24 MICROCOMPUTERS AND MICROPROCESSORS

Learning objectives

After reading this chapter, the reader will be able to

- Understand the history and structure of microcomputer
- Identify the basic architecture of microprocessor
- Realize the bus structure of the 8085
- Use the instruction set of 8085
- Enter a program into the 8085 microprocessor
- Understand the difference between 8085 and 8086 microprocessors

24.1 EVOLUTION OF MICROPROCESSOR

A microcomputer is a small computer with a microprocessor as its Central Processing Unit (CPU). It includes a microprocessor, memory, and interfacing facilities to communicate with the peripheral devices as shown in Figure 24.1.The communication of the data from the CPU to memory and peripheral devices is done through address bus and data bus. The control signals are given through the control bus. The basic building block of the microcomputer is the microprocessor.

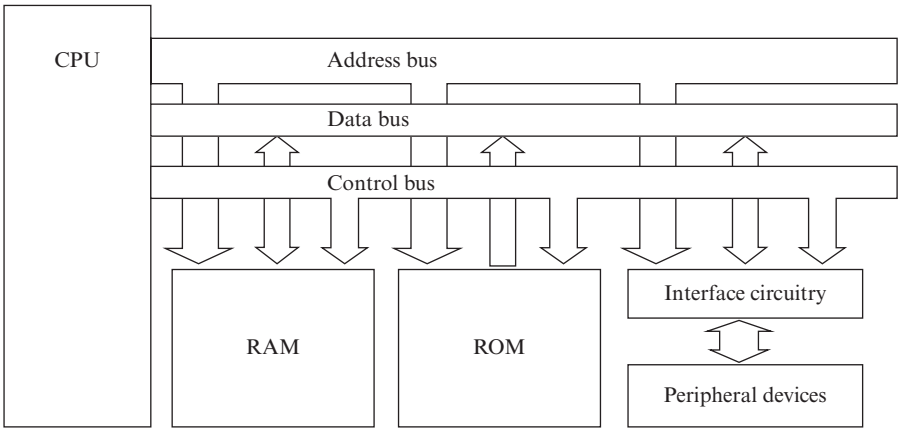


Figure 24.1 *Block diagram of the microcomputer*

Microprocessors are commonly classified by the number of bits that their Arithmetic Logic Unit (ALU) can work with at a time. The first commercial microprocessor, Intel 4004, was produced in 1971 and was known as 4-bit processor. It contained 2300 PMOS transistors and was mostly used with calculators. An 8-bit microprocessor, 8008, was produced in 1972, but it required 20 or more additional devices to form a functional CPU. Another version of the 8-bit microprocessor that has much larger instruction set than 8008 was introduced by Intel in 1974. The improvement was that the newer chip required only two additional devices to form a functional CPU. Motorola also developed MC6800 as 8-bit general purpose CPU. The advantage is that it requires only +5V power supply. In 1977, Intel produced 8085 microprocessor. It was implemented with 6200 transistors on a single chip using NMOS devices. Subsequently, during 1978 to 1982, Intel introduced 16-bit-8086 series microprocessors. The 80186/80286 microprocessors were used for embedded control applications. These processors were implemented with NMOS technology that was faster than PMOS. Later, during 1985 to 1988, 32-bit microprocessors 80386/80486, which are capable of directly accessing up to 4 GB of memory, were developed. Additional features such as multiuser and multitasking facilities were incorporated in the 32-bit processors. In addition, 80486 has parallel execution capability with five pipeline stages. It also has built-in math coprocessor and 8 KB code and data cache. Pentium produced a processor in 1993, which has superscalar, super-pipelined architecture. It has two pipelines where each one is a four-stage pipeline. Then, Pentium Pro P-I, P-II, P-III, and P-IV are developed, which are 64-bit microprocessors.

24.2 SALIENT FEATURES OF 8085 MICROPROCESSOR

(i) 8085 microprocessor can read or write or perform arithmetic and logical operations on 8-bit data at a time.
(ii) It is a single-chip NMOS device implemented with 6200 transistors.
(iii) It requires +5V dc power supply.
(iv) It provides on chip clock generator.
(v) Maximum clock frequency is 3 MHz and minimum clock frequency is 500 kHz.
(vi) It provides 74 instructions with five addressing modes.
(vii) It provides five hardware interrupts and eight software interrupts.
(viii) It has 8 data lines and 16 address lines; hence, the capacity is 2^{16} = 64 KB of memory.
(ix) It can generate 8-bit I/O address, therefore 2^8 = 256 input and 256 output ports can be accessed.
(x) It provides two serial I/O lines SID and SOD so that serial peripherals can be interfaced directly with 8085 microprocessor. SID stands for serial input data, where the data enters serially and SOD stands for serial output data, where the data leaves the microprocessor serially.

24.3 8085 BUS STRUCTURE

The basic operations carried out by the microprocessor unit are memory read/write and I/O read/write. To perform these tasks, microprocessor needs three sets of communication lines called "buses" to communicate with memory and other devices (input and output). These are as follows:

(i) Address Bus
(ii) Data Bus
(iii) Control Bus

The bus structure for 8085 microprocessor is shown in Figure 24.2

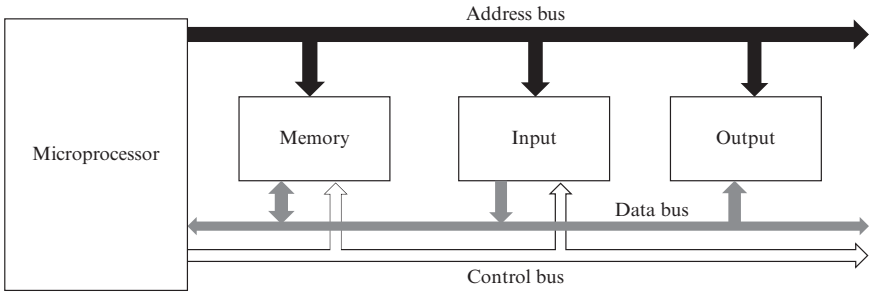


Figure 24.2 *8085 Bus structure*

Address Bus It is a unidirectional bus in which the information flows in only one direction, that is, from microprocessor to the memory or I/O elements. For an 8-bit microprocessor such as 8085, this is a group of 16 lines (16-bit long), that is, CPU can generate 2^{16}= 65,536 ≈ 64 k different possible addresses on this bus, that is, 0000000000000000 to 1111111111111111. Memory consists of locations, each with a unique address; the size of the address bus determines the size of memory, which can be used. To communicate with memory, the microprocessor sends an address on the address bus, for example, 0000000000000011 (3 in decimal), to the memory. The memory then selects location number 3 for reading or writing data.

