Computer Architecture
(CSC-3501)
Lecture 23
(17 April 2008)

Seung-Jong Park (Jay)
http://www.csc.lsu.edu/~sjpark

Announcement

- Homework #8 and #9 are uploaded at class website

Chapter 7 Objectives

- Understand how I/O systems work, including I/O methods and architectures.
- Become familiar with storage media, and the differences in their respective formats.
- Understand how RAID improves disk performance and reliability, and which RAID systems are most useful today.
- Be familiar with emerging data storage technologies and the barriers that remain to be overcome.

7.1 Introduction

- Data storage and retrieval is one of the primary functions of computer systems.
  - One could easily make the argument that computers are more useful to us as data storage and retrieval devices than they are as computational machines.
- All computers have I/O devices connected to them, and to achieve good performance I/O should be kept to a minimum!
- In studying I/O, we seek to understand the different types of I/O devices as well as how they work.

7.2 I/O and Performance

- Sluggish I/O throughput can have a ripple effect, dragging down overall system performance.
  - This is especially true when virtual memory is involved.
- The fastest processor in the world is of little use if it spends most of its time waiting for data.
- If we really understand what's happening in a computer system we can make the best possible use of its resources.

7.3 Amdahl's Law

- The overall performance of a system is a result of the interaction of all of its components.
- System performance is most effectively improved when the performance of the most heavily used components is improved.
- This idea is quantified by Amdahl's Law:

$$S = \frac{1}{(1-f) + \frac{f}{k}}$$

where S is the overall speedup;
f is the fraction of work performed by a faster component; and
k is the speedup of the faster component.

7.3 Amdahl's Law

- Amdahl's Law gives us a handy way to estimate the performance improvement we can expect when we upgrade a system component.
- On a large system, suppose we can upgrade a CPU to make it 50% faster for $10,000 or upgrade its disk drives for $7,000 to make them 150% faster.
- Processes spend 70% of their time running in the CPU and 30% of their time waiting for disk service.
- An upgrade of which component would offer the greater benefit for the lesser cost?

7.3 Amdahl's Law

- The processor option offers a 30% performance improvement: 1.3

$$f = 0.70,\quad k = 1.5 \qquad S = \frac{1}{(1 - 0.7) + 0.7/1.5}$$

- And the disk drive option gives a 22% performance improvement: 1.22

$$f = 0.30,\quad k = 2.5 \qquad S = \frac{1}{(1 - 0.3) + 0.3/2.5}$$

- Each 1% of improvement for the processor costs $333, and for the disk a 1% improvement costs $318.

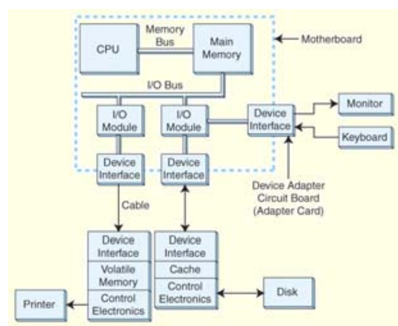
Should price/performance be your only concern?

7.4 I/O Architectures

- We define input/output as a subsystem of components that moves coded data between external devices and a host system.
- I/O subsystems include:
  - Blocks of main memory that are devoted to I/O functions.
  - Buses that move data into and out of the system.
  - Control modules in the host and in peripheral devices
  - Interfaces to external components such as keyboards and disks.
  - Cabling or communications links between the host system and its peripherals.

7.4 I/O Architectures

This is a model I/O configuration.



7.4 I/O Architectures

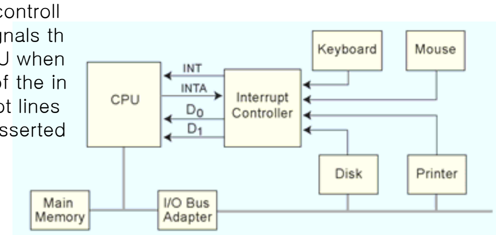
- I/O can be controlled in four general ways.
- Programmed I/O reserves a register for each I/O device. Each register is continually polled to detect data arrival.
- Interrupt-Driven I/O allows the CPU to do other things until I/O is requested.
- Direct Memory Access (DMA) offloads I/O processing to a special-purpose chip that takes care of the details.
- Channel I/O uses dedicated I/O processors.

7.4 I/O Architectures

This is an idealized I/O subsystem that uses interrupts.

Each device connects its interrupt line to the interrupt controller.

The controller signals the CPU when any of the interrupt lines are asserted.



