**Personal computer (PC)**: A computer designed for use by an individual, usually incorporating a graphics display, a keyboard, and a mouse.

**Server**: A computer used for running larger programs for multiple users, often simultaneously, and typically accessed only via a network.

**Supercomputer**: A class of computers with the highest performance and cost; they are configured as servers and typically cost tens to hundreds of millions of dollars.

**Embedded computer**: A computer inside another device used for running one predetermined application or collection of software.

**Personal mobile devices (PMDs)**: Small wireless devices used to connect to the Internet; they rely on batteries for power, and software is installed by downloading apps. Conventional examples are smart phones and tablets.

**Cloud Computing**: Refers to large collections of servers that provide services over the Internet; some providers rent dynamically varying numbers of servers as a utility.

**Software as a Service (SaaS)**: Delivers software and data as a service over the Internet, usually via a thin program such as a browser that runs on local client devices, instead of binary code that must be installed, and runs wholly on that device. Examples include web search and social networking.

**Multicore Microprocessor**: A microprocessor containing multiple processors (“cores”) in a single integrated circuit.

**Acronym**: A word constructed by taking the initial letters of a string of words. For example: RAM is an acronym for Random Access Memory, and CPU is an acronym for Central Processing Unit.

Eight Great Ideas in Computer Architecture

1. **Design for Moore’s Law**

It states that integrated circuit resources double every 18–24 months. Therefore, computer architects must anticipate where the technology will be when the design finishes rather than design for where it starts.

1. **Use Abstraction to Simplify Design**

Lower-level details are hidden to offer a simpler model at higher levels.

1. **Make the Common Case Fast**

Making the common case fast will tend to enhance performance better than optimizing the rare case.

1. **Performance via Parallelism**

Computer architects have offered designs that get more performance by performing operations in parallel.

1. **Performance via Pipelining**

A particular pattern of parallelism is so prevalent in computer architecture that it merits its own name: pipelining. The pipeline icon is a sequence of pipes, with each section representing one stage of the pipeline.

1. **Performance via Prediction**

In some cases, it can be faster on average to guess and start working rather than wait until you know for sure, assuming that the mechanism to recover from a misprediction is not too expensive and the prediction is relatively accurate.

1. **Hierarchy of Memories**

The fastest, smallest, and most expensive memory per bit is placed at the top of the hierarchy and the slowest, largest, and cheapest per bit at the bottom.

1. **Dependability via Redundancy**

Since any physical device can fail, systems are made dependable by including redundant components that can take over when a failure occurs and to help detect failures.

**Systems software**: Software that provides services that are commonly useful, including operating systems, compilers, loaders, and assemblers.

**Operating system**: Supervising program that manages the resources of a computer for the benefit of the programs that run on that computer.

**Compiler**: A program that translates high-level language statements into assembly language statements.

**Binary Digit**: Also called a bit. One of the two numbers in base 2 (0 or 1) that are the components of information.

**Instruction**: A command that computer hardware understands and obeys.

**Assembler**: A program that translates a symbolic version of instructions into the binary version.

**Assembly Language**: A symbolic representation of machine instructions.

**Machine Language**: A binary representation of machine instructions.

**High-Level Programming Language**: A portable language such as C, C++, Java, or Visual Basic that is composed of words and algebraic notation that can be translated by a compiler into assembly language.

**Input Device**: A mechanism through which the computer is fed information, such as a keyboard.

**Output Device**: A mechanism that conveys the result of a computation to a user, such as a display, or to another computer.

**Liquid Crystal Display**: A display technology using a thin layer of liquid polymers that can be used to transmit or block light according to whether a charge is applied.

**Active-Matrix Display**: A liquid crystal display using a transistor to control the transmission of light at each individual pixel.

**Pixel**: The smallest individual picture element. Screens are composed of hundreds of thousands to millions of pixels, organized in a matrix.

**Integrated Circuit**: Also called a chip. A device combining dozens to millions of transistors.

**Central Processor Unit (CPU)**: Also called processor. The active part of the computer, which contains the data-path and control and which adds numbers, tests numbers, signals I/O devices to activate, and so on.

**Datapath**: The component of the processor that performs arithmetic operations.

**Control**: The component of the processor that commands the data-path, memory, and I/O devices according to the instructions of the program.

**Memory**: The storage area in which programs are kept when they are running and that contains the data needed by the running programs.

**Dynamic-Random-Access-Memory (DRAM)**: Memory built as an integrated circuit; it provides random access to any location.

**Cache Memory**: A small, fast memory that acts as a buffer for a slower, larger memory.

**Static Random Access Memory (SRAM)**: Memory built as an integrated circuit, but faster and less dense than DRAM.

**Instruction Set Architecture**: Also called architecture. An abstract interface between the hardware and the lowest-level software that encompasses all the information necessary to write a machine language program that will run correctly, including instructions, registers, memory access, I/O, and so on.

**Application Binary Interface (ABI)**: The user portion of the instruction set plus the operating system interfaces used by application programmers. It defines a standard for binary portability across computers.

**Implementation**: Hardware that obeys the architecture abstraction.

**Volatile Memory**: Storage, such as DRAM, that retains data only if it is receiving power.

**Non-Volatile Memory**: A form of memory that retains data even in the absence of a power source and that is used to store programs between runs. A DVD disk is non-volatile.

**Main Memory**: Also called primary memory. Memory used to hold programs while they are running; typically consists of DRAM in today’s computers.

**Secondary Memory**: Non-volatile memory used to store programs and data between runs; typically consists of flash memory in PMDs and magnetic disks in servers.