Data Bus It is a bidirectional bus as the data in binary form can flow in both the directions between microprocessor and peripherals. This bus is a group of 8 lines (8-bits long). Data bus is used to transmit data, that is, information, results of arithmetic, and so on, between memory and the microprocessor. Since this bus is of 8-bit length, it can transmit 2^8 combinations of binary digits, that is from 00000000 to 11111111 (255 in decimal). The size of the bus therefore limits the number of possible instructions to 256, each specified by a separate number.

Control Bus It carries various lines that have specific functions for coordinating and controlling microprocessor operations, for example, Read/Write line. The control operation indicates whether data are being "written to" or "read from" a memory location or an I/O device. A logic 1 on the line indicates a read operation and a logic 0 indicates a write operation. Typically, microprocessor has 10 control lines. The control bus carries control signals partly unidirectional and partly bidirectional.

24.4 INTERNAL ARCHITECTURE OF 8085 MICROPROCESSOR

Figure 24.3 shows the internal architecture of 8085 microprocessor. The 8085 microprocessor mainly consists of ALU, timing and control unit, register array, interrupt control, instruction register, serial I/O control, and bus system.

24.4.1 Register Array

The register array is an important part of microprocessor's architecture. 8085 microprocessor has six general purpose registers that act as temporary storage devices and are identified as B, C, D, E, H, and L. Each of these registers is capable of holding 8 bits of information. The registers can be paired as B–C, D–E, and H–L to hold 16 bits of information. B, D, and H registers hold the higher order byte and C, E, and L hold the lower order byte respectively. Further, 8085 has a

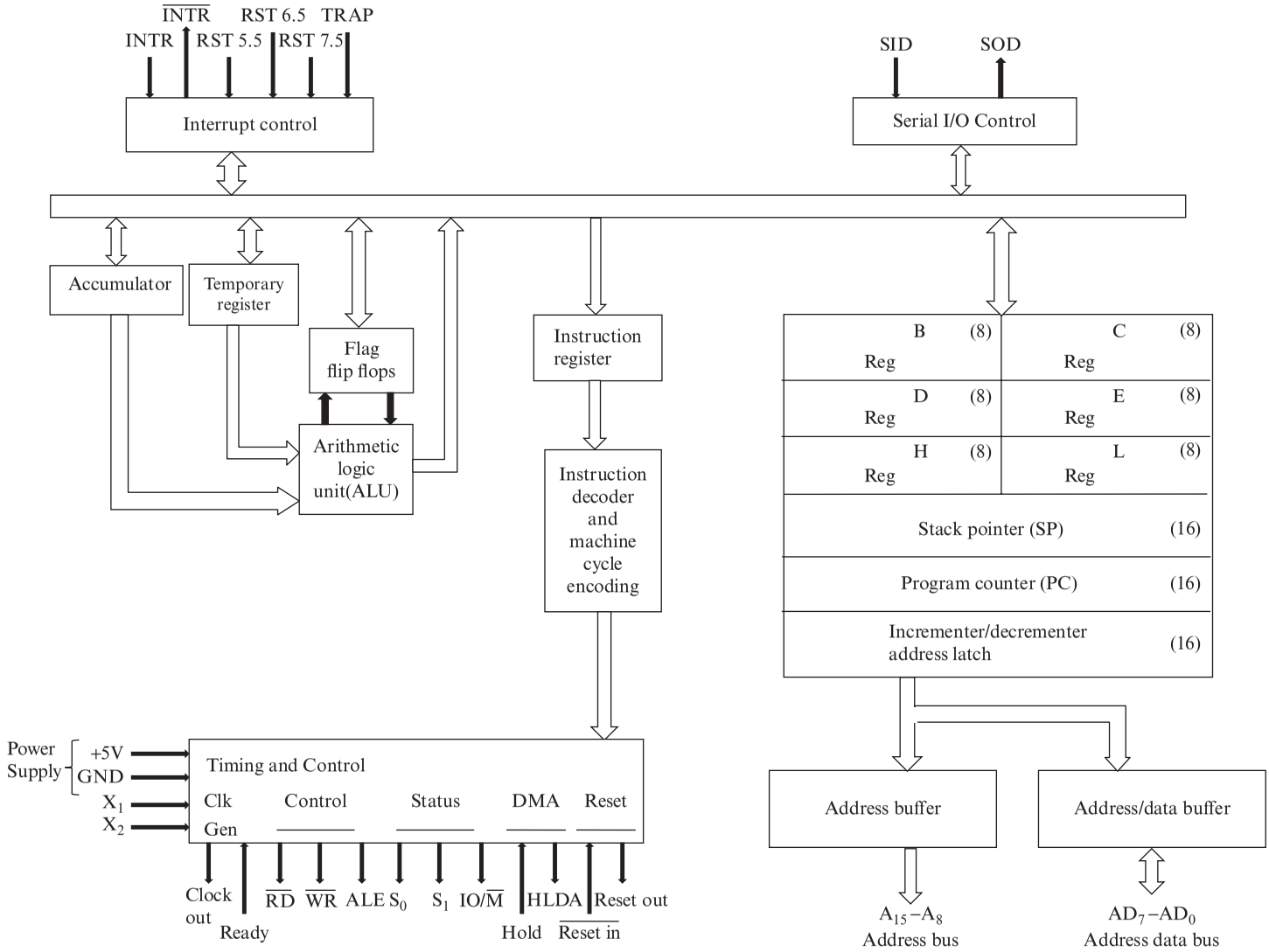


Figure 24.3 *Internal architecture of 8085 microprocessor*

special register called an accumulator (A) in which all the arithmetic and logical operations are performed. The accumulator is capable of holding 8 bits of information. 8085 has a flag register or status flags that are a set of flip-flops used to check the status of accumulator after an arithmetic or logical operation is performed. The various flags are sign flag (S), zero flag (Z), auxiliary carry (AC), parity (P), and carry flag (CY). The microprocessor uses these flags to test data conditions.

Table 24.1 *Status register*

D7	D6	D5	D4	D3	D2	D1	D0
S	Z		AC		P		CY

Table 24.1 shows an 8-bit register, called the flag or status register, adjacent to the accumulator. The flags are stored in the 8-bit register so that the programmer/processor can examine these flags (data conditions) by accessing the register through an instruction. Flags have critical importance in the decision-making process of the microprocessor.

