

---

## CHAPTER EIGHT

---

# Central Processing Unit

### IN THIS CHAPTER

- 8-1 Introduction
- 8-2 General Register Organization
- 8-3 Stack Organization
- 8-4 Instruction Formats
- 8-5 Addressing Modes
- 8-6 Data Transfer and Manipulation
- 8-7 Program Control
- 8-8 Reduced Instruction Set Computer (RISC)

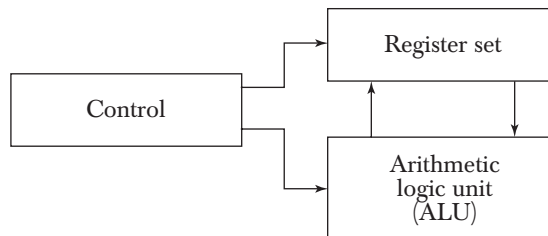
### 8-1 Introduction

---

*CPU*

The part of the computer that performs the bulk of data-processing operations is called the central processing unit and is referred to as the CPU. The CPU is made up of three major parts, as shown in Fig. 8-1. The register set stores intermediate data used during the execution of the instructions. The arithmetic logic unit (ALU) performs the required microoperations for executing the instructions. The control unit supervises the transfer of information among the registers and instructs the ALU as to which operation to perform.

The CPU performs a variety of functions dictated by the type of instructions that are incorporated in the computer. Computer architecture is sometimes defined as the computer structure and behavior as seen by the programmer that uses machine language instructions. This includes the instruction formats, addressing modes, the instruction set, and the general organization of the CPU registers leading to two computer architectures as reduced instruction set computer (RISC) and complex instruction set computer (CISC). Based on memory usage for programs and data, two architectures, namely nonembedded and embedded are evolved. Nonembedded computer architectures are basically stored program computer (SPC) architectures in which programs and data reside in the same memory system. Embedded architectures are basically Harvard computer architectures in which programs and data



**Figure 8-1** Major components of CPU.

reside in different memory systems leading to doubling the memory bandwidth. Example of nonembedded computers are all desktop systems such as personal computers. Example of embedded computers are microcontroller-based systems and digital signal processor-based (DSP) systems.

One boundary where the computer designer and the computer programmer see the same machine is the part of the CPU associated with the instruction set. From the designer's point of view, the computer instruction set provides the specifications for the design of the CPU. The design of a CPU is a task that in large part involves choosing the hardware for implementing the machine instructions. The user who programs the computer in machine or assembly language must be aware of the register set, the memory structure, the type of data supported by the instructions, and the function that each instruction performs.

Design examples of simple CPUs are carried out in Chaps. 5 and 7. This chapter describes the organization and architecture of the CPU with an emphasis on the user's view of the computer. We briefly describe how the registers communicate with the ALU through buses and explain the operation of the memory stack. We then present the type of instruction formats available, the addressing modes used to retrieve data from memory, and typical instructions commonly incorporated in computers. The last section presents the concept of reduced instruction set computer (RISC).

## 8-2 General Register Organization

In the programming examples of Chap. 6, we have shown that memory locations are needed for storing pointers, counters, return addresses, temporary results, and partial products during multiplication. Having to refer to memory locations for such applications is time consuming because memory access is the most time-consuming operation in a computer. It is more convenient and more efficient to store these intermediate values in processor registers. When a large number of registers are included in the CPU, it is most efficient to connect them through a common bus system. The registers communicate with each other not only for direct data transfers, but also while performing various microoperations. Hence it is necessary to provide a common unit that can perform all the arithmetic, logic, and shift microoperations in the processor.

*bus system*

A bus organization for seven CPU registers is shown in Fig. 8-2. The output of each register is connected to two multiplexers (MUX) to form the two buses *A* and *B*. The selection lines in each multiplexer select one register or the input data for the particular bus. The *A* and *B* buses form the inputs to a common arithmetic logic unit (ALU). The operation selected in the ALU determines the arithmetic or logic microoperation that is to be performed. The result of the microoperation is available for output data and also goes into the inputs of all the registers. The register that receives the information from the output bus is selected by a decoder. The decoder activates one of the register load inputs, thus providing a transfer path between the data in the output bus and the inputs of the selected destination register.

