CHAPTER 1

Introduction to Microprocessor

1.1 Introduction

- The most powerful tool that a man has created is computer.
- A digital computer is programmable. Fig. 1.1.1 shows the block diagram of a digital computer.
- The main parts of the computer are the central processing unit (CPU) input device, output device and memory.
- The CPU performs the task of executing instructions as per the users requirement.

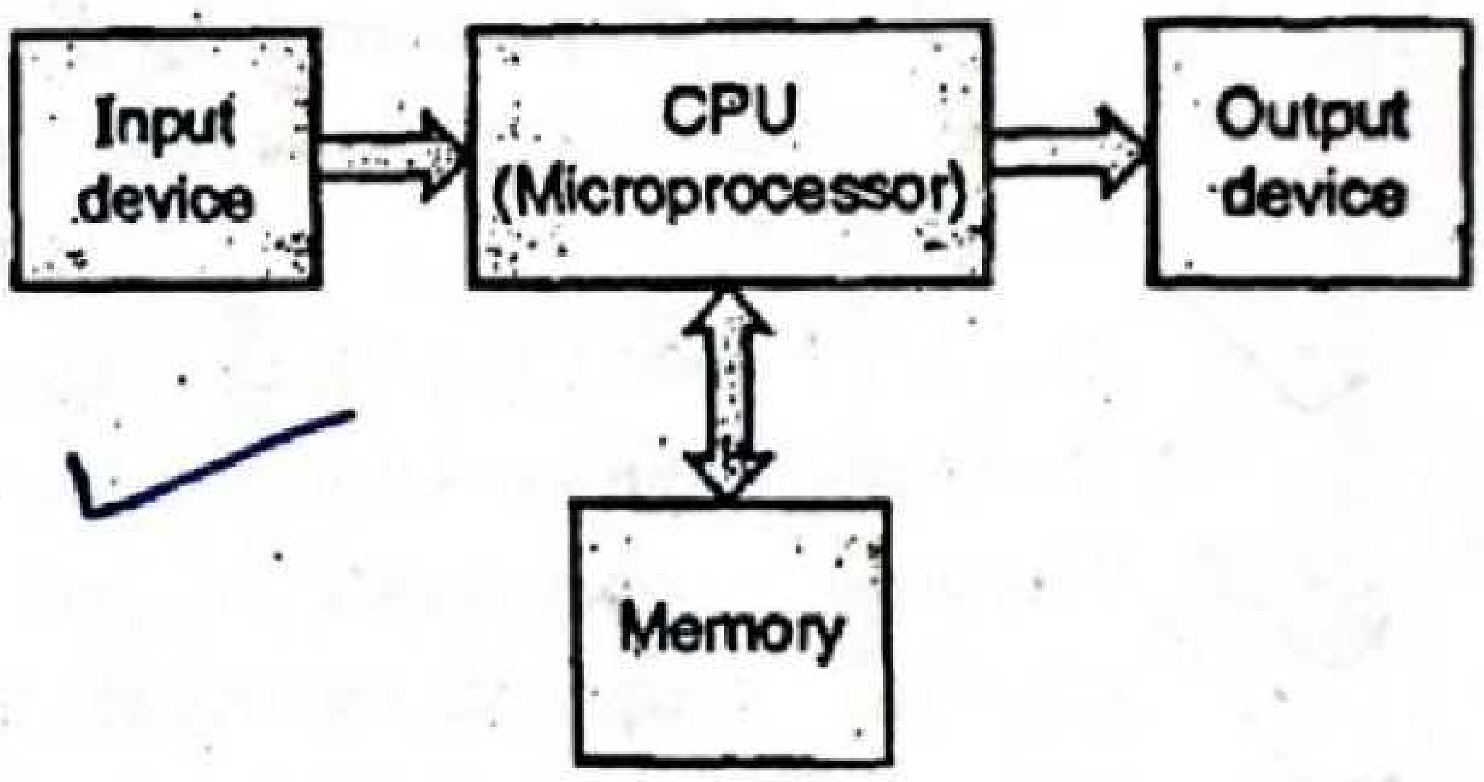


Fig. 1.1.1 : Block diagram of a digital computer

- The input device is used to send data and programs to the computer, whereas the output device is used to send data from the computer to the display unit, printer etc.
- The CPU is built on single IC called **microprocessor**. A microprocessor is a CPU. It is a programmable integrated device that has computing and decision making capability.
- The microprocessor has a wide range of applications beginning from small smart sewing machine, washing machines and other domestic appliances to computer aided system design.
- The microprocessor can be embedded into a larger system. It may be stand alone unit controlling processes or it may function like the CPU of a computer called **microcomputer**.
- The microprocessor communicates and operates in the binary numbers 0 and 1 called bits. Each microprocessor has a fixed number of instructions. The instructions are in the form of binary patterns referred to as machine language. Machine language is the language of 0's and 1's. Hence, the binary instructions are given abbreviated names (**mnemonics**) that contribute the assembly language for a microprocessor.