Sign flag: It is used to indicate whether the result of the last operation is positive or negative based on the most significant bit (MSB). If the MSB of the last operation is 1, then this flag is set to 1 and indicates the result as negative; otherwise, it is reset.

Zero flag: This sets to 1 if the result after an arithmetic or logical operation is zero, and it is set to 0 if the result is non-zero.

Carry flag: It indicates the condition of the word size of the result produced by certain arithmetic/logical operation. If the word size of the result is greater than the microprocessor's word size, the flag sets to 1; otherwise it sets to 0.

Auxiliary carry flag: It is generated when a carry is produced by 1 digit and passed on to next digit.

Parity flag: It sets to 1 indicating that the result of the last operation contains either an even number of 1s (even parity) or an odd number of 1s (odd parity).

Instruction register/decoder: The instruction decoder decodes the instruction.

Address buffer and address/data buffer: These are loaded with the contents of the stack pointer or program counter.

Control generator: It generates signals within microprocessor to carry out the instruction that has been decoded.

Stack pointer: 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. It is a last in–first out (LIFO) memory. It always points to the top of the stack.

Program counter: This 16-bit register deals with sequencing the execution of instructions. This register is a memory pointer. The microprocessor uses this 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.

Incrementer/decrementer: It allows the content of any of the registers to be incremented or decremented by logic 1.

Temporary register: ALU has no data storage, and any data applied to its input immediately appears at its output. Hence, temporary registers are essential to the internal functioning of the microprocessor in order to support ALU's operations. Users cannot access the temporary registers.

Table 24.2 shows the list and capacity of various registers in 8085 microprocessor.

Table 24.2 *List of registers in 8085 microprocessor*

Sl. No.	Name of the register	Quantity	Capacity
1.	Accumulator (register A)	1	8-bit
2.	General-purpose registers (B, C, D, E, H, and L)	6	8-bit each
3.	Stack pointer	1	16-bit
4.	Program counter	1	16-bit
5.	Incrementer/decrementer address latch	1	16-bit
6.	Temporary register	1	8-bit
7.	Flags	5 flip-flops	1-bit each
8.	Instruction register	1	8-bit

24.4.2 Arithmetic Logic Unit

The ALU performs the arithmetic and logic operations such as addition, subtraction, logical AND, OR, Ex–OR, complement, increment, decrement, shifting, and clearing on 8-bit quantities. It uses data from accumulator and the temporary register to perform operations, and results are then stored back in the accumulator.

24.4.3 Timing and Control Unit

It is a part of CPU and generates timing and control signals to carry out the instructions, which have been decoded. This section contains an oscillator and a controller-sequencer. The oscillator produces clock signals to synchronize all registers. The controller-sequencer generates the control signals required for the unit to control the data flow between CPU and peripherals, and controls the entire operations of the microprocessor.

24.4.4 Interrupt Control

An interrupt is an emergency signal to which the microprocessor is required to respond as soon as possible. When an interrupt signal is received by the microprocessor, it may suspend the program currently being executed and responds to the interrupt signal. The 8085 has 5 interrupt inputs: INTR, RST5.5, RST6.5, RST 7.5, and TRAP. Each of the three RESTART inputs 5.5, 6.5, and 7.5 has a programmable mask. This means that the processor finishes the current execution and then serves the interrupt. TRAP is also a RESTART interrupt except for the fact that it is non-maskable. A non-maskable interrupt is one that cannot be blocked or masked by the processor. The three RESTART interrupts cause the internal execution if the interrupts are enabled and if the interrupt mask is not set.

The interrupts are arranged in a fixed priority and the interrupt control determines as to which interrupt is to be given priority if more than one is pending as follows: TRAP, a non-maskable interrupt is of highest priority, then RST 7.5, RST 6.5, RST 5.5, and INTR.INTR has the lowest priority This priority scheme does not take into account the priority of a routine that was started by a higher priority interrupt. For example, RST 5.5 can interrupt an RST 7.5 routine if the interrupts were re-enabled before the end of the RST 7.5 routine. The TRAP interrupt is useful for catastrophic errors such as power failure or bus error. The TRAP input is recognized just as any other interrupt but has the highest priority.

24.4.5 Serial I/O Control

Generally, the data transfer from input/output (I/O) devices take place either in serial or in parallel form. In some cases, I/O devices work with serial data rather than parallel. Hence, before processing, the serial data from input device must be converted to 8-bit parallel form. Similarly, 8-bit parallel data out must be converted to serial form before reaching the serial output device.

8085 Pin Description The 8085 microprocessor is a 40 pin dual inline package (DIP). The pin out diagram and signals are depicted in Figure 24.4

