

MICROPROCESSOR AND ITS APPLICATION

38.1 INTRODUCTION

As you have already learned, the invention of transistor by John Bardeen, W. H. Brattain and W. B. Shockley at Bell Laboratories in 1947 was a breakthrough which brought an industrial revolution. Soon after the commercial availability of the transistor, almost every electronic system started to carry the label "solid state" signifying the conquest of the transistor over its rival - the vacuum tube, it was possible to construct entire circuits, consisting of many transistors, on a single chip of silicon. This was known as an integrated circuit (IC). The growth of integrated circuit was phenomenal. It started with small scale integrated (SSI) circuit consisting of fewer than ten gates per chip. The complexity of the electronic circuits that can be incorporated into one IC i.e. the packaging density, doubled in about every two years. Progressing at a staggering rate, resulted in the medium scale integrated (MSI) circuit with ten to hundred gates on a chip, Large scale integrated (LSI) circuit with hundred to few thousand gates and very large scale integrated (VLSI) with several thousand gates on a chip. By the early 1990's, the IC technology progressed to the point where it was possible that all the functional elements of the central processing unit (CPU); the arithmetic logic unit (ALU), register section and control unit which make a microprocessor system could be put together on an IC. This resulted in the 'computer on a chip', a dream of science fiction writers since long. This miracle device was named "Microprocessor". It is available in word lengths of 4, 8, 12, 16 or 32 bits, the details of which will be discussed in this lesson.

38.2 OBJECTIVES

After studying this lesson, you should be able to,

- explain the basic structure (architecture) of general purpose microprocessor;
 - recognize the various functional blocks of the microprocessor;
 - communicate with microprocessor using various instructions like data transfer, control, branch and input - output (I/O);
 - appreciate the utility of microprocessor in systems like data acquisition system, process control systems and microcomputers;
-

38.3 MICROPROCESSOR ARCHITECTURE

First of all we make you familiar with a microcomputer. The fig. 38.1 shows the basic architecture of a microcomputer. We start from the *input devices* which convert input signals into the proper binary form for the microprocessor. Typical input devices are keyboards, teleprinters, analog to digital converters, cassette - tape disks etc. An interface network is necessary to transform the input data into a compatible digital form for the processor by the input bus. The input device is selected by the processor by sending out an address on the I / O address bus.

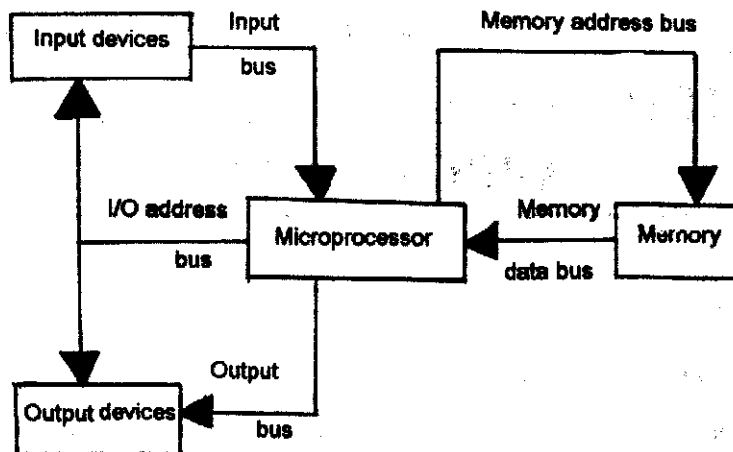


Fig. 38.1 : Basic structure of a micro computer

Memory stores binary numbers which describe in detail the instructions to be executed by the computer. It also stores in binary form, the data to be operated on and to be available at the output. Now question arises as to how Memory operation called *accessing* is performed by the microprocessor? It works in following manner. The processor first sends out a request for a memory location by transmitting an address on the memory address bus. The memory in turn responds by either accepting the data from the memory data bus into this location (called *write operation*) or sending data from this location to the processor over the memory data bus (called *read operation*). The instruction being executed by the processor determine whether access is a read or write operation.

