COMPUTER ORGANIZATION & INSTRUCTIONS

1.1 INTRODUCTION

Computer architecture acts as the interface between the hardware and the lowest level software. Computer architecture refers to:

- Attributes of a system visible to programmers like data type of variables.
- Attributes that have a direct impact on the execution of programs like clock cycle.

Computer Architecture is defined as study of the structure, behavior, and design of computers.

Computer Organization: It refers to the operational units and their interconnections that realize the architectural specifications. It describes the function of and design of the various units of digital computer that store and process information. The attributes in computer organization refers to:

- Control signals
- Computer/peripheral interface
- Memory technology

Computer hardware: Consists of electronic circuits, displays, magnetic and optical storage media, electromechanical equipment and communication facilities.

Computer Architecture: It is concerned with the structure and behavior of the computer.

It includes the information formats, the instruction set and techniques for addressing memory. The attributes in computer architecture refers to the:

- Instruction set
- Data representation
- I/O mechanisms
- Addressing techniques

The basic distinction between architecture and organization is: the attributes of the former are visible to programmers whereas the attributes of the later describes how features are implemented in the system.

BASICS OF A COMPUTER SYSTEM

The modern day computer system's functional unit is given by Von Neumann Architecture.

Fig 1 Von Neumann Architecture

Input Unit

Computers accepts the coded information through input unit. Computer must receive both data and program statements to function properly and must be able to solve problems. The method of feeding data and programs to a computer is accomplished by an input device. Input devices read data from a source, such as magnetic disks, and translate that data into electronic impulses for transfer into the CPU. Whenever a key is pressed, the corresponding letter or digit is automatically translated into its corresponding binary code and transmitted over a cable to either the memory or the processor.

Central Processing Unit (CPU)

The CPU processes data transferred to it from one of the various input devices. It then transfers either an intermediate or final result of the CPU to one or more output devices. A central control section and work areas are required to perform calculations or manipulate data. The CPU is the computing center of the system. It consists of a control section, an arithmetic- logic section, and an internal storage section (memory unit). Each section within the CPU serves a specific function and has a particular relationship with the other sections within the CPU.

Memory Unit

It stores the programs and data. Memory unit is broadly classified into two types: Primary memory and Secondary memory.

1. **Primary Memory:**

It is a fast memory that operates at electronic speeds. Programs must be stored in the memory while they are being executed. The memory contains large no of semiconductor storage cells. Each cell carries 1 bit of information. The cells

are processed in a group of fixed size called **Words**. To provide easy access to any word in a memory, a distinct address is associated with each word location. Addresses are numbers that identify successive locations. The number of bits in each word is called the **word length.** The word length ranges from 16 to 64 bits. There are 3 types of primary memory:

I. **RAM:** Memory in which any location can be reached in short and fixed amount of time after specifying its address is called RAM. Time required to
access 1 word is called Memory Access Time.

II. **Cache Memory:** The small, fast, RAM units are called Cache. They are tightly coupled with processor to achieve high performance.

III. **Main Memory:** The largest and the slowest unit is the main memory.

Arithmetic & Logic Unit

Most computer operations are executed in ALU. The arithmetic-logic section performs arithmetic operations, such as addition, subtraction, multiplication, and division. Through internal logic capability, it tests various conditions encountered during processing and takes action based on the result. Data may be transferred back and forth between these two sections several times before processing is completed. Access time to registers is faster than access time to the fastest cache unit in memory.

Output Unit

Its function is to send the processed results to the outside world.

Control Unit

The operations of Input unit, output unit, ALU are co-ordinate by the control unit. The control unit is the Nerve centre that sends control signals to other units and senses their states. The control section directs the flow of traffic (operations) and data. It also maintains order within the computer. The control section selects one program statement at a time from the program storage area, interprets the statement, and sends the appropriate electronic impulses to the arithmetic-logic and storage sections so they can carry out the instructions. The control section does not perform actual processing operations on the data.

The control section instructs the input device on when to start and stop transferring data to the input storage area. It also tells the output device when to start and stop receiving data from the output storage area. Data transfers between the

processor and the memory are controlled by the control unit through **timing signals**. Information stored in the memory is fetched, under program control into an arithmetic and logic unit, where it is processed.

