

Unit 8 : Microprocessor Architecture

Lesson 1 : Microcomputer Structure

1.1. Learning Objectives

On completion of this lesson you will be able to :

- ◆ draw the block diagram of a simple computer
- ◆ understand the function of different units of a microcomputer
- ◆ learn the basic operation of microcomputer bus system.

1.2. Digital Computer

A digital computer is a multipurpose, programmable machine.

A digital computer is a multipurpose, programmable machine that reads binary instructions from its memory, accepts binary data as input and processes data according to those instructions, and provides results as output.

1.3. Basic Computer System Organization

Basic computer system organization.

Every computer contains five essential parts or units. They are

- i. the arithmetic logic unit (ALU)
- ii. the control unit
- iii. the memory unit
- iv. the input unit
- v. the output unit.

1.3.1. The Arithmetic and Logic Unit (ALU)

The arithmetic and logic unit (ALU) is that part of the computer that actually performs arithmetic and logical operations on data.

The arithmetic and logic unit (ALU) is that part of the computer that actually performs arithmetic and logical operations on data. All other elements of the computer system - control unit, register, memory, I/O - are there mainly to bring data into the ALU to process and then to take the results back out.

An arithmetic and logic unit and, indeed, all electronic components in the computer are based on the use of simple digital logic devices that can store binary digits and perform simple Boolean logic operations. Data are presented to the ALU in registers. These registers are temporary storage locations within the CPU that are connected by signal paths of the ALU.

The ALU is the area of the computer in which arithmetic and logic operations are performed on data. The type of operation that is to be performed is determined by signals from the control unit. The data that are to be operated on by the ALU can come from either the memory unit or the input unit. Results of operations performed in the ALU can be transferred to either the memory unit for storage or the output unit.

1.3.2. Control Unit

Control unit is like the conductor of an orchestra, who is responsible for keeping each of the orchestra members in proper synchronization. This unit contains logic and timing circuits that generate the proper signals necessary to execute each instruction in a program.

The control unit fetches an instruction from memory by sending an address and read command.

The control unit fetches an instruction from memory by sending an address and read command. The instruction word stored at the memory location is then transferred to the control unit. This instruction word, which is in some form of binary code, is then decoded by logic circuitry in the control unit to determine which instruction is being called for. The control unit uses this information to send the proper signals to the rest of the units in order to execute the specified operation.

This sequence of fetching an instruction code and then executing the indicated operation is repeated over and over by the control unit.

1.3.3. Memory Unit

The memory section usually consists of a mixture of RAM and ROM.

The memory section usually consists of a mixture of RAM and ROM. It may also have magnetic floppy disks, magnetic hard disk or laser optical disks, magnetic hard disk or laser optical disks. Memory has two purposes. The first purpose is to store the instructions (Program) that the computer is to perform. The second purpose of the memory is to store the data that are to be operated on by the program. The memory stores these instruction and data as groups of binary digits (words).

Bus

Bus

Two different-size arrows are used; the larger arrows represent data or information that actually consists of a relatively large number of parallel lines, and the smaller arrows represent control signals that are normally only one or a few lines. The various arrows are also numbered to allow easy reference to them in the following descriptions. These components are common communication path called a bus.

Address Bus

Address bus is a unidirectional bus.

This is a unidirectional bus, because information flows over it in only one direction, from the CPU to the memory or I/O elements. The CPU alone can place logic levels on the lines of the address bus, thereby generating $2^{16} = 65,536$ different possible addresses. Each of these addresses corresponds to one memory location or one I/O element.

When the CPU wants to communicate (read or write) with a certain memory location or I/O device, it places the appropriate 16-bit address code on its 16 address pin outputs, A_0 through A_{15} , and onto the address bus. These address bits are then decoded to select the desired memory location or I/O device.

Data Bus

Data bus is a bi-directional bus.

This is a bi-directional bus, because data can flow to or from the CPU. The CPU's eight data pins, D_0 through D_7 , can be either inputs or outputs, depending on whether the CPU is performing a read or a write operation. During data bus by the memory or I/O element. During a write operation the CPU's data pins act as outputs and place data on the data bus, which are then sent to the selected memory or I/O element. In all cases, the transmitted data words are 8bits wide because the CPU handles 8-bit data words, making this an 8-bit μC .

