I/O Systems

- Features of I/O devices: performance, benchmarks and types
- How to connect I/O devices: bus
- Bus: features, performance, types, connections, timing, arbitration,

Common Framework for Memory Hierarchy

- Question 1: Where can a Block be Placed
  - Cache:
    - direct mapped, n-way set associative
  - VM:
    - fully associative
- Question 2: How is a block found
  - index, index the set and search among elements
  - search all cache entries or separate lookup table
- Question 3: Which block be replaced
  - Random, LRU, NRU
- What happens on a write
  - write through vs write back
  - write allocate vs write no-allocate on a write miss

I/O System Design Issues (§ 8.1)

- Performance
- Expandability
- Resilience in the face of failure

I/O System Performance (§ 8.1)

- I/O System performance depends on many aspects of the system (§ 8.9)
  - The CPU
    - The memory system:
      - Internal and external caches
      - Main Memory
    - The underlying interconnection (buses)
    - The I/O controller
    - The I/O device
    - The speed of the I/O software
    - The efficiency of the software’s use of the I/O devices
- Two common performance metrics:
  - Throughput: I/O bandwidth
  - Response time: Latency

Example: Pentium System Organization
**Producer-Server Model**

Throughput:
- The number of tasks completed by the server in unit time
- In order to get the highest possible throughput:
  - The server should never be idle
  - The queue should never be empty

Response time:
- Begins when a task is placed in the queue
- Ends when it is completed by the server
- In order to minimize the response time:
  - The queue should be empty
  - The server will be idle

**Throughput versus Response Time**

- Tradeoff between response time and throughput
- Example: grouping access requests that are close may increase throughput but also increase the response time for some requests.

**Throughput Enhancement**

- In general throughput can be improved by:
  - Throwing more hardware at the problem
- Response time is much harder to reduce:
  - Ultimately it is limited by the speed of light
  - You cannot bribe God!

**I/O Benchmarks for Magnetic Disks (§ 8.2)**

- Supercomputer application:
  - Large-scale scientific problems
- Transaction processing:
  - Examples: Airline reservations systems and banks
- File system:
  - Example: UNIX file system

**Supercomputer I/O**

- Supercomputer I/O is dominated by:
  - Access to large files on magnetic disks
- Supercomputer I/O consists of one large read (read in the data)
  - Many writes to snapshot the state of the computation
- Supercomputer I/O consists of more output than input
- The overriding supercomputer I/O measures is data throughput:
  - Bytes/second that can be transferred between disk and memory

**Transaction Processing I/O**

- Transaction processing:
  - Examples: airline reservations systems, bank ATMs
  - A lot of small changes to a large body of shared data
- Transaction processing requirements:
  - Throughput and response time are important
  - Must be gracefully handling certain types of failure
- Transaction processing is chiefly concerned with I/O rate:
  - The number of disk accesses per second
- Each transaction in typical transaction processing system takes:
  - Between 2 and 10 disk I/Os
  - Between 5,000 and 20,000 CPU instructions per disk I/O
File System I/O

- Measurements of UNIX file systems in an engineering environment:
  - 80% of accesses are to files less than 10 KB
  - 90% of all file accesses are to data with sequential addresses on the disk
  - 67% of the accesses are reads
  - 27% of the accesses are writes
  - 6% of the accesses are read-write accesses

Types and Characteristics of I/O Devices

- Behavior: how does an I/O device behave?
  - Input: read only
  - Output: write only, cannot read
  - Storage: can be reread and usually rewritten

- Partner:
  - Either a human or a machine is at the other end of the I/O device
  - Either feeding data on input or reading data on output

- Data rate:
  - The peak rate at which data can be transferred:
    - Between the I/O device and the main memory
    - Or between the I/O device and the CPU

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 or Output</td>
<td>Machine</td>
<td>200.00</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>2,000.00</td>
</tr>
</tbody>
</table>