Evolution of Computers

- The word "computer" is an old word that has changed its meaning several times in the last few centuries.
- Today, the word computer refers to computing devices, whether or not they are electronic, programmable, or capable of storing and retrieving data.

The Mechanical Era (1623-1945)

- Wilhelm Schick hard, Blaise Pascal, and Gottfried Leibnitz were among mathematicians who designed and implemented calculators that were capable of addition, subtraction, multiplication, and division during the seventeenth century.
- The first multi-purpose or programmable computing device was probably **Charles**

  Babbage's Difference Engine, which was begun in 1823 but never completed.
- In 1842, Babbage designed a more ambitious machine, called the **Analytical Engine** but unfortunately it also was only partially completed.
- Babbage, together with Ada Lovelace recognized several important programming techniques, including conditional branches, iterative loops and index variables.
- Babbage designed the machine which is the first to be used in computational science.
- In 1933, George Scheutz and his son, Edvard began work on a smaller version of the difference engine and by 1853 they had constructed a machine that could process 15- digit numbers and calculate fourth-order differences.
- The US Census Bureau was one of the first organizations to use the mechanical computers which used punch-card equipment designed by Herman Hollerith to tabulate data for the 1890 census.
- In 1911 (ollerith's company merged with a competitor to found the corporation which in 1924 became **International Business Machines** (IBM).

First Generation Electronic Computers (1937-1953)

- These devices used electronic switches, in the form of **vacuum tubes**, instead of electromechanical relays.
- The earliest attempt to build an electronic computer was by J. V. Atanasoff, a professor of physics and mathematics at Iowa State in 1937.
- Atanasoff set out to build a machine that would help his graduate students solve systems of partial differential equations.
- By 1941 he and graduate student Clifford Berry had succeeded in building a machine that could solve 29 simultaneous equations with 29 unknowns.
- However, the machine was not programmable, and was more of an electronic calculator.
- A second early electronic machine was Colossus, designed by Alan Turing for the British military in 1943.
- The first general purpose programmable electronic computer was the Electronic Numerical Integrator and Computer **(ENIAC),** built by J. Presper Eckert and John V. Mauchly at the University of Pennsylvania.
- ENIAC was controlled by a set of external switches and dials; to change the program required physically altering the settings on these controls.
- Research work began in 1943, funded by the Army Ordinance Department, which needed a way to compute ballistics during World War II.
- The machine was completed in 1945 and it was used extensively for calculations during the design of the hydrogen bomb.
- Eckert, Mauchly, and John von Neumann, a consultant to the ENIAC project, began work on a new machine before ENIAC was finished.
- The next development was **EDVAC**- Electronic Discrete Variable Computer.
- The main contribution of EDVAC, their new project, was the notion of a **stored program.**
- EDVAC was able to run orders of magnitude faster than ENIAC and by storing instructions in the same medium as data, designers could concentrate on improving the internal structure of the machine without worrying about matching it to the speed of an external control.
- Eckert and Mauchly later designed the first commercially successful computer, the **UNIVAC (**Universal Automatic Computer**)**; in 1952.
- Software technology during this period was very primitive.

- The instructions were written in machine language that could be executed directly.

Second Generation (1954-1962)

- The second generation witnessed several important developments at all levels of computer system design, ranging from the technology used to build the basic circuits to the programming languages used to write scientific applications.
- Electronic switches in this era were based on **discrete diode and transistor technology** with a switching time of approximately 0.3 microseconds.
- The first machines to be built with this technology include **TRADIC** at Bell Laboratories in 1954 and TX-ど at M)T┼s Lincoln Laboratory.
- Index registers were designed for controlling loops and floating point units for calculations based on real numbers.
- A number of high level **programming languages** were introduced and these include FORTRAN (1956), ALGOL (1958), and COBOL (1959).
- **Batch processing systems** came to existence.
- Important commercial machines of this era include the IBM 704 and its successors, the 709 and 7094.
- In the 1950s the first two supercomputers were designed specifically for numeric processing in scientific applications.
- Multi programmed computers that serve many users concurrently came to existence.
- This is otherwise known as **time-sharing systems**.

