Input / Output

Reading: Chapter 9

Basic Model

- Processing speed or program execution
  - determined primarily by ability of I/O operations to stay ahead of processor.

Input → Process → Output

Goal is to have the processor not wait for I/O to complete

Copyright 2011 John Wiley & Sons, Inc. & Robert F. Kelly
Examples of I/O Devices

<table>
<thead>
<tr>
<th>Device</th>
<th>Input/Output</th>
<th>Data rate</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td>Input</td>
<td>100 bps</td>
<td>char</td>
</tr>
<tr>
<td>Mouse</td>
<td>Input</td>
<td>3800 bps</td>
<td>char</td>
</tr>
<tr>
<td>Voice input/output</td>
<td>Input/Output</td>
<td>26.4 Kbps</td>
<td>block burst</td>
</tr>
<tr>
<td>Sound input</td>
<td>Input</td>
<td>3 Mbps</td>
<td>block burst or steady</td>
</tr>
<tr>
<td>Scanner</td>
<td>Input</td>
<td>3.2 Mbps</td>
<td>block burst</td>
</tr>
<tr>
<td>Laser printer</td>
<td>Output</td>
<td>3.2 Mbps</td>
<td>block burst</td>
</tr>
<tr>
<td>Sound output</td>
<td>Output</td>
<td>8 Mbps</td>
<td>block burst or steady</td>
</tr>
<tr>
<td>Flash drive</td>
<td>Storage</td>
<td>400-500 Mbps read; 80 Mbps write</td>
<td>block burst</td>
</tr>
<tr>
<td>USB</td>
<td>Input or output</td>
<td>1.6-480 Mbps</td>
<td>block burst</td>
</tr>
<tr>
<td>Network/Wireless LAN</td>
<td>Input or output</td>
<td>11-100 Mbps</td>
<td>block burst</td>
</tr>
<tr>
<td>Network/LAN</td>
<td>Input or output</td>
<td>100-1000 Mbps</td>
<td>block burst</td>
</tr>
<tr>
<td>Graphics display</td>
<td>Output</td>
<td>80-8000 Mbps</td>
<td>block burst or steady</td>
</tr>
<tr>
<td>Optical disk</td>
<td>Storage</td>
<td>4-400 Mbps</td>
<td>block burst or steady</td>
</tr>
<tr>
<td>Magneto tape</td>
<td>Storage</td>
<td>32-90 Mbps</td>
<td>block burst or steady</td>
</tr>
<tr>
<td>Magnetic disk</td>
<td>Storage</td>
<td>240-3000 Mbps</td>
<td>block burst</td>
</tr>
</tbody>
</table>

I/O Requirements

- Means for addressing different peripheral devices
- A way for peripheral devices to initiate communication with the CPU
- An efficient means of transferring data directly between I/O and memory for large data transfers since programmed I/O (i.e., word at a time) is suitable only for slow devices and individual word transfers
- Buses that interconnect high-speed I/O devices with the computer must support high data transfer rates
- Means for handling devices with extremely different control requirements
I/O Interfaces

- Are necessary because of
  - Different formats required by the devices
  - Incompatibilities in speed between the devices and the CPU make synchronization difficult
  - Bursts of data vs. streaming data
  - Device control requirements that would tie up too much CPU time

Simple I/O Configuration

[Diagram showing CPU connected to I/O module and I/O device, also referred to as a device controller]
More Complex I/O Module

Advanced I/O Techniques

- Programmed I/O
  - CPU controlled I/O
  - Direct movement from controller to CPU register (similar to memory access in CPU)
- Interrupt Driven I/O
  - External input controls
- Direct Memory Access Controllers
  - Method for transferring data between main memory and a device that bypasses the CPU
Programmed I/O

- I/O data and address registers in CPU
- One word transfer per I/O instruction
- Address information for each I/O device
  - LMC I/O capability for 100 devices
- Full instruction fetch/execute cycle
- Primary use:
  - keyboards
  - communication with I/O modules (see DMA)

Interrupts

- Signal that causes the CPU to alter its normal flow of instruction execution
  - frees CPU from waiting for events
  - provides control for external I/O initiation
- Examples
  - unexpected input
  - abnormal situation
  - illegal instructions
  - multitasking, multiprocessing
Interrupt Terminology

- Interrupt lines (hardware)
  - One or more special control lines to the CPU
- Interrupt request
- Interrupt handlers
  - Program that services the interrupt
  - Also known as an interrupt routine or device driver
- Context
  - Saved registers of a program before control is transferred to the interrupt handler
  - Allows program to resume exactly where it left off when control returns to interrupted program

Use of Interrupts

- Notify that an external event has occurred
  - real-time or time-sensitive
- Signal completion
  - printer ready or buffer full
- Allocate CPU time
  - time sharing
- Indicate abnormal event (CPU originates for notification and recovery)
  - illegal operation, hardware error
- Software interrupts
**Servicing the Interrupt**

1. Lower priority interrupts are held until higher priority interrupts are complete
2. Suspend program in progress
3. Save context, including last instruction executed and data values in registers, in the PCB or the stack area in memory
4. Branch to interrupt handler program (aka, device driver)

*PCB is the process control block*

---

**Interrupt Processing Methods**

- **Vectored interrupt**
  - Address of interrupting device is included in the interrupt
  - Requires additional hardware to implement

- **Polling**
  - Identifies interrupting device by polling each device
  - General interrupt is shared by all devices
**Vectored Interrupts**

Interrupt K occurs → Address of interrupt A → Address of interrupt B → ... → Address of interrupt K → Interrupt A service routine → ... → Interrupt K service routine → Jump to K service routine

---

**Polled Interrupts**

Interrupt K occurs → General interrupt polling routine → ... → Interrupt A service routine → ... → Interrupt K service routine → Polls devices to determine which device, then Jump to K service routine

---

Copyright 2011 John Wiley & Sons, Inc. & Robert F. Kelly
Direct Memory Access

- Transferring large blocks of data
- Direct transfer to and from memory
- CPU not actively involved in transfer itself
- Required conditions for DMA
  - The I/O interface and memory must be connected
  - The I/O module must be capable of reading and writing to memory
  - Conflicts between the CPU and the I/O module must be avoided
  - Interrupt required for completion

DMA Instructions

- Application program requests I/O service from operating system
  - e.g., Halt on the LMC was a privileged instruction
  - privileged programmed I/O instructions
- To initiate DMA, programmed I/O is used to send the following information:
  1. location of data on I/O device
  2. the starting location in memory
  3. the size of the block
  4. read/write
- Interrupt to CPU upon completion of DMA
DMA

- Examples
  - ISA (becoming obsolete)
  - PCI
  - I/O Acceleration

- Devices
  - Disk
  - Graphics cards
  - Network cards
  - Sound cards

Copyright 2011 John Wiley & Sons, Inc. & Robet F. Kelly