Bahria University
Discovering Knowledge

Computer Architecture and Logic Design (CALD)

Lecture 02

Dr. Sorath Hansrajani

Assistant Professor

Department of Software Engineering

Bahria University Karachi Campus

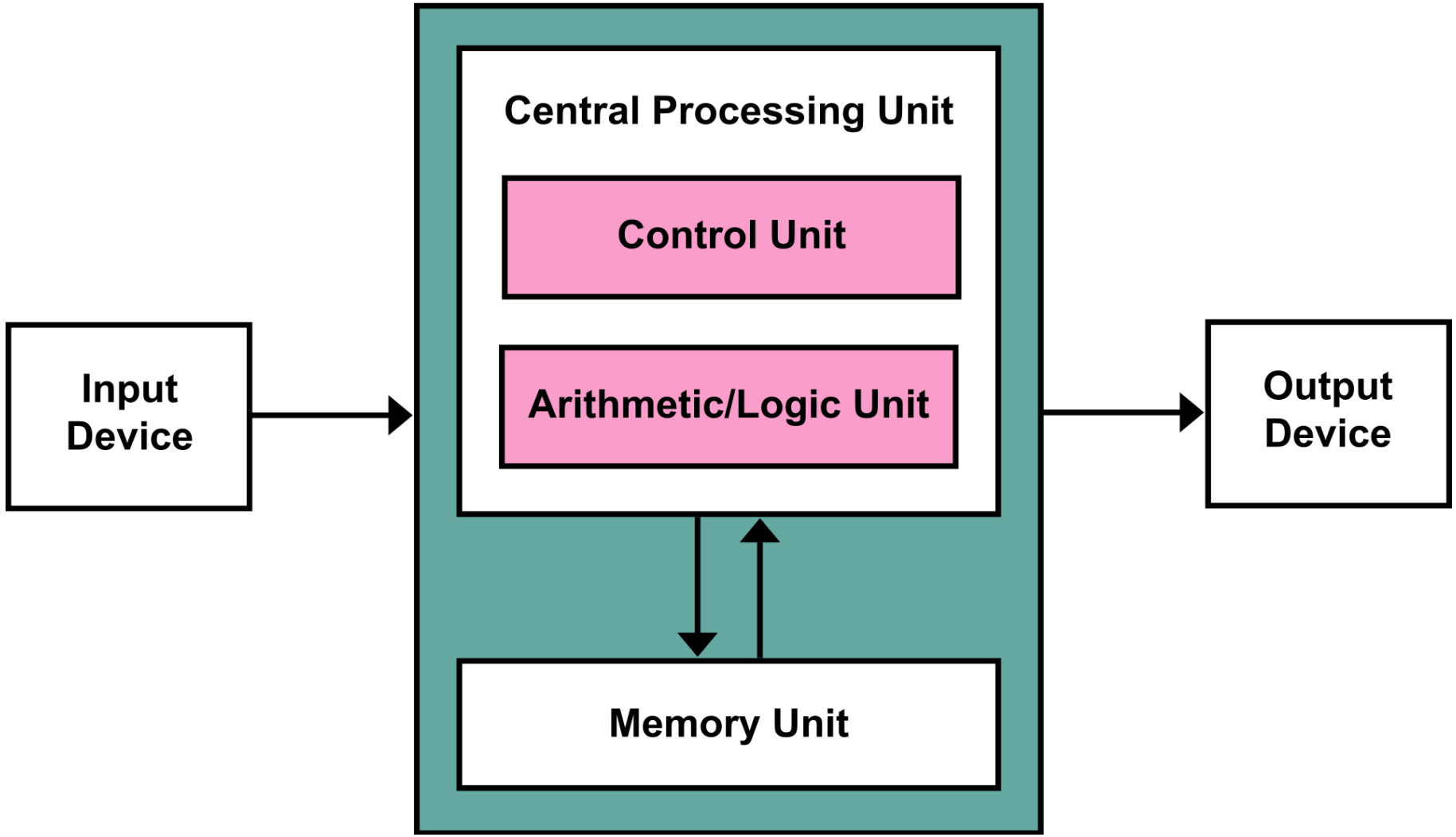
Email: sorathhansrajani.bukc@bahria.edu.pk

Computer Architecture and Performance Parameters

Von Neumann Architecture

- Introduced by John Von Neumann in 1945.
- The first generation electronic computer.
- Also known as IAS computer.
- Designed at the Princeton Institute for Advanced Studies.
- Completed in 1952 and is the prototype of all subsequent general-purpose computers.