Third Generation (1963-1972)

- Technology changes in this generation include the use of **integrated circuits**, or ICs.
- This generation led to the introduction of **semiconductor memories**, **micro programming** as a technique for efficiently designing complex processors and the introduction of **operating systems** and time-sharing.
- The first ICs were based on small-scale integration (SSI) circuits, which had around 10 devices per circuit ゅor ┬chip┬ょ, and evolved to the use of medium-scale integrated (MSI) circuits, which had up to 100 devices per chip.
- Multilayered printed circuits were developed and core memory was replaced by faster, solid state memories.

- In 1964, Seymour Cray developed the CDC 6600, which was the first architecture to use functional parallelism.
- By using 10 separate functional units that could operate simultaneously and 32 independent memory banks, the CDC 6600 was able to attain a computation rate of one million floating point operations per second (Mflops).
- Five years later CDC released the 7600, also developed by Seymour Cray.
- The CDC 7600, with its pipelined functional units, is considered to be the first vector processor and was capable of executing at ten Mflops.
- The IBM 360/91, released during the same period, was roughly twice as fast as the CDC 660.
- Early in this third generation, Cambridge University and the University of London cooperated in the development of CPL (Combined Programming Language, 1963).
- CPL was an attempt to capture only the important features of the complicated and sophisticated ALGOL.
- However, like ALGOL, CPL was large with many features that were hard to learn.
- In an attempt at further simplification, Martin Richards of Cambridge developed a subset of CPL called BCPL (Basic Computer Programming Language, 1967).
- In 1970 Ken Thompson of Bell Labs developed yet another simplification of CPL called simply B, in connection with an early implementation of the UNIX operating system.

Fourth Generation (1972-1984)

- **Large scale integration** (LSI - 1000 devices per chip) and **very large scale integration**

  (VLSI - 100,000 devices per chip) were used in the construction of the fourth generation computers.
- Whole processors could now fit onto a single chip, and for simple systems the entire computer (processor, main memory, and I/O controllers) could fit on one chip.
- Gate delays dropped to about 1ns per gate. Core memories were replaced by semiconductor memories.
- Large main memories like CRAY 2 began to replace the older high speed

vector processors, such as the CRAY 1, CRAY X-MP and CYBER.

- In 1972, Dennis Ritchie developed the C language from the design of the CPL and Thompson┼s B.
- Thompson and Ritchie then used C to write a version of UNIX for the DEC PDP-11.
- Other developments in software include very high level languages such as FP (functional programming) and Prolog (programming in logic).
- IBM worked with Microsoft during the 1980s to start what we can really call PC (Personal Computer) life today.
- IBM PC was introduced in October 1981 and it worked with the operating system

  ゆsoftwareよ called┼Microsoft Disk Operating System ゆMS DOSよ な.ど.
- Development of MS DOS began in October 1980 when IBM began searching the market for an operating system for the then proposed IBM PC and major contributors were Bill Gates, Paul Allen and Tim Paterson.
- In 1983, the Microsoft Windows was announced and this has witnessed several improvements and revision over the last twenty years.

Fifth Generation (1984-1990)

- This generation brought about the introduction of machines with hundreds of processors that could all be working on different parts of a single program.
- The scale of integration in semiconductors continued at a great pace and by 1990 it was possible to build chips with a million components - and semiconductor memories became standard on all computers.
- Computer networks and single-user workstations also became popular. Parallel processing started in this generation.
- The Sequent Balance 8000 connected up to 20 processors to a single shared memory module though each processor had its own local cache.
- The machine was designed to compete with the DEC VAX-780 as a general purpose UNIX system, with each processor working on a different user's job
- However Sequent provided a library of subroutines that would allow programmers to write programs that would use more than one processor,

and the machine was widely used to explore parallel algorithms and programming techniques.

- The **Intel iPSC-1**, also known as ┬the hypercube┬ connected each processor to its own
  memory and used a network interface to connect processors.
- This distributed memory architecture meant memory was no longer a problem and large systems with more processors (as many as 128) could be built.
- Also introduced was a machine, known as a **data-parallel** or SIMD where there were
  several thousand very simple processors which work under the direction of a single control unit.
- Both wide area network (WAN) and local area network (LAN) technology developed rapidly.

