Input / Output

CPS 104
Lecture 22

Administrivia

- Projects due Friday Midnight
- HW 6 Due Friday Dec 5 (no late homework)
- Pipelining, Superscalar
- P4, Emotion Engine, etc.
- Review

Final

- Monday Dec 9, 2-5pm
- Closed Book, closed notes
- Cumulative

System Organization

I/O Device Examples

<table>
<thead>
<tr>
<th>Device</th>
<th>Behavior</th>
<th>Partner</th>
<th>Data Rate (KB/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td>Input</td>
<td>Human</td>
<td>0.01</td>
</tr>
<tr>
<td>Mouse</td>
<td>Input</td>
<td>Human</td>
<td>0.02</td>
</tr>
<tr>
<td>Line Printer</td>
<td>Output</td>
<td>Human</td>
<td>1.00</td>
</tr>
<tr>
<td>Laser Printer</td>
<td>Output</td>
<td>Human</td>
<td>100.00</td>
</tr>
<tr>
<td>Graphics Display</td>
<td>Output</td>
<td>Human</td>
<td>30,000.00</td>
</tr>
<tr>
<td>Network-LAN</td>
<td>Input/Output</td>
<td>Machine</td>
<td>10,000.00</td>
</tr>
<tr>
<td>Floppy disk</td>
<td>Storage</td>
<td>Machine</td>
<td>50.00</td>
</tr>
<tr>
<td>Optical Disk</td>
<td>Storage</td>
<td>Machine</td>
<td>500.00</td>
</tr>
<tr>
<td>Magnetic Disk</td>
<td>Storage</td>
<td>Machine</td>
<td>5,000.00</td>
</tr>
</tbody>
</table>

Types of Storage Devices

- Magnetic Disks
- Magnetic Tapes
- Compact Flash
- CD ROM
- Juke Box (automated tape library, robots)

Magnetic Disks

- Long term nonvolatile storage
- Another slower, less expensive level of memory hierarchy
**Organization of a Hard Magnetic Disk**

- Typical numbers (depending on the disk size):
  - 500 to 2,000 tracks per surface
  - 32 to 128 sectors per track
  - A sector is the smallest unit that can be read or written
- Traditionally all tracks have the same number of sectors:
  - Constant bit density: record more sectors on the outer tracks
  - Recently relaxed: constant bit size, speed varies with track location

**Magnetic Disk Characteristic**

- Cylinder: all the tracks under the head at a given point on all surfaces
- Read/write data is a three-stage process:
  - Seek time: position the arm over the proper track
  - Rotational latency: wait for the desired sector to rotate under the read/write head
  - Transfer time: transfer a block of bits (sector) under the read/write head
- Average seek time as reported by the industry:
  - Typically in the range of 8 ms to 12 ms
  - (Sum of the time for all possible seek) / (total # of possible seeks)
- Due to locality of disk reference, actual average seek time may:
  - Only be 25% to 33% of the advertised number

**Typical Numbers of a Magnetic Disk**

- Rotational Latency:
  - Most disks rotate at 3,600 to 10,000 RPM
  - Approximately 16ms to 3.5ms per revolution, respectively
  - An average latency to the desired information is halfway around the disk:
    - 8 ms at 3600 RPM, 4 ms at 7200 RPM
- Transfer Time is a function of:
  - Transfer size (usually a sector): 1 KB / sector
  - Rotation speed: 3600 RPM to 7200 RPM
  - Recording density: bits per inch on a track
  - Diameter typical ranges from 2.5 to 5.25 in
  - Typical values: 2 to 12 MB per second

**Disk Access**

- Access time =
  queue + seek + rotational + transfer + overhead
- Seek time
  - move arm over track
  - average is confusing (startup, slowdown, locality of accesses)
- Rotational latency
  - wait for sector to rotate under head
  - average ~ 0.5*(7200 RPM) = 4.15 ms
- Transfer Time
  - f(size, BW bytes/sec)

**Disk Access Time Example**

- Disk Parameters:
  - Transfer size is 8K bytes
  - Advertised average seek is 12 ms
  - Disk spins at 7200 RPM
  - Transfer rate is 4 MB/sec
- Controller overhead is 2 ms
- Assume that disk is idle so no queuing delay
- What is Average Disk Access Time for a Sector?
  - Ave seek + ave rot delay + transfer time + controller overhead
  - 12 ms + 0.5*(7200 RPM/60) + 8 KB/4 MB/s + 2 ms
  - 12 + 4.15 + 2 + 2 = 20 ms
- Advertised seek time assumes no locality: typically 1/4 to 1/3 advertised seek time: 20 ms => 12 ms