The microprocessor contains a *CPU (central processing unit)* which has the circuitry required to access the locations in the memory, interpret resulting instructions and execute these instructions. It also contains an Arithmetic Logic Unit (ALU) to perform arithmetic and logic operations on the data. Besides these, CPU has a control section to control the operations of the computer and various data registers for temporary storage and manipulation of the data and instructions. The output device converts the binary data into the required form. Typical *output devices* are printers, CRT displays and digital to analog converters. Data is transmitted from the microprocessor to the output devices through the output bus. The output device is selected by the processor via the I/O address bus.

In many microcomputer systems the input and output devices, and the memory share the buses. The memory data bus input bus and output bus share a signal, a signal data bus, which is bi-directional bus as shown in fig. 38.1. Similarly, the memory address bus, the I/O address bus are merged and they share a common address bus. The control bus is used to synchronize the operations of the microprocessor and external circuitry. The control bus is used to inform whether the address is meant for memory or I/O, since there is a common

address bus. Also, an "interrupt" facility is required for interrupting the normal execution of the program. The control bus is used for transmitting this signal. Suppose, an external input device such as an A/D (analog to digital) converter has to communicate to the processor the fact that the conversion is over and data is available. The *control bus* is used to inform the processor, which after completing the current instruction, interrupts the program execution, sends out a signal through the control bus acknowledging the request of the interrupting device and services the interrupt. A block diagram, showing the common bus arrangement is shown in fig. 38.2.

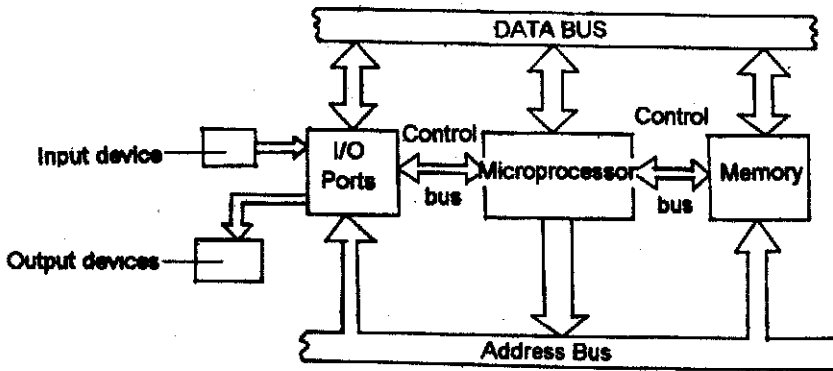


Fig. 38.2 : Block diagram for a common bus arrangement

During reading or writing into a memory location, the address must be present till the operation is completed. Latches are present at the address input lines of the memory and they latch the address as soon as it arrives. Once the address is latched, the address bus is not required and can be released for any other purpose. Usually, the data bus is used for this dual purpose. This type of bus usage is called " Multiplexing" and results in a saving of lines and hence number of pins on the IC chip. The pin diagram of a typical IC is shown in fig. 38.3.

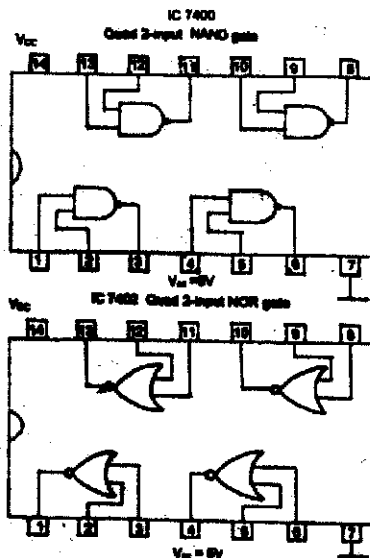


Fig. 38.3 : Pin diagram of IC 7400 and IC 7402

Usually for a 16 bit memory address, the 8-bit microprocessor transmit the least significant 8 bits of the address on the 8 - lines on which data is transmitted. Thus an 8 - bit processor is enabled for a 16 bit address transmission. This implies that a total of 2^{16} (=65536) words can be directly addressed. Each address location is a byte containing 8 bits of data. Therefore, the processor can directly address 64 kbytes of memory (where 1kbytes = 2^{10} = 1024).