Sixth Generation (1990 -)

- Most of the developments in computer systems since 1990 have not been fundamental changes but have been gradual improvements over established systems.
- This generation brought about gains in parallel computing in both the hardware and in improved understanding of how to develop algorithms to exploit parallel architectures.
- Workstation technology continued to improve, with processor designs now using a combination of RISC, pipelining, and parallel processing.
- Wide area networks, network bandwidth and speed of operation and networking capabilities have kept developing tremendously.
- Personal computers (PCs) now operate with Gigabit per second processors, multi- Gigabyte disks, hundreds of Mbytes of RAM, color printers, high-resolution graphic monitors, stereo sound cards and graphical user interfaces.
- Thousands of software (operating systems and application software) are existing today and Microsoft Inc. has been a major contributor. Microsoft is said to be one of the biggest companies ever, and its chairman – Bill Gates has been rated as the richest man for several years.

- Finally, this generation has brought about micro controller technology. Micro controllers are ┼embedded┼ inside some other devices so that they can control the features or actions of the product.
- They work as small computers inside devices and now serve as essential components in most machines.

Great Ideas in Computer Architecture

The ideas that marked tremendous improvement in the field of computer architecture are briefly discussed here.

Moore's Law

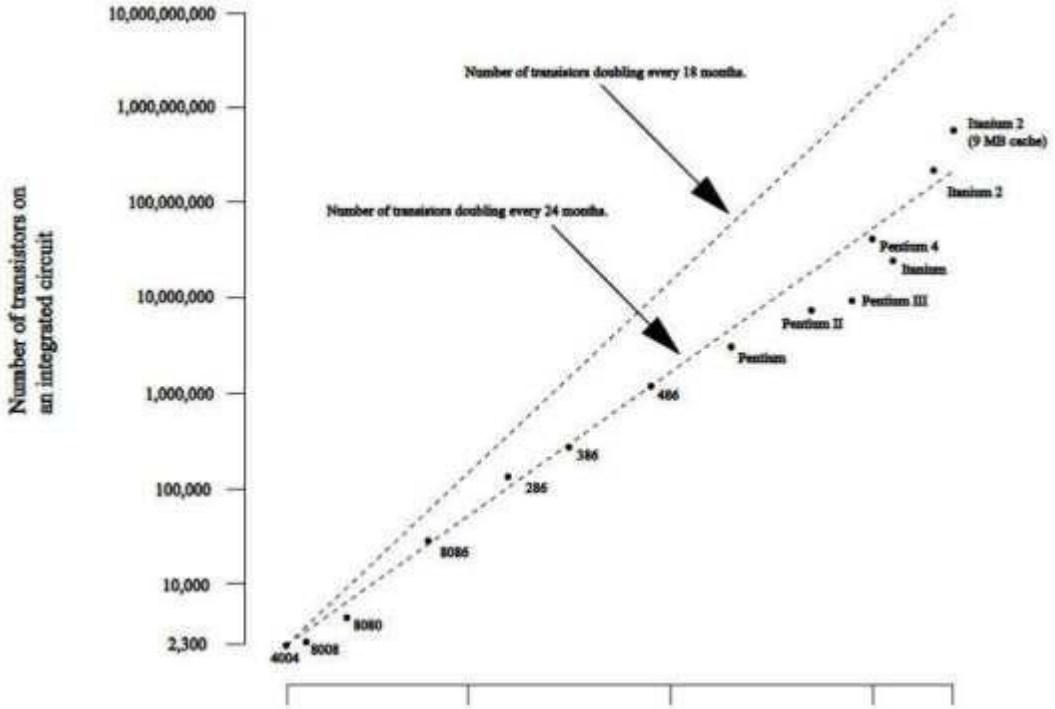

Fig 2 Illustration of Moore's Law

Moore's law states that the numbers of transistors will double every 18 months.

It is an observation that the number of transistors in a dense integrated circuit doubles about every two years. It is an observation and projection of a historical trend and not a physical or natural law.

2. Abstract Design

It is a major productivity technique for hardware and software. Abstractions are used to represent the design at different levels of representation. The detailed lower-level design details from the higher levels.