1.1.1 Microprocessors

- The microprocessor is a clock driven, programmable, register based device.
- The microprocessor reads the data i.e. binary instructions from the memory. It accepts this binary data as input and processes the data according to the instructions and computes the result as output.
- The microprocessor is a **programmable machine** i.e. the microprocessor can be instructed to completed certain tasks e.g. A remote is a programmable machine that is capable of changing the channels. A viewer selects the channel depending on the number printed in the manual. Similarly, a microprocessor understand and executes many instructions. The microprocessor can perform complicated computing functions e.g. switching on / off of a device.

  The programmer can select the instruction and ash the microprocessor to complete the specific tasks on given data set.
- The person who designs the remote determines a frequency for a given channel. Similarly the engineers who design the microprocessor determine the tasks that are required to be processed by the microprocessor. Accordingly, the logic is designed with a view to provide the user with a list of instructions that a user can understand.
- Fig. 1.1.1 shows the block diagram of a microprocessor based system. It consists of four components microprocessor, memory, input and output.
- These four components interact with each other to complete a given task. These physical components contribute to the microprocessor system **hardware.**

  The set of instructions written for the microprocessor to complete a task is called as a program. A group of programs is called **software.**

1.1.2 Binary Digits

- The microprocessor works in binary digits 0 and 1. Each bit is a binary digit.
- The digits are representated in terms of electrical voltages in the machine, "0" represents one voltage level and bit "1" represents other voltage level.
- A processor with an 8 bit word is called as a 8 bit **microprocessor** and a processor with 16 / 32 bit word is called as **16 / 32 bit microprocessor.**

1.1.3 Memory

- It stores the binary instructions and data for the microprocessor.
- The memory is similar to the pages of notebook.

  Each line on the page has a fixed number of binary numbers. The pages of memory are made up of semiconductor material.
- Each line is an 8 bit register that is capable of storing 8 binary bits. Each of these registers are arranged in a sequence referred to as memory. The registers are grouped in powers of two e.g. a group of 4096 (2^{12}) 8 bit registers on a semiconductor chip is called as **4 Kbytes of memory.**

- The user writes the essential instructions and data in the memory through an input device and requests the microprocessor to compute the task and compute result. The result is displayed on the output device or stored in memory.

1.1.4 Input / Output Device

- The user can enter the instructions and data into memory through devices like keyboard or switches. Such devices are called as **input devices.**
- The microprocessor reads the instructions from the memory and processes the result according to the instructions.
- The result can be displayed by a device e.g. printer, CRT screen, seven segment display, LCDs. Such devices are called as output devices.

1.2 Evolution of Microprocessors

- The first microprocessor was introduced in the year 1971 by Intel. It was a 4 bit PMOS microprocessor named as Intel 4004.After that an enhanced version of Intel 4004 was developed. Other companies like Toshiba, Rockwell also developed 4 bit microprocessors.
- In 1972 Intel introduced the first 8 bit microprocessor Intel 8008 .It also used the PMOS technology. These processors were slow and not compatible with the TTL logic. So, Intel introduced a faster NMOS microprocessor Intel 8080.
- But, the main drawback of Intel 8080 was that it required three power supplies. Hence, in 1975 Intel developed an improved version of microprocessor 8080 called **Intel 8085.** The other manufacturers of 8 bit microprocessor are Zilog, National Semiconductors, Motorola etc.
- The first x86 processor was developed in 1979 by Intel and was called the 8086 **microprocessor.** It is a 16 bit microprocessor with 16 bit data bus and 20 bit address bus. It allows 1 MB of addressing space.
- A re-engineered version of the 8086 is the 8088 microprocessor, which is identical to the 8086 except the data bus is 8 bit wide.
- The 8088 and the 8086 could run the same programs but one could not fit into the other's socket. The 8086 got its success mainly from the IBM-PC although IBM preferred implementing the 8088 into its computers because it was less expensive and less complex. Depending on the manufacturer, the 8086 and 8088 processors would run at a speed ranging from 4 MHz to16 MHz.
- The next advancements over the 8086 and the 8088 were the 80186 and the 80188 processors. It contains special hardware like programmable timers, interrupt controllers and address decoders. The 80186 and the 80188 were very advanced for their time but they were not adopted by hardware manufacturers. These processors were never used in the PC, but these processors are ideal for the systems that require minimum hardware. Depending on the manufacturer, you could find 80186/80188 processors running at a speed ranging between 6 and 40 MHz.