Von Neumann Architecture – Block Diagram



Von Neumann Architecture

- Historically there have been 2 types of Computers:
  - **Fixed Program Computers –** Their function is very specific and they couldn't be re-programmed, for example: Calculators.
  - **Stored Program Computers –** These can be programmed to carry out many different tasks, applications are stored on them.
- Von Neumann Architecture is based on stored-program computer concept.
- It consists of:
  - A **main memory**, which stores both data and instructions.
  - An **Arithmetic and Logic Unit (ALU)** capable of operating on binary data.
  - A **control unit**, which interprets the instructions in memory and causes them to be executed.
  - **Input-Output (I/O)** equipment operated by the control unit.

Von Neumann Architecture

- Instruction and data are stored in same memory.
- Three key concepts:
  - Data and instructions are stored in a single read-write memory.
  - The contents of this memory are addressable by location, without regard to the type of data contained there.
  - Execution occurs in a sequential fashion, from one instruction to the next.

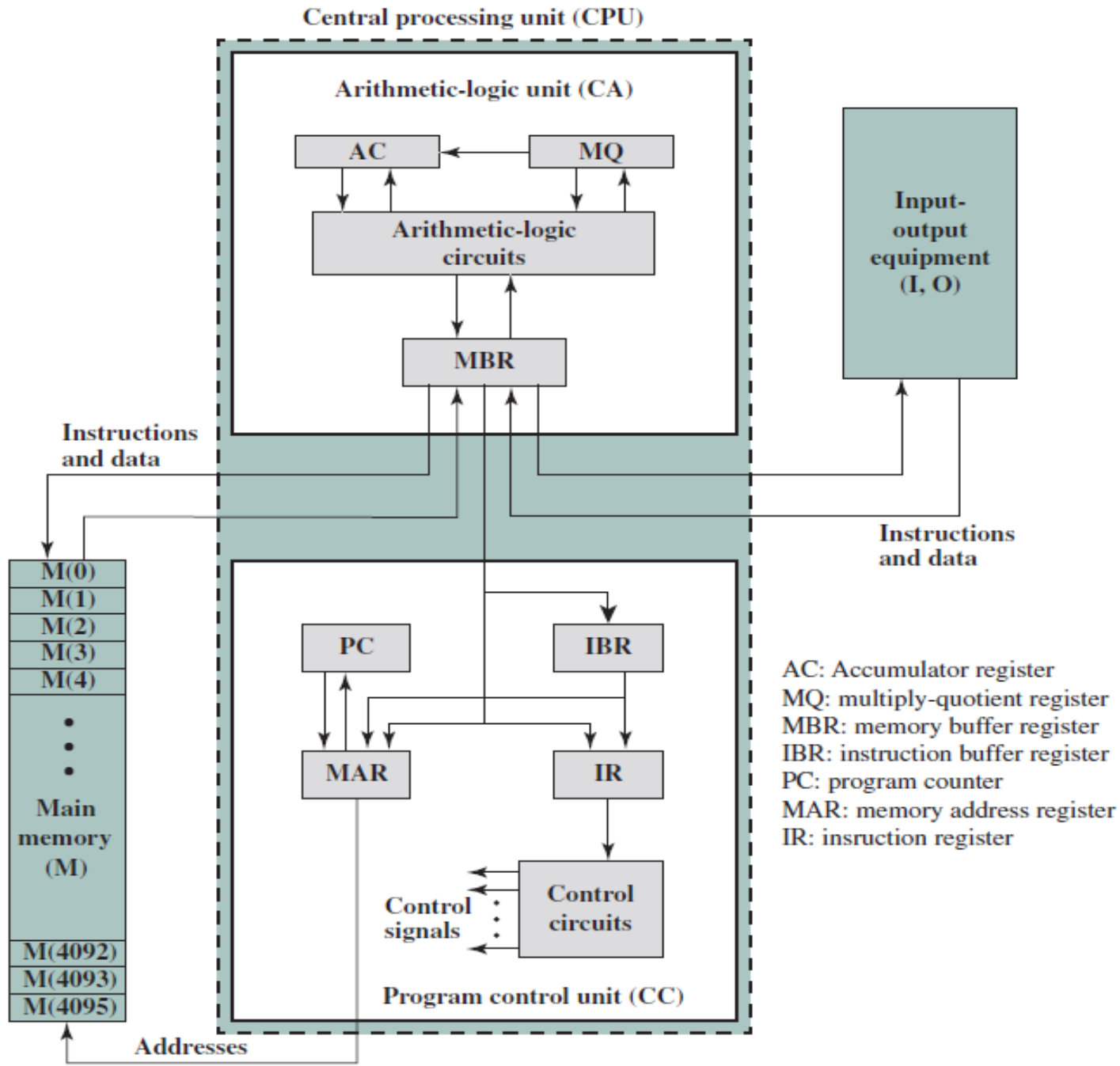
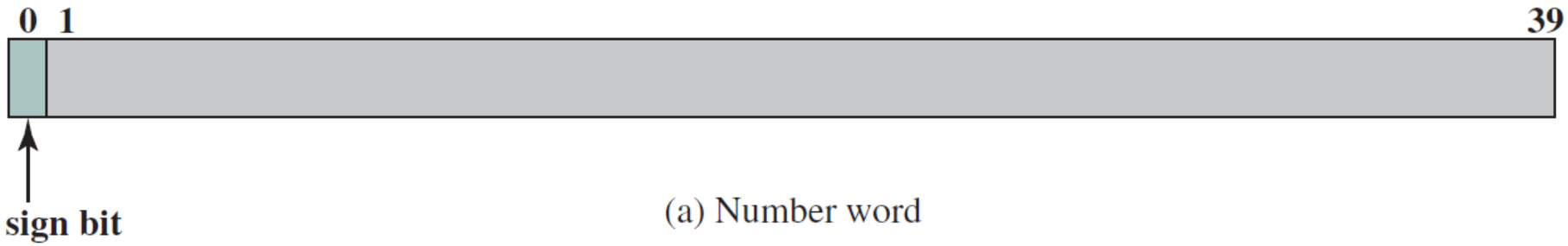