INTEXT QUESTIONS 38.1

1. *Is control bus bi-directional or unidirectional ?*
.....
 2. *How a 16 bit address is transmitted by 8 bit microprocessor?*
.....
 3. *How many words can be directly addressed by a 8 bit microprocessor ?*
.....
 4. *How many bytes of memory can be addressed by a 8 bit processor?*
.....
-

38.4 FUNCTIONAL BLOCKS OF MICROPROCESSOR

The functional block diagram of a microprocessor is given in Fig. 38.1. A number of registers are included in CPU. These registers are used for temporary storage of data and instructions. The accumulator accumulates the results of the ALU operation. It also serves as one input to the ALU. The size of the accumulator equals size of the data word. There are Six eight-bit general purpose registers labeled A, B, C, D, E, H addresses. These are known as "index registers" or scratch pad memories.

The "program counter" is a 16 bit register and is used to keep track of the instructions that has to be executed next. The "stack pointer" is another 16 bit registers which is used to maintain a stack in the memory. When a word is placed in the stack, all words previously stored are moved down one location. This operation is called "PUSH" operation. When a word is retrieved from the stack, all words are moved up by one location. This is called "POP" operation. The stack is also used to store return addresses for the subroutines part of the program and to store the condition of internal registers during the processing of "interrupts". The number of registers in the stack is limited say typically seven. The stack pointer is used to hold the address of the top element of the stack. The instruction register contains the instruction being decoded and executed. The input to this register is from the memory as the program steps are read sequentially.

A status register consisting of flip-flops called "Flags" (beyond scope of the book) is used to provide indication of certain conditions, such as the over flow, carry presence of 0's in the accumulator, sign of a number in the accumulator etc. arising during arithmetic and logical operations. These are one bit registers. Their information is vital and is used for the conditional branching or the jump during program execution.

INTEXT QUESTIONS 38.2

1. *What do you understand by Registers ?*
.....
2. *How many 8 bit general purpose registers are there in 8 bit microprocessor ?*
.....
3. *What do you understand by 'stack pointer' ?*
.....
4. *What are flags & define their function ?*
.....

38.5 INSTRUCTION

This is the most important aspect in any system. When you buy a TV, VCR washing machine or any gadget, you are supplied with instruction for current. So is done here also. Therefore, the potential of using any microprocessor depends on a thorough knowledge of the instruction set and addressing modes and how to use them in conjunction in order to accomplish a given task.

38.5.1 Instruction Set

The instructions can be divided into the following five functional Categories:

- a) Data transfer
- b) Control and branch
- c) Subroutine linking
- d) Operation and
- e) Input/output

Data transfer instructions are move, load, store or exchange. They are used for moving data between memory locations or between the register and the memory. *Control and branch* instructions may be conditional depending upon the condition of a flag flip-flop or unconditional. Such instructions are jump and conditional jump. *Subroutine linking* instructions may also be conditional or unconditional. These instructions are the Call-subroutine and return. The *operation instruction* are meant to perform on a particular register or status flip-flop. No data transfer occurs. Typical instructions in this category are clear (0's are stored in the register), increment (the value stored is increased by '1'), decrement (the value stored is decreased by '1'), complement (each bit is complemented) and rotate (contents are rotated one bit to the left or right), reset flip flop and decimal-adjust accumulator (correction for BCD addition). The *input* instruction transfers the contents of the I/O bus to the accumulator. The *output* instruction transfers the contents of the accumulator to the I/O bus.

The instruction word consists of an 'OPCODE'(operation code) describing the operation to be performed and the address of the memory location or registers containing the data on which the operation is to be performed (operand). An n-bit instruction is divided into three parts (a) operation code (b) address mode and (c) operand address. Writing the instruction in machine language is tedious. The binary opcodes are difficult to remember. The binary addresses are too long and difficult to handle. Both these difficulties are

removed by the use of "mnemonic opcodes" and symbolic addresses. Let us illustrate with an example as shown below.