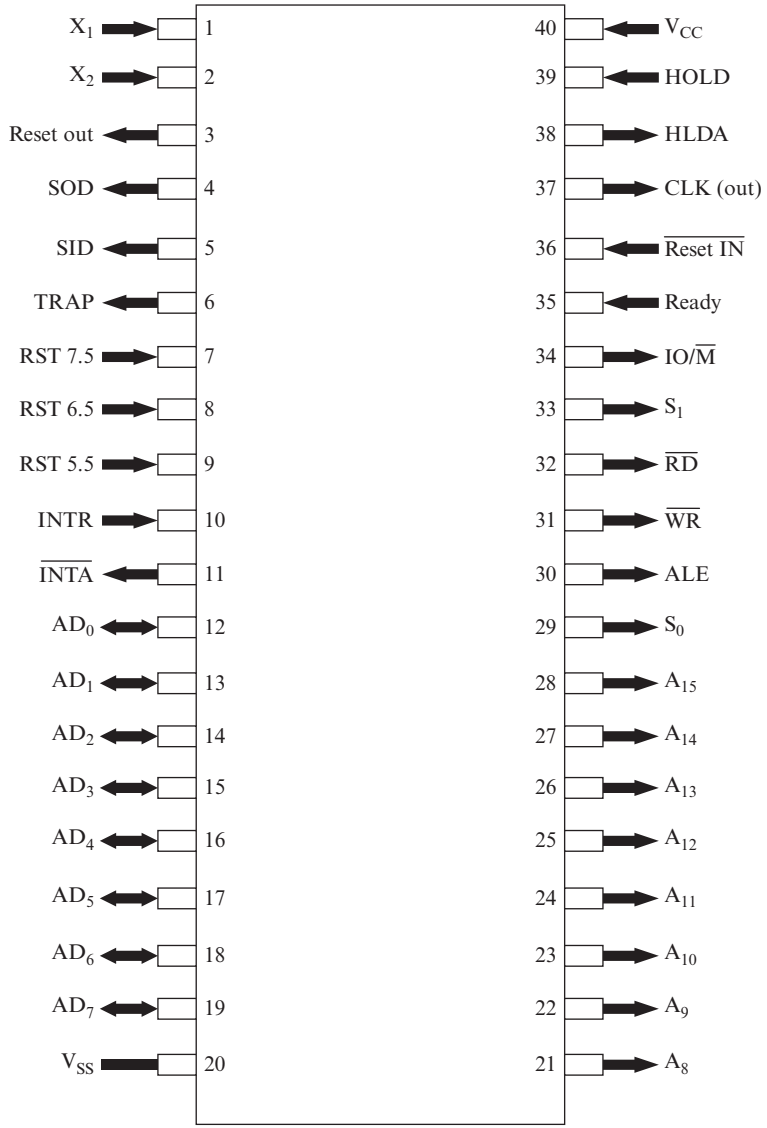


Figure 24.4 *Pin out diagram of 8085 microprocessor*

Description of Each Pin

Pins 1 and 2 [(X_1, X_2) (Input)]:Between these two pins, a crystal oscillator-*LC* or *RC* network is connected to control the clock frequency whose maximum value is 3.125 MHz and minimum value is 500 KHz.

Pin 3 [RESET OUT (Input)]: When the signal is high, it indicates that CPU is being reset. After the RESET OUT goes low, the process begins.

Pins 4 and 5 [SOD and SID]: SOD converts accumulator data into a serial data that comes out of pin 4, which can be connected to a serial output device. Pin 5 is the input for SID.

Pins 6 to 11: These pins are used for interrupt control unit. The priority of five interrupts is in the following order: TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. Pin 11 (INTA) is an output pin that responds to an INTR interrupt.

Pins 12 to 19: These pins are multiplexed, that is, they are used to carry the lower 8 address bits or data bits (AD_0–AD_7).

Pin 20: It is used for the system grounding.

Pins 21 to 28: These pins are used to carry the upper 8 address bits (A_8–A_{15}).

Pins 29 and 33 (S_0 and S_1): These two pins are used to carry status signals that indicate which operation is taking place. For example, encoded status of the bus cycle is given below:

S1	S0	
00		HALT
0	1	WRITE
1	0	READ
1	1	FETCH

Pin 30 [ALE (Output)]: Address latch enable (ALE) occurs during the first clock cycle of a machine state and enables the address to get latched into the on-chip latch of peripherals. The falling edge of ALE is set to guarantee setup and hold times for the address information. ALE can also be used to strobe the status information.

Pins 31, 32, and 34 [$\overline{\text{WR}}$, $\overline{\text{RD}}$, and IO/$\overline{\text{M}}$]:

$\overline{\text{WR}}$ **[WRITE]** It indicates the data on the data bus is to be written into the selected memory or I/O location. Data are set up at the trailing edge of WR. The bar over WR signal indicates that it is an active low signal, that is, it is activated on receiving a low value of signal.

$\overline{\text{RD}}$ **[READ]** It indicates the selected memory or I/O device is to be read and that the data bus is available for the data transfer.

IO/$\overline{\text{M}}$ It indicates a memory (read/write) operation.

Pin 35 [READY]: If Ready is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If Ready is low, the CPU will wait for Ready to go high before completing the read or write cycle.

Pins 36 and 3 [$\overline{\text{RESETIN}}$ and RESETOUT**]:** This signal will reset program counter, instruction register, and others. This resets the CPU. The pin 3 then gives out high RESET OUT signal indicating that the CPU is being reset. The processing starts only after $\overline{\text{RESET IN}}$ goes high (and RESET OUT goes low).

Pin 37 [CLK (Output)]: Clock output for use as a system clock when a crystal or *RC* network is used as an input to the CPU.

Pins 38 and 39 [HLDA and HOLD]: These pins are used to support the fast data transfer. The CPU, upon receiving the Hold request, floats the address, data and control buses. The HLDA (HOLD ACKNOWLEDGE) signal indicates that the CPU has received the Hold request. HLDA goes low after the Hold request is removed.

Pin 40 [V_{CC}]: This is the supply pin connected to a+ 5 V.

24.5 MICROCOMPUTER LANGUAGES

A program is a set of statements or instructions that direct the computer to perform a required task. Each processor has its own set of instructions. These instructions can be written in a number of programming languages. However, programs written in any language must be converted into the binary form or machine level. Microcomputer programming languages are of three types. These are as follows:

(i) Machine language (binary, hexa, and octal)
(ii) Assembly language (symbols and mnemonics)
(iii) High-level language (FORTRAN, COBOL, Pascal, C, etc.)

24.5.1 Machine Language

This is a machine-dependent programming language in which programs are written in the binary or in 0s and 1s. Some of the disadvantages of using machine language are as follows:

(i) It is difficult to understand and debug.
(ii) It takes more time to write and also to enter the program and therefore uses a tiresome procedure.
(iii) The programmer should have thorough knowledge of internal organization of the computer.
(iv) The programmer must keep track of all the addresses of the main memory.

The only advantage of using machine language is that it eliminates the process of translation of program to binary form.

24.5.2 Assembly Language

It uses mnemonics and symbols to write a program. Assembler software translates assembly language program to machine language. Some of the advantages of using machine language are as follows:

(i) It is easy to read, write, debug, and modify the program than machine language.
(ii) It is a machine-dependent language.
(iii) Programmer need not keep track of memory.
(iv) Programming efficiency is greater than machine language.

Some of the disadvantages of using machine language are as follows:

(i) It needs an assembler that increases the cost.
(ii) Different manufacturers use different mnemonics for the same operation.

24.5.3 High-Level Language

It uses words, phrases, and sentences of English to write a program. The program written in high-level language can be used for any other type of computer also. Each statement in this corresponds to a number of assembly language or machine language instructions.

Some of the advantages of using high-level language are as follows:

(i) It is a user-friendly and machine-independent language.
(ii) Writing as well as modifying a program is easier and faster.

The disadvantage of using high-level language is that it requires more memory and a translator.

24.6 INSTRUCTION SET

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.

24.6.1 Operation Code and Operand

Instruction word can be divided into two parts as follows:

(i) Op-code (operation code) represents the operation or task that the processor is being instructed to perform such as addition, subtraction, etc.
(ii) Operand (address or operand address) represents the location in memory where the data are stored.