- The next advancement was the 80286 microprocessor that provides memory management and protection.
- The 80286 has a 24 bit address bus that gives the capability of accessing 16 MB of storage. The internal memory management unit makes it possible to access up to 1 GB of memory in the virtual mode.
- The 80286's speed could range between 6 MHz and 25 MHz depending on the manufacturer. It didn't go faster than the 80186 but it had several improvements. This processor was used in the IBM PC-AT in 1985 (three years after the introduction of the 80286).
- The 80386 was introduced by Intel in 1985. It had several significant improvements compared to 8086, 8088, 80186, 80188, 80286 processors. It is a 32-bit processor. Although it was not adopted immediately by computer manufacturers, it enjoyed a long life span.
- The 80386 was later renamed 80386 DX, for Intel developed another 80386 processor, the 80386 SX. This was a downgraded version of the 80386 DX.
- The 80386 supports 32 bit address and data bus. It can access 4 GB of physical memory and 64 terabytes of memory in the virtual mode.
- The 80386 can operate in three different modes. These modes are in the order of their increased complexity. They are :
  (i) Real Address mode.
  (ii) Virtual 8086 mode.
  (iii) Protected Virtual Address mode.
- The 80386 has an 8086/8088 compatibility in which half of the processor shuts down and the other half becomes an embedded 8086. This allows us to run a fast 8086.
- The 80486's major improvement over the 80386 was the integration of an 80387 math- coprocessor into the 80486. Intel followed the pattern of the 80386 for the 80486 by introducing a 80486SX and renaming its original processor 80486DX.
- The difference between the two was the presence or absence of the math coprocessor (in the earlier versions of the SX, the math-coprocessor was present but did not function). Intel then introduced the 80487SX (a math coprocessor) which was used as an expensive upgrade for the 80486SX making it a 80486DX. This 80487SX was in fact a fully operational 80486DX on which a few pins had been relocated so as not to allow the customer to use the cheaper 80486DX as an upgrade.
- The Pentium was a big step in the evolution of the processor. The major difference was that it contained more than one execution unit. An execution unit receives the instructions and activates them. Having more than one execution unit meant that the processor could do more than one thing at a time.
- The Pentium came in different versions: 60 MHz, 66 MHz, 75 MHz, 90 MHz, 100 MHz, 120 MHz, 133 MHz, 166 MHz, 200 MHz and 233 MHz. Then Intel introduced the MMX-enhanced processors.

1.2.1 Comparison of Features of 16, 32 and 64-bit Microprocessors

Sr. No.	Feature	8086	80286	80386 DX	80486 DX	Pentium - IV
1.	Evolution in	1978	1983	1986	1989	2000
2.	Memory	1 MB	16 MB	4 GB	4 GB	64 GB
3.	Address bus	20 bit	24 bit	32 bit	32 bit	36 bit
4.	Data bus	16 bit	16 bit	32 bit	32 bit	64 bit
5.	Multiplexed Address/ Data bus	Yes	No	No	No	No
6.	n-bit microprocessor	8086 is a 16 bit microprocessor	80286 is a 24 bit microprocessor	80386 is a 32 bit microprocessor	80486 is a 32 bit microprocessor	Pentium-IV is a 64 bit microprocessor
7.	Cache memory	Not present	Not present	Not present	8 KB on chip cache	16 KB Data cache and 16 KB code cache } L_1 256 KB } L_2
8.	Floating point unit	Not present	Not present	Not present	Present	Present
9.	Processing units	(i) Bus interface unit (ii) Execution unit	(i) Bus interface unit (ii) Instruction unit (iii) Execution unit	(i) Bus interface unit (ii) Prefetch unit (iii) Decode unit (iv) Execution unit	(i) Bus interface unit (ii) Prefetch unit (iii) Decode unit (iv) Execution unit	(i) Bus interface unit (ii) Prefetch unit (iii) Decode unit (iv) Execution unit (v) Segmentation unit

Sr. No.	Feature	8086	80286	80386 DX	80486 DX	Pentium - IV
				(v) Segmentation unit (vi) Paging unit	(v) Segmentation unit (vi) Paging unit (vii) Cache unit (viii) Floating point unit (ix) Parity generator and checker unit	(vi) Paging unit (vii) Instruction cache unit (viii) Data cache unit (ix) Floating point unit (x) Data integrity and error detection unit (xi) Functional redundancy checking unit.
10.	Branch prediction logic	Not supported	Not supported	Not supported	Not supported	Supported
11.	Operating modes supported	(i) Real mode	(i) Real mode (ii) Protected virtual address mode	(i) Real mode (ii) Protected mode (iii) Virtual mode	(i) Real mode (ii) Protected mode (iii) Virtual mode	(i) Real mode (ii) Protected mode (iii) Virtual mode
12.	Superscalar execution	Not supported	Not supported	Not supported	Not supported	Supported
13.	Data validity	Unavailable	Unavailable	Unavailable	Achieved by parity generator/checker	Achieved by data and address parity checking as functional redundancy checking.