Control Bus

Control Bus

This is the set of signals that is used to synchronize the activities of the separate μC elements. Some of these control signals, such as RD and WR are sent by the CPU to the other elements to tell them what type of operation is currently in progress. The I/O elements can send control signals to the CPU. An example is the reset input (RES) of the CPU which, when driven LOW, causes the CPU to reset to a particular starting state.

Memory Unit

Operation of the memory is controlled by the control unit which signals for either a read or a write operation. A given location in memory is accessed by the control unit that provides the appropriate address code. Information can be written into the memory from the ALU or the input unit, again under control of the control unit. Information can be read from memory into the ALU or into the output unit.

1.3.4. Input Unit

The input unit consists of all of the devices used to take information and data that are external to the computer and put them into the memory unit or the ALU. The control unit determines where the input information is sent. The input unit is used to enter the program and data into the memory unit or into the ALU from an external device during the execution of a program. Some of the common input devices are keyboards, Joysticks, mouse, OCR, OMR etc.

1.3.5. Output Unit

The output unit consists of the devices used to transfer data and information from the computer to the “outside world.” The output devices are directed by the control unit and can receive data from memory or the ALU. Examples of common output devices are printers, disk or tape units, video monitors, and digital-to-analog converters (DACs).

Input Unit
Output Unit
Interfacing

1.3.6. Interfacing

The devices that make up the input and output units are called peripherals because they are external to the rest of the computer. The most important aspect of peripherals involves interfacing. Computer interfacing is specifically defined as transmitting digital information between a computer and its peripherals in a compatible and synchronized way.

Many input/output devices are not directly compatible with the computer because of differences in such characteristics as operating speed, data format (e.g., BCD, ASCII, binary), data transmission mode (e.g., serial, parallel), and logic signal level. Such I/O devices require special interface circuits which allow them to communicate with the control, memory, and ALU portions of the computer system. A common example is the video display terminal (VDT), which can operate both as an input and an output device. The VDT transmits and receives data serially (one bit at a time) while most computers handle data in parallel form. Thus, a VDT requires interface circuitry in order to send data to or receive data from a computer.

Central Processing Unit (CPU)

Central Processing Unit.

In the ALU and control units are combined into one unit called the central processing unit (CPU). This is commonly done to separate the actual “brains” of the computer from the other units. In a microcomputer the CPU is usually implemented on a single chip silicon wafer.

1.4. Exercise

1.4.1. Multiple choice questions

- a) VDT stands for
 - i) Video processing unit
 - ii) Video display terminal
 - iii) Input output terminal
 - iv) all of the above.

- b) A digital computer is
 - i) a programmable machine
 - ii) a digital machine
 - iii) a digital calculator
 - iv) none of the above.

1.4.2. Questions for short answers

- a) What is the CPU?
- b) What is meant by interfacing in a computer system?
- c) What basic operations occur repeatedly in computer?

1.4.3. Analytical questions

- a) Name the five basic units of a computer and describe the major functions of each.
- b) Describe about the computer interfacing.

Lesson 2 : Microprocessor Architecture

2.1. Learning Objectives

On completion of this lesson you will be able to :

- ◆ define the address bus, the data bus, and the control bus, and explain their functions in reference to the 8085/8080A microprocessor.
- ◆ list the registers in the 8085/8080A microprocessor, and explain their functions.

The microprocessor is usually a single IC that contains all of the circuitry of the control and arithmetic-logic units-in other words, the CPU.

It is important to understand the difference between the microcomputer (μC) and the microprocessor (μP). A microcomputer contains several elements, the most important of which is the microprocessor. The microprocessor is usually a single IC that contains all of the circuitry of the control and arithmetic-logic units-in other words, the CPU. It is common to refer to the microprocessor as the MPU (microprocessor unit), since it is the CPU (central processing unit) of the microcomputer.

The microprocessor unit of the computer consists of various registers to store data, the arithmetic logic unit (ALU) to perform arithmetic and logical operations, instruction decoders, counters, and control lines. The CPU reads instructions from the memory and performs the tasks specified. The CPU communicates with input/ output devices to accept or to send data. The input and output devices are known also as peripherals. The CPU is the primary and central player in communicating with various devices such as memory, input, and output; however, the timing of the communication process is controlled by the group of circuits called the control unit.

2.2. Microprocessor

A microcomputer is a computer similar to any other computer, except that the CPU functions of the microcomputer are performed by the microprocessor.

