I/O devices can be characterized by
- Behavior: input, output, storage
- Partner: human or machine
- Data rate: bytes/sec, transfers/sec

I/O bus connections
## I/O Device Summary

<table>
<thead>
<tr>
<th>Device</th>
<th>Behavior</th>
<th>Partner</th>
<th>Data rate (Mbit/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td>Input</td>
<td>Human</td>
<td>0.0001</td>
</tr>
<tr>
<td>Mouse</td>
<td>Input</td>
<td>Human</td>
<td>0.0038</td>
</tr>
<tr>
<td>Voice input</td>
<td>Input</td>
<td>Human</td>
<td>0.2640</td>
</tr>
<tr>
<td>Sound input</td>
<td>Input</td>
<td>Machine</td>
<td>3.0000</td>
</tr>
<tr>
<td>Scanner</td>
<td>Input</td>
<td>Human</td>
<td>3.2000</td>
</tr>
<tr>
<td>Voice output</td>
<td>Output</td>
<td>Human</td>
<td>0.2640</td>
</tr>
<tr>
<td>Sound output</td>
<td>Output</td>
<td>Human</td>
<td>8.0000</td>
</tr>
<tr>
<td>Laser printer</td>
<td>Output</td>
<td>Human</td>
<td>3.2000</td>
</tr>
<tr>
<td>Graphics display</td>
<td>Output</td>
<td>Human</td>
<td>800.0000–8000.0000</td>
</tr>
<tr>
<td>Cable modem</td>
<td>Input or output</td>
<td>Machine</td>
<td>0.1280–6.0000</td>
</tr>
<tr>
<td>Network/LAN</td>
<td>Input or output</td>
<td>Machine</td>
<td>100.0000–10000.0000</td>
</tr>
<tr>
<td>Network/wireless LAN</td>
<td>Input or output</td>
<td>Machine</td>
<td>11.0000–54.0000</td>
</tr>
<tr>
<td>Optical disk</td>
<td>Storage</td>
<td>Machine</td>
<td>80.0000–220.0000</td>
</tr>
<tr>
<td>Magnetic tape</td>
<td>Storage</td>
<td>Machine</td>
<td>5.0000–120.0000</td>
</tr>
<tr>
<td>Flash memory</td>
<td>Storage</td>
<td>Machine</td>
<td>32.0000–200.0000</td>
</tr>
<tr>
<td>Magnetic disk</td>
<td>Storage</td>
<td>Machine</td>
<td>800.0000–3000.0000</td>
</tr>
</tbody>
</table>
I/O System Characteristics

Dependability is important
   Particularly for storage devices

Performance measures
   Latency (response time)
   Throughput (bandwidth)

Desktops & embedded systems
   Mainly interested in response time & diversity of devices

Servers
   Mainly interested in throughput & expandability of devices
Fault: failure of a component
- May or may not lead to system failure
Dependability Measures

Reliability: mean time to failure (MTTF)

Service interruption: mean time to repair (MTTR)

Mean time between failures
\[ \text{MTBF} = \text{MTTF} + \text{MTTR} \]

Availability = \( \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \)

Improving Availability

Increase MTTF: fault avoidance, fault tolerance, fault forecasting
Reduce MTTR: improved tools and processes for diagnosis and repair
Disk Storage

Nonvolatile, rotating magnetic storage
Disk Sectors and Access

Each sector records
  Sector ID
  Data (512 bytes, 4096 bytes proposed)
  Error correcting code (ECC)
    Used to hide defects and recording errors
  Synchronization fields and gaps

Access to a sector involves
  Queuing delay if other accesses are pending
  Seek: move the heads
  Rotational latency
  Data transfer
  Controller overhead
Disk Access Example

Given
512B sector, 15,000rpm, 4ms average seek time, 100MB/s transfer rate, 0.2ms controller overhead, idle disk

Average read time
4ms seek time
+ ½ / (15,000/60) = 2ms rotational latency
+ 512 / 100MB/s = 0.005ms transfer time
+ 0.2ms controller delay
= 6.2ms

If actual average seek time is 1ms
Average read time = 3.2ms
Disk Performance Issues

Manufacturers quote average seek time
   Based on all possible seeks
   Locality and OS scheduling lead to smaller actual average seek times

Smart disk controller allocate physical sectors on disk
   Present logical sector interface to host
   SCSI, ATA, SATA

