

# CPU Structure

The CPU or processor acts as the controller of all actions or services provided by the system. It is the Brain and the Heart of the Computer. The operations of a CPU can be reduced into the following:

1. Fetch the next instruction from memory.

2. Decode the instruction.

3. Execute the instruction.

4. Store the result of the instruction into main memory.

In general, the CPU has three main units: (See the figure (1) below)

1. Arithmetic and Logic Unit (ALU): Performs arithmetic and logical operations. For example, it can add together two binary numbers either from memory or from some of the CPU registers.

2. Control Unit: controls the action of the other computer components so that instructions are executed in the correct sequence.

3. Registers - Temporary storage inside CPU. Registers can be read and written at high speed as they are inside the CPU.



# Fig.(1) block diagram of CPU Structure

#### • Internal Architecture of 8086µP

The architecture of 8086µP is shown in figure (2) below. It has two separate functional units: Bus Interface Unit (BIU) and Execution Unit (EU). The 8086µP architecture employs parallel processing—i.e., both the units (BIU and EU) work at the same time. Parallel processing makes the fetch and execution of instructions independent operations. This results in efficient use of the system bus and higher performance for 8086µP systems.

• The BIU has segment registers, instruction pointer, address generation and bus control logic block, instruction queue.

## The main jobs performed by BIU are:

1. The BIU performs all bus operations such as instruction fetching, reading and writing operands for memory and calculating the addresses of the memory operands.

2. Input/output of data from/to input/output peripherals.

• The EU has general purpose registers, ALU, control unit, instruction register, flags (or status) register.

# The main jobs performed by the execution unit EU are:

1. Decoding/execution of instructions.

2. It accepts instructions from the instruction queue and data from the general purpose registers or memory.

3. EU tests the status of flags in the control register and updates them

## when executing instructions.



Fig. (2) block diagram of CPU architecture

#### <u>Register organization</u>

Register organization is the arrangement of the registers in the processor. The processor designers decide the organization of the registers in a processor. Different processors may have different register organization. Depending on the roles played by the registers they can be categorized into two types, user-visible register and control and status register.

#### What is Register?

**Registers** are the *smaller* and the *fastest* accessible **memory units** in the central processing unit (CPU). According to memory hierarchy, the registers in the processor, function a **level above** the **main memory** and **cache memory**. The registers used by the central unit are also called as **processor registers**.

A register can hold the *instruction, address location, or operands*. Sometimes, the instruction has register as a part of itself.

#### • **Types of Registers**

As we have discussed above, registers can be organized into two main categories i.e. the User-Visible Registers and the Control and Status Registers. Although we can't separate the registers in the processors clearly among these two categories.



This is because in some processors, a register may be user-visible and in some, the same may not be user-visible. But for our rest of discussion regarding register organization, we will consider these two categories of register.

#### 1. User Visible Registers

- General Purpose Register
- Data Register
- Address Register
- Condition Codes

## 2. Control and Status Registers

- Program Counter
- Instruction Register
- Memory Address Register
- Memory Buffer Register

#### • User-Visible Registers

These registers are visible to the assembly or machine language programmers and they use them effectively to minimize the memory references in the instructions. Well, these registers can only be referenced using the machine or assembly language.



The registers that fall in this category are discussed below:

## 1. General Purpose Register

The general-purpose registers detain both the addresses or the data. Although we have separate data registers and address registers. The general purpose register also accepts the intermediate results in the course of program execution.

Well, the programmers can restrict some of the general-purpose registers to specific functions. Like, some registers are specifically used for stack operations or for floating-point operations. The general-purpose register can also be employed for the addressing functions.

# 2. Data Register

The term itself describes that these registers are employed to hold the data. But the programmers can't use these registers for calculating operand address.

# 3. Address Register

Now, the address registers contain the address of an operand or it can also act as a general-purpose register. An address register may be dedicated to a certain addressing mode. Let us understand this with the examples.

# (a) Segment Pointer Register

A memory divided in segments, requires a segment register to hold the base address of the segment. There can be multiple segment registers. As one segment register can be employed to hold the base address of the segment occupied by the operating system. The other segment register can hold the base address of the segment allotted to the processor.

# (b) Index Register

The index register is employed for indexed addressing and it is initial value is 0. Generally, it used for traversing the memory locations. After each reference, the index register is incremented or decremented by 1, depending upon the nature of the operation. Sometime the index register may be auto indexed.

# (c) Stack Pointer Register

The stack register has the address that points the stack top.

7

## 4. Condition Code

Condition codes are the flag bits which are the part of the control register. The condition codes are set by the processor as a result of an operation and they are implicitly read through the machine instruction.

The programmers are not allowed to alter the conditional codes. Generally, the condition codes are tested during conditional branch operation.

Control and Status Registers

The control and status register holds the address or data that is important to control the processor's operation. The most important thing is that these registers are not visible to the users. Below we will discuss all the control and status registers are essential for the execution of an instruction.

# <u>Control and Status Registers</u>

# 1. Program Counter

The program counter is a processor register that holds the address of the instruction that has to be executed next. It is a processor which updates the program counter with the address of the next instruction to be fetched for execution.

# 2. Instruction Register

Instruction register has the instruction that is currently fetched. It helps in analyzing the opcode and operand present in the instruction.

8

#### 3. Memory Address Register (MAR)

Memory address register holds the address of a memory location.

# 4. Memory Buffer Register (MBR)

The memory buffer register holds the data that has to be written to a memory location or it holds the data that is recently been read.

The memory address registers (MAR) and memory buffer registers (MBR) are used to move the data between processor and memory.

Apart from the above registers, several processors have a register termed as Program Status Word (PSW). As the word suggests it contains the status information.