# Chapter 1

## Introduction To Computer Systems

# 1.1 Historical Background

- The first program-controlled computer ever built was the Z1 (1938).
- This was followed in 1939 by the Z2 as the first operational program-controlled computer with fixed-point arithmetic.
- The Z3, has been reported in Germany in 1941.
- The Electronic Numerical Integrator and Calculator (ENIAC) machine in 1944 was the first operational general-purpose machine built using vacuum tubes.

# 1.1 Historical Background

- An improved version of the ENIAC was proposed and called the Electronic Discrete Variable Automatic Computer (EDVAC): it was an attempt to improve the way programs are entered and explored the concept of stored-programs.
- In 1946, a stored-program computer, known as the Electronic Delay Storage Automatic Calculator (EDSAC) was built. In 1949, the EDSAC became the world's first full-scale, stored-program, fully operational computer.

# 1.1 Historical Background

- A spin off the EDSAC resulted in a series of machines introduced at Harvard. The series consisted of MARK I, II, III and IV. The later two machines introduced the concept of separate memories for instructions and data. The term *Harvard Architecture* was given to such machines to indicate the use of separate memories.
- The first general-purpose commercial computer, the UNIVersal Automatic Computer (UNIVAC I), was on the market by the middle of 1951.
- In 1952, IBM announced its first computer, the IBM701.

# 1.1 Historical Background

- In 1964 IBM announced a line of products under the name IBM 360 series. The series included a number of models that varied in price and performance.
- Digital Equipment Corporation (DEC) introduced the first *minicomputer*, the PDP-8. It was considered a remarkably low-cost machine.
- Intel introduced the first *microprocessor*, the Intel 4004, in 1971.
- The world witnessed the birth of the first *personal computer (PC)* in 1977 when Apple computer series were first introduced.

# 1.1 Historical Background

- In parallel with small-scale machines, supercomputers were coming into play.
- The first such supercomputer, the CDC 6600, was introduced in 1961 by Control Data Corporation.
- Cray Research Corporation introduced the best cost/performance supercomputer, the Cray-1, in 1976.
- The 1980s and 1990s witnessed the introduction of many commercial parallel computers with multiple processors. They can generally be classified into two main categories: (1) shared memory and (2) distributed memory systems.

# 1.1 Historical Background

- In 1977, the world also witnessed the introduction of the VAX-11/780 by DEC.
- Intel followed suit by introducing the first of the most popular microprocessor, the 80x86 series.
- PCs from Compaq, Apple, IBM, Dell, and many others, soon became pervasive, and changed the face of computing.

# 1.1 Historical Background

- The number of processors in a single machine ranged from several in a shared memory computer to hundreds of thousands in a massively parallel system.
- Examples of parallel computers during this era include Sequent Symmetry, Intel iPSC, nCUBE, Intel Paragon, Thinking Machines (CM-2, CM-5), MsPar (MP), Fujitsu (VPP500), and others.
- Local Area Networks (LAN) of powerful personal computers and workstations began to replace mainframes and minis by 1990.
- These individual desktop computers were soon to be connected into larger complexes of computing by wide area networks (WAN).

# 1.1 Historical Background

- The pervasiveness of the Internet created interest in network computing and more recently in grid computing.
- Grids are geographically distributed platforms of computation.
- They should provide dependable, consistent, pervasive, and inexpensive access to high-end computational facilities.

# 1.1 Historical Background

| Feature | Batch | Time-Sharing | Desktop | Network |
|---|---|---|---|---|
| **Decade** | 1960s | 1970s | 1980s | 1990s |
| **Location** | Computer Room | Terminal Room | Desktop | Mobile |
| **Users** | Experts | Specialists | Individuals | Groups |
| **Data** | Alphanumeric | Text, numbers | Fonts, Graphs | Multimedia |
| **Objective** | Calculate | Access | Present | Communicate |
| **Interface** | Punched Card | Keyboard & CRT | See & Point | Ask & Tell |
| **Operation** | Process | Edit | Layout | Orchestrate |
| **Connectivity** | None | Peripheral Cable | LAN | Internet |
| **Owners** | Corporate Computer Centers | Divisional IS Shops | Departmental End-Users | Everyone |

# 1.2 Architectural Development & Styles

- Computer architects have always been striving to increase the performance of their architectures.
- One philosophy was that by doing more in a single instruction, one can use a smaller number of instructions to perform the same job.
- The immediate consequence of this is the need for a less memory read/write operations and an eventual speedup of operations.

# 1.2 Architectural Development & Styles

- It was also argued that increasing the complexity of instructions and the number of addressing modes have the theoretical advantage of reducing the “semantic gap” between the instructions in a high level language and those in the low level (machine) language.
- Machines following this philosophy have been referred to as *complex instructions set computers (CISCs).* Examples of CISC machines include the Intel *Pentium,* the Motorola *MC*68000, and the *IBM & Macintosh PowerPC* .

# 1.2 Architectural Development & Styles

- A number of studies from the mid 70s and early 80s also identified that in typical programs, more than 80% of the instructions executed are those using assignment statements, conditional branching and procedure calls.
- Simple assignment statements constitute almost 50% of those operations. These findings caused a different philosophy to emerge.
- This philosophy promotes the optimization of architectures by speeding up those operations that are most frequently used while reducing the instruction complexities and the number of addressing modes.