Table 38.1 : Equivalence of opcodes

Machine code	Mnemonic code	Function
00111 010	LDA	load the accumulator with contents of a memory cell
00 11 0010	STA	store the accumulator contents in a memory cell.
10000 110	ADD	add the contents of a memory cell to the contents of accumulation.
0111 0110	HLT	halt

Let us illustrate the use of symbolic address and consider the programme when we have to add two integers stored in memory locations 7 and 8 and place the result in memory location 9 and halt.

The program will have the following steps.

1. Load the contents of location 7 into the accumulator.
2. Add to the accumulator the contents of location 8.
3. Store the contents of the accumulator in location 9.
4. Halt.

The operand is to be specified by its address which will be 16 bit long. Writing such an address is tedious. Therefore, we use symbols for our address. Let, A denote location 7, B location 8 and C location 9. We use symbol STRT to indicate the first instruction to be executed and END for signifying end of the program. DW statement is used to define the values of the operands to be stored in the symbolic addresses. Suppose, we have to add integers 4 and 6. The location C has to be first cleared and have 0. The program thus written in terms of mnemonic codes and symbolic addresses is called the "*assembly language Program*" for example discussed above, the assembly language program is as given below:

Assembly language Programme

A	DW	4
B	DW	6
C	DW	0
STRT	LDA	A
	ADD	B
	STA	C
	HLT	
	END	

As explained already, one has to use an *assembler* for the conversion of the assembly language program to the machine language. If we use higher level languages, a *compiler*

has to be used. Normally, compilers are available for FORTRAN, PL/M, COBOL and BASIC languages.

38.5.2 Addressing Modes

You know that the data to be operated on, is usually stored in memory. The microprocessor, has four distinct addressing modes to address the data stored in memory or in the internal CPU registers. They are as follows.

1. **Register Addressing:** The data to be operated is contained within CPU registers, and this addressing specifies the register or register pair where the data is to be located.

Example: MOV A, B : move contents of register B to A;
XCHG : Exchange contents of HL with DE registers.

2. **Register Indirect Addressing :** When the address of the data is contained in a register pair it is called register indirect addressing.

Example: MOV A, M : Move contents of location M referenced by HL register pair to A.
LDAXB : Load A with contents of location X referenced by BC register pair.

3. **Immediate Addressing :** When the data to be operated is a part of the instruction and is contained in byte 2 or 3 depending on 8-or 16 bit data, then immediate addressing mode is used.

Example: MVIA : Load A with contents of the second byte which is 75 in this case.

4. **Direct Addressing:** Bytes 2 and 3 of the instruction contain the direct address of the data.

Example: STA A : Store accumulator A at location 20 BO
BO low, 20 High

Example 38.1 : A simple data manipulation algorithm, using the 8085 instruction set, is given below:

Location	Mnemonic	Opcode	Address mode	Comment
2001	MVIB	06	Immediate	move immediate data of 0 to register B
02	—	—	—	—
03	MOV A, B	78	Register	move B register to A register
04	LXIH	21	Immediate	load immediate HL register with data of 20 A
05	A0 Low	—	—	—
06	20 High	—	—	—
07	INR A	3C	Register	increment register

08	MOV M, A	77	Reg. Indirect specified by HL	move A to location M
09	NOP	00	Register	
0A	INR B	04	Register	increment register B
0B	ADD B	80	Register	add B to register A
0C	NOP	00	Register	no operation
200 D	STA A	32	Direct	store A to location
20 B	0 E	BO Low	—	—
0 F	20 High	—	—	—
2010	JMP	C 3	Direct	jump to location 200
11	07	—	—	—
12	20	—	—	—

- a) What is accomplished between 2001 and 2006 ?
- b) What is accomplished between 2007 and 2012?
- c) What are the contents of A, B, H, L registers after the program is run once and four times.
- d) Will the program advance beyond 2012?

Solution

- a) 2001 to 2003 B register is loaded with 0 0 and is transferred to A register. The amounts to clearing A and B to all-Zero conditions 2004 to 2006: H and L registers are loaded with 200 and A0 to form address vector (20A0) Hex.
- b) 2007 and 2008 : A register is incremented by 1 and is stored at location 20 A0, as referenced by HL registers.

