

THE 8085 MPU

The term **microprocessing unit** (MPU) is similar to the term **central processing unit** (CPU) used in traditional computers. We define the MPU as a device or a group of devices (as a unit) that can communicate with peripherals, provide timing signals, direct data flow, and perform computing tasks as specified by the instructions in memory. The unit will have the necessary lines for the address bus, the data bus, and the control signals, and would require only a power supply and a crystal (or equivalent frequency-determining components) to be completely functional.

Using this description, the 8085 microprocessor can almost qualify as an MPU, but with the following two limitations.

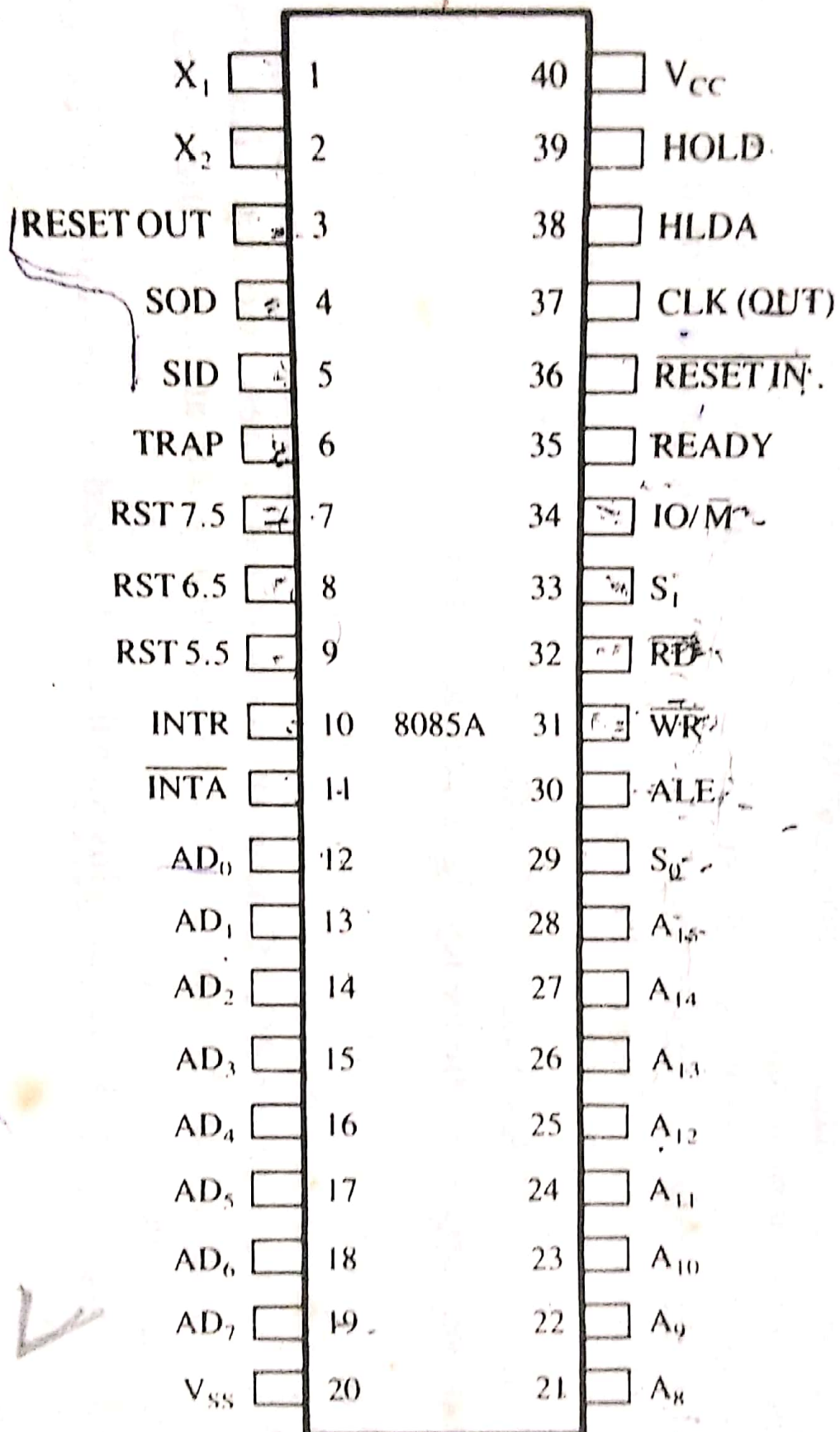
1. The low-order address bus of the 8085 microprocessor is **multiplexed** (time-shared) with the data bus. The buses need to be demultiplexed.
2. Appropriate control signals need to be generated to interface memory and I/O with the 8085. (Intel has some specialized memory and I/O devices that do not require such control signals.)