Figure 1.6 IAS Structure

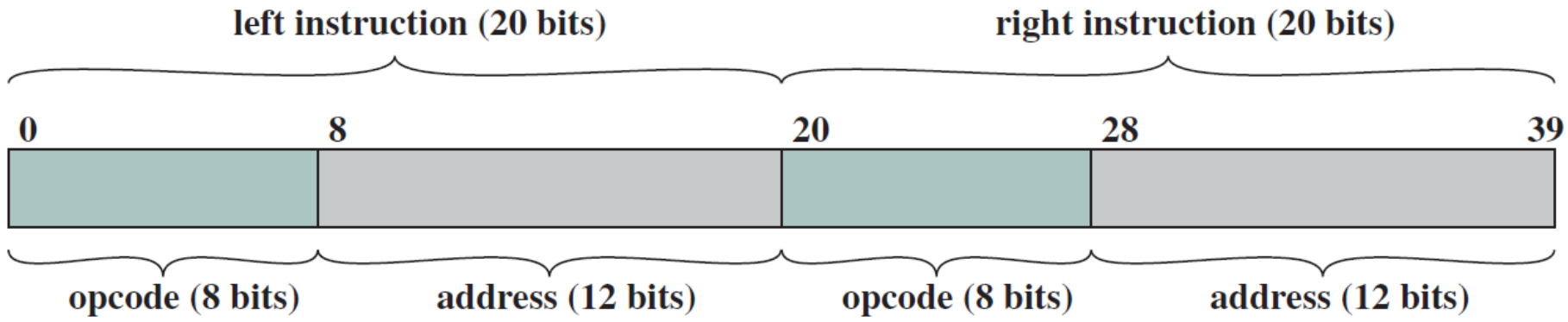
IAS Computer

- Memory consists of 4096 storage locations, called words, of 40 bits each.
- Instructions and data are stored in binary format.
- Each number is represented by a sign bit and a 39-bit value.
- A word may contain two 20-bit instructions.
- Each instruction with 8-bit opcode (operation code) specifying the operation to be performed, and 12-bit address.

IAS Memory Formats



(a) Number word



(b) Instruction word

Figure 1.7 IAS Memory Formats

IAS Registers

- **Memory buffer register (MBR):** Contains a word to be stored in memory or sent to the I/O unit or is used to receive a word from memory or from the I/O unit.
- **Memory address register (MAR):** Specifies the address in memory of the word to be written from or read into the MBR.
- **Instruction register (IR):** Contains the 8-bit opcode instruction being executed.
- **Instruction buffer register (IBR):** Employed to hold temporarily the right-hand instruction from a word in memory.
- **Program counter (PC):** Contains the address of the next instruction pair to be fetched from memory.
- **Accumulator (AC)** and **multiplier quotient (MQ):** Employed to hold temporarily operands and results of ALU operations.

IAS Instruction Set

- Total of 21 instructions, grouped in 5 categories as follows:
  - **Data transfer:** Move data between memory and ALU registers or between two ALU registers.
  - **Unconditional branch:** Normally, the control unit executes instructions in sequence from memory. This sequence can be changed by a branch instruction, which facilitates repetitive operations.
  - **Conditional branch:** The branch can be made dependent on a condition, thus allowing decision points.
  - **Arithmetic:** Operations performed by the ALU.
  - **Address modify:** Permits addresses to be computed in the ALU and then inserted into instructions stored in memory. This allows a program considerable addressing flexibility.

Table 1.1 The IAS Instruction Set

Instruction Type	Opcode	Symbolic Representation	Description
Data transfer	00001010	LOAD MQ	Transfer contents of register MQ to the accumulator AC
	00001001	LOAD MQ,M(X)	Transfer contents of memory location X to MQ
	00100001	STOR M(X)	Transfer contents of accumulator to memory location X
	00000001	LOAD M(X)	Transfer M(X) to the accumulator
	00000010	LOAD –M(X)	Transfer –M(X) to the accumulator
	00000011	LOAD |M(X)|	Transfer absolute value of M(X) to the accumulator
	00000100	LOAD –|M(X)|	Transfer –|M(X)| to the accumulator
Unconditional branch	00001101	JUMP M(X,0:19)	Take next instruction from left half of M(X)
	00001110	JUMP M(X,20:39)	Take next instruction from right half of M(X)
Conditional branch	00001111	JUMP + M(X,0:19)	If number in the accumulator is nonnegative, take next instruction from left half of M(X)
	00010000	JUMP + M(X,20:39)	If number in the accumulator is nonnegative, take next instruction from right half of M(X)
Arithmetic	00000101	ADD M(X)	Add M(X) to AC; put the result in AC
	00000111	ADD |M(X)|	Add |M(X)| to AC; put the result in AC
	00000110	SUB M(X)	Subtract M(X) from AC; put the result in AC
	00001000	SUB |M(X)|	Subtract |M(X)| from AC; put the remainder in AC
	00001011	MUL M(X)	Multiply M(X) by MQ; put most significant bits of result in AC, put least significant bits in MQ
	00001100	DIV M(X)	Divide AC by M(X); put the quotient in MQ and the remainder in AC
	00010100	LSH	Multiply accumulator by 2; that is, shift left one bit position
	00010101	RSH	Divide accumulator by 2; that is, shift right one position
Address modify	00010010	STOR M(X,8:19)	Replace left address field at M(X) by 12 rightmost bits of AC
	00010011	STOR M(X,28:39)	Replace right address field at M(X) by 12 rightmost bits of AC