24.7 CLASSIFICATION OF INSTRUCTION SET

There are 74 instructions in 8085 microprocessor. These instructions in an instruction set can be classified into the following functional categories:

(i) Data transfer group
(ii) Arithmetic group
(iii) Logical group
(iv) Branch group
(v) Stack, I/O, and machine-control group

These instructions are common to both 8080 and 8085 microprocessors. The set of instructions is permanently fixed while designing the microprocessor chip.

24.7.1 Data-Transfer Group

The data are copied or transferred from a location called a source to another location called a destination, without modifying the contents of the source.

Some examples of data-transfer instructions are illustrated in Table 24.3.

Table 24.3 *Some operations of data-transfer instructions*

Op-code	Operand	Description	Comments
MOV	r_1, r_2	Move (1-byte instruction)	Copies data from source register r_2 to destination register r_1
MVI	r_1, 8-bit	Move immediate (2-byte instruction)	Loads the 8-bits of the second byte into the register r_1

IN	8-bit port Address	Input from port (2-byte instruction)	Accepts/reads data from the specified input port and loads into the accumulator
OUT	8-bit port Address	Output from port (2-byte instruction)	Sends/copies the accumulator contents to the specified output port

Example 24.1 Consider the register H that contains 10_H and register C that contains 32_H. Write the suitable instruction to transfer the contents of register C to register H and explain how the action is initiated by the instruction.

Solution: The instruction used for this operation is MOV H, C. The first operand specifies the destination and the second operand C specifies the source.

Register contents before instruction Register contents after instruction

Let the contents of the register H before the execution of the instruction be 10_H and 32_H in register C. After the instruction MOV H, C is executed, the contents of register C, are copied into the register H. Hence, at the end of the instruction, the contents of both register H and C are 32_H.

24.7.2 Arithmetic Operations

These instructions perform arithmetic operations, such as addition, subtraction, increment, and decrement, etc., on data in registers or memory. There is no multiplication and division operation in 8085. Depending on the operation, all these instructions will affect various flags.

Addition The contents of any register or memory location are added to the contents of the accumulator, and the result is stored in the accumulator. The instruction DAD is an exception; it adds 16-bit data directly in register pairs. Ex: ADD B, adds the contents of the register B to the accumulator and the result is stored in the accumulator and DAD D adds the contents of DE register pair to HL register pair and the result is in HL register pair.

Subtraction The contents of any register or memory location are subtracted from the contents of the accumulator, and the result is stored in the accumulator. The subtraction is performed in 2's compliment, and the results if negative, are expressed in 2's complement. Ex: SUB B, the contents of the register B are subtracted from the contents of the accumulator and the result is stored in the accumulator.

Increment|decrement The 8-bit contents of a register or a memory location can be incremented or decrement by 1. Similarly, the 16-bit contents of a register pair can be incremented or decrement by 1. These increment and decrement operations differ primarily from addition and subtraction in that they can be performed in any one of the registers or in a memory location. Ex: INR C increments the contents of the register C by 1.

Table 24.4 illustrates various instructions for the arithmetic operations.

Table 24.4 *Various instructions for the arithmetic operations*

Op-code	Operand	Description	Comments
ADD	B	Add (1-byte instruction)	Adds the contents of register B to the contents of accumulator
ADI	8-bit	Add Immediate (2-byte instruction)	Adds the second byte to the contents of accumulator
SUB	C	Subtracts (1-byte instruction)	Subtracts the contents of register C from the contents of accumulator
SUI	8-bit	Subtract immediate (2-byte instruction)	Subtracts the second byte from the contents of accumulator
INR	B	Increment (1-byte instruction)	Increases the content of register B by 1
DCR	B	Decrement (1-byte instruction)	Decreases the content of register B by 1

Example 24.2 The contents of register C and accumulator are 45_H and $A7_H$, respectively. Add the contents of accumulator to the contents of register C.

Solution: The suitable instruction for this operation is ADD C.

		Addition operation			
A	A7	$A7_H$ = 1010 0111	**A**	EC	
C	45	45_H = 0100 0101	**C**	45	
Register contents before instruction		EC_H = 1110 1100 (Flags: S=1, Z=0, AC=0, P=1, CY=0)	Register contents after instruction		

Let the initial contents of the accumulator A be $A7_H$ and contents of register C be 45_H. The contents of register A are added to the contents of register C, and the result is stored in the accumulator. The flags that are affected are sign flag and parity flag that are set and all other flags are reset. The result EC_H is stored in the accumulator in binary form

24.7.3 Logical Operations

These instructions perform various logical operations with the contents of the accumulator, register, and memory location. The results are stored in the accumulator. Unless indicated otherwise, all flags will be affected.

The description of these instructions is illustrated in Table 24.5.

Table 24.5 *Some of the logical operations*

Op-code	Operand	Description	Comments
ANA	C	Logical AND with accumulator (1-byte instruction)	Logically ANDs the contents of the register C with the contents of accumulator
ANI	8-bit	AND immediate with accumulator (2-byte instruction)	Logically ANDs the second byte with the contents of the accumulator
ORA	E	Logically OR with accumulator (1-byte instruction)	Logically ORs the contents of the register E with the contents of accumulator
ORI	8-bit	OR immediate with accumulator (2-byte instruction)	Logically ORs the second byte with the contents of the accumulator
XRA	B	Logically exclusive OR with accumulator (1-byte instruction)	Exclusive ORs the contents of register B with the contents of the accumulator
XRI	8-bit	Exclusive-OR immediate with accumulator (2-byte instruction)	Exclusive ORs the second byte with the contents of the accumulator
CMA		Complement (1-byte instruction)	Complements the accumulator content. Flags are not affected

Example 24.3 The contents of register C and accumulator are 74_H and 84_H, respectively. Perform OR operation of the register C contents with the contents of accumulator.

Solution: The suitable instruction for this operation is ORA C

		Addition operation		
A	A7	$84_H = 1000\ 0100$	**A**	F4
C	45	$74_H = 0111\ 0100$	**C**	45
Register contents before instruction		$F4_H = 1111\ 0100$ (Flags: S=0, Z=0, AC=0, P=1, CY=0)	Register contents after instruction	

Let the initial contents of the accumulator A be 84_H and contents of register C be 74_H. The contents of register A are logically ORed (bit by bit) with the contents of register C, and the result is stored in the accumulator. The flags that are affected are parity flag that is set, and all other flags are reset. The result $F4_H$ is stored in the accumulator in binary form.

