Computer Organization 1

Types of Computers

1. Personal Computers (PCs)
    - Intended for a single user at a stationary location
    - Notebooks and workstations
    - Emphasize good performance to single users at low cost
2. Servers
    - Accessed by other computers to provide computation and/or data
    - Typically only accessed via a network
    - Greater computing, storage, and I/O capacity
    - Emphasis on performing well under large workloads with enhanced dependability
3. Embedded Computers
    - **Most Prevalent type of computer/computer class**
    - Computers contained in other devices
    - Usually a small number of predetermined applications
    - Emphasis on cost and low power
4. Personal Mobile Device
    - Battery-powered wireless devices with multimedia user interfaces
    - Smart phones and tablets
    - Reliance on touch screens
    - Emphasis on cost and energy efficiency
5. Large Cluster/Warehouse-Scale-Computers (WSCs)
    - Large collections of servers connected by a network to act as a single powerful computer
    - Scalability and availability handled through the network

Eight Great Architecture Ideas

1. Design for Moore's Law
2. Abstraction
3. Make the common case fast
4. Parallelism
5. Pipelining
6. Prediction
7. Hierachy
8. Improve dependability via redundancy

Steps for executing a program

1. Input device loads the machine code from the executable

2. The machine code is stored in memory
3. Processor fetches an instruction
4. Control decodes the instruction
5. Datapath executes the instruction
6. If application does not complete, then go to step 3

Remember: QTSpim is **not** a compiler, it is an **assembler.**

Formulas to remember:

$$\text{Dies per Wafer} \approx \frac{\text{Wafer Area}}{\text{Die Area}}$$

$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area})(\frac{\text{Die Area}}{2}))^2}$$

$$\text{Cost per Die} = \frac{\text{Wafer Cost}}{(\text{Die per Wafer}) * \text{yield}}$$

When comparing performance between Computer_x and Computer_y :

$$\text{Performance} = \frac{1}{\text{Execution Time}}$$

$$\text{Performance}_x > \text{Performance}_y$$

$$\frac{1}{\text{Execution Time}_x} > \frac{1}{\text{Execution Time}_y}$$

$$\text{Execution Time}_y > \text{Execution Time}_x$$

Finding CPU Time:

$$\text{CPU Time} = \text{CPU Clock Cycles} * \text{CPU Clock Cycle Time} = \frac{\text{CPU Clock Cycles}}{\text{CPU clock rate}}$$

$$\text{CPI} = \frac{\text{CPU Clock Cycles}}{\text{Instruction Count}}$$

$$\text{CPU Time} = \text{Instruction Count} * \text{CPI} * \text{CPU Clock Cycles}$$

Terms to know:

1. Latency (Response, or execution time): The time between the start and completion of an event or task.
2. Bandwidth/Throughput: The total amount of work done in a given period of time.
3. Clock cycles Per Instruction (CPI): Average number of clock cycles per instruction for a program or process
4. Amdahl's Law: The performance improvement gained from using an enhanced component is limited by the portion improved.