I/O System Architecture -
Connecting the Processor to the Rest of the World

- Low speed I/O
  - audio
  - video

- High speed I/O
  - networks
  - disks

Mem

P

Operates concurrently with the processor

- How does (special) software on the processor deal with lots of peripherals while still running programs
  - issue commands
  - get responses
  - transfer data
  - asynchronous notification

- How do we connect peripherals to the system?
- How do we exchange data & commands?
Connecting the Pieces
- Bus: collection of wires, data & control
  - Protocol for exchange
  - Multiple Parties

KB/s

Low speed

MT/s

Asynch handshake
- Recognizes master, responds to protocol
- 100 MHz (10 ns)
- 16 x 8
- 1.6 GB/s

10^6 GB/s

Sync

33 1/3 MHz
84 MHz

5/6 Bus (PCI)

Memory bus

Net, Ex, I/O, BA

AT 100 MHz
Exchanging Information

Old world's In/Out instruction

Most Machines' Ld/St instruction

- Memory mapped I/O
- Control registers
- Status registers
- Data registers

- Real/Write location as if memory
- Device controller monitors I/O regs &节省 action.

Device Driver Software
- Implements processor side if device protocol

+ Interrupt
Interrupt

- Normal interrupt occurs, access to execution page
- Sequence
  - Jump
  - Branch
- Exception
- \( PC \leftarrow f(PC) \)

Device driver software

- Interrupt handle
- Dispatch

- Where?
- What happens to old PC?
- Other registers?
- How do you figure out what to do?
- How to restore interrupted PC?
- Enable/disable interrupts
MIPS Pro State

Exception
EPC <= PC (restart loc)
PC <= 0x80000080

- user -> kernel
- interrupts disabled
- save status

Interrupt handler

Save regs
- detect cause
- specific
- entry
- dispatch
- handle
- specific
- interrupt
- R/W CTRL reg
- R/W data reg

EE or EPC
- kernel -> user
- enable interrupts
- restore status

Some times a register or vector of addresses

Interrupt handler

Int. Address

Int. #
Exceptions:

- Interrupts (asynchronous)
  - between instruction
- Traps/Faults (synchronous)
  - within an instruction

Ex: Page Fault

```inst
LD R3, 16 (R2)
```

![Diagram of page fault processing]

- Invalid / Non-existent mappings
- Suspend inst
  - Trap to OS
  - Load page from disk
    - mil/ths of inst
  - Update page table
- Retry inst (From scratch)
Direct Memory Access

- Processor sets up transfer using PIO
  - writing device register
- Start DMA transfer
  - PIO write CTRL
- I/O device read/write memory
- Issues interrupt when complete

- KB/s ms
- MB/s μs
- GB/s ns

P ns  Millions inst
Pio ns  times per unit
DMA ns  than total

15