24.7.4 Branching Operations

This group of instructions alters the normal sequence of program flow. There are two types of branch instructions: conditional branching and unconditional branching. Conditional branching checks the status of the flag registers to branch. All flags are used in conditional branching except auxiliary carry. This group includes the instructions for conditional and unconditional jump, subroutine call and return, and restart. Examples include JMP, JC, JZ, CALL, CZ, RST, RET, etc.

24.7.5 Stack, I/O, and Machine Control Group

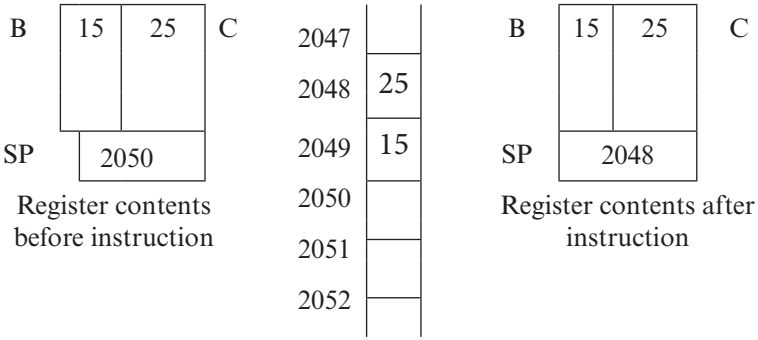
This group includes the instructions for input/output ports, manipulates the stack, and alters internal control flags. Unless indicated, flags are not affected by any instruction.

Examples include IN, OUT, PUSH, POP, HLT, etc.

Example 24.4 The stack pointer register contains 2050_H, resister B contains 15_H, and register C contains 25_H. Save the contents of BC register pair on stack.

Solution:

The suitable instruction for this task is PUSH B.



Before the instruction is executed the stack pointer points to the top of the stack 2050_H. The content of the register B is 15 and that of C is 25. After the PUSH B instruction is executed, the SP points to address location SP-1 that is 2049_H. The contents of C register are loaded in the memory location 2049_H. The SP points to address location SP-1, that is 2048_H, the contents of B register are loaded in memory location 2048_H.

24.8 THE 8085 ADDRESSING MODES

The instructions MOV B, A or MVI A, and 82H are to copy data from a source into a destination. To execute these instructions, the source can be a register, an input port, or an 8-bit number (00H to FFH). Similarly, the destination can be a register or an output port. The source and destination are operands. The various formats for specifying operands are called the Addressing Modes. For 8085, these are as follows:

(i) Direct addressing
(ii) Register addressing
(iii) Register indirect addressing
(iv) Immediate addressing
(v) Implicit addressing

24.8.1 Direct Addressing

In this mode, the address of the operand is given in the instruction itself. Example: LDA 8050, the content of 8050 is loaded into the accumulator.

24.8.2 Register Addressing

In this mode, the operand is in general-purpose register. Example: MOV A, B moves the contents of register B to A.

24.8.3 Register Indirect Addressing

In this mode, the address of operand is specified by a register pair. Example: MOV A, M moves data from memory location specified by H–L pair to accumulator.

24.8.4 Immediate Addressing

In this mode, the operand is specified within the instruction itself. Example: MVI A,05H Move 05H to accumulator

24.8.5 Implicit Addressing

If the address of source of data as well as address of destination of result is fixed, then there is no need to give any operand along with the instruction. Example: CMA – Complement accumulator

24.9 INSTRUCTION FORMATS

According to instruction word size of an instruction, the 8085 instruction set can be classified into the following three groups:

(i) One-word or 1-byte or single-byte instructions
(ii) Two-word or 2-byte or 2-byte instructions
(iii) Three-word or 3-byte or 3-byte instructions

24.9.1 Single-Byte Instructions

A single-byte instruction includes the opcode and operand in the same byte for the operation to be performed. Examples include MOV, ADD, SUB, ANA, ORA, INR, DCR, etc.

Table 24.6 explains examples of single-byte instructions.

Table 24.6 *Examples of single-byte instructions*

Op-code	Operand	Binary code	Hex code	Task
MOV	C, A	0100 1111	4FH	Copy the contents of the accumulator in the register C
ADD	B	1000 0000	80 H	Add the contents of register B to the contents of the accumulator
CMA		0010 1111	2 FH	Invert (compliment) each bit in the accumulator

24.9.2 Two-Byte Instructions

In a 2-byte instruction, the first byte specifies the operation code and the second byte specifies the operand. Source operand is a data byte immediately following the Op-code. Examples include MVI, ANI, ADI, ORI, etc.

Table 24.7 explains example of 2-byte instruction

Table 24.7 *Example of 2-byte instruction*

Op-code	Operand	Binary code	Hex code	Task
MVI	A, 10H	0011 1110	3E First byte	Load an 8-bit data (10H) to accumulator
		0001 0000	10 Second byte	

The corresponding assembly language instruction is as follows:

Mnemonics	Hex code
MVI A, 10H	3E 10

24.9.3 Three-Byte Instructions

In a 3-byte instruction, the first byte specifies the opcode and the following 2-bytes specify the 16-bit address. The second byte is the low-order address and the third byte is the high-order address. Examples include LXI, JMP, CALL, etc.

Table 24.8 explains example of 2-byte instruction.

Table 24.8 *Example of 3-byte instructions*

Op-code	Operand	Binary code	Hex code		Task
JMP	2085H	1100 0011	C3	First byte	Transfer the program sequence to memory location 2085H
		1000 0101	85	Second byte	
		0010 0000	20	Third byte	

24.10 16-BIT MICROPROCESSOR

8086 is an improved version of 8085 and was developed by Intel in 1976. It is a 16-bit microprocessor. It has a powerful instruction set, and it is capable of providing multiplication and division operations directly. It has 20 address lines and 16 data lines. Therefore, it can access up to 1 MB of memory. It supports two modes of operation: The first is the maximum mode and the second is the minimum mode. Minimum mode is applicable for systems that have a single processor, and maximum mode is used for the multiprocessor system. 8086 provides additional features in that it has an instruction queue capable to store six instruction bytes from the memory. The next instruction is fetched while the present instruction is being executed. Hence, it makes the processor fast.