This section shows how to demultiplex the bus and generate the control signals after describing the 8085 microprocessor and illustrates the bus timings.

3.11 The 8085 Microprocessor

The 8085A (commonly known as the 8085) is an 8-bit general-purpose microprocessor capable of addressing 64K of memory. The device has forty pins, requires a +5 V single power supply, and can operate with a 3-MHz single-phase clock. The 8085A-2 version can operate at the maximum frequency of 5 MHz. The 8085 is an enhanced version of its predecessor, the 8080A; its instruction set is upward-compatible with that of the 8080A, meaning that the 8085 instruction set includes all the 8080A instructions plus some additional ones.

Figure 3.1 shows the logic pinout of the 8085 microprocessor. All the signals can be classified into six groups: (1) address bus, (2) data bus, (3) control and status signals, (4) power supply and frequency signals, (5) externally initiated signals, and (6) serial I/O ports.



8085 Pinout

ADDRESS BUS

The 8085 has eight signal lines, $A_{15}-A_8$, which are unidirectional and used as the high-order address bus.

MULTIPLEXED ADDRESS/DATA BUS

The signal lines AD_7-AD_0 are bidirectional: they serve a dual purpose. They are used as the low-order address bus as well as the data bus. In executing an instruction, during the earlier part of the cycle, these lines are used as the low-order address bus. During the later part of the cycle, these lines are used as the data bus. (This is also known as multiplexing the bus.) However, the low-order address bus can be separated from these signals by using a latch.

CONTROL AND STATUS SIGNALS

This group of signals includes two control signals (\overline{RD} and \overline{WR}), three status signals (IO/\overline{M} , S_1 , and S_0) to identify the nature of the operation, and one special signal (ALE) to indicate the beginning of the operation. These signals are as follows:

- ☐ ALE—Address Latch Enable: This is a positive going pulse generated every time the 8085 begins an operation (machine cycle); it indicates that the bits on AD_7-AD_0 are address bits. This signal is used primarily to latch the low-order address from the multiplexed bus and generate a separate set of eight address lines, A_7-A_0 .
- ☐ \overline{RD} —Read: This is a Read control signal (active low). This signal indicates that the selected I/O or memory device is to be read and data are available on the data bus.
- ☐ \overline{WR} —Write: This is a Write control signal (active low). This signal indicates that the data on the data bus are to be written into a selected memory or I/O location.
- ☐ IO/\overline{M} : This is a status signal used to differentiate between I/O and memory operations. When it is high, it indicates an I/O operation; when it is low, it indicates a memory operation. This signal is combined with \overline{RD} (Read) and \overline{WR} (Write) to generate I/O and memory control signals.
- ☐ S_1 and S_0 : These status signals, similar to IO/\overline{M} , can identify various operations, but they are rarely used in small systems. (All the operations and their associated status signals are listed in Table 3.1 for reference.)

POWER SUPPLY AND CLOCK FREQUENCY

The power supply and frequency signals are as follows:

- ☐ V_{CC} : +5 V power supply.
- ☐ V_{SS} : Ground Reference.
- ☐ X_1, X_2 : A crystal (or RC, LC network) is connected at these two pins. The frequency is internally divided by two; therefore, to operate a system at 3 MHz, the crystal should have a frequency of 6 MHz.
- ☐ CLK (OUT)—Clock Output: This signal can be used as the system clock for other devices.

