Interfacing Processors and Peripherals

Diagram showing the interfacing of processors and peripherals, including:
- Processor
- Cache
- Main memory
- I/O controller
- Disk
- I/O controller
- Graphics output
- Network
- Interrupts
- Memory–I/O bus
Buses

- Bus lines
  - Control lines
  - Data lines (data, commands, addresses)
- Bus transactions
  - Read (output): memory to I/O device
  - Write (input): I/O device to memory
- Types of buses
  - Processor-memory (specific)
  - I/O buses (standard)
  - Backplane bases (standard)
Bus transactions: output

- Request a read on control lines and supply data at the data lines
- Memory access
- Memory signals data availability and transfer them on the data line to I/O
Bus transactions: input

- Write request for memory on the control lines and address on the data lines
- Memory signal the I/O device that is ready and transfers starts
Types of buses

a. Processor → Backplane bus → Memory
   - I/O devices

b. Processor
   - Processor-memory bus
   - Bus adapter
   - Bus adapter
   - Bus adapter
   - I/O bus
   - I/O bus
   - I/O bus

C. Processor
   - Processor-memory bus
   - Bus adapter
   - Bus adapter
   - Backplane bus
   - I/O bus
   - I/O bus
Synchronous and Asynchronous buses

- Synchronous buses: Using a clock synchronized communication protocol
  - All devices use a single clock
  - Cannot be long
- Asynchronous buses: handshaking protocol
  - Reading a word from memory:
    - Read request (*ReadReq*)
    - Data are ready to read (*DataRdy*)
    - Acknowledge the ReadReq and DataRdy signals of the other party (*Ack*)
Implementing handshaking

1. Record from data lines and assert Ack
2. Release data lines; deassert ReadReq
3. Drop Ack; put memory data on data lines; assert DataRdy
4. Read memory data from data lines; assert Ack
5. Read data lines; deassert ReadReq
6. Release data lines and DataRdy
7. Deassert Ack

New I/O request
Increasing the bus bandwidth

- Increasing the width of the data bus lines
- Using separate address and data lines
- Transferring multiple words (blocks)
Bus access: master and slave

- I/O device generates a request to the processor to use the bus
- The processor responds and generates the corresponding bus control signals
- The processor notifies the device and the device places the address on the bus
Bus arbitration schemes

- **Daisy chain arbitration:** A single bus grant line is run through the devices from highest priority to lowest. A higher priority device intercepts the bus grant signal, not allowing a lower priority device to see it.
- **Centralized, parallel arbitration:** Using multiple request lines managed by a centralized arbiter.
- **Distributed arbitration by self-selection:** Multiple request lines, but the device requesting the bus access determine who will be granted access. Each bus requesting device places its identity code on the bus.
- **Distributed arbitration by collision detection:** Each device independently requests the bus. Multiple simultaneous requests result in a collision, which is detected and solved.
## Bus standards

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>PCI</th>
<th>SCSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Backplane</td>
<td>I/O</td>
</tr>
<tr>
<td>Data bus width</td>
<td>32-64</td>
<td>8-32</td>
</tr>
<tr>
<td>Address/data multiplexed?</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of bus masters</td>
<td>multiple</td>
<td>multiple</td>
</tr>
<tr>
<td>Arbitration</td>
<td>Centralized, parallel</td>
<td>Self-selection</td>
</tr>
<tr>
<td>Clocking</td>
<td>Synchronous 33-66MHz</td>
<td>Asynchronous or synchronous 5-10MHz</td>
</tr>
<tr>
<td>Theoretical peak bandwidth</td>
<td>133-512 MB/sec</td>
<td>5-40 MB/sec</td>
</tr>
<tr>
<td>Estimated achievable bandwidth</td>
<td>80 MB/sec</td>
<td>2.5-40 MB/sec</td>
</tr>
<tr>
<td>Maximum number of devices</td>
<td>1024</td>
<td>7-31</td>
</tr>
<tr>
<td>Maximum bus length</td>
<td>0.5 meter</td>
<td>25 meters</td>
</tr>
</tbody>
</table>