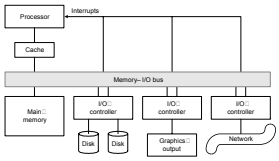
Interfacing Processors and Peripherals

- I/O Design affected by many factors (expandability, resilience)
- Performance:
  - — access latency
  - — throughput
  - — connection between devices and the system
  - — the memory hierarchy
  - — the operating system
- A variety of different users (e.g., banks, supercomputers, engineers)



I/O

- Important but neglected

  "The difficulties in assessing and designing I/O systems have often relegated I/O to second class status"

  "courses in every aspect of computing, from programming to computer architecture often ignore I/O or give it scanty coverage"

  "textbooks leave the subject to near the end, making it easier for students and instructors to skip it!"

- Somewhat GUILTY!
  - — We won't be looking at I/O in much detail
  - — Read Chapter 8
  - — Recommendation: You should take a networking class!

I/O Devices

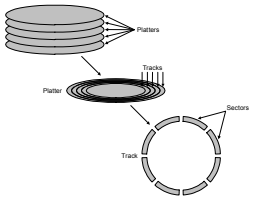
- Very diverse devices
  - — behavior (i.e., input vs. output vs. both)
  - — partners (who is at the other end?)
  - — data rates

Device	Behavior	Partner	Data rate (KB/sec)
Keyboard	input	human	0.01
Mouse	input	human	0.02
Voice input	input	human	0.02
Scanner	input	human	400.00
Voice output	output	human	0.60
Line printer	output	human	1.00
Laser printer	output	human	200.00
Graphics display	output	human	60,000.00
Modem	input or output	machine	2.00-8.00
Network/LAN	input or output	machine	500.00-6000.00
Floppy disk	storage	machine	100.00
Optical disk	storage	machine	1000.00
Magnetic tape	storage	machine	2000.00
Magnetic disk	storage	machine	2000.00-10,000.00

I/O Techniques

- Three main modes
  - – Programmed I/O
    - CPU checks and reads and writes device buffers
  - – Interrupt Mode
    - CPU asks devices to let it know when they are read
  - – DMA
    - CPU asks devices to perform directly from memory
    - CPU sets the memory buffers
    - CPU is informed when I/O is done (or in case of an error) through an interrupt

I/O Example: Disk Drives



To access data we need the following steps
- – Set DMA controller with address and count of words to be read
- – Ask disk to be on the right track and right position
- – Transfer Data to/from memory

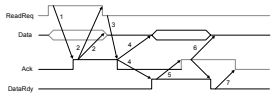
DISK I/O Time

- – Preparation time: OS prepares/delivers the transaction controller
  - 1-2 ms.
- – Seek: position head over the proper track
  - On average takes 8 to 10 ms.
- – Rotational latency: wait for desired sector (.5 / RPM)
  - At 7200 RPM, time for 0.5 revolution is 4.2 ms.
- – Transfer time: grab the data and transfer to memory
  - At 4MB/sec, 4Kb takes 4KB/(4MB/sec) = 1 ms.
- – Total time: 14.2 ms to 17.2ms (min to max)
  - Say 15ms on average
- – Keeps bus busy for 1 ms out of 15 ms

I/O Example: Buses

- Shared communication link (one or more wires)
- Difficult design:
  - — may be bottleneck
  - — length of the bus
  - — number of devices
  - — tradeoffs (buffers for higher bandwidth increases latency)
  - — support for many different devices
  - — cost
- Types of buses:
  - — processor-memory (short high speed, custom design)
  - — backplane (high speed, often standardized, e.g., PCI)
  - — I/O (lengthy, different devices, standardized, e.g., SCSI)
- Synchronous vs. Asynchronous
  - — use a clock and a synchronous protocol, fast and small but every device must operate at same rate and clock skew requires the bus to be short
  - — don't use a clock and instead use handshaking

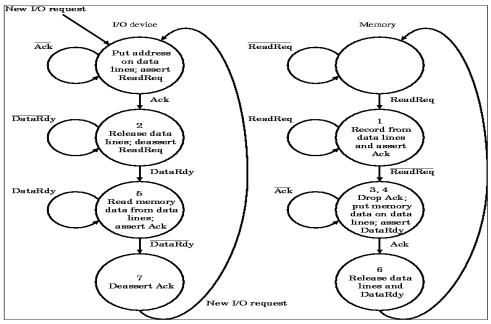
Some Example Problems



Bus Arbitration:

- — daisy chain arbitration (not very fair)
- — centralized arbitration (requires an arbiter), e.g., PCI
- — self selection, e.g., NuBus used in Macintosh, HPIB
- — collision detection, e.g., Ethernet

I/O State Machine



Some performance examples

- Let's look at some examples from the text
  - – We are not going to include bus acquisition time
  - – Processor runs at 200MHz, clock cycle time 5 nsec
  - – Address transfer takes 1 cycles
  - – Memory reads first 4 word block in 200 nsec (40 cycles)
  - – Reads successive 4 word blocks in 20 nsec (4 cycles)
  - – transfer of a 4 word block takes 10 nsec (2 cycles)
  - – Successive read and transfers can be overlapped
  - – There should be at least two cycles delay between bus cycles
  - – To transfer 4 words it will take 1 + 40 + 2 + 2 = 45 cycles
  - – To transfer 16 words it will take (successive reads and transfers can be overlapped) 1 + 40 + 4 + 4 + 4 + 2 + 2= 57 cycles
  - – To transfer 256 words using
    - 4 words at a time will take 64 * 45 = 2880 cycles
    - 16 words at a time will take 16 * 57 = 912 cycles