The microprocessor is one component of the microcomputer. A microcomputer is a computer similar to any other computer, except that the CPU functions of the microcomputer are performed by the microprocessor. Similarly, the term peripheral is used for input/ output devices; however, occasionally memory is also included in this term. The various components of the microcomputer and their function are described in the following paragraphs.

The MPU (microprocessor unit) is the heart of every microcomputer. It performs a number of functions, including.

1. Providing timing and control signals for all elements of the μC
2. Fetching instructions and data from memory
3. Transferring data to and from memory and I/O devices

The MPU microprocessor unit.

4. Decoding instructions
5. Performing arithmetic and logic operations called for by instructions
6. Responding to I/O generated control signals such as RESET and INTERRUPT.

The MPU contains all of the logic circuitry for performing these functions, but its internal logic is generally not externally accessible. Instead, we can control what happens inside the MPU by the program of instructions that we put in memory for the MPU to execute. This is what makes the MPU so versatile and flexible - when we want to change its operation, we simply change the programs stored in RAM (software) or ROM (firmware) rather than rewire the electronics (hardware).

The microprocessor is a semiconductor device consisting of electronic logic circuits manufactured by using either a large-scale (LSI) or very-large-scale integration (VLSI) technique. The microprocessor is capable of performing computing functions and making decisions to change the sequence of program execution. In large computers, the CPU performs these computing functions and it is implemented on one or more circuit boards. The microprocessor is in many ways similar to the CPU; however, the microprocessor includes all the logic circuitry (including the control unit) on one chip. For clarity, the microprocessor can be divided into three segments, arithmetic/logic unit (ALU), register unit, and control unit.

Very-large-scale integration (VLSI)

- ◆ **Arithmetic and Logic Unit** : In this area of the microprocessor, computing functions are performed on data. The CPU performs arithmetic operations such as addition and subtraction, and logic operations such as AND, OR, and exclusive OR. Results are stored either in register or in memory or sent to output devices.
- ◆ **Register Unit** : This area of the microprocessor consists of various registers. The registers are used primarily to store data temporarily during the executing of a program. Some of the registers are accessible to the user through instructions.
- ◆ **Control Unit** : The control unit provides the necessary timing and control signals to all the operations in the microcomputer. It controls the flow of data between the microprocessor and peripherals (including memory).

Input

The input section transfers data and instructions in binary from the outside world to the microprocessor. It includes devices such as keyboards, teletypes, and analog-to-digital converters. Typically, a microcomputer includes a keyboard as an input device. The keyboard has sixteen data keys (0 to 9 and A to F) and some additional function keys to perform operations such as storing data and executing programs.

The input section transfers data and instructions in binary from the outside world to the microprocessor.

Output

The output section transfers data from the microprocessor to output devices Output.

The output section transfers data from the microprocessor to output devices such as light emitting diodes (LEDs), cathode-ray-tubes (CRTs), printers, magnetic tape, or another computer. Typically, single-board computers include LEDs and seven-segment LEDs as output devices.

Memory

Memory stores binary information such as instructions and data, and provides that information to the microprocessor whenever necessary.

Memory stores binary information such as instructions and data, and provides that information to the microprocessor whenever necessary. To execute programs, the microprocessor reads instructions and data from memory and performs the computing operations in its ALU section. Result are either transferred to the output section for display or stored in memory for later use. The memory block has two sections : Read - Only Memory (ROM) and Read / Write Memory (R/WM), popularly known as Random Access Memory (RAM).

The ROM is used to store programs that do not need alterations. The monitor program of a single - board microcomputer is generally stored in the ROM. Program stored in the ROM can only be read; they cannot be altered.

The Read / Write memory (R/WM) is also known as user memory. It is used to store user programs and data. The information stored in this memory can be read and altered easily.

System Bus

The system bus is a communication path between the microprocessor and the peripherals.

The system bus is a communication path between the microprocessor and the peripherals; it is nothing but a group of wires that carries bits. The microcomputer bus is in many ways similar to a one-track, express subway, the microcomputer bus carries bits between the microprocessor and only one peripheral at a time. The same bus is time - shared to communicate with various peripherals, with the timing provided by the control section of the microprocessor.

The Bus System

The μC has three buses which carry all the information and signals involved in the system operation.

The μC has three buses which carry all the information and signals involved in the system operation. These buses connect the microprocessor (CPU) to each of the memory and I/O elements so that and information can flow between the CPU and any of these other elements. The buses involved in all the data transfers have functions that are described as follows.