The control unit that operates the CPU bus system directs the information flow through the registers and ALU by selecting the various components in the system. For example, to perform the operation

$$R1 \leftarrow R2 + R3$$

the control must provide binary selection variables to the following selector inputs:

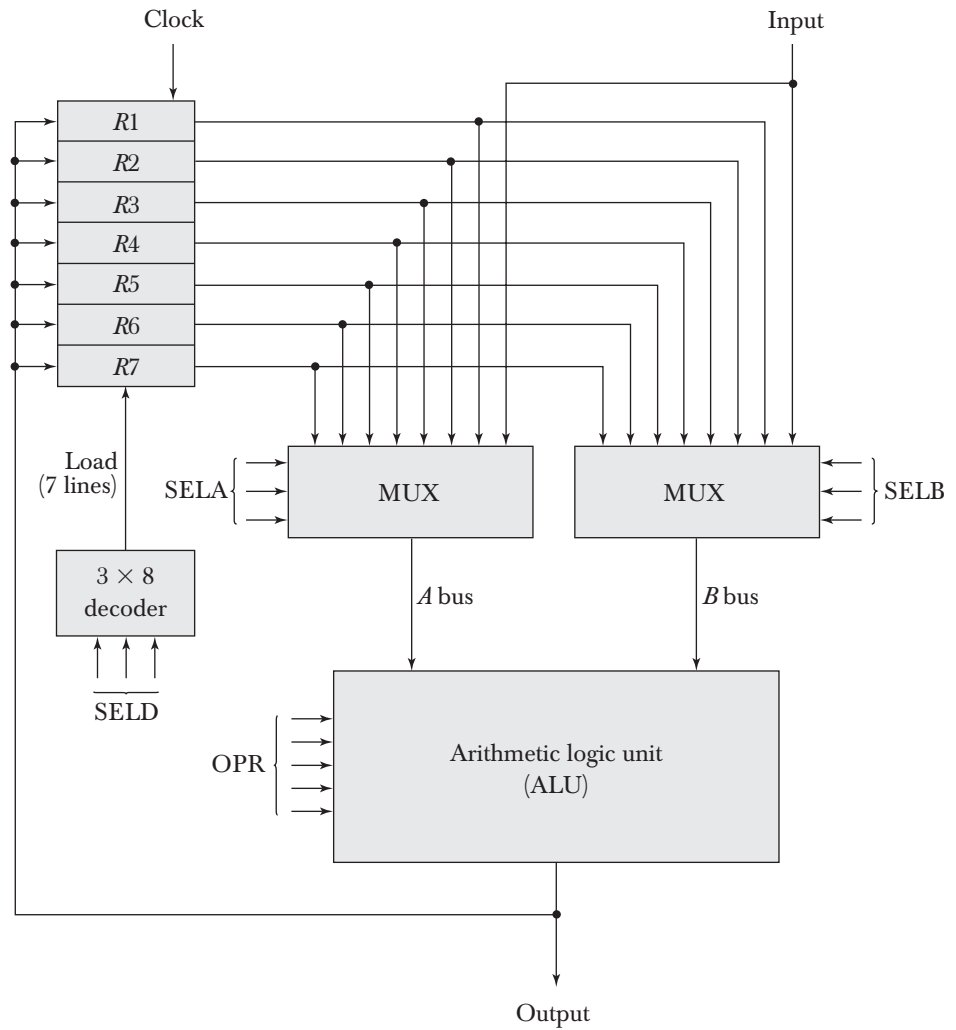
1. MUX A selector (SELA): to place the content of *R2* into bus *A*.
2. MUX B selector (SELB): to place the content of *R3* into bus *B*.
3. ALU operation selector (OPR): to provide the arithmetic addition  $A + B$ .
4. Decoder destination selector (SELD): to transfer the content of the output bus into *R1*.

The four control selection variables are generated in the control unit and must be available at the beginning of a clock cycle. The data from the two source registers propagate through the gates in the multiplexers and the ALU, to the output bus, and into the inputs of the destination register, all during the clock cycle interval. Then, when the next clock transition occurs, the binary information from the output bus is transferred into *R1*. To achieve a fast response time, the ALU is constructed with high-speed circuits. The buses are implemented with multiplexers or three-state gates, as shown in Sec. 4-3.

### Control Word

*control word*

There are 14 binary selection inputs in the unit, and their combined value specifies a *control word*. The 14-bit control word is defined in Fig. 8-2(b). It consists of four fields. Three fields contain three bits each, and one field has five bits. The three bits of SELA select a source register for the *A* input of the ALU. The three bits of SELB select a register for the *B* input of the ALU. The three bits of SELD select a destination register using the decoder and its seven load outputs. The five bits of OPR select one of the operations in the ALU. The 14-bit control word when applied to the selection inputs specify a particular microoperation.



(a) Block diagram

3	3	3	5
SELA	SELB	SELD	OPR

(b) Control word

Figure 8-2 Register set with common ALU.