The primary objectives of this 16-bit microprocessor can be summarized as follows:

(i) Increases memory addressing capability
(ii) Increases execution speed
(iii) Provides a powerful instruction set
(iv) Facilitates programming in high-level languages

INTEL 8086/8088 It is a 16-bit microprocessor housed in a 40-pin DIP and is capable of addressing 1 MB memory. Various versions of this chip can operate with different clock frequencies:

(a) 8086 (5 MHz)
(b) 8086-2 (8 MHz)
(c) 8086-1 (10 MHz)

It contains approximately 29,000 transistors and is fabricated using the HMOS technology. The term 16-bit means that it is arithmetic logic unit, its internal registers, and most of its instructions are designed to work with 16-bit binary word. The 8086 microprocessor has a 16-bit data bus; therefore, it can read from or write either 16-bits or 8-bits data to memory and ports at a time. The 8086 microprocessor has 20-bit address bus; hence, it can address any one of 2^{20} or 1,048,576 memory locations. Here, 16-bit words will be stored in two consecutive memory locations. If the first byte of a word is at an even address, the 8086 can read entire word in one operation. Alternately, if the first byte of the word is at an odd address, then 8086 will read the first byte with one bus operation and the second byte with another bus operation.

24.11 DIFFERENCES BETWEEN 8085 AND 8086

(i) **Size**: 8085 is 8-bit microprocessor, whereas 8086 is 16-bit microprocessor.
(ii) **Address bus**: 8085 has 16-bit address bus, whereas 8086 has 20-bit address bus.
(iii) **Memory**: 8085 can access up to 2^{16}(64 kB) of memory, whereas 8086 can access up to 2^{20} = 1 MB of memory.
(iv) **Instruction queue**: 8085 does not have an instruction queue, whereas 8086 has instruction queue.
(v) **Multiprocessing support**: 8085 does not support multiprocessing, whereas 8086 supports.
(vi) **I/O**: 8085 can address 2^{8}(256) I/Os and 8086 can access 2^{16}(64 kB)I/Os
(vii) **Arithmetic support**: 8085 only supports integer and decimal arithmetic, whereas 8086 supports integer, decimal, and ASCII arithmetic.
(viii) **Multiplication and division**: 8085 does not support, whereas 8086 supports.
(ix) **External hardware**: 8085 requires less external hardware, whereas 8086 requires more external hardware.
(x) **Cost**: The cost of 8085 is low, whereas the cost of 8086 is high.
(xi) **Memory segmentation**: In 8085, the memory space is not segmented; but in 8086, the memory space is segmented.

Additional Solved Examples

Example 24.5 Write an assembly program to add two hexadecimal numbers.

Solution: Let the numbers to be added are $8B_H$ and $6F_H$
Algorithm:
Step 1: Load the first number $8B_H$ in register D.
Step 2: Load the second number $6F_H$ in register C.
Step 3: Move the contents of register C into register A.
Step 4: Add the contents of register D with the contents of the accumulator A.
Step 5: The contents of the accumulator are stored in the memory location 8051.

Code:

```
MVI D, 8B_H
MVI C, 6F_H
MOV A, C
ADD D
STA 8051
HLT
```

Example 24.6 Write an assembly program to find the larger number between two numbers

Solution: Let the two numbers 30_H and 40_H be compared

Algorithm:
Step 1: Load the first number 30_H in register B.
Step 2: Load the second number 40_H in register C.
Step 3: Move the contents of register B into register A.
Step 4: Compare the contents of register C with the contents of the accumulator A.
Step 5: If the both are equal then zero flag is set.
Step 6: If the contents of register C are greater than the contents of accumulator, then carry flag is set.

The contents of the accumulator are stored in the memory location 8051.

Code:

```
        MVI B, 30H
        MVI C, 40H
        MOV A, B
        CMP C
        JZ   EQU
        JC   GRT
        STA 8051
        HLT
EQU:    MVI A, 01H
        STA 8051
        HLT
GRT:    MOV A, C
        STA 8051
        HLT
```

Example 24.7 Write an assembly program to perform the multiplication of two 8-bit numbers.

Solution: Let the two numbers that are to be multiplied be 30_H and 40_H, respectively.

Algorithm:
Step1: Load the first number 30_H in 4150
Step 2: Load the second number 40_H in 4151.
Step 3: Initialize D register and Accumulator to zero.

Step 4: Initialize HL register with 4150 such that H register contains 41 and L register contains 50.
Step 5: The contents of the memory location 4150 are loaded into B register.
Step 6: The contents of the memory location is incremented by 1.
Step 7: The contents of the memory location 4150 are loaded into C register
Step 8: The contents of B register are added to accumulator by the contents of C register times.
Step 9: The contents of the accumulator are stored in the memory location 4052 and the contents of D register are incremented if carry is produced. The contents of D register are loaded in 4053.

Code:

```
        MVID, 00
        MVIA, 00
        LXIH, 4150
        MOVB,M
        INXH
        MOVC,M
LOOP:   ADDB
        JNC   NEXT
        INR D
NEXT:   DCRC
        JNZLOOP
        STA 4152
        MOVA, D
        STA 4153
        HLT
```

Summary

- A microprocessor is a digital IC that can be programmed, using a set of instructions, to perform desired operations on the data. It consists of an ALU, a register unit and a control unit. A microprocessor is one component of a microcomputer.
- A microcomputer is a small computer with a microprocessor as its CPU. It includes a microprocessor, memory, and interfacing facilities to communicate with the peripheral devices.
- The microprocessor performs the following operations on the memory, namely, memory read/write, and I/O read/write. Address bus, data bus, and control bus communicate with memory and other devices (I/O).
- The ALU performs the arithmetic and logic operations such as addition, subtraction, logical AND, OR, Ex-OR, complement, increment, decrement, shifting, and clearing on 8-bit quantities
- An interrupt is an emergency signal to which the microprocessor is required to respond as soon as possible. When an interrupt signal is received by the microprocessor, it may suspend the program currently being executed and responds to the interrupt signal.
- The interrupts are arranged in a fixed priority that determines which interrupt is to be recognized if more than one is pending. TRAP gets the highest priority, then RST 7.5, RST 6.5, RST 5.5, and INTR.INTR has the lowest priority.

- 8085 microprocessor has six general-purpose registers that act as temporary storage devices called B, C, D, E, H, and L. In addition, 8085 has a special register called an accumulator (A), in which all the arithmetic and logical operations are carried out.
- 8085 has a flag register or status flags that are a set of flip-flops that are used to check the status of accumulator after arithmetic or logical operation is performed. The various flags are sign flag (S), zero flag (Z), auxiliary carry (AC), parity (P), and carry flag (CY). The microprocessor uses these flags to test data conditions.
- Instruction word can be divided into two parts: Op-code (operation code) and operand. Op-code represents the operation or task that the processor is being instructed to perform such as addition, subtraction, etc. Operand represents the location in memory where the data are stored.
- The instructions in an instruction set can be classified into five functional categories, namely, data transfer group, arithmetic group, logical group, branch group and stack, I/O and machine-control group.