Magnetic Disk

- Purpose:
  - Long term, nonvolatile storage
  - Large, inexpensive, and slow
  - Lowest level in the memory hierarchy

- Two major types:
  - Floppy disk
  - Hard disk

- Both types of disks:
  - Rely on a rotating platter coated with a magnetic surface
  - Use a moveable read/write head to access the disk

- Advantages of hard disks over floppy disks:
  - Platters are more rigid (metal or glass) so they can be larger
  - Higher density because it can be controlled more precisely
  - Higher data rate because it spins faster
  - Can incorporate more than one platter

Organization of a Hard Magnetic Disk

- A stack of platters, a surface with a magnetic coating
- 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

Magnetic Disk Characteristic

- Disk head: each side of a platter has separate disk head
- Cylinder: all the tracks under the head at a given point on all surface
- 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 15 ms
  - (Sum of the time for all possible seeks) / (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 3,600/5400/7200 RPM
  - Approximately 16 ms per revolution
  - An average latency to the desired information is halfway around the disk: 8 ms

- **Transfer Time is a function of:**
  - Transfer size (usually a sector): 1 KB / sector
  - Rotation speed: 3600 RPM to 5400 RPM to 7200
  - Recording density: typical diameter ranges from 2 to 14 in
    - Typical values: 2 to 4 MB per second

Disk I/O Performance

- Disk Access Time = Seek time + Rotational Latency + Transfer time + Controller Time + Queuing Delay

  - Estimating Queue Length:
    - Utilization = U = Request Rate / Service Rate
    - Mean Queue Length = U / (1 - U)
    - As Request Rate → Service Rate → Mean Queue Length → Infinity

Magnetic Disk Examples (page 650)

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>IBM 3090</th>
<th>IBM0663</th>
<th>Integral 1820</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk diameter (inches)</td>
<td>10.88</td>
<td>3.50</td>
<td>1.80</td>
</tr>
<tr>
<td>Formatted data capacity (MB)</td>
<td>22,700</td>
<td>1,000</td>
<td>21</td>
</tr>
<tr>
<td>MTTF (hours)</td>
<td>50,000</td>
<td>400,000</td>
<td>100,000</td>
</tr>
<tr>
<td>Number of arms/box</td>
<td>12</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Rotation speed (RPM)</td>
<td>3,600</td>
<td>4,318</td>
<td>3,800</td>
</tr>
<tr>
<td>Transfer rate (MB/sec)</td>
<td>4.2</td>
<td>4</td>
<td>1.9</td>
</tr>
<tr>
<td>Power/watt (watts)</td>
<td>2,900</td>
<td>12</td>
<td>2</td>
</tr>
<tr>
<td>MB/watt</td>
<td>8</td>
<td>102</td>
<td>10.5</td>
</tr>
<tr>
<td>Volume (cubic feet)</td>
<td>97</td>
<td>0.13</td>
<td>0.02</td>
</tr>
<tr>
<td>MB/cubic foot</td>
<td>234</td>
<td>7692</td>
<td>1050</td>
</tr>
</tbody>
</table>

These disks represent the newest products of 1993. Compare with the newest disks of 1997 at page 650 to see how fast the disks are developed.

Example (pp.648-649)

- 512 byte sector, rotate at 5400 RPM, advertised seeks is 12 ms, transfer rate is 4 MB/sec, controller overhead is 1 ms, queue idle so no service time

  - Disk Access Time = Seek time + Rotational Latency + Transfer time + Controller Time + Queuing Delay

  - Disk Access Time = 12 ms + 0.5 / 5400 RPM + 0.5 KB / 4 MB/s + 1 ms + 0
  - Disk Access Time = 12 ms + 0.5 / 90 RPS + 0.125 / 1024 s + 1 ms + 0
  - Disk Access Time = 12 ms + 5.5 ms + 0.1 ms + 1 ms + 0 ms
  - Disk Access Time = 18.6 ms

  - If real seeks are 1/3 advertised seeks, then it is 10.6 ms, with rotation delay at 50% of the time!