Disk drives include caches
   Prefetch sectors in anticipation of access
   Avoid seek and rotational delay
<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Seagate ST33000655SS</th>
<th>Seagate ST31000340NS</th>
<th>Seagate ST973451SS</th>
<th>Seagate ST9160821AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk diameter (inches)</td>
<td>3.50</td>
<td>3.50</td>
<td>2.50</td>
<td>2.50</td>
</tr>
<tr>
<td>Formatted data capacity (GB)</td>
<td>147</td>
<td>1000</td>
<td>73</td>
<td>160</td>
</tr>
<tr>
<td>Number of disk surfaces (heads)</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Rotation speed (RPM)</td>
<td>15,000</td>
<td>7200</td>
<td>15,000</td>
<td>5400</td>
</tr>
<tr>
<td>Internal disk cache size (MB)</td>
<td>16</td>
<td>32</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>External interface, bandwidth (MB/sec)</td>
<td>SAS, 375</td>
<td>SATA, 375</td>
<td>SAS, 375</td>
<td>SATA, 150</td>
</tr>
<tr>
<td>Sustained transfer rate (MB/sec)</td>
<td>73–125</td>
<td>105</td>
<td>79–112</td>
<td>44</td>
</tr>
<tr>
<td>Minimum seek (read/write) (ms)</td>
<td>0.2/0.4</td>
<td>0.8/1.0</td>
<td>0.2/0.4</td>
<td>1.5/2.0</td>
</tr>
<tr>
<td>Average seek read/write (ms)</td>
<td>3.5/4.0</td>
<td>8.5/9.5</td>
<td>2.9/3.3</td>
<td>12.5/13.0</td>
</tr>
<tr>
<td>Mean time to failure (MTTF) (hours)</td>
<td>1,400,000 @ 25°C</td>
<td>1,200,000 @ 25°C</td>
<td>1,600,000 @ 25°C</td>
<td>—</td>
</tr>
<tr>
<td>Annual failure rate (AFR) (percent)</td>
<td>0.62%</td>
<td>0.73%</td>
<td>0.55%</td>
<td>—</td>
</tr>
<tr>
<td>Contact start-stop cycles</td>
<td>—</td>
<td>50,000</td>
<td>—</td>
<td>&gt;600,000</td>
</tr>
<tr>
<td>Warranty (years)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Nonrecoverable read errors per bits read</td>
<td>&lt;1 sector per $10^{16}$</td>
<td>&lt;1 sector per $10^{15}$</td>
<td>&lt;1 sector per $10^{16}$</td>
<td>&lt;1 sector per $10^{14}$</td>
</tr>
<tr>
<td>Size: dimensions (in.), weight (pounds)</td>
<td>1.0&quot; x 4.0&quot; x 5.8&quot;, 1.5 lbs</td>
<td>1.0&quot; x 4.0&quot; x 5.8&quot;, 1.4 lbs</td>
<td>0.6&quot; x 2.8&quot; x 3.9&quot;, 0.5 lbs</td>
<td>0.4&quot; x 2.8&quot; x 3.9&quot;, 0.2 lbs</td>
</tr>
<tr>
<td>Power: operating/idle/standby (watts)</td>
<td>15/11/—</td>
<td>11/8/1</td>
<td>8/5.8/—</td>
<td>1.9/0.6/0.2</td>
</tr>
<tr>
<td>GB/cu. in., GB/watt</td>
<td>6 GB/cu.in., 10 GB/W</td>
<td>43 GB/cu.in., 91 GB/W</td>
<td>11 GB/cu.in., 9 GB/W</td>
<td>37 GB/cu.in., 84 GB/W</td>
</tr>
<tr>
<td>Price in 2008, $/GB</td>
<td>~ $250, ~ $1.70/GB</td>
<td>~ $275, ~ $0.30/GB</td>
<td>~ $350, ~ $5.00/GB</td>
<td>~ $100, ~ $0.60/GB</td>
</tr>
</tbody>
</table>
Flash Storage

Nonvolatile semiconductor storage

100× – 1000× faster than disk

Smaller, lower power, more robust

But more $/GB (between disk and DRAM)
Flash Types

NOR flash: bit cell like a NOR gate
   Random read/write access
   Used for instruction memory in embedded systems

NAND flash: bit cell like a NAND gate
   Denser (bits/area), but block-at-a-time access
   Cheaper per GB
   Used for USB keys, media storage, …

Flash bits wears out after 10,000 to 1,000,000+ accesses
   Not suitable for direct RAM or disk replacement
   Wear leveling: remap data to less used blocks
## Contemporary Examples

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Kingston SecureDigital (SD) SD4/8 GB</th>
<th>Transend Type I CompactFlash TS16GCF133</th>
<th>RiDATA Solid State Disk 2.5 inch SATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formatted data capacity (GB)</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>Bytes per sector</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>Data transfer rate (read/write MB/sec)</td>
<td>4</td>
<td>20/18</td>
<td>68/50</td>
</tr>
<tr>
<td>Power operating/standby (W)</td>
<td>0.66/0.15</td>
<td>0.66/0.15</td>
<td>2.1/—</td>
</tr>
<tr>
<td>Size: height x width x depth (inches)</td>
<td>0.94 x 1.26 x 0.08</td>
<td>1.43 x 1.68 x 0.13</td>
<td>0.35 x 2.75 x 4.00</td>
</tr>
<tr>
<td>Weight in grams (454 grams/pound)</td>
<td>2.5</td>
<td>11.4</td>
<td>52</td>
</tr>
<tr>
<td>Mean time between failures (hours)</td>
<td>&gt; 1,000,000</td>
<td>&gt; 1,000,000</td>
<td>&gt; 4,000,000</td>
</tr>
<tr>
<td>GB/cu. in., GB/watt</td>
<td>84 GB/cu.in., 12 GB/W</td>
<td>51 GB/cu.in., 24 GB/W</td>
<td>8 GB/cu.in., 16 GB/W</td>
</tr>
<tr>
<td>Best price (2008)</td>
<td>~ $30</td>
<td>~ $70</td>
<td>~ $300</td>
</tr>
</tbody>
</table>
Interconnecting Components

