WEEK- 9

Chapter 3

An Introduction to a Simple Computer

3-1 Introduction

Designing a computer now a day is a job for a computer engineer. However, in this chapter, we first look at a very simple computer called MARIE: A Machine Architecture that is Really Intuitive and Easy. The objective of this chapter is to give you an understanding of how a computer functions. We have, therefore, kept the architecture as uncomplicated as possible.

3.2 CPU Basics and Organization

A computer must manipulate binary-coded data. Memory is used to store both data and program instructions (also in binary). Somehow, the program must be executed and the data must be processed correctly.

The central processing unit (CPU) is responsible for fetching program instructions, decoding each instruction that is fetched, and performing the indicated sequence of operations on the correct data. All computers have a central processing unit. This unit can be divided into two pieces. The first is the **data path**, which is a network of storage units (registers) and arithmetic and logic units (for performing various operations on data) connected by buses (capable of moving data from place to place) where the timing is controlled by clocks. The second CPU component is the **control unit**: A module responsible for sequencing operations and making sure the correct data is where it needs to be at the correct time. Together, these components perform the tasks of the CPU: fetching instructions, decoding them, and finally performing the indicated sequence of operations. The performance of a machine is directly affected by the design of the data path and the control unit. Therefore, we cover these components of the CPU in detail in the following sections.

The Registers.

Registers are used in computer systems as places to store a wide variety of data, such as addresses, program counters, or data necessary for program execution.

A register is a hardware device that stores binary data. Registers are located on the processor so information can be accessed very quickly. To build a 16-bit register, we need to connect 16 D flip-flops together. At each pulse of the clock, input enters the register and cannot be changed (and thus is stored) until the clock pulses again.

Data processing on a computer is usually done on fixed size binary words that are stored in registers. Therefore, most computers have registers of a certain size. Common sizes include 16, 32, and 64 bits. The number of registers in a machine varies from architecture to architecture, but is typically a power of 2, with 16 and 32 being most common. Registers contain data, addresses, or control information. Some registers are specified as “special purpose” and may contain only data, only addresses, or only control information. Other registers are more generic and may hold data, addresses, and control information at various times.

In modern computer systems, there are many types of specialized registers:

Registers to store information, registers to shift values, registers to compare values, and registers that count. There are “scratchpad” registers that store temporary values, index registers to control program looping, stack pointer registers to manage stacks of information for processes, status registers to hold the status or mode of operation (such as overflow, carry, or zero conditions), and general purpose registers that are the registers available to the programmer.

The ALU.

The arithmetic logic unit (ALU) carries out the logic operations (such as comparisons) and arithmetic operations (such as add or multiply) required during the program execution. Generally an ALU has two data inputs and one data output. Operations performed in the ALU often affect bits in the status register (bits are set to indicate actions such as whether an overflow has occurred). The ALU knows which operations to perform because it is controlled by signals from the control unit.

The Control Unit.

The control unit is the “policeman” or “traffic manager” of the CPU. It monitors the execution of all instructions and the transfer of all information. The control unit extracts instructions from memory, decodes these instructions, making sure data is in the right place at the right time, tells the ALU which registers to use, services interrupts, and turns on the correct circuitry in the ALU for the execution of the desired operation. The control unit uses a program counter register to find the next instruction for execution and a status register to keep track of overflows, carries, borrows, and the like.