3. Performance through parallelism

Parallelism executes programs faster by performing several computations at the same time. This requires hardware with multiple processing units. The overall performance of the system is significantly increased by performing operations in parallel.

4. Performance through Pipelining

Pipelining increases the CPU instruction throughput. **Throughput** is a performance metric which is the number of instructions completed per unit of time. But it does not reduce the execution time of an individual instruction. It increases the execution time of each instruction due to overhead in the pipeline control. The increase in instruction throughput means that a program runs faster and has lower total execution time.

5. Make the Common Case Fast

Making the common case fast will tend to enhance performance better than optimizing the rare case. Ironically, the common case is often simpler than the rare case and hence is often easier to enhance. In making a design trade-off, favor the frequent case over the infrequent case.

Amdahl's Law can be used to quantify this principle. This also applies when determining how to spend resources, since the impact on making some occurrence faster is higher if the occurrence is frequent. This will:

- Helps performance
- Is simpler and can be done faster

6. Performance via prediction

The computer can perform better (on average) by making rational guesses on the decisions. Instead of wasting clock cycles for certain results, the computers can remarkably improve the performance

7. Hierarchy of memories

Programmers want memory to be fast, large, and cheap. The memory speed is a primary factor in determining the performance of the system. The memory capacity limits the size of problems that can be solved.

Architects have found that hierarchy of memories will be a solution for all these issues. The fastest, smallest, and most expensive memory per bit is placed the top of the hierarchy and the slowest, largest, and cheapest per bit is at the bottom. **Caches** give the illusion that main memory is nearly as fast as the top of the hierarchy and nearly as big and cheap as the bottom of the hierarchy.

8. Dependability via Redundancy

Computers need to be fast and dependable. Since any physical device can fail, we make systems dependable by including redundant components that can take over when a failure occurs and help detect failures. Restoring the state of the system is done by redundancy.

Technologies

Up until the early なひばど十s computers used magnetic core memory, which was slow, cumbersome, and expensive and thus appeared in limited quantities. The situation improved with the introduction of transistor-based dynamic random-access memory (DRAM, invented at IBM in 1966) and static random-access memory (SRAM). A transistor is simply an on/off switch controlled by electricity. The integrated circuit (IC) combined dozens to hundreds of transistors into a simple chip. Very large-scale integrated (VLSI) circuit is a device containing hundreds of thousands to millions of transistors.

Manufacturing of IC:

