21/10/2021

MICROPROCESSOR

8085 µP : Word-length is 8 bit

Hardware model of 8085
$\underbrace{\text{model}}$ → Conceptual representation of real object.

A ↑ is a 8-bit register which is a part of ALU

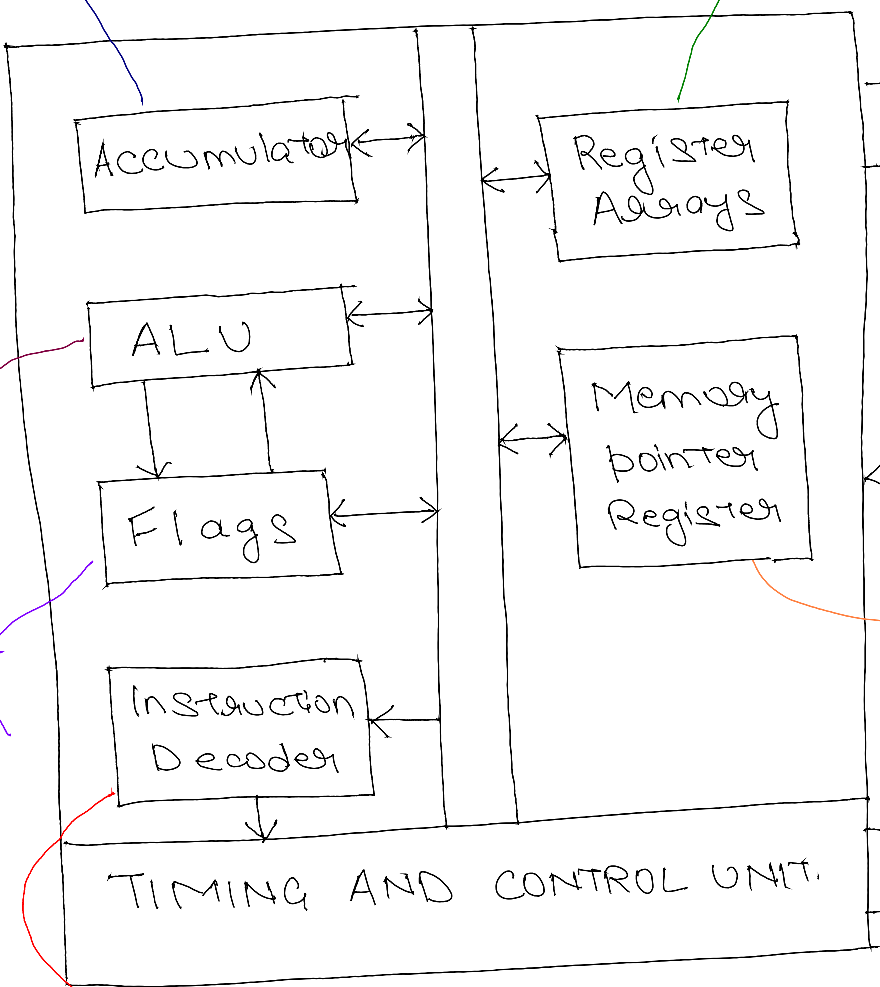
↓ Stores result

There are six general purpose register ↑

(8-bit)
B, C, D, E
H, L

(Unidirectional) Address Bus.
16 bit
To carry address of memory / I/O

Data bus Bi-dir.
8 bit



Control Signals

ALU → Arithmetic and logical Unit

Flags → is Combination of flip-flops
↓
based on result of ALU

Memory pointer Register → To hold addresses of memory location

Instruction Decoder → After instruction fetch, instruction goes to instruction decoder

Flags indicate certain Conditions → Importance in decision-making

☐ Write logical steps to solve a simple programming problem.

☐ Draw a flowchart from the logical steps of a given programming problem.

☐ Translate the flowchart into mnemonics and convert the mnemonics into Hex code for a given programming problem.