8085 MICROPROCESSOR ARCHITECTURE AND MEMORY INTERFACING

TABLE 3.1
8085 Machine Cycle Status and Control Signals

Machine Cycle	Status			Control Signals
	$\text{IO}/\overline{\text{M}}$	S_1	S_0	
Opcode Fetch	0	1	1	$\overline{\text{RD}} = 0$
Memory Read	0	1	0	$\overline{\text{RD}} = 0$
Memory Write	0	0	1	$\overline{\text{WR}} = 0$
I/O Read	1	1	0	$\overline{\text{RD}} = 0$
I/O Write	1	0	1	$\overline{\text{WR}} = 0$
Interrupt Acknowledge	1	1	1	$\overline{\text{INTA}} = 0$
Halt	Z	0	0	$\overline{\text{RD}}, \overline{\text{WR}} = \text{Z and } \overline{\text{INTA}} = 1$
Hold	Z	X	X	
Reset	Z	X	X	

NOTE: Z = Tri-state (high impedance)
X = Unspecified

EXTERNALLY INITIATED SIGNALS, INCLUDING INTERRUPTS

The 8085 has five interrupt signals (see Table 3.2) that can be used to interrupt a program execution. One of the signals, INTR (Interrupt Request), is identical to the 8080A microprocessor interrupt signal (INT); the others are enhancements to the 8080A. The microprocessor acknowledges an interrupt request by the $\overline{\text{INTA}}$ (Interrupt Acknowledge) signal. (The interrupt process is discussed in Chapter 12.)

In addition to the interrupts, three pins—RESET, HOLD, and READY—accept the externally initiated signals as inputs. To respond to the HOLD request, the 8085 has one

TABLE 3.2

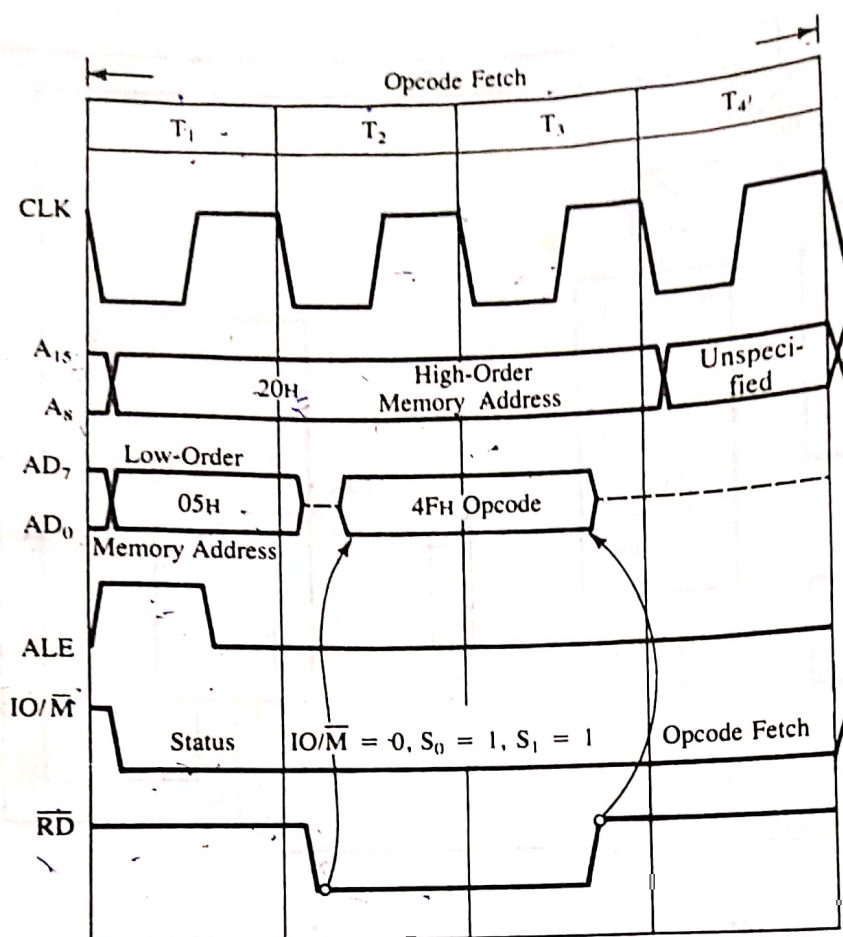


FIGURE 3.3

Timing: Transfer of Byte from Memory to MPU

Step 3: The byte from the memory location is placed on the data bus.

When the memory is enabled, the instruction byte (4FH) is placed on the bus AD_7 – AD_0 and transferred to the microprocessor. The \overline{RD} signal causes 4FH to be placed on bus AD_7 – AD_0 (shown by the arrow), and when \overline{RD} goes high, it causes the bus to go into high impedance.