2.3. How does the Microcomputer Work?

At first program and data are entered in the R/W memory. The program includes binary instructions to add given data and to display the answer at the monitor. When the microcomputer is given a command to execute the program, it reads and executes one instruction at a time and finally sends the result to the display.

The microprocessor fetches the first instruction from its memory sheet, decodes it, and executes that instruction.

This process of program execution can best be described by comparing it to the process of assembling a radio kit. The instructions for assembling the radio are printed on a sheet of paper in sequence. One reads the first instruction, then picks up the necessary components of the radio and performs the task. The sequence the instructions are stored sequentially in the memory. The microprocessor fetches the first instruction from its memory sheet, decodes it, and executes that instruction. The sequence of fetch, decode, and execute is continued until the microprocessor comes across the instruction, Stop. During the entire process, the microprocessor uses the system bus to fetch the binary instructions and data from the memory. It uses registers from the register section to store data temporarily, and it performs the computing function in the ALU section. Finally it sends out the result in binary, using the same bus lines, to the monitor.

This lesson describes the internal architecture and various operations of the microprocessor in the context of the 8085/8080A. It also expands on topics such as memory and I/O .

2.4. Microprocessor Architecture and its Operations

The microprocessor is a programmable logic device, designed with registers, flip-flops.

The microprocessor is a programmable logic device, designed with registers, flip-flops. The microprocessor has a set of instructions designed internally, to manipulate data and communicate with peripherals. This process of data manipulation and communication is determined by the logic design of the microprocessor, called the architecture.

All the various functions performed by the microprocessor can be classified in three general categories.

The microprocessor can be programmed to perform functions on given data by selecting necessary instructions from its set. These instructions are given to the microprocessor by writing them into its memory. Writing (Or entering) instruction and data is done through an input device such as a keyboard. The microprocessor reads or transfers each instruction one at a time, matches it with its instruction set, and performs the data manipulation indicated by the instruction. The result can be stored in memory or sent to such output devices as LEDs or a CRT terminal. In addition, the microprocessor can respond to external signals. Its can be interrupted, reset, or asked to wait to synchronize with

slower peripherals. All the various functions performed by the microprocessor can be classified in three general categories :

- ◆ Microprocessor-initiated operations
- ◆ Internal data operations
- ◆ Peripheral (or externally) initiated operations.

To perform these functions, the microprocessor requires a group of logic circuits and a set of signals called control signals. However, early processors do not have the necessary circuitry on one chip; the complete units are made up more than one chip. Therefore, the term Micro Processing Unit (MPU) is defined here as a group of devices that can perform these functions with the necessary set of control signals. This term is similar to the term Central Processing Unit (CPU). However, later microprocessors include most of the necessary circuitry to perform these operations on a single chip. Therefore, the terms MPU and microprocessor often are used synonymously.

The microprocessor functions are explained here in relation to the 8085 or 8080A MPU.

2.5. Microprocessor-Initiated Operations and 8085/8080A Bus Organization

The MPU performs primarily four operations.

The MPU performs primarily four operations.

1. Memory Read: Reads data from memory.
2. Memory Write: Writes data into memory.
3. I/O read: Accepts data from input devices.
4. I/O Write: Sends data to output devices.

All these operations are part of the communication process between the MPU and peripheral devices (including memory). To communicate with a peripheral (or a memory location), the MPU needs to perform the following steps:

- Step 1 : Identify the peripheral or the memory location (with its address).
Step 2 : Transfer data
Step 3 : Provide timing or synchronization signals.

The 8085/8080A MPU performs these functions using three sets of communication lines called buses : The address bus, the data bus, and the control bus.

2.5.1. Internal Data Operations and the 8085/8080A Registers

Internal data operations and the 8085/8080A registers.

The internal architecture of the 8085/8080A microprocessor determines how and what operations can be performed with the data. These operations are

1. Store 8-bits data.
2. Perform arithmetic and logical operations.
3. Test for conditions.
4. Sequence the execution of instructions.
5. Store data temporarily during execution in the defined R/W memory locations called the stack.

To perform these operations, the microprocessor requires registers, an arithmetic logic unit (ALU) and control logic, and internal buses (path for information flow).

