**Unit I: Introduction**

**Syllabus** :Basic functional blocks of a computer: CPU, memory, input-output subsystems, control unit.Instruction set architecture of a CPU –registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set, Instruction set architecture CISC, RISC, Case study –instruction sets of common CPUs

**Basic functional blocks of a computer**

**Computer**: A computer is a combination of hardware and software resources which integrate together and provides various functionalities to the user. Hardware are the physical components of a computer like the processor, memory devices, monitor, keyboard etc. while software is the set of programs or instructions that are required by the hardware resources to function properly.

There are a few basic components that aids the working-cycle of a computer i.e. the Input- Process- Output Cycle and these are called as the functional components of a computer. It needs certain input, processes that input and produces the desired output. The input unit takes the input, the central processing unit does the processing of data and the output unit produces the output. The memory unit holds the data and instructions during the processing.



**Central Processing Unit (CPU) :** Once the information is entered into the computer by the input device, the processor processes it. The CPU is called the brain of the computer because it is the control center of the computer. It first fetches instructions from memory and then interprets them so as to know what is to be done. If required, data is fetched from memory or input device. Thereafter CPU executes or performs the required computation and then either stores the output or displays on the output device. The CPU has three main components which are responsible for different functions – Arithmetic Logic Unit (ALU), Control Unit (CU) and Memory registers

**Memory** : Memory attached to the CPU is used for storage of data and instructions and is called internal memory The internal memory is divided into many storage locations, each of which can store data or instructions. Each memory location is of the same size and has an address. With the help of the address, the computer can read any memory location easily without having to search the entire memory. when a program is executed, it’s data is copied to the internal memory and is stored in the memory till the end of the execution. The internal memory is also called the Primary memory or Main memory. This memory is also called as RAM, i.e. Random Access Memory. The time of access of data is independent of its location in memory, therefore this memory is also called Random Access memory (RAM). Read this for different types of RAMs

**Input Unit :**The input unit consists of input devices that are attached to the computer. These devices take input and convert it into binary language that the computer understands. Some of the common input devices are keyboard, mouse, joystick, scanner etc.

**Output** Unit : The output unit consists of output devices that are attached with the computer. It converts the binary data coming from CPU to human understandable form. The common output devices are monitor, printer, plotter etc.

**Arithmetic and Logic Unit (ALU) :** The ALU, as its name suggests performs mathematical calculations and takes logical decisions. Arithmetic calculations include addition, subtraction, multiplication and division. Logical decisions involve comparison of two data items to see which one is larger or smaller or equal.

**Control Unit** : The Control unit coordinates and controls the data flow in and out of CPU and also controls all the operations of ALU, memory registers and also input/output units. It is also responsible for carrying out all the instructions stored in the program. It decodes the fetched instruction, interprets it and sends control signals to input/output devices until the required operation is done properly by ALU and memory.

**Memory Registers** : A register is a temporary unit of memory in the CPU. These are used to store the data which is directly used by the processor. Registers can be of different sizes(16 bit, 32 bit, 64 bit and so on) and each register inside the CPU has a specific function like storing data, storing an instruction, storing address of a location in memory etc. The user registers can be used by an assembly language programmer for storing operands, intermediate results etc. Accumulator (ACC) is the main register in the ALU and contains one of the operands of an operation to be performed in the ALU.

**Instruction set architecture of a CPU :**

An Instruction Set Architecture (ISA) is part of the abstract model of a computer that defines how the CPU is controlled by the software. The ISA acts as an interface between the hardware and the software, specifying both what the processor is capable of doing as well as how it gets done.

The ISA provides the only way through which a user is able to interact with the hardware. It can be viewed as a programmer’s manual because it’s the portion of the machine that’s visible to the assembly language programmer, the compiler writer, and the application programmer

The ISA defines the supported data types, the registers, how the hardware manages main memory, key features (such as virtual memory), which instructions a microprocessor can execute, and the input/output model of multiple ISA implementations. The ISA can be extended by adding instructions or other capabilities, or by adding support for larger addresses and data values.

**Registers :** Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. The registers used by the CPU are often termed as Processor registers.

A processor register may hold an instruction, a storage address, or any data (such as bit sequence or individual characters).

The computer needs processor registers for manipulating data and a register for holding a memory address. The register holding the memory location is used to calculate the address of the next instruction after the execution of the current instruction is completed.

**Registers Involved In Each Instruction Cycle:**

**Memory address registers(MAR)** : It is connected to the address lines of the system bus. It specifies the address in memory for a read or write operation.

**Memory Buffer Register(MBR)** : It is connected to the data lines of the system bus. It contains the value to be stored in memory or the last value read from the memory.

**Program Counter(PC)** : Holds the address of the next instruction to be fetched.

**Instruction Register(IR)** : Holds the last instruction fetched.

**The Instruction Cycle –**

Each phase of Instruction Cycle can be decomposed into a sequence of elementary micro-operations. In the above examples, there is one sequence each for the Fetch, Indirect, Execute and Interrupt Cycles.