Western Digital WD205BA

- Setup parameters: 16383 Cylinders, 63 sectors per track
- 3 platters, 6 heads
- Bytes per sector: 512
- RPM: 7200
- Transfer mode: 66.6MB/s
- Average Read Seek time: 9.0ms (read), 9.5ms (write)
- Average latency: 4.17ms
- Physical dimension: 1" x 4" x 5.75"
- Interleave: 1:1

Disk Arrays (p.692, 709)

- A new organization of disk storage:
  - Arrays of small and inexpensive disks
  - Increase potential throughput by having many disk drives:
    - Data is spread over multiple disk
    - Multiple accesses are made to several disks

  - Reliability is lower than a single disk:
    - But availability can be improved by adding redundant disks:
      - Lost information can be reconstructed from redundant information
    - MTTR: mean time to repair is in the order of hours
    - MTTF: mean time to failure of disks is three to five years
**The Big Picture: Where are We Now?**

- How to connect I/O to the rest of the computer?

**Buses: Connecting I/O to Processor and Memory (§ 8.4)**

- A bus is a shared communication link
- Multiple sources and multiple destinations
- It uses one set of wires to connect multiple subsystems
- Different uses: data, address and control
- A definition from dictionary:
  - An electrical connection between the components of a computer system along which the signals or power is transmitted. Information is transferred along the buses from any one of many sources to any one of many destinations. The bus consists of several parallel wires, with separate wires serving various functions: addresses, data, synchronization, control and power, etc.

**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**

- It creates a communication bottleneck
  - The bandwidth of that bus 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
- A bus transaction includes two parts:
  - Sending the address
  - Receiving or sending the data

**Master versus Slave**

- A bus transaction includes two parts:
  - Sending the address
  - Receiving or sending the data
- Master is the one who starts the bus transaction by:
  - Sending the address
- Slave is the one who 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 is defined as the Processor sending data to the I/O device:

1. **Step 1: Request Memory**
   - Processor sends a memory read request to the I/O device (Disk).
   - Memory.

2. **Step 2: Read Memory**
   - Data (Memory Address).

3. **Step 3: Send Data to I/O Device**
   - Control (Device Write Request).
   - Memory.
   - Data (I/O Device Address and then Data).

### Input Operation

Input is defined as the Processor receiving data from the I/O device:

1. **Step 1: Request Memory**
   - Processor sends a memory write request to the I/O device (Disk).
   - Memory.

2. **Step 2: Receive Data**
   - Data (I/O Device Address and then Data).

### Types of Buses

- **Processor-Memory Bus** (design specific or proprietary):
  - Short and high speed
  - Only need to match the memory system
  - Connects directly to the processor

- **I/O Bus** (industry standard):
  - Usually is 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
  - Allow processors, memory, and I/O devices to coexist
  - Cost advantage: one single bus for all components

### 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: IBM PC

### A Two-Bus System

- I/O buses tap into the processor-memory bus via bus adaptors:
  - Processor-memory bus: mainly for processor-memory traffic
  - I/O buses: provide expansion slots for I/O devices

- 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

- I/O system can be easily expanded
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, they cannot be long if they are 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

Simplest bus paradigm
- All agents operate synchronously
- All can source/sink data at the same rate
- => simple protocol
  - just manage the source and target

Simple Synchronous Protocol
- Even memory busses are more complex than this
  - memory (slave) may take time to respond
  - it need to control data rate

Typical Synchronous Protocol
- Slave indicates when it is prepared for data transfer
- Actual transfer goes at bus rate

A Handshaking Protocol (page 661)
- Three control lines
  - ReadReq: indicate a read request for memory
  - Address is put on the data lines at the same line
  - DataRdy: indicate the data word is now ready on the data lines
  - Data is put on the data lines at the same time
  - Ack: acknowledge the ReadReq or the DataRdy of the other party

