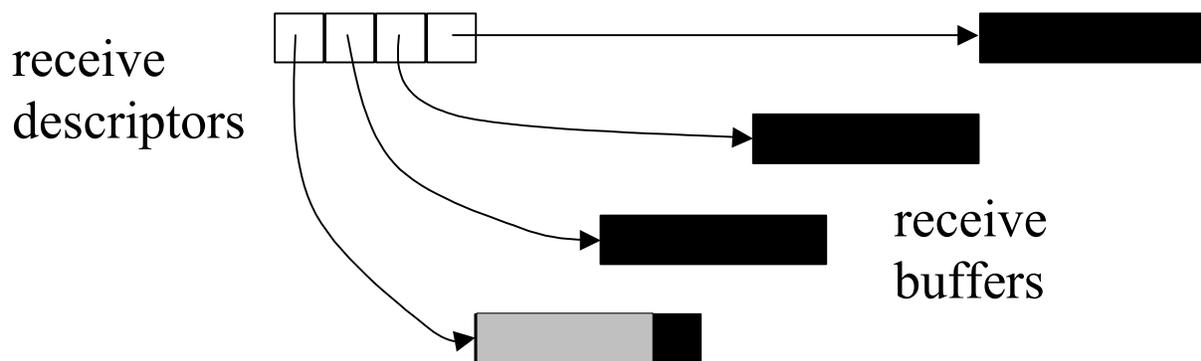
*continued*

## **device** on receipt of packet

Stores packet in buffer

Marks buffer as owned by host

Interrupts host



## **OS** on handling interrupt

Determines interrupt is for received packet

Invokes appropriate protocol routine

Marks buffer as owned by device