Step 4: The byte is placed in the instruction decoder of the microprocessor, and the task is carried out according to the instruction.

The machine code or the byte (4FH) is decoded by the instruction decoder, and the contents of the accumulator are copied into register C. This task is performed during the period T_4 in Figure 3.3.

The above four steps are similar to the steps listed in our analogy of the package pickup.

3.13 Demultiplexing the Bus AD_7 – AD_0

The need for demultiplexing the bus AD_7 – AD_0 becomes easier to understand after examining Figure 3.3. This figure shows that the address on the high-order bus (20H) remains on the bus for three clock periods. However, the low-order address (05H) is lost after the

first clock period. This address needs to be latched and used for identifying the memory address. If the bus AD_7-AD_0 is used to identify the memory location (2005H), the address will change to 204FH after the first clock period.

Figure 3.4 shows a schematic that uses a latch and the ALE signal to demultiplex the bus. The bus AD_7-AD_0 is connected as the input to the latch 74LS373. The ALE signal is connected to the Enable (G) pin of the latch, and the Output control (\overline{OC}) signal of the latch is grounded.

Figure 3.3 shows that the ALE goes high during T_1 . When the ALE is high, the latch is transparent; this means that the output changes according to input data. During T_1 , the output of the latch is 05H. When the ALE goes low, the data byte 05H is latched until the next ALE, and the output of the latch represents the low-order address bus A_7-A_0 after the latching operation.