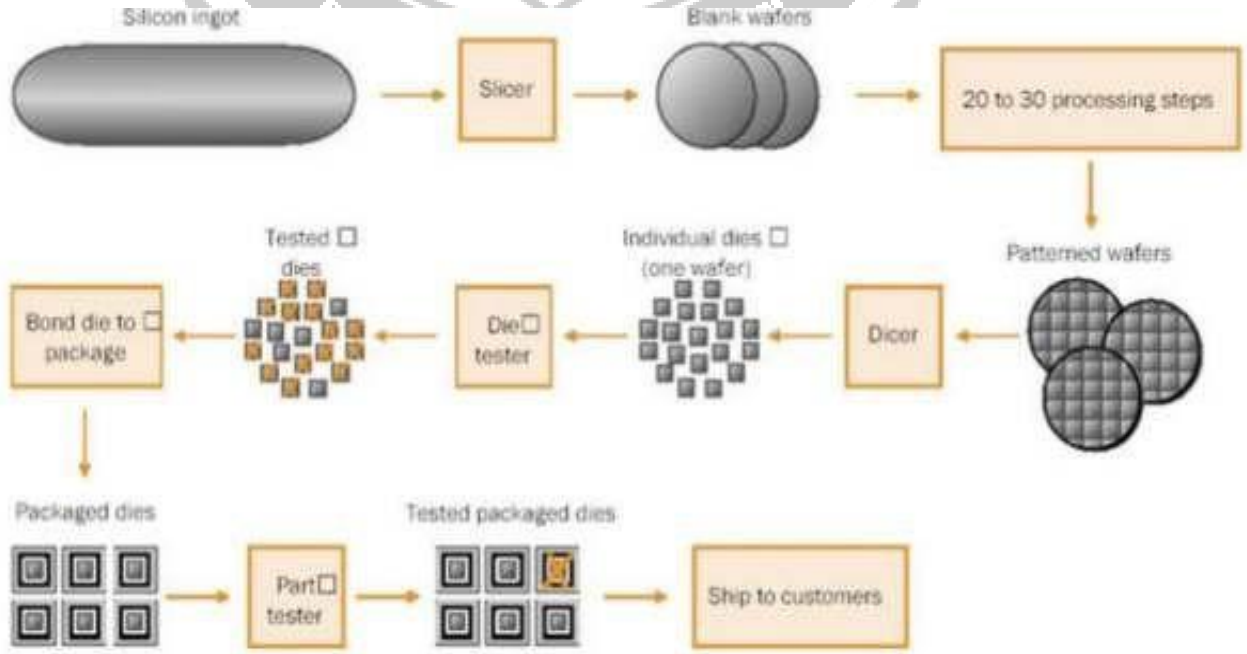

Fig 3 Chip manufacturing process

Integrated circuits are chips manufactured on silicon wafers. Transistors are placed on wafers through a chemical etching process. Each wafer is cut into chips which are packed individually.

After being sliced from the silicon ingot, blank wafers are put through 20 to 40 steps to create patterned wafers. These patterned wafers are then tested with a wafer tester, and a map of the good parts is made. Then, the wafers are diced into dies. The good dies are then bonded into packages and tested one more time before shipping the packaged parts to customers.

Cost of an IC is found from:

- Cost per die= (cost per wafer) / ((dies per wafer)*yield) Yield refers the fraction of dies that pass testing.
- Dies / wafer= wafer area / die area
- Yield=1 / (1 + (defects per area * die area)/2)^2

Programmable Logic Device (PLD)

A programmable logic device (PLD) is an electronic component used to build reconfigurable digital circuits. Unlike a logic gate, which has a fixed function, a PLD has an undefined function at the time of manufacture. Before the PLD can be used in a circuit it must be programmed, that is, reconfigured.

The major limitations of PLD:

- Consume space due to large number of switches for programmability
- Low speed due to the presence of many switches.

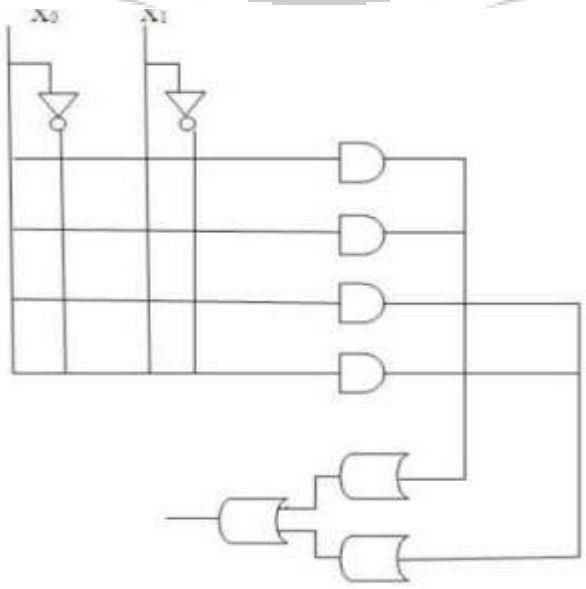

Fig 4 Programmable Logic Device

Custom chips

An Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use. Application-Specific Standard Products (ASSPs) are intermediate between ASICs and industry standard integrated circuits.

Performance

Elapsed time and throughput are two different ways of measuring speed.

- **Elapsed time** or wall-clock time or response time is the total time to complete a task, including disk accesses, memory accesses, input/output (I/O) activities, operating system overhead. It is the better measure for processor speed because it is less dependent on other system components.
- **CPU execution time** is the actual time the CPU spends computing for a specific task.
- The **User CPU time** is the CPU time spent in a program itself. **System CPU time** is the CPU time spent in the operating system performing tasks on behalf of the program.
- **CPU execution time** is the actual time the CPU spends computing for a specific task.
- The **User CPU time** is the CPU time spent in a program itself. **System CPU time** is the CPU time spent in the operating system performing tasks on behalf of the program.
- The **CPU Performance** equation (CPU Time) is the product of number of instructions executed, Average CPI of the program and CPU clock cycle.