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
  - Example: SPARCstation 20’s memory bus is 128 bit wide
  - Cost: more bus lines
- Data bus width:
  - By increasing the width of the data bus, transfers of multiple words require fewer bus cycles
  - Example: SPARCstation 20’s memory bus is 128 bit wide
  - 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: increased complexity
    - (b) decreased response time for request
Asynchronous Handshake – Write Operation

Write Transaction

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Read</th>
<th>Req</th>
<th>Ack</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

1. Master has obtained control and asserts address, direction, data
2. Waits a specified amount of time for slaves to decode target
3. Master asserts request line
4. Slave asserts ack, indicating data received
5. Master releases req
6. Slave releases ack

Asynchronous Handshake – Read Operation

Read Transaction

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
<th>Read</th>
<th>Req</th>
<th>Ack</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
</tbody>
</table>

1. Master has obtained control and asserts address, direction, data
2. Waits a specified amount of time for slaves to decode target
3. Master asserts request line
4. Slave asserts ack, indicating ready to transmit data
5. Master releases req, data received
6. Slave releases ack

Example: Performance Analysis (page 665)

Consider a system with the following characteristics:
- Its memory and bus supporting block access of 4 to 16 32-bit words
- 64-bit synchronous bus clocked at 200MHz, with each 64-bit transfer taking 1 clock cycle, and 1 clock cycle required to send an address to memory
- Two clock cycles needed between each bus operation
- A memory access time for the first four words of 200ns; each additional set of four words can be read in 20ns. Assume that a bus transfer of the most recently read data and a read of the next four words can be overlapped.

Find the sustained bandwidth and the latency for a read of 256 words that use 4-word blocks;

- Compute the effective number of bus transactions per second

Repeat the question for transfers that use 16-word blocks

<table>
<thead>
<tr>
<th>I/O</th>
<th>Memory</th>
<th>Block</th>
<th>address</th>
<th>Read</th>
<th>Data</th>
<th>transfer</th>
<th>Bus</th>
<th>idle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latency = 2880 clock cycles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bandwidth = 71.11MB/sec</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Obtaining Access to the Bus

One of the most important issues in bus design:
- How is the bus reserved by a device 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

Split Bus Transaction (page 666: elaboration)

- Request-Reply
  - CPU initiates a read or write transaction
  - address, data, and command
  - then waiting for reply from memory
- Split bus transaction
  - CPU initiates a read or write transaction
  - address, data, and command
  - Memory initiates a reply transaction
  - data (read) or acknowledge (write)
- Bandwidth is improved
- Latency for an individual read/write ??

Multiple Potential Bus Masters: the Need for Arbitration

- Bus arbitration scheme:
  - A bus master wanting to use the bus asserts 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: see next slide.
  - Centralized, parallel arbitration: see next-next slide
The Daisy Chain Bus Arbitrations Scheme

- Advantage: simple
- Disadvantages:
  - Cannot assure fairness: A low-priority device may be locked out indefinitely
  - The use of the daisy chain grant signal also limits the bus speed

Centralized Parallel Arbitration

- Used in essentially all processor-memory busses and in high-speed I/O busses

Centralized Arbitration with a Bus Arbiter

Simple Implementation of a Bus Arbiter

Priority Logic

JK Flip Flop

- JK Flip Flop can be implemented with a D-Flip Flop

<table>
<thead>
<tr>
<th>J</th>
<th>K</th>
<th>Q_{t-1}</th>
<th>Q_t</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
**1993 MP Server Memory Bus Survey: GTL revolution**