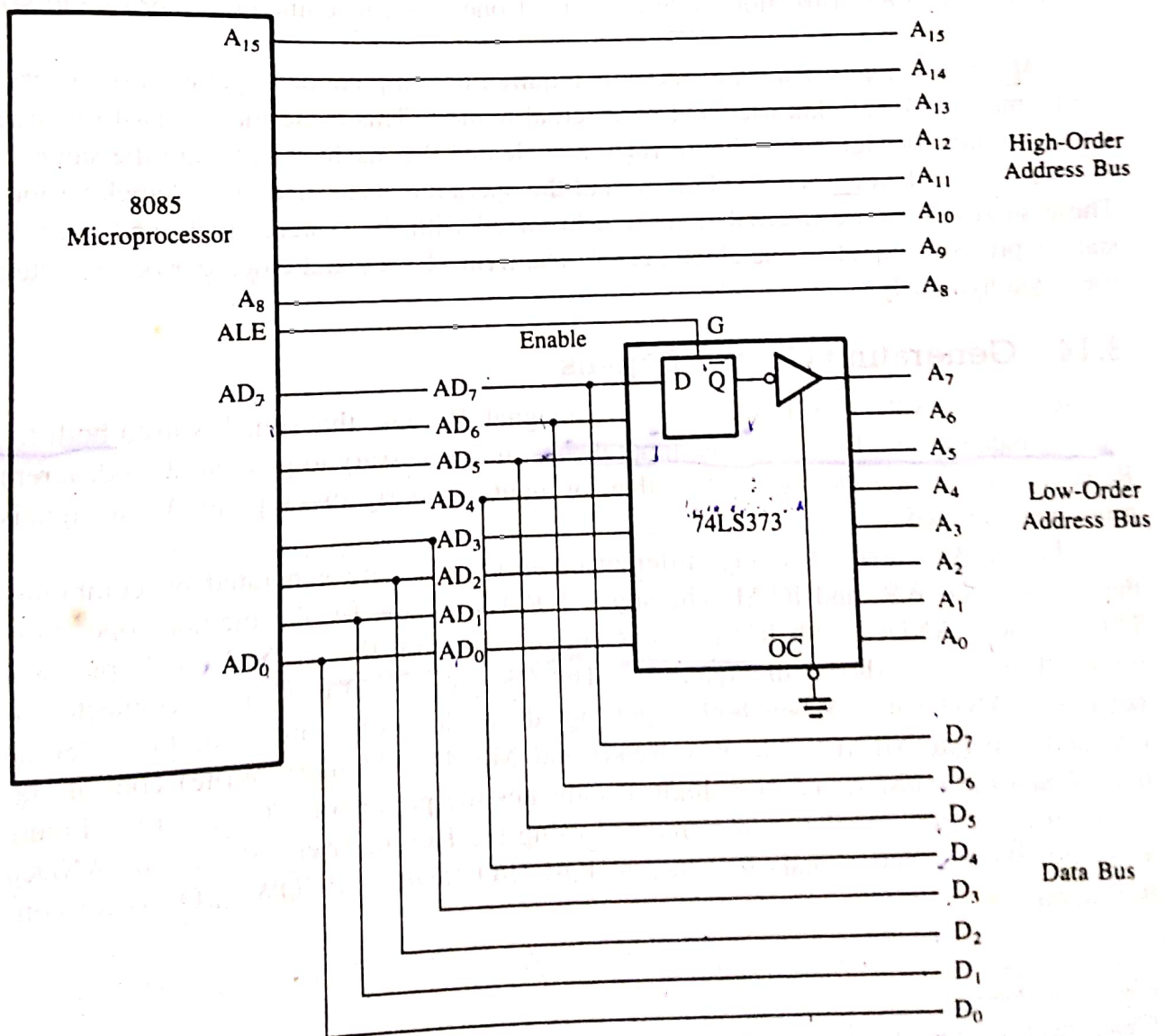


FIGURE 3.4
Schematic of Latching Low-Order Address Bus

Intel has circumvented the problem of demultiplexing the low-order bus by designing special devices such as the 8155 (256 bytes of R/W memory + I/Os), which is compatible with the 8085 multiplexed bus. These devices internally demultiplex the bus using the ALE signal (see Figures 3.18 and 3.19).

After carefully examining Figure 3.3, we can make the following observations:

1. The machine code 4FH (0100 1000) is a one-byte instruction that copies the contents of the accumulator into register C.
2. The 8085 microprocessor requires one external operation—fetching a machine code* from memory location 2005H.
3. The entire operation—fetching, decoding, and executing—requires four clock periods.

Now we can define three terms—instruction cycle, machine cycle, and T-state—and use these terms later for examining timings of various 8085 operations (Section 3.2).

Instruction cycle is defined as the time required to complete the execution of an instruction. The 8085 instruction cycle consists of one to six machine cycles or one to six operations.

Machine cycle is defined as the time required to complete one operation of accessing memory, I/O, or acknowledging an external request. This cycle may consist of three to six T-states. In Figure 3.3, the instruction cycle and the machine cycle are the same.

T-state is defined as one subdivision of the operation performed in one clock period. These subdivisions are internal states synchronized with the system clock, and each T-state is precisely equal to one clock period. The terms T-state and clock period are often used synonymously.

3.14 Generating Control Signals

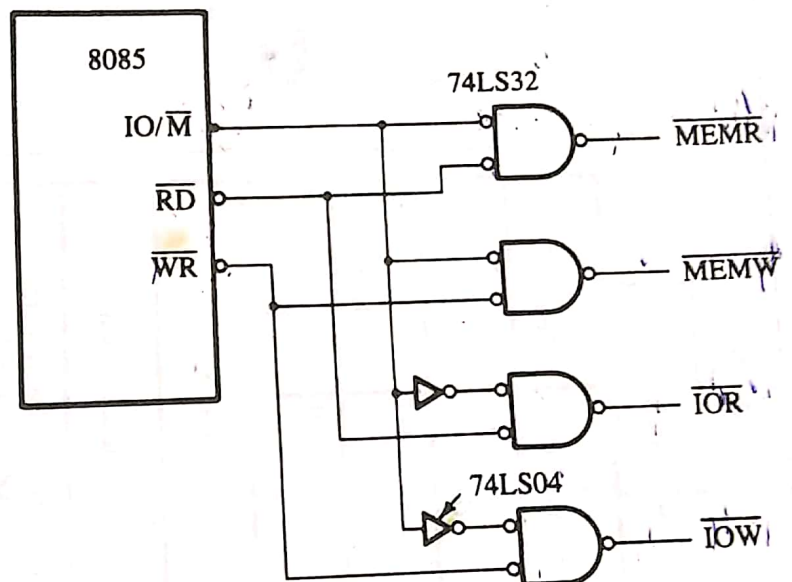
Figure 3.3 shows the \overline{RD} (Read) as a control signal. Because this signal is used both for reading memory and for reading an input device, it is necessary to generate two different Read signals: one for memory and another for input. Similarly, two separate Write signals must be generated.