$$CPU\ TIME = \frac{Seconds}{Program} X \frac{Instruction}{Program} X \frac{Cycles}{Instruction} X \frac{Seconds}{Cycles}$$

- Performance is inversely proportional to execution time. Performance ratios are inverted from time ratios.

$$Performance\ improve\ mentation = \frac{Performance\ after\ change}{Performance\ before\ change} x \frac{\text{Execution } time\ before\ change}{\text{Execution } time\ after\ change}$$

- Clock cycle is the time for one clock period, usually of the processor clock, which runs at a constant rate.
- Clock period is the length of each clock cycle.

- The CPU clock rate depends on CPU organization and hardware implementation.

$$Clock\ Rate = \frac{1}{Clock\ Cycle}$$

Cycles per Instruction (CPI) is count clock cycles taken by an instruction to complete its execution.

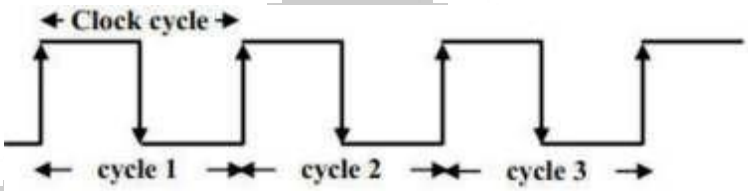

Fig 5 Count Clock Cycle

$$Instructions\ per\ Cycle\ (IPC) = \frac{1}{Cycles\ per\ Instruction}$$

- Performance is improved by reducing number of clock cycles, increasing clock rate and hardware designer must often trade off clock rate against cycle count.
- Workload is a set of programs run on a computer that is either the actual collection of applications run by a user or is constructed from real programs to approximate such a mix. A typical workload specifies both the programs as well as the relative frequencies.
- To evaluate two computer systems, a user would simply compare the execution time of the workload on the two computers.
- Alternatively, set of benchmarks containing several typical engineering or scientific applications can be used. A CPU benchmark (CPU benchmarking) is a series of tests designed to measure the performance of a computer or device CPU. A set of standards, or baseline measurements are used to compare the performance of different systems, using the same methods and circumstances.

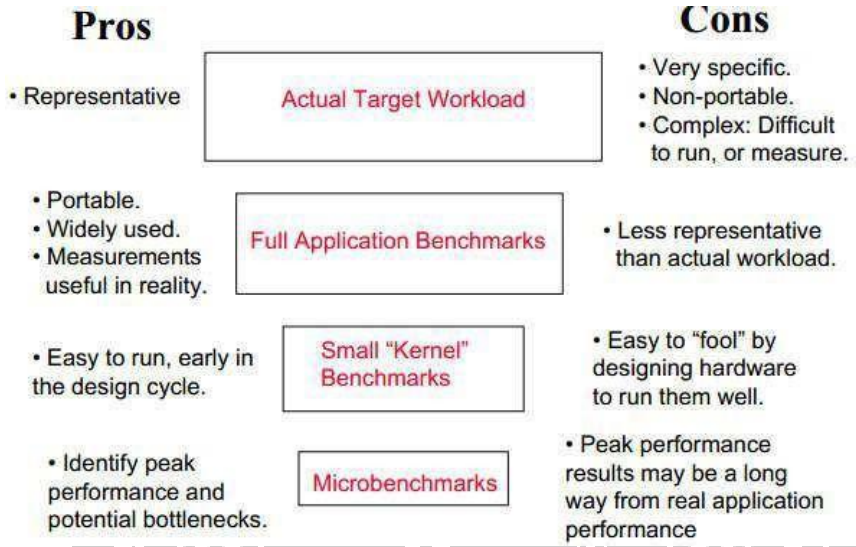

Fig 6 Types of Benchmark Programs

- The use of benchmarks whose performance depends on very small code segments encourages optimizations in either the architecture or compiler that target these segments.
- The arithmetic mean is proportional to execution time, assuming that the programs in the workload are each run an equal number of times.
- **Weighted arithmetic mean** is an average of the execution time of a workload with weighting factors designed to reflect the presence of the programs in a workload; computed as the sum of the products of weight.