2009 to 2012: B register is incremented by 1 and is added to A, with the result in A and stored at location (20B0) Hex.

NOP instructions are just a way to generate momentary delays without modifying CPU contents JMP instruction at 2010, puts back the program in a loop starting at (2007) Hex.

- (c) Contents of registers A, B, H, L (for one run) contents of

(A) -02 (Since it gets incremented and added to B)

(B) - 01

(HL) - 20A0 .

Contents of A, B, H, L (for 4 runs)

Contents of (A) - 0E; (B) - 04; (HL) - 20A0 unchanged

- d) **Program Loop** :- The program will not advance beyond 2012 because of the indefinite loop between 2007 and 2012. Only an interrupt or reset will pull the microprocessor out of the loop.

INTEXT QUESTIONS 38.2

1. Name the instruction set for 8 bit microprocessor and give examples for each category?

.....

2. What do you understand by OPCODE ?

.....

3. What do you understand by mnemonic opcode ?

.....

4. Explain assembly language program and give one example ?

.....

38.6 MICROPROCESSOR APPLICATIONS

In the present day age microprocessor finds a number of applications. Microprocessor can be used in two classes of applications- those applications where a microprocessor is used to replace a micro computer which is under utilized and those cases where a microprocessor replaces a network of random logic used for a particular dedicated digital applications. Most of the application are in later category. The advantage in this type of application is that the hardware of the random logic is replaced by the software of the programme. The software is much more adoptable to change than the hardware.

36.6.1 Data Acquisition System (DAS) :

Data Acquisition system is one which accepts data from various processes and monitors it. For example, suppose there are five chambers in furnace which are to be maintained at different temperatures. levels such as 400 °C, 700 °C, we put temperature sensors such as thermocouple in each chamber and the other end is connected to data acquisition system(DAS). DAS continuously gets data and gives signals to the user that what are the various temperatures at a given instant of time. Suppose, the first chamber acquires a temperature 600° C, which is much higher than the desired temperature, then DAS will give a warning signal to the user, so that he can take necessary corrective measure. So, in your absence, if system is complex, DAS plays a very important role in monitoring. A typical example is shown in figure 38.4

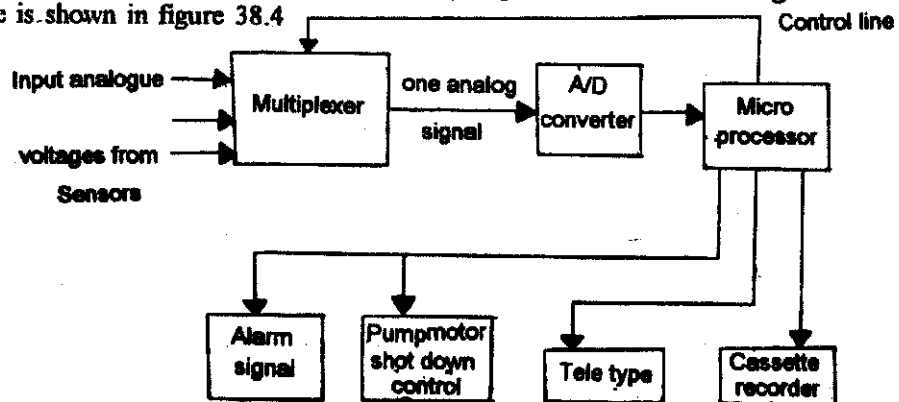


Fig. 38.4 : An example of DAS

Let us consider a data-acquisition system needed in a process control where many process variables have to be monitored. Analog data for these variables (obtained from the various sensors) can be converted to digital form and transmitted to a microprocessor. The process can suitably sample the sensors at periodic intervals, compute the averages at preselected intervals and output these on a teletype and cassette recorder. The multiplexer will allow only one of the many input analog signals to pass through to the A/D converter. The microprocessor sends to the multiplexer the address of the signal it desires to record and the multiplexer will let this particular signal to pass. The microprocessor will scale or average the value of the variable, print it and record it for future use.