Multiple Choice Questions

1. The first microprocessor was
(a) 4001 (b) 4002
(c) 4003 (d) 4004

2. The memory capacity of 8085 microprocessor is
(a) 64 kB (b) 640 B
(c) 1 MB (d) 16 MB

3. 8085 microprocessor is a
(a) 8-bit microprocessor (b) 4-bit microprocessor
(c) 16-bit microprocessor (d) 32-bit microprocessor

4. 8085 microprocessor is a -------- device
(a) PMOS (b) NMOS
(c) CMOS (d) QMOS

5. Which of the following processors has an in-built math processor?
(a) 8085 (b) 8086
(c) 8088 (d) Pentium-4

6. A general-purpose microprocessor requires the following devices to operate properly:
(a) RAM (b) ROM
(c) I/O ports (d) All of the above

7. 8085 μP enters into a wait state after the recognition of
(a) READY (b) RESET IN
(c) HOLD (d) INTR

8. Which of the following is an 8085 hardware interrupt?
(a) TRAP (b) RST 6.5
(c) RST 7.5 (d) All of the above

9. Which of the following is correct decreasing order of interrupts
(a) TRAP, RST 7.5, RST 6.5, RST 5.5, INTR
(b) TRAP, RST 6.5, RST 5.5, RST 7.5, INTR
(c) INTR, RST 6.5, RST 5.5, RST 7.5, TRAP
(d) INTR, TRAP, RST 6.5, RST 5.5, RST 7.5

10. Maximum number of I/O devices that can be addressed by Intel 8085 is
(a) 256 (b) 255
(c) 512 (d) 65,536

11. If the clock frequencies of 8085 μP1 is 5 MHz and 8085 μP2 is 4 MHz, that means
(a) μP1 is faster than μP2 (b) Both will work at same speed
(c) μP2 is faster than μP1 (d) None of the above

12. The data bus of 8085 processor is multiplexed with
(a) Lower-order address lines (b) Status lines
(c) Higher order address lines (d) None of the above

13. Which of the following is non-maskable?
(a) TRAP (b) RST 7.5
(c) INTR (d) RST 6.5

14. Which of the following instructions is used to clear accumulator content?
(a) MVI A, 00H (b) SUB A
(c) ORA A (d) CLR A

15. Which of the following instructions is used to shift the accumulator content right by 1-bit through carry flag bit?
(a) RLC (b) RAL
(c) RRC (d) RAR

16 Which of the following instructions may be used to save the accumulator content onto stack?
(a) PUSH PSW (b) PUSH A
(c) POP PSW (d) PUSH SP

17. Which of the following instructions is not an example of 8085 instruction category?
(a) Data transfer (b) Cache memory transfer
(c) Arithmetic and logic (d) Branching

18. MVI A, 10H is an example of
(a) 1-byte instruction (b) 3-byte instruction
(c) 2-byte instruction (d) 4-byte instruction

19. Which of the following instructions is not an operation performed on data of 8085?
(a) Perform arithmetic and logical operation
(b) Sequence the execution of instructions
(c) Store 8-bit data
(d) None of the above

20. Microprocessor speed depends on
(a) Clock (b) Data bus width
(c) Register size (d) Address bus width

21. RRC is a
(a) 2-byte instruction (b) Single byte instruction
(c) 3-byte instruction (d) All of the above

22. The flag that cannot be operated by direct instructions is
(a) CY (b) P
(c) AC (d) Z

23. Consider the following set of 8085 instructions

```
MVI A, 82H
ORA A
JP DISPLAY
XRA A
DISPLAY:OUT PORT1
HLT
```

the output OF PORT1 is
(a) 01H (b) FFH
(c) 11H (d) 00H

24. In an 8085-based system, the maximum number of input output devices that can be connected using I/O mapped I/O method is
(a) 64 (b) 65,536
(c) 512 (d) 256

25. When referring to instruction words, a mnemonic is
(a) A short abbreviation for the operand address
(b) A short abbreviation for the data word stored at the operand address
(c) Shorthand for machine language
(d) A short abbreviation for the operation to be performed

26. Which bus is a bidirectional one?
(a) Data bus (b) Address bus and data bus
(c) Address bus (d) None of the above

27. 8086 microprocessor is a
(a) 8-bit microprocessor (b) 4-bit microprocessor
(c) 16-bit microprocessor (d) 32-bit microprocessor

28. Which of the following is a special component used in 8086 for block transfer of data to or from a port?
(a) BMA (b) BNA
(c) DNA (d) DMA

29. Which of the following ICs is used as clock generator for 8086

(a) 8087 (b) 8284
(c) 8088 (d) 8294

30. 8086 microprocessor has --------address pins out of which --------number of pins are used as data pins

(a) 16,8 (b) 20,16
(c) 16,14 (d) 20,8

Short Answer Questions

1. What is microprocessor? Give the power supply and clock frequency of 8085?

2. List few applications of a microprocessor-based system.

3. What are the functions of an accumulator?

4. List the 16-bit registers of 8085 microprocessor.

5. Mention the purpose of SID and SOD lines.

6. What is an Op-code?

7. What is the function of IO/$\overline{M}$ signal in the 8085?

8. What is an Operand?

9. How many operations are there in the instruction set of 8085 microprocessor?

10. List out the five categories of the 8085 instructions. Give examples of the instructions for each group.

11. Explain the difference between a JMP instruction and CALL instruction.

12. Explain the purpose of the I/O instructions IN and OUT.

13. What is the difference between the shift and rotate instructions?

14. What are the control signals used for DMA operation?

15. List the four instructions that control the interrupt structure of the 8085 microprocessor.

16. What is meant by interrupt?

17. Explain priority interrupts of 8085.

18. Explain the different instruction formats with examples.

19. What is the use of addressing modes? List down the different types of addressing modes.

20. What is the use of bidirectional buffer?

Unsolved Problems

1. Write an assembly language program using 8085 to multiply two 16-bit hexadecimal numbers.
2. Write an assembly language program using 8085 to check if the bits in the given 8-bit hexadecimal number are in palindrome.
3. Write an assembly language program to check if the given 8-bit number is even or odd.
4. Write an assembly language program to divide two 8-bit hexadecimal numbers
5. Write an assembly language program to convert an 8-bit hexadecimal number to equivalent decimal number.