Figure 3.5 shows that four different control signals are generated by combining the signals \overline{RD} , \overline{WR} , and $\overline{IO/\overline{M}}$. The signal $\overline{IO/\overline{M}}$ goes low for the memory operation. This signal is ANDed with \overline{RD} and \overline{WR} signals by using the 74LS32 quadruple two-input OR gates, as shown in Figure 3.5. The OR gates are functionally connected as negative NAND gates. When both input signals go low, the outputs of the gates go low and generate \overline{MEMR} (Memory Read) and \overline{MEMW} (Memory Write) control signals. When the $\overline{IO/\overline{M}}$ signal goes high, it indicates the peripheral I/O operation. Figure 3.5 shows that this signal is complemented using the Hex inverter 74LS04 and ANDed with the \overline{RD} and \overline{WR} signals to generate \overline{IOR} (I/O Read) and \overline{IOW} (I/O Write) control signals.

8085 MICROPROCESSOR ARCHITECTURE AND MEMORY INTERFACING

FIGURE 3.5

Schematic to Generate Read/Write Control Signals for Memory and I/O



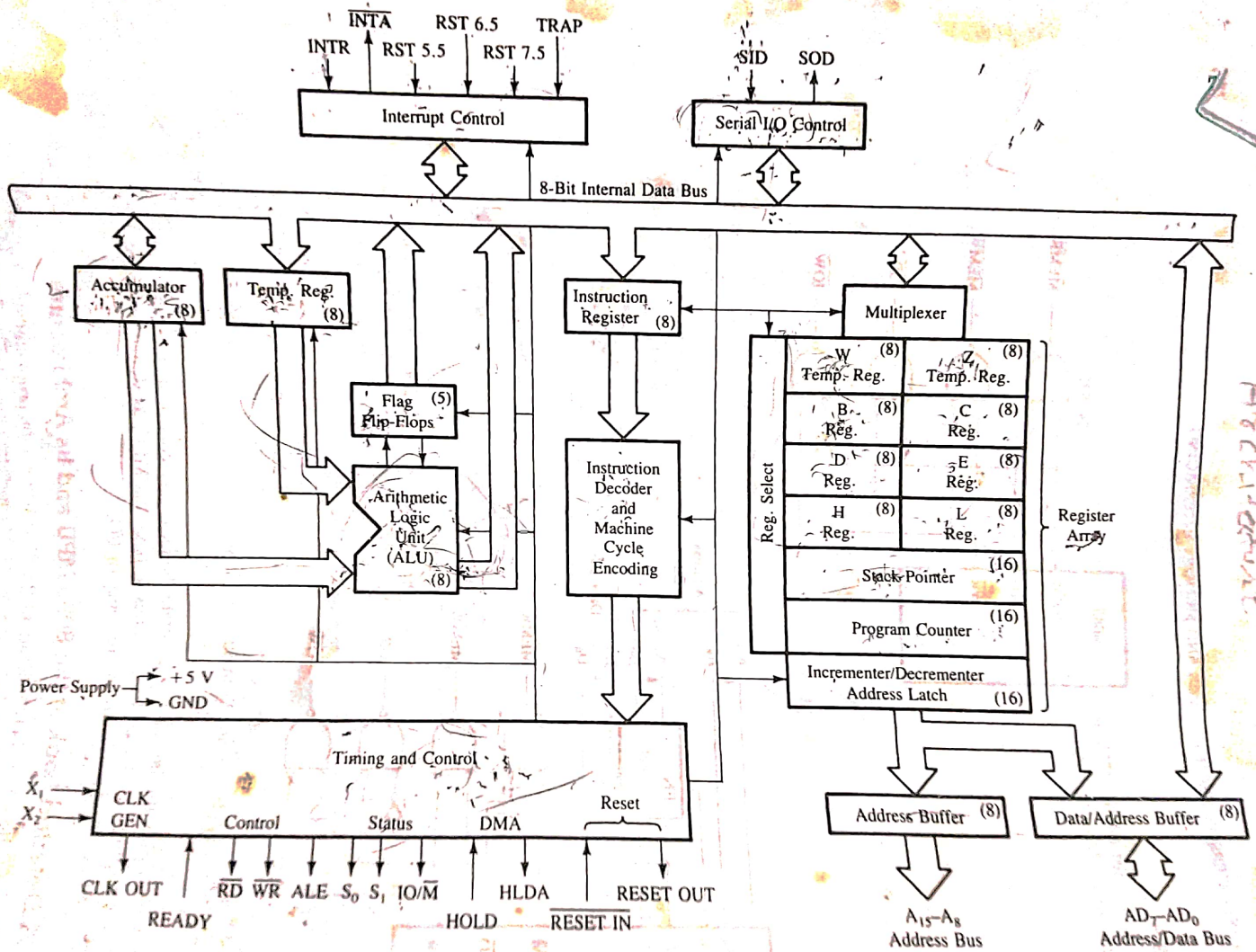


FIGURE 3.7

The 8085A Microprocessor: Functional Block Diagram

NOTE: The 8085A microprocessor is commonly known as the 8085.

Control Unit, Instruction Register and Decoder, Register Array, Interrupt Control, and Serial I/O Control. We will discuss the first four units below; the last two will be discussed later in the book.

THE ALU

The arithmetic/logic unit performs the computing functions; it includes the accumulator, the temporary register, the arithmetic and logic circuits, and five flags. The temporary register is used to hold data during an arithmetic/logic operation. The result is stored in the accumulator, and the flags (flip-flops) are set or reset according to the result of the operation.

The flags are affected by the arithmetic and logic operations in the ALU. In most of these operations, the result is stored in the accumulator. Therefore, the flags generally reflect data conditions in the accumulator—with some exceptions. The descriptions and conditions of the flags are as follows:

- **S—Sign flag:** After the execution of an arithmetic or logic operation, if bit D_7 of the result (usually in the accumulator) is 1, the Sign flag is set. This flag is used with signed numbers. In a given byte, if D_7 is 1, the number will be viewed as a negative number; if it is 0, the number will be considered positive. In arithmetic operations with signed numbers, bit D_7 is reserved for indicating the sign, and the remaining seven bits are used to represent the magnitude of a number. (See Appendix A2 for a discussion of signed numbers.)