2.1 THE 8085 PROGRAMMING MODEL

A model is a conceptual representation of a real object. It can take many forms, such as text description, a drawing, or a built structure. Most of us have seen an architectural model of a building. Similarly, the microprocessor can be represented in terms of its hardware (physical electronic components) and a programming model (information needed to write programs). In Chapter 1, we described a simplified hardware model of a microprocessor as a part of the microprocessor-based system (Figure 1.3). It showed three components: ALU, register array, and control. Figure 2.1 shows a hardware model and a programming model specific to the 8085 microprocessor.

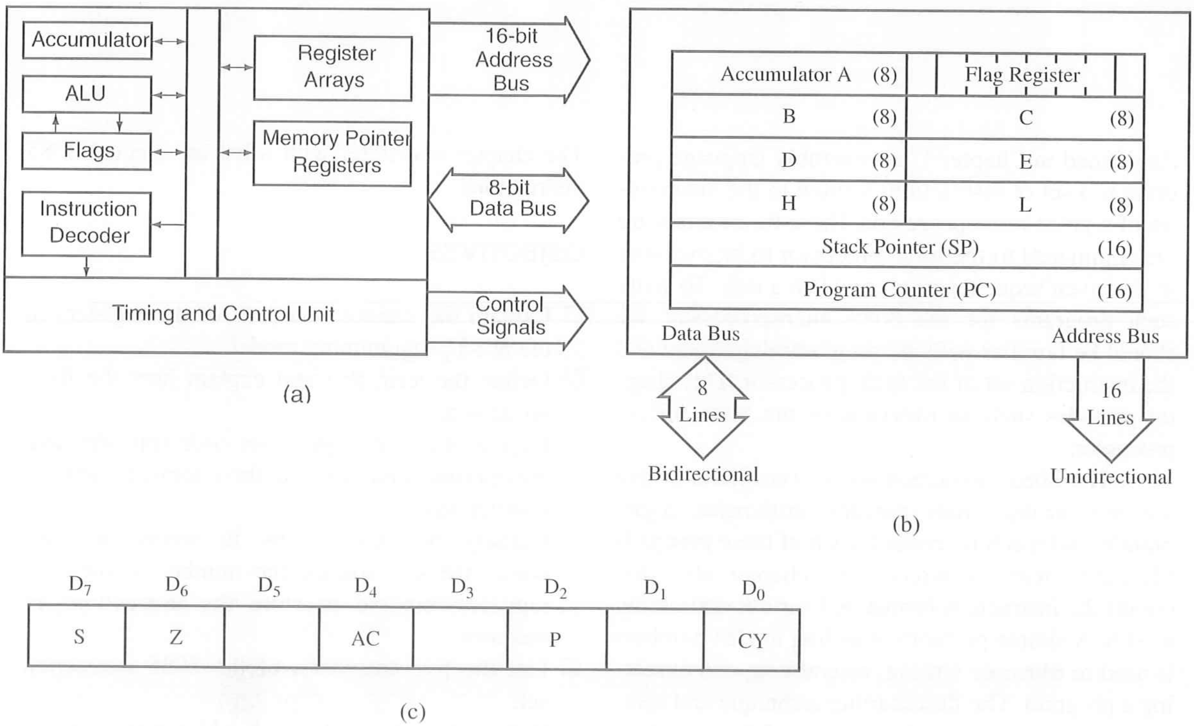


FIGURE 2.1
8085 Hardware Model (a), Programming Model (b), and Flag Register (c)

2.1.1 8085 Hardware Model

The hardware model in Figure 2.1(a) shows two major segments. One segment includes the arithmetic/logic unit (ALU) and an 8-bit register called an accumulator, instruction decoder, and flags. The second segment shows 8-bit and 16-bit registers. Both segments are connected with various internal connections called an internal bus. The arithmetic and logical operations are performed in the ALU. Results are stored in the accumulator, and flip-flops, called flags, are set or reset to reflect the results (see Figure 2.1a). There are three buses: a 16-bit unidirectional address bus, an 8-bit bidirectional data bus, and a control bus. In Chapter 1, these three buses were shown as one system bus. The 8085 processor uses the 16-bit address bus to send out memory addresses, the 8-bit data bus to transfer data, and the control bus for timing signals. The details of the hardware model are included in later chapters.