38.6.2 Traffic -light Control System

Figure 38.5 shows how a microprocessor can be used in a traffic - light control system. Let us consider a simple problem for illustration. The requirement is to control a set of traffic lights at the intersection of two roads, one major with heavy traffic flow and the other minor with light traffic flow. Signals from car sensors embedded at appropriate location in the roads are fed to the microprocessor via a peripheral interface adapter which converts the signal to a form compatible with microprocessor logic. The processor continuously monitors the signals from the sensors and decides the control signals for traffic lights.

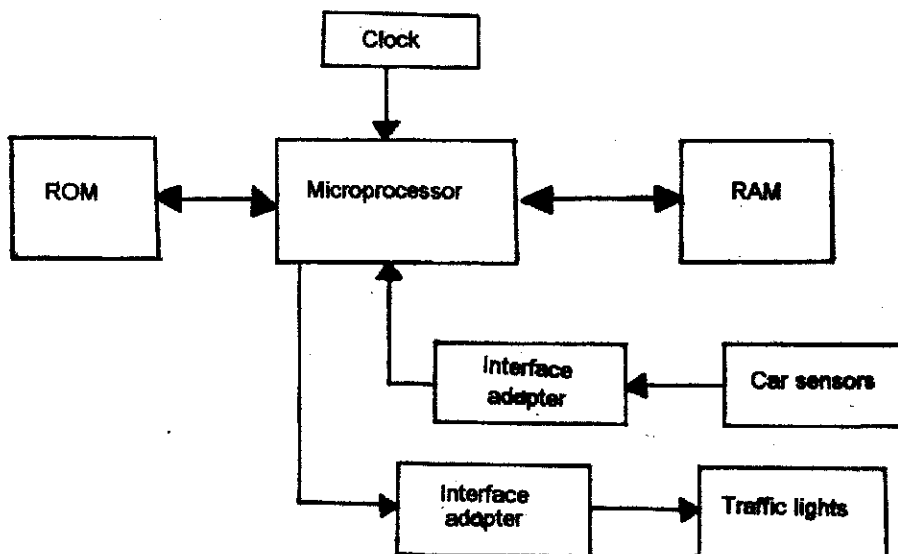


Fig. 38.5 : Traffic light control system

A running tally of the relative locations of the different cars is made and stored in the RAM along with other temporary information, such as how long a particular car has been waiting to enter the major road from the minor one and how many cars have been passed through the green light since it turned green. The ROM contains permanently wired, instructions which are used to perform the logic. A specific instruction in the ROM can be : *If 100 cars have passed through the intersection along the major road since the light turned green but none have passed for last 10 seconds, the major road signal should be turned yellow and then red and the minor road signal be turned to green.*

Based on the information received from sensors and the information accumulated in the RAM, the microprocessor would obtain an appropriate instruction from the ROM using the pre-programmed response to the particular traffic situation at the intersection. The

processor would feed appropriate signals to the peripheral interface adapter which will in turn control the traffic lights. If EPROM is used, one can periodically review traffic patterns and reprogram it or one can temporarily re-program it to meet variations in traffic patterns which are seasonal or dependent on temporary special situation.

INTEXT QUESTIONS 38.3

1. *Mention two real life situation where Data acquisition system can play an important role ?*

.....

2. *How microprocessor is useful in traffic light control ?*

.....

38.7 WHAT HAVE YOU LEARNT

- What is microprocessor and basic structure of a 8 bit microprocessor
- Recognise the various functional block of microprocessor.
- Communicate with microprocessor using various instructions like data transfer, control, branch and input -output.
- Utility of microprocessor based systems like data acquisition system, traffic control system, micro computers etc.
- Basic principle of operation of calculator and electronic watch.