**Magnetic Disk**: Also called hard disk. A form of non-volatile secondary memory composed of rotating platters coated with a magnetic recording material.

**Flash Memory**: A non-volatile semi-conductor memory. It is cheaper and slower than DRAM but more expensive per bit and faster than magnetic disks.

Advantages of Networked Computers:

1. *Communication*: Information is exchanged between computers at high speeds.
2. *Resource Sharing*: Rather than each computer having its own I/O devices, computers on the network can share I/O devices.
3. *Non*­*-Local Access*: By connecting computers over long distances, users need not be near the computer they are using.

**Local Area Network (LAN)**: A network designed to carry data within a geographically confined area, typically within a single building.

**Wide Area Network (WAN)**: A network extended over hundreds of kilo-meters that can span a continent.

**Transistor**: An on/off switch controlled by an electric signal.

**Very large-scale integrated (VLSI) circuit**: A device containing hundreds of thousands to millions of transistors.

**Silicon**: A natural element (found in sand) that is a semiconductor.

**Semiconductor**: A substance that does not conduct electricity well.

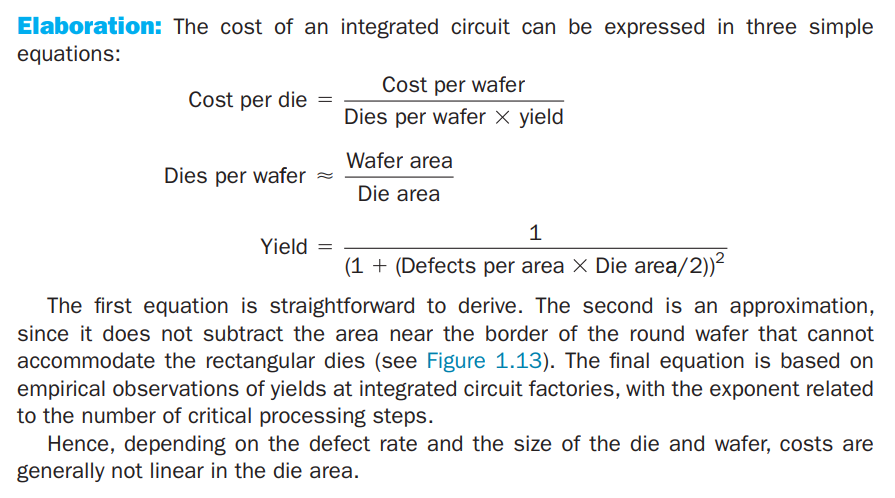
**Silicon Crystal Ingot**: A rod composed of a silicon crystal that is between 8 and 12 inches in diameter and about 12 to 24 inches long.

**Wafer**: A slice from a silicon ingot no more than 0.1 inches thick, used to create chips.

**Defect**: A microscopic flaw in a wafer or in patterning steps that can result in the failure of the die containing that defect.

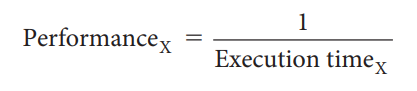
**Die**: The individual rectangular sections that are cut from a wafer, more informally known as chips.

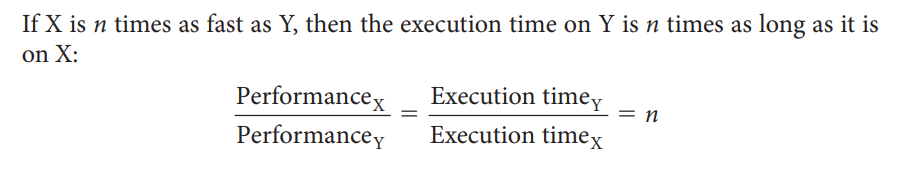
**Yield**: The percentage of good dies from the total number of dies on the wafer.



**Response Time**: Also called execution time. The total time required for the computer to complete a task, including disk accesses, memory accesses, I/O activities, operating system overhead, CPU execution time, and so on.

**Throughput**: Also called bandwidth. Another measure of performance, it is the number of tasks completed per unit time.





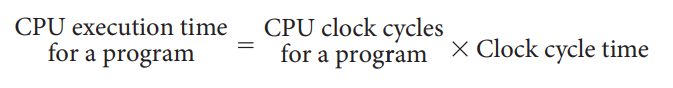
**CPU Execution Time**: Also called CPU time. The actual time the CPU spends computing for a specific task.

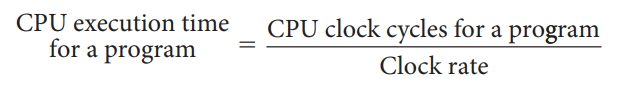
**User CPU Time**: The CPU time spent in a program itself.

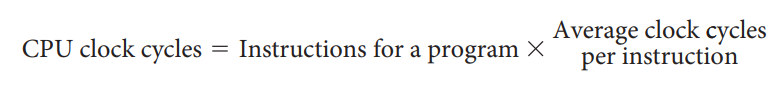
**System CPU Time**: The CPU time spent in the operating system performing tasks on behalf of the program.

**Clock Cycle**: Also called tick, clock tick, clock period, clock, or cycle. The time for one clock period, usually of the processor clock, which runs at a constant rate.

**Clock Period**: The length of each clock cycle.







**Clock Cycles Per Instruction (CPI)**: Average number of clock cycles per instruction for a program or program fragment.

**Instruction Count**: The number of instructions executed by the program.

**Instruction Mix**: A measure of the dynamic frequency of instructions across one or many programs.