7.4 I/O Architectures

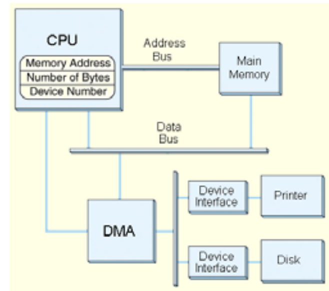
- Recall from Chapter 4 that in a system that uses interrupts, the status of the interrupt signal is checked at the top of the fetch-decode-execute cycle.
- The particular code that is executed whenever an interrupt occurs is determined by a set of addresses called *interrupt vectors* that are stored in low memory.
- The system state is saved before the interrupt service routine is executed and is restored afterward.

7.4 I/O Architectures

This is a DMA configuration.

Notice that the DMA and the CPU share the bus.

The DMA runs at a higher priority and steals memory cycles from the CPU.



7.4 I/O Architectures

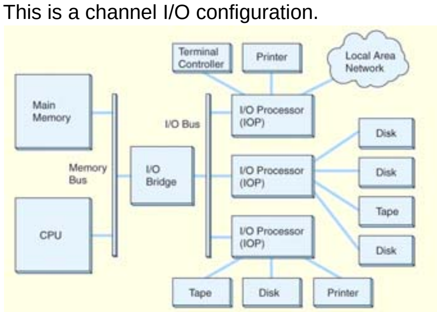
- Very large systems employ channel I/O.
- Channel I/O consists of one or more I/O processors (IOPs) that control various channel paths.
- Slower devices such as terminals and printers are combined (*multiplexed*) into a single faster channel.
- On IBM mainframes, multiplexed channels are called *multiplexor channels*, the faster ones are called *selector channels*.

7.4 I/O Architectures

- Channel I/O is distinguished from DMA by the intelligence of the IOPs.
- The IOP negotiates protocols, issues device commands, translates storage coding to memory coding, and can transfer entire files or groups of files independent of the host CPU.
- The host has only to create the program instructions for the I/O operation and tell the IOP where to find them.

7.4 I/O Architectures

- This is a channel I/O configuration.



7.4 I/O Architectures

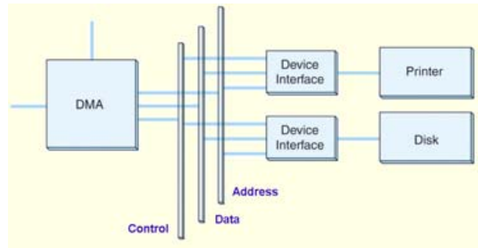
- Character I/O devices process one byte (or character) at a time.
  - Examples include modems, keyboards, and mice.
  - Keyboards are usually connected through an interrupt-driven I/O system.
- Block I/O devices handle bytes in groups.
  - Most mass storage devices (disk and tape) are block I/O devices.
  - Block I/O systems are most efficiently connected through DMA or channel I/O.

7.4 I/O Architectures

- I/O buses, unlike memory buses, operate asynchronously. Requests for bus access must be arbitrated among the devices involved.
- Bus control lines activate the devices when they are needed, raise signals when errors have occurred, and reset devices when necessary.
- The number of data lines is the *width* of the bus.
- A bus clock coordinates activities and provides bit cell boundaries.

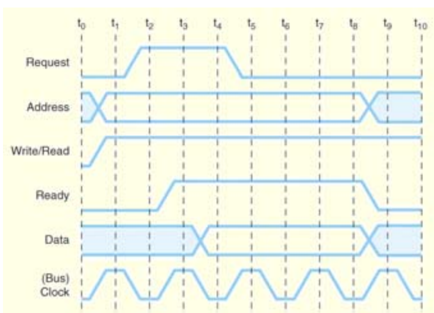
7.4 I/O Architectures

This is a generic DMA configuration showing how the DMA circuit connects to a data bus.



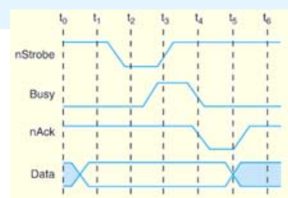
7.4 I/O Architectures

Timing diagrams, such as this one, define bus operation in detail.



7.5 Data Transmission Modes

- Bytes can be conveyed from one point to another by sending their encoding signals simultaneously using *parallel data transmission* or by sending them one bit at a time in *serial data transmission*.
  - Parallel data transmission for a printer resembles the signal protocol of a memory bus:



7.5 Data Transmission Modes

- In parallel data transmission, the interface requires one conductor for each bit.
- Parallel cables are fatter than serial cables.
- Compared with parallel data interfaces, serial communications interfaces:
  - Require fewer conductors.
  - Are less susceptible to attenuation.
  - Can transmit data farther and faster.