Registers

Registers

The 8085/8080A has six general - purpose registers to perform the first operation listed above, that is, to store 8-bit data during a program execution. These registers are identified as B, C, E, H, and L. They can be combined as register pairs - BC, DE, and HL - to perform some 16-bit operation.

These registers are programmable, meaning that a programmer can use them to load or transfer data from the registers by using instructions. For example, the instruction MOV B, C transfers the data from register C to register B. Conceptually, the registers can be viewed as memory locations, except they are built inside the microprocessor and identified by specific names. Some microprocessors do not have these types of registers; instead, they use memory space as their registers.

Accumulator

*Accumulator
Flags*

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 that are set or reset according to data conditions in the accumulator and other registers. The microprocessor uses them to perform the third operation; namely testing for data conditions.

For example, after an addition of two numbers, if the sum in the accumulator is larger than eight bits, the flip-flop that is used to indicate a carry, carry the Carry flag (CY), is set to one. When an arithmetic operation results in zero, the flip-flop called the Zero flag (Z) is set to one. The 8085/8080A has five flags to indicate five different types of data conditions. They are called Zero (Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags. The most commonly used flags are Zero and Carry; the others will be explained as necessary.

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 implemented to flags cannot be emphasized enough; they will be discussed again in applications of conditional jump instructions.

Program Counter (PC)

Program Counter

This 16-bit register deals with the fourth operation, sequencing the execution of instructions. This register is a memory pointer. Memory locations have 16-bit address, and that is why this is a 16-bit register.

The microprocessor uses this register to sequence the execution of 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.

Stack Pointer (SP)

Stack Pointer

The stack pointer is also a 16-bit register used as a memory pointer; initially, it will be called the stack pointer register to emphasize that it is a register. 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 (register).*

2.6. Exercise

2.6.1. Multiple choice questions

- a) Which one of the following is not the function of memory unit?
 - i) Stores data awaiting to process
 - ii) To store instruction
 - iii) Perform arithmetic operation
 - iv) Stores processed data.

Microprocessor Architecture

- b) The central processing unit which is the ‘brains’ of the computer is the combination of
 - i) the ALU and control unit
 - ii) the input unit and ALU
 - iii) the control unit and bus structure
 - iv) the ALU and bus structure.

- c) A CPU with 16 address lines can address
 - i) 16,384 memory locations
 - ii) 32,768 memory locations
 - iii) 64,000 memory locations
 - iv) 65,536 memory locations.

2.6.2. Questions for short answers

- a) What is meant by interfacing in a computer system?
- b) What is the function of central processing unit?

2.6.3. Analytical question

- a) Draw the typical structure of a micro computer and show its bus organization

Lesson 3 : 8085 Microprocessor Architecture

3.1. Learning Objectives

On completion of this lesson you will be able to :

- ◆ recognize the functions of various pins of the 8085 microprocessor.
- ◆ list the various internal units that make up 8085 architecture, and explain their functions decoding and executing an instruction.
- ◆ draw the block diagram of an 8085 base microcomputer.

3.2. The 8085 MPU

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 term Micro Processing 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).

The 8085 Microprocessor

The 8085 is an 8-bit general purpose microprocessor capable of addressing 64K of memory.

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 8085 is an enhanced version of 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. Programs written for the 8080A will be executed by the 8085, but the 8085 and the 8080A are not pin compatible.