Need interconnections between
- CPU, memory, I/O controllers

Bus: shared communication channel
- Parallel set of wires for data and synchronization of data transfer
- Can become a bottleneck

Performance limited by physical factors
- Wire length, number of connections

More recent alternative: high-speed serial connections with switches
- Like networks
Bus Types

Processor-Memory buses
- Short, high speed
- Design is matched to memory organization

I/O buses
- Longer, allowing multiple connections
- Specified by standards for interoperability

Connect to processor-memory bus through a bridge
Bus Signals and Synchronization

Data lines
- Carry address and data
- Multiplexed or separate

Control lines
- Indicate data type, synchronize transactions

Synchronous
- Uses a bus clock

Asynchronous
- Uses request/acknowledge control lines for handshaking
## I/O Bus Examples

<table>
<thead>
<tr>
<th></th>
<th>Firewire</th>
<th>USB 2.0</th>
<th>USB 3.0</th>
<th>PCI Express</th>
<th>Serial ATA</th>
<th>Serial Attached SCSI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intended use</strong></td>
<td>External</td>
<td>External</td>
<td>External</td>
<td>Internal</td>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td><strong>Devices per channel</strong></td>
<td>63</td>
<td>127</td>
<td>127</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td><strong>Data width</strong></td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2/lane</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td><strong>Peak bandwidth</strong></td>
<td>400 Mb/s or 800 Mb/s</td>
<td>1.6 Mb/s, 12 Mb/s, or 480 Mb/s</td>
<td>5 Gb/s, 10 Gb/x</td>
<td>2 Gb/s/lane 1×, 2×, 4×, 8×, 16×, 32×</td>
<td>2.4 Gb/s</td>
<td>2.4 Gb/s</td>
</tr>
<tr>
<td><strong>Hot pluggable</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Depends</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Max length</strong></td>
<td>4.5m</td>
<td>5m</td>
<td>?</td>
<td>0.5m</td>
<td>1m</td>
<td>8m</td>
</tr>
<tr>
<td><strong>Standard</strong></td>
<td>IEEE 1394</td>
<td>USB Implementers Forum</td>
<td>USB Implementers Forum</td>
<td>PCI-SIG</td>
<td>SATA-IO</td>
<td>INCITS TC T10</td>
</tr>
</tbody>
</table>
I/O Management

I/O is mediated by the OS
   Multiple programs share I/O resources
       Need protection and scheduling
   I/O causes asynchronous interrupts
       Same mechanism as exceptions
I/O programming is fiddly
   OS provides abstractions to programs
I/O Commands

I/O devices are managed by I/O controller hardware
  Transfers data to/from device
  Synchronizes operations with software

Command registers
  Cause device to do something

Status registers
  Indicate what the device is doing and occurrence of errors

Data registers
  Write: transfer data to a device
  Read: transfer data from a device
I/O Register Mapping

Memory mapped I/O
- Registers are addressed in same space as memory
- Address decoder distinguishes between them
- OS uses address translation mechanism to make them only accessible to kernel

I/O instructions
- Separate instructions to access I/O registers
- Can only be executed in kernel mode
- Example: x86
Polling

Periodically check I/O status register
   If device ready, do operation
   If error, take action

Common in small or low-performance real-time embedded systems
   Predictable timing
   Low hardware cost

In other systems, wastes CPU time
Interrupts

When a device is ready or error occurs
  Controller interrupts CPU

Interrupt is like an exception
  But not synchronized to instruction execution
  Can invoke handler between instructions
  Cause information often identifies the interrupting device

Priority interrupts
  Devices needing more urgent attention get higher priority
  Can interrupt handler for a lower priority interrupt
I/O Data Transfer

Polling and interrupt-driven I/O
   CPU transfers data between memory and I/O data registers
   Time consuming for high-speed devices

Direct memory access (DMA)
   OS provides starting address in memory
   I/O controller transfers to/from memory autonomously
   Controller interrupts on completion or error
DMA/Cache Interaction

If DMA writes to a memory block that is cached
   Cached copy becomes stale

If write-back cache has dirty block, and DMA reads memory block
   Reads stale data

Need to ensure cache coherence
   Flush blocks from cache if they will be used for DMA
   Or use non-cacheable memory locations for I/O
OS uses virtual addresses for memory
   DMA blocks may not be contiguous in physical memory

Should DMA use virtual addresses?
   Would require controller to do translation

If DMA uses physical addresses
   May need to break transfers into page-sized chunks
   Or chain multiple transfers
   Or allocate contiguous physical pages for DMA