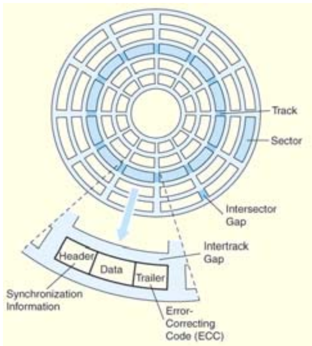
Serial communications interfaces are suitable for time-sensitive (*isochronous*) data such as voice and video.

7.6 Magnetic Disk Technology

- Magnetic disks offer large amounts of durable storage that can be accessed quickly.
- Disk drives are called *random* (or *direct*) *access storage devices,* because blocks of data can be accessed according to their location on the disk.
  - This term was coined when all other durable storage (e.g., tape) was sequential.
- Magnetic disk organization is shown on the following slide.

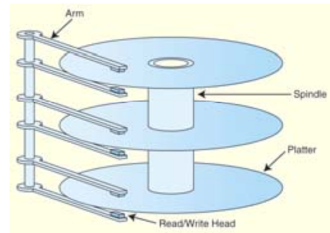
7.6 Magnetic Disk Technology

Disk tracks are numbered from the outside edge, starting with zero.



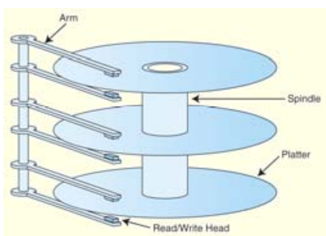
7.6 Magnetic Disk Technology

- Hard disk platters are mounted on spindles.
- Read/write heads are mounted on a comb that swings radially to read the disk.



7.6 Magnetic Disk Technology

- The rotating disk forms a logical cylinder beneath the read/write heads.
- Data blocks are addressed by their cylinder, surface, and sector.



7.6 Magnetic Disk Technology

- There are a number of electromechanical properties of hard disk drives that determine how fast its data can be accessed.
- *Seek time* is the time that it takes for a disk arm to move into position over the desired cylinder.
- *Rotational delay* is the time that it takes for the desired sector to move into position beneath the read/write head.
- Seek time + rotational delay = *access time*.

7.6 Magnetic Disk Technology

- *Transfer rate* gives us the rate at which data can be read from the disk.
- *Average latency* is a function of the rotational speed:

$$\frac{\frac{60 \text{ seconds}}{\text{disk rotation speed}} \times \frac{1000 \text{ ms}}{\text{second}}}{2}$$

- *Mean Time To Failure (MTTF)* is a statistically-determined value often calculated experimentally.
  - It usually doesn't tell us much about the actual expected life of the disk. *Design life* is usually more realistic.

Figure 7.11 in the text shows a sample disk specification.

7.6 Magnetic Disk Technology

- Floppy (flexible) disks are organized in the same way as hard disks, with concentric tracks that are divided into sectors.
- Physical and logical limitations restrict floppies to much lower densities than hard disks.
- A major logical limitation of the DOS/Windows floppy diskette is the organization of its file allocation table (FAT).
  - The FAT gives the status of each sector on the disk: Free, in use, damaged, reserved, etc.

7.6 Magnetic Disk Technology

- On a standard 1.44MB floppy, the FAT is limited to nine 512-byte sectors.
  - There are two copies of the FAT.
- There are 18 sectors per track and 80 tracks on each surface of a floppy, for a total of 2880 sectors on the disk. So each FAT entry needs at least 12 bits (2^{11}= 2048 < 2880 < 2^{12} = 4096).
  - Thus, FAT entries for disks smaller than 10MB are 12 bits, and the organization is called FAT12.
  - FAT 16 is employed for disks larger than 10MB.

7.6 Magnetic Disk Technology

- The disk directory associates logical file names with physical disk locations.
- Directories contain a file name and the file's first FAT entry.
- If the file spans more than one sector (or cluster), the FAT contains a pointer to the next cluster (and FAT entry) for the file.
- The FAT is read like a linked list until the <EOF> entry is found.

7.6 Magnetic Disk Technology

- A directory entry says that a file we want to read starts at sector 121 in the FAT fragment shown below.

FAT Index →	120	121	122	123	124	125	126	127
FAT Contents	97	124	<EOF>	1258	126	<BAD>	122	577

  - Sectors 121, 124, 126, and 122 are read. After each sector is read, its FAT entry is to find the next sector occupied by the file.
  - At the FAT entry for sector 122, we find the end-of-file marker <EOF>.

How many disk accesses are required to read this file?