TERMINAL QUESTIONS

1. Name the various functional blocks of a micro computer?
2. Is data bus bi-directional or unidirectional?
3. Is addition bus bi-directional or unidirectional ?
4. What do you understand by "bus"?
5. What do you mean by "latch"?
6. What do you understand by "multiplexing"?
7. Give schematic of functional blocks of a general purpose microprocessor?
8. Explain the concept of multiplexing and demultiplexing?
9. Give block diagram of Data Acquisition System and explain how it works?
10. Give block diagram of traffic - light control system and explain its working with an example ?
11. What advantages are achieved if ROM is replaced by an EPROM?
12. What is a program counter ?
13. Explain 'PUSH' operation ?
14. Explain 'POP' operation ?
15. Draw functional block diagram of a 8 bit microprocessor ?

16. Write short notes on following :

- (i) Register addressing
- (ii) Register indirect addressing
- (iii) Immediate addressing
- (iv) Direct addressing

17. Identify the addressing mode in the examples listed below -

- (i) LDAXB
- (ii) MOVA, M
- (iii) MOVA, B
- (iv) MVI A
- (v) STA A
- (vi) JMP
- (vii) LXI H

What will you require when you want to convert the assembly language program to the machine language ?

18. What is the use of the compiler ?

ANSWER TO THE INTEXT QUESTIONS

Intext Question 38.1

- 1. Bidirectional
- 2. Usually for a 16 bit memory address, the 8 bit microprocessor transmit the least significant 8 bits of the address on the 8- lines on which data is transmitted. Thus an 8-bit processor is enabled for a 16 bit address transmission.
- 3. $2^{16} = 65536$
- 4. 64 kBytes

Intext Question 38.2

- 1. Registers are devices used for temporary storage of data and instructions.
- 2. There are six eight bit general purpose registers labelled A,B,C,D,E & H.
- 3. A stack pointer is 16 bit register which is used to maintain a stack in the memory.
- 4. Flags are essentially status register consisting of flip flops. Its function is to provide indicating of certain conditions like overflow, sign of a member etc.
- 5. See Fig in the text.

Intext Question 38.3

- 1. Instruction set are :
 - (a) Data transfer : Example Move, load, store.
 - (b) Control and branch : Example Halt, Jump :
 - (c) Subroutine linking : Example Call subroutine, return

- (d) Operation : Example clear, Increment,decrement
- (e) Input/ output : Example start, end.
- 2 OPCODE stands for operation code. It describes the operation to be performed and the address of the memory location or registers containing the data on which the operation is to be performed.
 - 3 MNEMONIC operation code are essentially symbolic address. Generally the binary addresses are too long and difficult to handle and that is why Mnemonic operation codes are being used.
 - 4 Assembly language program is written in terms of mnemonic codes and symbolic address. For example: Add two integers 4 and 6 stored in memory location 7 and 8 and place the result in memory location 9 and halt. Assembly language program will be

A	DW	4
B	DW	6
C	DW	0
STR	LDA	A
	ADD	B
	STA	C
	HLT	END.

5. See text section 38.5.2
- 6
 - Register direct Addressing
 - Register Indirect Addressing
 - Register Addressing
 - Immediate Addressing
 - Direct Addressing
 - Direct Addressing
 - Immediate Addressing
- 7 "ASSEMBLER" is required for conversion of the assembly language program to the machine language.
- 8 Compilers are used for higher level languages.

Intext Question 38.4

- I Example are : (1) Maintenance of furnace temperatures
(2) Weather Monitoring.

See text section no 38.6.2

ANSWER TO TERMINAL QUESTIONS

1. Input device (like keyboard, teleprinters), Microprocessor, Memory and output device.

Physics

2. Bidirectional
3. Unidirectional
4. A "bus" is a set of lines carrying information.
5. Latch is used to signify storage of data.
6. Multiplexing is a process in which a information is taken to more than output lines as per our requirement.
7. The process of taking some data to various output lines as per our requirement is called multiplexing. If we have several data and we want any one of them to be connected to output, the device required is called Demultiplexer and the process called demultiplexing.
9. If ROM is replaced by EPROM, we will have facility to program it also as per our requirement.
10. A program counter is a 16 bit register and is used to keep track of the instructions that has to be executed next.
11. When a word is placed in the stack, all words previously stored are moved down one location. This operation is called PUSH operation.
12. When a word is retrieved from the stack, all words are moved up by one location. This is called "POP" operation.