- **Z—Zero flag:** The Zero flag is set if the ALU operation results in 0, and the flag is reset if the result is not 0. This flag is modified by the results in the accumulator as well as in the other registers.
- **AC—Auxiliary Carry flag:** In an arithmetic operation, when a carry is generated by digit D_3 and passed on to digit D_4 , the AC flag is set. The flag is used only internally for BCD (binary-coded decimal) operations and is not available for the programmer to change the sequence of a program with a jump instruction.
- **P—Parity flag:** After an arithmetic or logical operation, if the result has an even number of 1s, the flag is set. If it has an odd number of 1s, the flag is reset. (For example, the data byte 0000 0011 has even parity even if the magnitude of the number is odd.)
- **CY—Carry flag:** If an arithmetic operation results in a carry, the Carry flag is set; otherwise it is reset. The Carry flag also serves as a borrow flag for subtraction.

The bit positions reserved for these flags in the flag register are as follows:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
S	Z		AC		P		CY

Among the five flags, the AC flag is used internally for BCD arithmetic; the instruction set does not include any conditional jump instructions based on the AC flag. Of the remaining four flags, the Z and CY flags are those most commonly used.

TIMING AND CONTROL UNIT

This unit synchronizes all the microprocessor operations with the clock and generates the control signals necessary for communication between the microprocessor and peripherals.

MICROPROCESSOR-BASED SYSTEMS: HARDWARE AND INTERFACING

The control signals are similar to a sync pulse in an oscilloscope. The \overline{RD} and \overline{WR} signals are sync pulses indicating the availability of data on the data bus.

INSTRUCTION REGISTER AND DECODER

The instruction register and the decoder are part of the ALU. When an instruction is fetched from memory, it is loaded in the instruction register. The decoder decodes the instruction and establishes the sequence of events to follow. The instruction register is not programmable and cannot be accessed through any instruction.

REGISTER ARRAY

The programmable registers were discussed in the last chapter. Two additional registers, called temporary registers W and Z, are included in the register array. These registers are used to hold 8-bit data during the execution of some instructions. However, because they are used internally, they are not available to the programmer.

3.16 Decoding and Executing an Instruction

Decoding and executing an instruction after it has been fetched can be illustrated with the example from Section 3.12.