Computer Architecture
(CSC-3501)
Lecture 10
(19 Feb 2008)

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

Announcement

Chapter 4. Objectives

- Learn the components common to every modern computer system.
- Be able to explain how each component contributes to program execution.
- Understand a simple architecture invented to illuminate these basic concepts, and how it relates to some real architectures.
- Know how the program assembly process works.

CPU

- The computer's CPU fetches, decodes, and executes program instructions.
- The two principal parts of the CPU are the *datapath* and the *control unit*.
  - The datapath consists of an arithmetic-logic unit and storage units (registers) that are interconnected by a data bus that is also connected to main memory.
  - Various CPU components perform sequenced operations according to signals provided by its control unit.

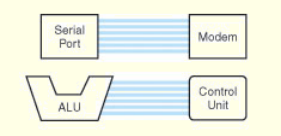
CPU

- Registers hold data that can be readily accessed by the CPU.
- They can be implemented using D flip-flops.
  - A 32-bit register requires 32 D flip-flops.
- The arithmetic-logic unit (ALU) carries out logical and arithmetic operations as directed by the control unit.
- The control unit determines which actions to carry out according to the values in a program counter register and a status register.

BUS

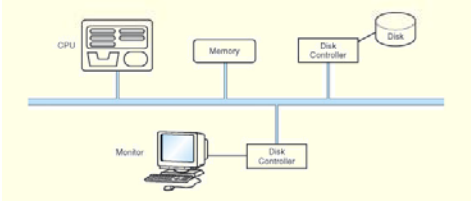
- The CPU shares data with other system components by way of a data bus.
  - A bus is a set of wires that simultaneously convey a single bit along each line.
- Two types of buses are commonly found in computer systems: *point-to-point*, and *multipoint* buses.

This is a point-to-point bus configuration:



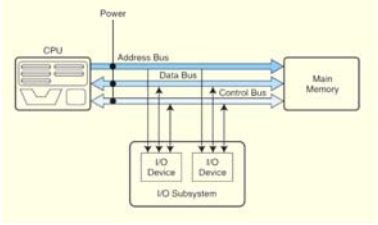
Multipoint Bus

- Because a multipoint bus is a shared resource, access t o it is controlled through protocols, which are built into th e hardware.



BUS

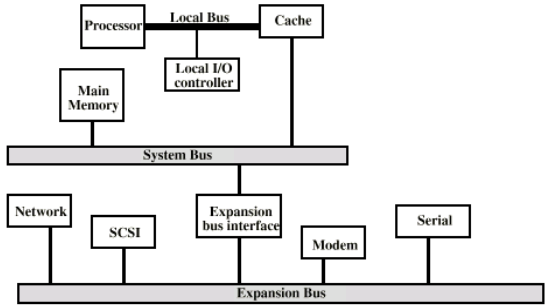
- Buses consist of data lines, control lines, and address lines.
- While the data lines convey bits from one device to another, control li nes determine the direction of data flow, and when each device can access the bus.
- Address lines determine the location of the source or destination of t he data.



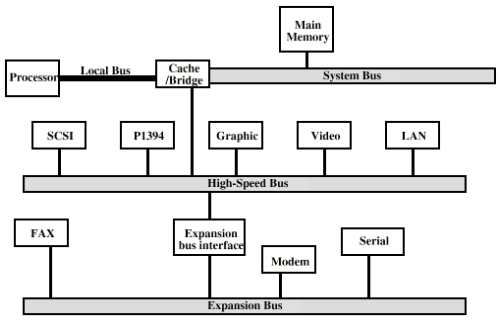
BUS Arbitration

- In a master-slave configuration, where more than one de vice can be the bus master, concurrent bus master requ ests must be arbitrated.
- Four categories of bus arbitration are:
  - **Daisy chain:** Permissions are passed from the highest-pri ority device to the lowest.
  - **Centralized parallel:** Each dev ice is directly connected to a n arbitration circuit.
  - **Distributed using self-detection:** Dev ices decide which gets the bus amo ng themselves.
  - **Distributed using collision-detection:** Any device can try to use the bus. I f its data collides with the data of a nother device, it tries again.

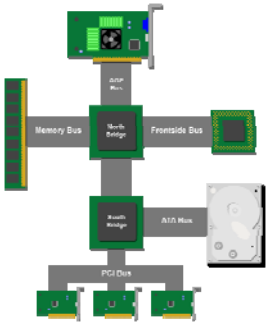
Traditional (ISA) (with cache)



High Performance Bus



Inside of Modern Computer



Clock

- Every computer contains at least one clock that synchro nizes the activities of its components.
- A fixed number of clock cycles are required to carry out e ach data movement or computational operation.
- The clock frequency, measured in megahertz or gigahert z, determines the speed with which all operations are car ried out.
- Clock cycle time is the reciprocal of clock frequency.
  - An 800 MHz clock has a cycle time of 1.25 ns.