**DRAM as Disk**

- Solid state disk, Expanded Storage, NVRAM
- Disk is slow, DRAM is fast => replace Disk with battery backed DRAM
- BUT, Disk is cheap, much cheaper than DRAM
Alternative Storage

- CD ROM
  - Read only: good distribution, archiving
- CD RW
  - Read and Write:
- Magnetic Tape
  - Sequential Access
  - R-DAT (Rotating Digital Audio Tape)
    - Helical Scan (angle to tape, high density ~5GB)
  - Tera to peta bytes of storage (NASA EOS)

I/O Data Flow

Impediment to high performance: multiple copies, complex hierarchy

Memory to Memory Copy
DMA over Peripheral Bus
Host Processor

Application Address Space

Disk Buffers (~10 MB/Sec)

HBA Buffers (1 M - 4 MB/sec) I/O Controller

Xfer over Disk Channel
Track Buffers (32K - 256KBytes) Embedded Controller

Xfer over Serial Interface
I/O Device Head/Disk Assembly

Buses: Connecting I/O to Processor and Memory

- A bus is a shared communication link
- It uses one set of wires to connect multiple subsystems

Advantages of Buses

- Versatility:
  - New devices can be added easily
  - Peripherals can be moved between computer systems that use the same bus standard
- Low Cost:
  - A single set of wires is shared in multiple ways

Disadvantages of Buses

- The bus creates a communication bottleneck
  - Bus bandwidth can limit the maximum I/O throughput
- The maximum bus speed is largely limited by:
  - The length of the bus
  - The number of devices on the bus
  - The need to support a range of devices with:
    - Widely varying latencies
    - Widely varying data transfer rates

The General Organization of a Bus

- Control lines:
  - Signal requests and acknowledgments
  - Indicate what type of information is on the data lines
- Data lines: carry information between the source and the destination:
  - Data and Addresses
  - Complex commands
Master versus Slave

- A bus transaction includes two parts:
  - Sending the address
  - Receiving or sending the data
- Master is the device that starts the bus transaction by:
  - Sending the address
- Slave is the device that responds to the address by:
  - Sending data to the master if the master asks for data
  - Receiving data from the master if the master wants to send data

Output Operation

- Output: Processor sending data to the I/O device:
  1. Request Memory
  2. Read Memory
  3. Send Data to I/O Device

Input Operation

- Input is defined as the Processor receiving data from the I/O device:
  1. Request Memory
  2. Receive Data

Types of Buses

- Processor-Memory Bus (design specific)
  - Short and high speed
  - Only need to match the memory system
  - Maximizes memory-to-processor bandwidth
  - Connects directly to the processor
- External I/O Bus (industry standard)
  - Usually lengthy and slower
  - Need to match a wide range of I/O devices
  - Connects to the processor-memory bus or backplane bus
- Backplane Bus (industry standard)
  - Backplane: an interconnection structure within the chassis
  - Allows processors, memory, and I/O devices to coexist
  - Cost advantage: one single bus for all components
- Bit-Serial Buses (New trend: USB, Firewire, ..)
  - Use high speed unidirectional point-to-point communication

A Computer System with One Bus: Backplane Bus

- A single bus (the backplane bus) is used for:
  - Processor to memory communication
  - Communication between I/O devices and memory
- Advantages: Simple and low cost
- Disadvantages: slow and the bus can become a major bottleneck
- Example: Early IBM PC

A Two-Bus System

- I/O buses tap into the processor-memory bus via bus adaptors:
  - I/O buses provide expansion slots for I/O devices
- Example: Apple Macintosh-II
  - NuBus: Processor, memory, and a few selected I/O devices
  - SCSI Bus: the rest of the I/O devices
A Three-Bus System

- A small number of backplane buses tap into the processor-memory bus.
  - Processor-memory bus is used for processor memory traffic.
  - I/O buses are connected to the backplane bus.
- Advantage: loading on the processor bus is greatly reduced.

Synchronous and Asynchronous Bus

- **Synchronous Bus:**
  - Includes a clock in the control lines.
  - A fixed protocol for communication that is relative to the clock.
  - Advantage: involves very little logic and can run very fast.
  - Disadvantages:
    - Every device on the bus must run at the same clock rate.
    - To avoid clock skew, bus must be short if it is fast.

