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”
- Performance measures for I/O systems: see pp.639
- Very diverse devices (Section 8.3)
  - behavior (i.e., input vs. output), partner (who is at the other end?)

<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>Voice input</td>
<td>input</td>
<td>human</td>
<td>0.02</td>
</tr>
<tr>
<td>Scanner</td>
<td>input</td>
<td>human</td>
<td>400.00</td>
</tr>
<tr>
<td>Voice output</td>
<td>output</td>
<td>human</td>
<td>0.60</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>200.00</td>
</tr>
<tr>
<td>Graphics display</td>
<td>output</td>
<td>human</td>
<td>60,000.00</td>
</tr>
<tr>
<td>Modem</td>
<td>input or output</td>
<td>machine</td>
<td>2.00-8.00</td>
</tr>
<tr>
<td>Network/LAN</td>
<td>input or output</td>
<td>machine</td>
<td>500.00-6000.00</td>
</tr>
<tr>
<td>Floppy disk</td>
<td>storage</td>
<td>machine</td>
<td>100.00</td>
</tr>
<tr>
<td>Optical disk</td>
<td>storage</td>
<td>machine</td>
<td>1000.00</td>
</tr>
<tr>
<td>Magnetic tape</td>
<td>storage</td>
<td>machine</td>
<td>2000.00</td>
</tr>
<tr>
<td>Magnetic disk</td>
<td>storage</td>
<td>machine</td>
<td>2000.00-10,000.00</td>
</tr>
</tbody>
</table>

I/O Example: Disk Drives

- To access data:
  - seek: position head over the proper track (8 to 20 ms. avg.)
  - rotational latency: wait for desired sector (.5 / RPM), see pp.647
  - transfer: grab the data (one or more sectors) 2 to 15 MB/sec
- Disk Read time: average seek time + average rotational delay + transfer time + controller overhead; see pp.648
8.4 Buses

- What is a bus? A bunch of wires; Shared communication link (one or more wires); Used to connect multiple subsystems; A Bus is also a fundamental tool for composing large, complex systems

- Difficult design:
  - may be bottleneck
  - length of the bus
  - number of devices
  - cost
  - tradeoffs (buffers for higher bandwidth increases latency)
  - support for many different devices

- Advantages of bus: versatility
  - New devices can be added easily
  - Peripherals can be moved between computer systems that use the same bus standard

Types of Buses

- Processor-Memory Bus (design specific)
  - Short and high speed
  - Only need to match the memory system
    - Maximize memory-to-processor bandwidth
  - Connects directly to the processor
  - Optimized for cache block transfers, custom design

- I/O Bus (industry standard, e.g., SCSI: small computer system interface)
  - 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 (standard or proprietary, e.g., PCI: Peripheral component interconnect bus)
  - Backplane: an interconnection structure within the chassis
  - Allow processors, memory, and I/O devices to coexist
  - Cost advantage: one bus for all components
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

• What defines a bus?
  - Transaction Protocol
  - Timing and Signaling Specification
  - Bunch of Wires
  - Electrical Specification
  - Physical / Mechanical Characteristics: the connectors
Master versus Slave

- A bus transaction includes three parts:
  - Arbitration: Which bus master gets to use the bus next
  - Request: Issuing the command (and address)
  - Action: Transferring the data
- Master is the one who starts the bus transaction by issuing the command (and address)
- Slave is the one who responds to the request by sending data to the master if the master asks for data or receiving data from the master if the master wants to send data

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 - AT
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
  - SCCI 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, 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

---

**Fig. 8.10/11: Asynchronous Handshaking Protocol**

1. **ReadReq**
2. **Data**
3. **Ack**
4. **DataRdy**
5. **Read memory data from data lines; assert Ack**
6. **Release data lines and DataRdy**
7. **Deassert Ack**

---

**Tsung-Han Tsai**
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: SPARCstation 20 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: (a) increased complexity (b) decreased response time for request

Arbitration: 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:
  - 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 finish 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:
  - Daisy chain arbitration: single device with all request lines.
  - Centralized, parallel arbitration: see next-next slide
  - 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.

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

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</td>
<td>muxed</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 (80)</td>
<td>960</td>
<td>1200</td>
<td>1056</td>
</tr>
<tr>
<td>Master</td>
<td>Multi</td>
<td>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>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>12? inches</td>
<td>17 inches</td>
<td></td>
</tr>
</tbody>
</table>
**1993 Backplane/IO Bus Survey**

<table>
<thead>
<tr>
<th>Bus</th>
<th>SBus</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</td>
<td>async</td>
<td>33</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</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

**Summary of Bus Options**

- **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

- **Protocol**
  - pipelined
  - Serial
Multimedia Bandwidth Requirements

- **High Quality Video**
  - Digital Data = (30 frames / second) (640 x 480 pels) (24-bit color / pel) = 221 Mbps (27.6 MB/s)

- **Reduced Quality Video**
  - Digital Data = (15 frames / second) (320 x 240 pels) (16-bit color / pel) = 18 Mbps (2.2 MB/s)

- **High Quality Audio**
  - Digital Data = (44,100 audio samples / sec) (16-bit audio samples) = 1.4 Mbps

- **Reduced Quality Audio**
  - Digital Data = (11,050 audio samples / sec) (8-bit audio samples) (1 audio channel for monaural) = 0.1 Mbps

- compression changes the whole story!

Other important issues

- **Interfacing I/O devices to the memory, processor: Operating system**
- **Devices communicate with processor; Transferring the data between a device and memory by**
  - polling: simplest way; processor is totally in control; can waste a lot of processor time; example: p676
  - interrupt-driven: like exception; control unit check for pending I/O interrupt at the time it starts a new instruction; example: p679
  - DMA: specialized controller is used to transfers data between an I/O device and memory independent of the processor; example: p681