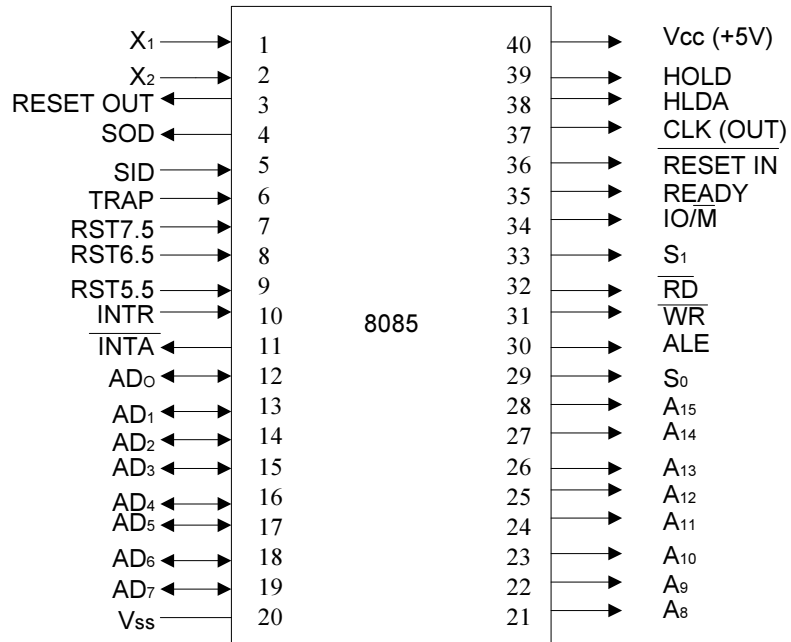


Fig. 8.1 : 8085 microprocessor signals and pin assignments.

Multiplexed Address / Data Bus

Multiplexed address / data bus.

The signal lines AD₇ to AD₀ 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 art 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

Control and status signals.

This group of signals includes two control signals (RD and WR), three status signals (IO/M, S₁ and S₀) to identify the nature of the operation, and one special signal (ALE) to indicate the beginning of the operation. These signals are as follows :

- ◆ ALU - 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 AD₇ - AD₀ 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, A₇ to A₀.

- ◆ 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.
- ◆ WR - Write : This is a Write control signal (active low). This signal indicates that the data on the bus are to be written into a selected memory or I/O location.
- ◆ IO/M : This is a status signal 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 RD (Read) and WR (Write) to generate I/O and memory control signals.
- ◆ S₁ and S₀ : These status signals, similar to IO/M, can identify various operations; but they are rarely used in small systems.

Power Supply and Clock Frequency

Power supply and clock frequency.

The power supply and frequency signals are as follows :

- ◆ V_{CC} : + 5 volt power supply.
- ◆ V_{SS} : Ground Reference.
- ◆ X₁, X₂ : 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 frequency of 6 MHz.
- ◆ CLK (OUT) - Clock Output : This signal can be used as the system clock for other devices.

Interrupts and Externally Initiated Operations

Interrupts and Externally Initiated Operations.

The 8085 has five interrupt signals 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 by the INTA (Interrupt Acknowledge) signal.

In addition to the interrupts, three pins - RESET, HOLD, and READY - accept the externally initiated signals as inputs. To respond to the HOLD request, it has one signal called HLDA (Hold Acknowledge). The RESET is again described below.

- ◆ **RESET IN** : When the signal on this pin goes low, the program counter is set to zero, the buses are tri-stated, and the MPU is reset.
- ◆ **RESET OUT** : This signal indicates that the MPU is being reset. The signal can be used to reset other devices.

Serial I/O Ports

The 8085 has two signals to implement the serial transmission : SID (Serial Input Data) and SOD (Serial Output Data).

3.3. Exercise

3.3.1. Questions for short answers

- a) What are the functions of 8085 microprocessor?
- b) What are control and status signals?

3.3.2. Analytical question

- a) List the internal units of 8085 architecture and explain their function.

Lesson 4 : Addressing Modes

4.1. Learning Objectives

On completion of this lesson you will be able to :

- ◆ learn about microprocessor typical addressing modes and instructions.

In this lesson some important characteristics and properties of microcomputer instruction sets are discussed. Topics also include addressing modes and instruction.

An instruction has two components.

An instruction manipulates the stored data, and a sequence of instructions constitutes a program. In general, an instruction has two components:

- ◆ Op-code field
- ◆ Address field

The purpose of the address field is to indicate the data address.

The op-code field specifies how data are to be manipulated. The data items may reside within a microprocessor register or in the main memory. The purpose of the address field is to indicate the data address. When operations require data to be read from or stored into two or more addresses, the address field may contain more than one address. For example, consider the following instruction:

ADD	R1, R0
op-code field	address field

Assume that the microcomputer uses R1 as the source register and R0 as the destination register. The preceding instruction then adds the contents of registers R0 and R1 and saves the sum in register R0. The number and types of instructions supported by a microcomputer vary from one microcomputer to another and depend primarily on the architecture of a particular machine.

Depending on the number of addresses specified, one can have the following instruction formats:

- ◆ Three-address
- ◆ Two-address
- ◆ One-address
- ◆ Zero-address.

4.2. Addressing Modes

The sequence of operations that a processor has to carry out while executing an instruction is called its instruction cycle. One of the activities in an instruction cycle is the determination of the addresses of the operands involved in that instruction.