<table>
<thead>
<tr>
<th>Bus</th>
<th>MBus</th>
<th>Summit</th>
<th>Challenge</th>
<th>XDBus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Originator</td>
<td>Sun</td>
<td>HP</td>
<td>SGI</td>
<td>Sun</td>
</tr>
<tr>
<td>Clock Rate (MHz)</td>
<td>40</td>
<td>60</td>
<td>48</td>
<td>66</td>
</tr>
<tr>
<td>Address lines</td>
<td>36</td>
<td>48</td>
<td>40 muxed</td>
<td></td>
</tr>
<tr>
<td>Data lines</td>
<td>64</td>
<td>128</td>
<td>256</td>
<td>144 (parity)</td>
</tr>
<tr>
<td>Data Sizes (bits)</td>
<td>256</td>
<td>512</td>
<td>1024</td>
<td>512</td>
</tr>
<tr>
<td>Clocks/transfer</td>
<td>4</td>
<td>5</td>
<td>4?</td>
<td></td>
</tr>
<tr>
<td>Peak (MB/s)</td>
<td>320</td>
<td>960</td>
<td>1200</td>
<td>1056</td>
</tr>
<tr>
<td>Master</td>
<td>Multi</td>
<td>Multi Multi</td>
<td>Multi</td>
<td></td>
</tr>
<tr>
<td>Arbitration</td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
</tr>
<tr>
<td>Slots</td>
<td>16</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Busses/system</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Length</td>
<td>13 inches</td>
<td>127 inches</td>
<td>17 inches</td>
<td></td>
</tr>
</tbody>
</table>

**1993 Backplane/IO Bus Survey**

<table>
<thead>
<tr>
<th>Bus</th>
<th>SRBus</th>
<th>TurboChannel</th>
<th>MicroChannel</th>
<th>PCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Originator</td>
<td>Sun</td>
<td>DEC</td>
<td>IBM</td>
<td>Intel</td>
</tr>
<tr>
<td>Clock Rate (MHz)</td>
<td>16-25</td>
<td>12.5-25 async</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Addressing</td>
<td>Virtual</td>
<td>Physical</td>
<td>Physical</td>
<td>Physical</td>
</tr>
<tr>
<td>Data Sizes (bits)</td>
<td>8,16,32</td>
<td>8,16,24,32</td>
<td>8,16,24,32,64</td>
<td>8,16,24,32,64</td>
</tr>
<tr>
<td>Master</td>
<td>Multi</td>
<td>Single Multi</td>
<td>Multi</td>
<td>Multi</td>
</tr>
<tr>
<td>Arbitration</td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
</tr>
<tr>
<td>32 bit read (MB/s)</td>
<td>33</td>
<td>25</td>
<td>20</td>
<td>33</td>
</tr>
<tr>
<td>Peak (MB/s)</td>
<td>89</td>
<td>84</td>
<td>75</td>
<td>111 (222)</td>
</tr>
<tr>
<td>Max Power (W)</td>
<td>16</td>
<td>26</td>
<td>13</td>
<td>25</td>
</tr>
</tbody>
</table>

**High Speed I/O Bus**
- Examples
  - graphics
  - fast networks
- Limited number of devices
- Data transfer bursts at full rate
- DMA transfers important
  - small controller spools stream of bytes to or from memory
- Either side may need to squelch transfer
  - buffers fill up

**PCI Transactions**
- All signals sampled on rising edge
- Centralized Parallel Arbitration
  - overlapped with previous transaction
- Bus Parking
  - retain bus grant for previous master until another makes request
  - granted master can start next transfer without arbitration
- All transfers are bursts; Arbitrary Burst length
  - initiator and target can exert flow control with xRDY
  - target can disconnect request with STOP (abort or retry)
  - master can disconnect by deasserting FRAME
  - arbiter can disconnect by deasserting GNT
- Delayed (pended, split-phase) transactions
  - free the bus after request to slow device

**Summary of Bus Options** (Fig 8.14, page 671)
- Option
  - High performance
  - Low cost
- Bus width
  - Separate address & data lines
  - Multiplex address & data lines
- Data width
  - Wider is faster (e.g., 32 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
- Clocking rate
  - Higher is faster (e.g., 66MHz for PCI)
  - Lower is easier (e.g., 5MHz for SCSI)
- Protocol
  - pipelined
  - Serial