Clock vs. Performance

- Clock speed should not be confused with CPU performa nce.
- The CPU time required to run a program is given by the general performance equation:

$$\text{CPU Time} = \frac{\text{seconds}}{\text{program}} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{avg. cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{cycle}}$$

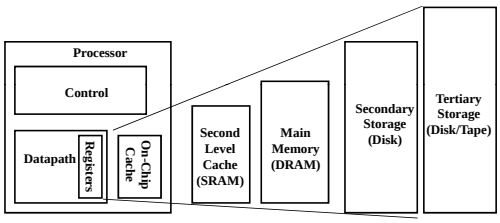
  - We see that we can improve CPU throughput when we reduce th e number of instructions in a program, reduce the number of cycl es per instruction, or reduce the number of nanoseconds per clo ck cycle.

Input/Output Systems

- A computer communicates with the outside world through its input/o utput (I/O) subsystem.
- I/O devices connect to the CPU through various interfaces.
- I/O can be memory-mapped-- where the I/O device behaves like mai n memory from the CPU's point of view.
- Or I/O can be instruction-based, where the CPU has a specialized I/ O instruction set.

A Modern Memory Hierarchy

- By taking advantage of the principle of locality:
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.



Speed (ns):	1s	10s	100s	10,000,000s (10s ms)	10,000,000,000s (10s sec)	
Size (bytes):	100s	Ks	Ms	Gs	Ts	

Main Memory Organization

- Computer memory consists of a linear array of addressa ble storage cells that are similar to registers.
- Memory can be byte-addressable, or word-addressable, where a word typically consists of two or more bytes.
- Memory is constructed of RAM chips, often referred to in terms of length × width.
- If the memory word size of the machine is 16 bits, then a 4M × 16 RAM chip gives us 4 megabytes of 16-bit memo ry locations.

Memory Organization

- How does the computer access a memory location corre sponds to a particular address?
- We observe that 4M can be expressed as $2^2 \times 2^{20} = 2^{22}$ words.
- The memory locations for this memory are numbered 0 t hrough $2^{22}-1$.
- Thus, the memory bus of this system requires at least 22 address lines.
  - The address lines "count" from 0 to $2^{22} - 1$ in binary. Each line is either "on" or "off" indicating the location of the desired memory e lement.

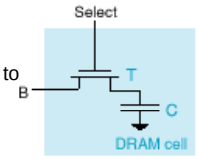
Memory Organization

- Physical memory usually consists of more than one RAM chip.
- Access is more efficient when memory is organized into banks of chips with the addresses interleaved across the chips
- With low-order interleaving, the low order bits of the address specify which memory bank contains the address of interest.
- Accordingly, in high-order interleaving, the high order address bits specify the memory bank.

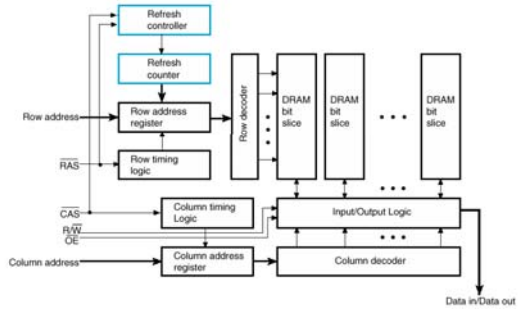
Module 0	Module 1	Module 2	Module 3	Module 4	Module 5	Module 6	Module 7
0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31

Dynamic RAM

- Capacitor can hold charge
- Transistor acts as gate
- No charge is a 0
- Can close switch & add charge to store a 1
- Then open switch (disconnect)
- Can read by closing switch
- Destructive Read
  - When cell read, charge removed
  - Must be restored after a read
- Refresh
  - Also, there's steady leakage
  - Charge must be restored periodically



DRAM Logical Diagram



Interrupt

- The normal execution of a program is altered when an event of higher-priority occurs. The CPU is alerted to such an event through an interrupt.
- Interrupts can be triggered by I/O requests, arithmetic errors (such as division by zero), or when an invalid instruction is encountered.
- Each interrupt is associated with a procedure that directs the actions of the CPU when an interrupt occurs.
  - Nonmaskable interrupts are high-priority interrupts that cannot be ignored.