2.1.2 8085 Programming Model

The programming model consists of some segments of the ALU and the registers. This model does not reflect the physical structure of the 8085 but includes the information that is critical in writing assembly language programs. The model includes six registers, one accumulator, and one flag register, as shown in Figure 2.1(b). In addition, it has two 16-bit registers: the stack pointer and the program counter. They are described briefly as follows.

REGISTERS

The 8085 has six general-purpose registers to store 8-bit data; these are identified as B, C, D, E, H, and L, as shown in Figure 2.1(b). They can be combined as register pairs—BC, DE, and HL—to perform some 16-bit operations. The programmer can use these registers to store or copy data into the registers by using data copy instructions.

ACCUMULATOR

The accumulator is an 8-bit register that is part of the arithmetic/logic unit (ALU). This register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is stored in the accumulator. The accumulator is also identified as register A.

FLAGS

The ALU includes five flip-flops, which are set or reset after an operation according to data conditions of the result in the accumulator and other registers. They are called Zero (Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags; they are listed in Table 2.1 and their bit positions in the flag register are shown in Figure 2.1(c). The most commonly used flags are Zero, Carry, and Sign. The microprocessor uses these flags to test data conditions.

These flags have critical importance in the decision-making process of the microprocessor. The conditions (set or reset) of the flags are tested through software instructions. For example, the instruction JC (Jump On Carry) is implemented to change the sequence of a program when the CY flag is set. The thorough understanding of flags is essential in writing assembly language programs.

TABLE 2.1
The 8085 Flags

The following flags are set or reset after the execution of an arithmetic or logic operation; data copy instructions do not affect any flags. See the instruction set (Appendix F) to find how flags are affected by an instruction.

- ☐ Z—Zero: The Zero flag is set to 1 when the result is zero; otherwise it is reset.
- ☐ CY—Carry: If an arithmetic operation results in a carry, the CY flag is set; otherwise it is reset.
- ☐ S—Sign: The Sign flag is set if bit D_7 of the result = 1; otherwise it is reset.
- ☐ P—Parity: If the result has an even number of 1s, the flag is set; for an odd number of 1s, the flag is reset.
- ☐ AC—Auxiliary Carry: In an arithmetic operation, when a carry is generated by digit D_3 and passed to digit D_4, the AC flag is set. This flag is used internally for BCD (binary-coded decimal) operations; there is no Jump instruction associated with this flag.

PROGRAM COUNTER (PC) AND STACK POINTER (SP)*

These are two 16-bit registers used to hold memory addresses. The size of these registers is 16 bits because the memory addresses are 16 bits.

The microprocessor uses the PC register to sequence the execution of the instructions. The function of the program counter is to point to the memory address from which the next byte is to be fetched. When a byte (machine code) is being fetched, the program counter is incremented by one to point to the next memory location.

The stack pointer is also a 16-bit register used as a memory pointer. It points to a memory location in R/W memory, called the stack. The beginning of the stack is defined by loading a 16-bit address in the stack pointer. The stack concept is explained in Chapter 9, "Stack and Subroutines."

This programming model will be used in subsequent chapters to examine how these registers are affected after the execution of an instruction.

2.2 INSTRUCTION CLASSIFICATION

An **instruction** is a binary pattern designed inside a microprocessor to perform a specific function. The entire group of instructions, called the **instruction set,** determines what functions the microprocessor can perform. The 8085 microprocessor includes the instruction set of its predecessor, the 8080A, plus two additional instructions.

*The concept of stack memory is difficult to explain at this time; it is not necessary for the reader to understand stack memory until subroutines are discussed. It is included here only to provide continuity in the discussion of programmable registers and microprocessor operations. This concept will be explained more fully in Chapter 9.