Microprocessor Speed

- Pipelining:
  - Multiple stages of operation including fetching the instruction, decoding the opcode, fetching operands, performing a calculation, and so on..
  - Pipelining enables a processor to work simultaneously on multiple instructions.
  - For example: while one instruction is being executed, the computer is decoding the next instruction.
- Branch prediction:
  - The processor looks ahead in the instruction code fetched from memory and predicts which branches or group of instructions, are likely to be processed next.
  - Prefetches the correct instructions and buffer them so that the processor is kept busy.
- Superscalar execution:
  - Ability to issue more than one instruction in every processor clock cycle.
  - Multiple parallel pipelines are used.

Microprocessor Speed

- Data flow analysis:
  - The processor analyses which instructions are dependent on each other's results, or data, to create an optimized schedule of instructions.
  - Prevents unnecessary delay.
- Speculative execution:
  - Using branch prediction and data flow analysis, some processors speculatively executes instructions ahead of their actual appearance in the program execution, holding the results in temporary locations.
  - Enabling processors to keep their execution engines as busy as possible by executing instructions that are likely to be needed.

System Clock

- Performance is one of the key parameters to consider, along with cost, size, security, reliability, and power consumption (in some cases).
- Operations performed by a processor (fetching, decoding, execution), are governed by a system clock.
- All operations begin with the pulse of the clock.

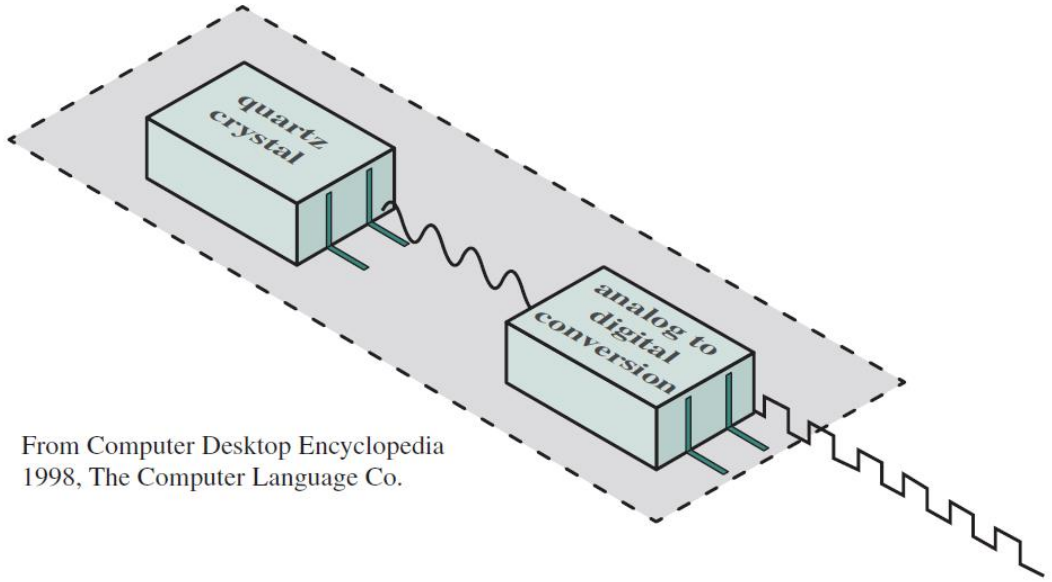


Figure 2.5 System Clock

Computer Performance Measures – Clock Speed

- Clock rate or clock speed:
  - Pulse frequency or the rate of pulses produced by the clock.
  - Measured in number of cycles per second or Hertz (Hz).
- Clock cycle or clock tick:
  - One increment, or pulse of the clock.
- Cycle time:
  - The time between pulses (t = 1/f).

Computer Performance Measures – Instruction Execution Rate

- CPI:
  - Average number of cycles per instruction.
- MIPS:
  - Millions of Instructions per second.
- FLOPS or MFLOPS:
  - Floating point operations per second or
  - Millions of floating-point operations per second.