# 1.2 Architectural Development & Styles

- Machines following this philosophy have been referred to as reduced instructions set computers *(RISCs*)*.* Examples of *RISCs* include the *Sun SPARC and MIPS* machines.
- The two philosophies in architecture design have lead to the unresolved controversy which architecture style is "best".
- It should however be mentioned that studies have indicated that RISC architectures would indeed lead to faster execution of programs.

# 1.3 Technological Development

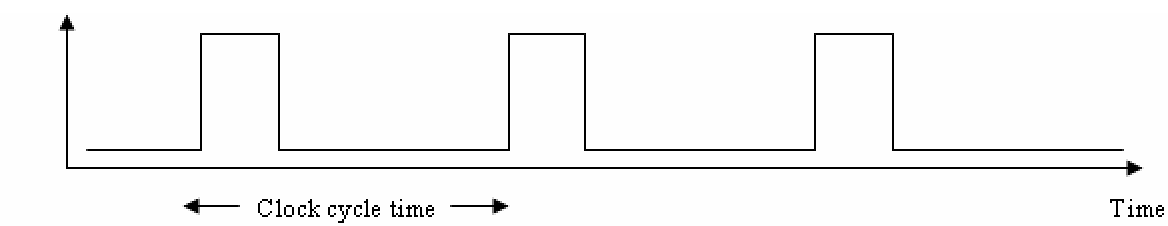
- Computer technology has shown an unprecedented rate of improvement. This includes the development of processors and memories.
- This impressive increase has been made possible by the advances in the fabrication technology of transistors.
- The scale of integration has grown from small scale (SSI) to medium scale (MSI) to large scale (LSI) to very large scale integration (VLSI) and currently to wafer scale integration (WSI).

# 1.3 Technological Development

| Integration | Technology | Typical number of devices | Typical functions |
|---|---|---|---|
| SSI | Bipolar | 10-20 | Gates and Flip-Flops |
| MSI | Bipolar & MOS | 50-100 | Adders & Counters |
| LSI | Bipolar & MOS | 100-10,000 | ROM & RAM |
| VLSI | CMOS (mostly) | 10,000-5,000,000 | Processors |
| WSI | CMOS | >5,000,000 | DSP & Special purposes |

# 1.4 Performance Measures

- There are various facets to the performance of a computer.
- A metric for assessing the performance of a computer helps comparing alternative designs.
- Performance analysis should help answering questions such as how fast can a program be executed using a given computer?
- The clock cycle time is defined as the time between two consecutive rising edges of a periodic clock signal.



# 1.4 Performance Measures

- We denote the number of CPU clock cycles for executing a job to be the cycle count (CC),
- the cycle time by CT, and the clock frequency by $f = 1 / CT$.
- The time taken by the CPU to execute a job can be expressed as: $\text{CPU time} = CC * CT = CC / f$
- It is easier to count the number of instructions executed in a given program as compared to counting the number of CPU clock cycles needed for executing that program.
- Therefore, the average number of clock cycles per instruction (CPI) has been used as an alternate performance measure.

# 1.4 Performance Measures

- The following equation shows how to compute the CPI:

$$\text{CPI} = \frac{\text{CPU clock cycles for the program}}{\text{Instruction count}}$$

$$CPU\ time = Instruction\ count \times CPI \times Clock\ cycle\ time$$

$$\text{CPU time} = \frac{\text{Instruction count} * \text{CPI}}{\text{Clock rate}}$$

- A different performance measure that has been given a lot of attention in recent years is *MIPS* - million instructions-per-second (the rate of instruction execution per unit time) and is defined as follows:

$$MIPS = \frac{\text{instruction count}}{\text{execution time} \times 10^6} = \frac{\text{clock rate}}{\text{CPI} \times 10^6}$$

# 1.4 Performance Measures

- MFLOP - *million floating-point instructions per second,* (rate of floating-point instruction execution per unit time) has also been used as a measure for machines' performance. MFLOPS is defined as follows:

$$\text{MFLOPS} = \frac{\text{Number of floating - point operations in a program}}{\text{Execution time} \ * 10^6}$$

- Speedup is a measure of how a machine performs after some enhancement relative to its original performance. The following relationship formulates Amdahl's law:

$$SU_o = \frac{\text{Performance after enhancement}}{\text{Performance before enhancement}}$$

$$Speedup = \frac{\text{Execution time before enhancement}}{\text{Execution time after enhancement}}$$

# 1.5 Summary

- A brief historical background for the development of computer systems was provided, starting from the first recorded attempt to build a computer, the Z1, in 1938, passing through the CDC 6600 and the Cray supercomputers, and ending up with today's modern high performance machines.
- Then a discussion was provided on the RISC versus CISC architectural styles and their impact on machine performance.

# 1.5 Summary

- This was followed by a brief discussion on the technological development and its impact on computing performance.
- This chapter was concluded with a detailed treatment of the issues involved in assessing the performance of computers.
- Possible ways of evaluating the speedup for given partial or general improvement measurements of a machine were also discussed.