The way in which a processor accomplishes this task is called the addressing mode.

Most common addressing techniques are :

Most common addressing techniques are.

- ◆ Inherent Addressing
- ◆ Immediate Addressing
- ◆ Absolute Addressing
- ◆ Register Addressing
- ◆ Register Indirect Addressing

4.2.1. Inherent Addressing

An instruction is said to have an inherent addressing mode if its op-code indicates the address of the operand, which is usually the contents of a register. For example, consider the following instruction :

*Inherent Addressing
Immediate Addressing*

STC : Set the carry flag in the status register. Since the op-code implies the address of the operand, the processor does not have to compute the operand address. This mode is very common with 8-bit microprocessors such as the 8085, Z80 and MC6809.

4.2.2. Immediate Addressing

Whenever an instruction contains the operand value, it is called an immediate mode instruction. For example, consider the following instruction :

ADD # 25 , R1 ; $R1 \leftarrow R1 + 25$

In this instruction, the symbol # indicates that it is an immediate-mode instruction. This convention is adopted in the assemblers for processors. In these systems, the machine representation of this instruction occupies two consecutive memory words. The first word holds the op-code, whereas the next word holds the data value. In this example, data value is 25. To execute this instruction, the processor has to access memory twice.

The advantage of immediate addressing is that no memory reference other than the instruction fetch is required to obtain the operand, thus saving one memory in the instruction cycle. The disadvantage is that the size of the number is restricted to the size of the address field, which, in most instruction sets, is small compared with the word length.

4.2.3. Absolute Addressing

An instruction is said to have an absolute addressing mode if it contains the address of the operand.

An instruction is said to have an absolute addressing mode if it contains the address of the operand. For example, consider the following move instruction :

MOVE 5000 , R2 ; R2 ← [5000]

This instruction copies the contents of memory location 5000 in the register R2.

4.2.4. Register Addressing

An instruction is said to have a register mode if it contains a register address as opposed to a memory address.

An instruction is said to have a register mode if it contains a register address as opposed to a memory address. For example, consider the following register mode and instruction :

ADD R2 , R3 ; R3 ← R2 + R3

4.2.5. Register Indirect Addressing

Whenever an instruction specifies a register that holds the address of an operand, the resulting addressing mode is known as the register indirect mode.

Whenever an instruction specifies a register that holds the address of an operand, the resulting addressing mode is known as the register indirect mode. From this definition, it follows that the EA of an operand in the register-indirect mode is the contents of the register R. More formally, this result is written as follows:

$$EA = [R]$$

To illustrate this idea clearly, consider the following instruction:

MOVE (R2) , (R3) ; [R3] ← [R2]

Assume that the following configuration exists :

[R2]	=	5000 ₁₆
[R3]	=	4000 ₁₆
[5000]	=	1256 ₁₆
[4000]	=	4629 ₁₆

This instruction copies the contents of the memory location, whose address is specified by the register R2, into the location whose address is specified by the register R3. Thus, after the execution of this instruction, the memory location 4000 will contain the value 1256.

4.3. Instruction Types

Instructions available in a processor may be broadly classified into five groups.

In general, instructions available in a processor may be broadly classified into five groups :

- ◆ Data transfer instructions
- ◆ Arithmetic instructions
- ◆ Logical instructions
- ◆ Program control instructions
- ◆ I/O instructions.

Data transfer instructions are primarily concerned with data transfers between the microprocessor registers to between register and memory. An example is MOVE R0, R1 which transfers the contents of register R0 to register R1.

Typical arithmetic instructions include ADD and SUBTRACT instructions. For example, ADD R0, R1 adds the contents of R0 to R1 and stores the results in R1.

Logical instructions perform Boolean AND, OR, NOT and EXCLUSIVE-OR operations on a bit-by-bit basis. An example is OR R0, R1 which logically ORes the contents of R0 with R1 and places the result in R1.

Typical program control instructions include unconditional and conditional branch and subroutine CALL instructions. For example, JMP 2035H unconditionally branches to the 16-bit address 2035H.

I/O instructions perform input and output operations. An example is IN PORTA which inputs the contents of an I/O port called port A into a microprocessor register such as the accumulator.

4.4. Exercise

4.4.1. Questions for short answers

- a) What are the components of an instruction?
- b) What are the types of instruction formats?

4.4.2. Analytical question

- a) Discuss different types of addressing mode with example.