The Indirect Cycle is always followed by the Execute Cycle. The Interrupt Cycle is always followed by the Fetch Cycle. For both fetch and execute cycles, the next cycle depends on the state of the system.

**Register Transfer Language (RTL)**

In symbolic notation, it is used to describe the micro-operations transfer among registers. It is a kind of intermediate representation (IR) that is very close to assembly language, such as that which is used in a compiler.The term “Register Transfer” can perform micro-operations and transfer the result of operation to the same or other register.

**Micro-operations :**

The operation executed on the data store in registers are called micro-operations. They are detailed low-level instructions used in some designs to implement complex machine instructions.

**Register Transfer** :

The information transformed from one register to another register is represented in symbolic form by replacement operator is called Register Transfer.

**Replacement Operator :**

In the statement, R2 <- R1, <- acts as a replacement operator. This statement defines the transfer of content of register R1 into register R2.

**There are various methods of RTL –**

1. General way of representing a register is by the name of the register enclosed in a rectangular box as shown in (a).
2. Register is numbered in a sequence of 0 to (n-1) as shown in (b).
3. The numbering of bits in a register can be marked on the top of the box as shown in (c).
4. A 16-bit register PC is divided into 2 parts- Bits (0 to 7) are assigned with lower byte of 16-bit address and bits (8 to 15) are assigned with higher bytes of 16-bit address as shown in (d).

**Addressing Modes**– The term addressing modes refers to the way in which the operand of an instruction is specified. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

**Microprocessor - 8085 Instruction Sets**

A binary command that is used to perform a function in the microprocessor over provided data is known as instruction. A set of instructions is known as an instruction set that decides the microprocessor function. Every instruction includes two parts like Opcode & the Operand where Opcode is used to specify the function to be executed & operand gives the data to be functioned on.

**Classification of Instruction Set of 8085**

The instruction set of 8085 microprocessor is classified into five types which include the following.



**Data Transfer Instruction :**

An instruction that is used to transfer the data from one register to another is known as data transfer instruction. So, the data transfer can be done from source to destination without changing the source contents. Data transfer mainly occurs from one register to another register, from memory location to register, register to memory, and between an I/O device & accumulator.

**Arithmetic Instruction of 8085**

The arithmetic instructions perform different operations like addition, subtraction, increment & decrement on the data within memory & register in the 8085 microprocessor.

**Logical Instruction:**

Logical instructions are mainly used to perform different operations like logical or Boolean over the data available in either memory or register. These instructions will modify the flag bits based on the operation executed.

**Branching Instruction:**

These types of instructions are mainly used to transfer or switch the microprocessor from one location to another. So, it simply changes the general sequential flow.

**Control Instruction:**

These instructions are mainly used to control the microprocessor operations.

**Reduced Instruction Set Architecture (RISC) –**

The main idea behind this is to make hardware simpler by using an instruction set composed of a few basic steps for loading, evaluating, and storing operations just like a load command will load data, a store command will store the data.

**Complex Instruction Set Architecture (CISC) –**

The main idea is that a single instruction will do all loading, evaluating, and storing operations just like a multiplication command will do stuff like loading data, evaluating, and storing it, hence it’s complex.

**Instruction sets of common CPUs**

A set of codes that can only be understood by a processor of the computer or CPU is known as an instruction set. These codes and machine languages are generally present as 1s and 0s. The movements of bits and bytes are controlled by these instruction sets present in the processor.

Some common examples of instruction sets are:

JUMP – jump instruction set is used to jump to any designated address of RAM.

ADD – add instruction set is used to add any two numbers together.

LOAD – load instruction set is used to load any required information from the RAM to the CPU.

**Types of instruction set**

Many popular and useful instruction sets are important and useful in computer science. These sets of instructions have their advantages as well as usages. Here are the types of instruction sets:

**1. Reduced instruction set computer (RISC)**

RISC has only a few cycles per instruction. It has a simpler form than a complex set of instructions. RISC is also used in many supercomputers. For example, it uses a summit, which is a supercomputer. It was the world’s fastest supercomputer as per data in 2018.

**2. Minimal instruction set computers (MISC)**

A few codes and a set of instructions are basic for any processor. They also include sub-codes. As a result, they are smaller and faster. A disadvantage of MISC is that it has more sequential dependencies.

**3. Complex instruction set computer (CISC)**

CISC is a set of instructions with a few instructions per program. A CISC has fewer instructions than RISC.

**4. Explicitly parallel instruction computing (EPIC)**

This is an instruction set that permits microprocessors that help to execute instructions in parallel software. EPIC intends to give a simpler performance.

**5. Very long instruction word (VLIW)**

VLIW exploits parallelism at the instruction level. By this set of instructions, instructions are processed in sequence only in the CPU. This set of instructions improves the performance of the CPU.

**6. Zero instruction set computer (ZISC)**

The instructions that do not include microinstructions are known as ZISC. They are based on the pattern matching and can be compared to networks of synapses and neurons.

**7. One instruction set computer (OISC)**

The OISC set of instructions uses only one instruction for a machine language. This set of instructions is used to teach computer architecture and to compute structural computing research.