- **Asynchronous Bus:**
  - It is not clocked.
  - It can accommodate a wide range of devices.
  - It can be lengthened without worrying about clock skew.
  - It requires a handshaking protocol.

Increasing the Bus Bandwidth

- **Separate versus multiplexed address and data lines:**
  - Address and data can be transmitted in one bus cycle if separate address and data lines are available.
  - Cost: (a) more bus lines, (b) increased complexity.

- **Data bus width:**
  - By increasing the width of the data bus, transfers of multiple words require fewer bus cycles.
  - Example: USB vs. 32-bit PCI vs. 64-bit PCI.
  - Cost: more bus lines.

- **Block transfers:**
  - Allow the bus to transfer multiple words in back-to-back bus cycles.
  - Only one address needs to be sent at the beginning.
  - The bus is not released until the last word is transferred.
  - Cost: (a) increased complexity.
  - (b) increased response time for request.

Obtaining Access to the Bus

- **One of the most important issues in bus design:**
  - How is the bus reserved by a devices that wishes to use it?
- **Chaos is avoided by a master-slave arrangement:**
  - Only the bus master can control access to the bus:
    - It initiates and controls all bus requests.
    - A slave responds to read and write requests.
- **The simplest system:**
  - Processor is the only bus master.
  - All bus requests must be controlled by the processor.
  - Major drawback: the processor is involved in every transaction.
Multiple Potential Bus Masters: the Need for Arbitration

- Bus arbitration scheme:
  - If a bus master wants to use the bus, they assert the bus request.
  - A bus master cannot use the bus until its request is granted.
  - A bus master must signal to the arbiter after finishing using the bus.

- Bus arbitration schemes usually try to balance two factors:
  - Bus priority: the highest priority device should be serviced first.
  - Fairness: Even the lowest priority device should never be completely locked out from the bus.

- Bus arbitration schemes can be divided into four broad classes:
  - Distributed arbitration by self-selection: each device wanting the bus places a code indicating its identity on the bus.
  - Distributed arbitration by collision detection: Ethernet uses this.
  - Daisy chain arbitration: single device with all request lines.
  - Centralized, parallel arbitration: see next slide.

Centralized Bus Arbitration

- Option
  - High performance
  - Low cost

- Bus width
  - Separate address & data lines
  - Multiplex address & data lines

- Data width
  - Wider is faster (e.g., 64 bits)
  - Narrower is cheaper (e.g., 8 bits)

- Transfer size
  - Multiple words has less bus overhead
  - Single-word transfer is simpler

- Bus masters
  - Multiple (requires arbitration)
  - Single master (no arbitration)

- Clocking
  - Synchronous
  - Asynchronous

Communication Networks

- Send/receive queues in processor memories
- Network controller copies back and forth via DMA
- No host intervention needed
- Interrupt host when message sent or received

Relationship to Processor Architecture

- Virtual memory frustrates DMA
  - Page faults during DMA?

- Synchronization between controller and CPU

- Caches required for processor performance cause problems for I/O
  - Flushing is expensive, I/O pollutes cache
  - Solution is borrowed from shared memory multiprocessors
    - “snooping” (coherent DMA)

- Caches and write buffers
  - Need uncached and write buffer flush for memory mapped I/O
Manufacturing Advantages of Disk Arrays

Disk Product Families

Conventional:
- 4 disk designs
- 3.5" 5.25" 10"

Disk Array:
- 1 disk design
- Low End → High End

Redundant Arrays of Disks

- Files are “striped” across multiple spindles
- Redundancy yields high data availability
- Disks will fail
- Contents reconstructed from data redundantly stored in the array
- Capacity penalty to store it
- Bandwidth penalty to update

Techniques:
- Mirroring/Shadowing (high capacity cost)
- Horizontal Hamming Codes (overkill)
- Parity & Reed-Solomon Codes
- Failure Prediction (no capacity overhead)

Summary

- I/O devices
  - Device controller
- Rotational media (disks)
  - Disk access time equation, etc.
- I/O bus
  - Memory bus
  - Standard interfaces, many devices from different vendors
- Buses
  - Asynchronous vs. synchronous
  - Bandwidth tradeoffs
  - Arbitration schemes
- I/O and virtual memory interaction