1.3 Microprocessor Characteristics

Q. Enlist the characteristics of a microprocessor.

The power of the microcomputer is determined by the characteristics of th microprocessor

- Processing capability
- Clock frequency
- Width of the address bus
- Data types
- Word length
- Width of the data bus
- I/O addressing capability
- Interrupt capability

1. **Processing Capability :** It depends upon the number of instructions and flexibili of each instruction .
2. **Word Length :** It depends upon the width of internal data bus, registers, ALU etc The word length of a microprocessor is given as n bit, where n may be 8,16,32,64 etc An 8 bit microprocessor can process 8 bits at a time, similarly a 16 bit processor ca process 16 bits at a time. A processor with longer word length is more powerful an can process data at a faster speed as compared to processor with shorter wor length.
3. **Clock frequency :** The processing speed of microprocessor depends upon cloc frequency.
   The program execution speed is also determined by this parameter. The maximu clock frequency depends upon technology adopted in microprocessor fabrication.
4. **Width of the data bus :** This parameter decides word length of the microcompute This is the width of the external data bus.
5. **Width of the address bus :** This parameter decides the decides the memo addressing capability of the microprocessor. The maximum size of the memory un is decided by this parameter.
6. **I/O addressing capability :** The maximum number of the I/O ports accessed b the microprocessor depends upon the width of the I/O address provided in the I/ instruction.
7. **Data types :** The microprocessor handles various types of data formats like binar BCD, ASCII, integers, real numbers, signed numbers and unsigned numbers etc.
8. **Interrupt Capability :** Interrupts are used to handle unpredictable and rando events in the microcomputer. It is used to interrupt the microprocessor. It is al used to speed up the I/O programs. It improves the throughput of the system.

Depending on the size and capabilities the computers are classified in various typ as shown in Fig. 1.3.1.

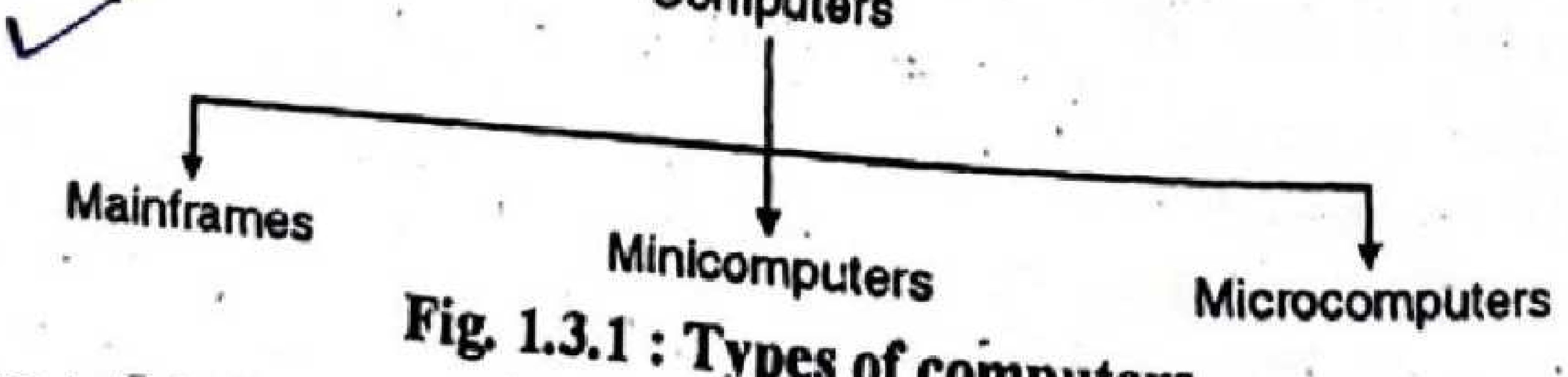


Fig. 1.3.1 : Types of computers

1.4 Basics of Microprocessor

When we hear the word microprocessor, what comes to our mind is a small (i micro) IC (integrated circuit) that can process data i.e. perform arithmetic and logi operations. But this can be done by an ALU.

ALU (Arithmetic and Logic Unit)

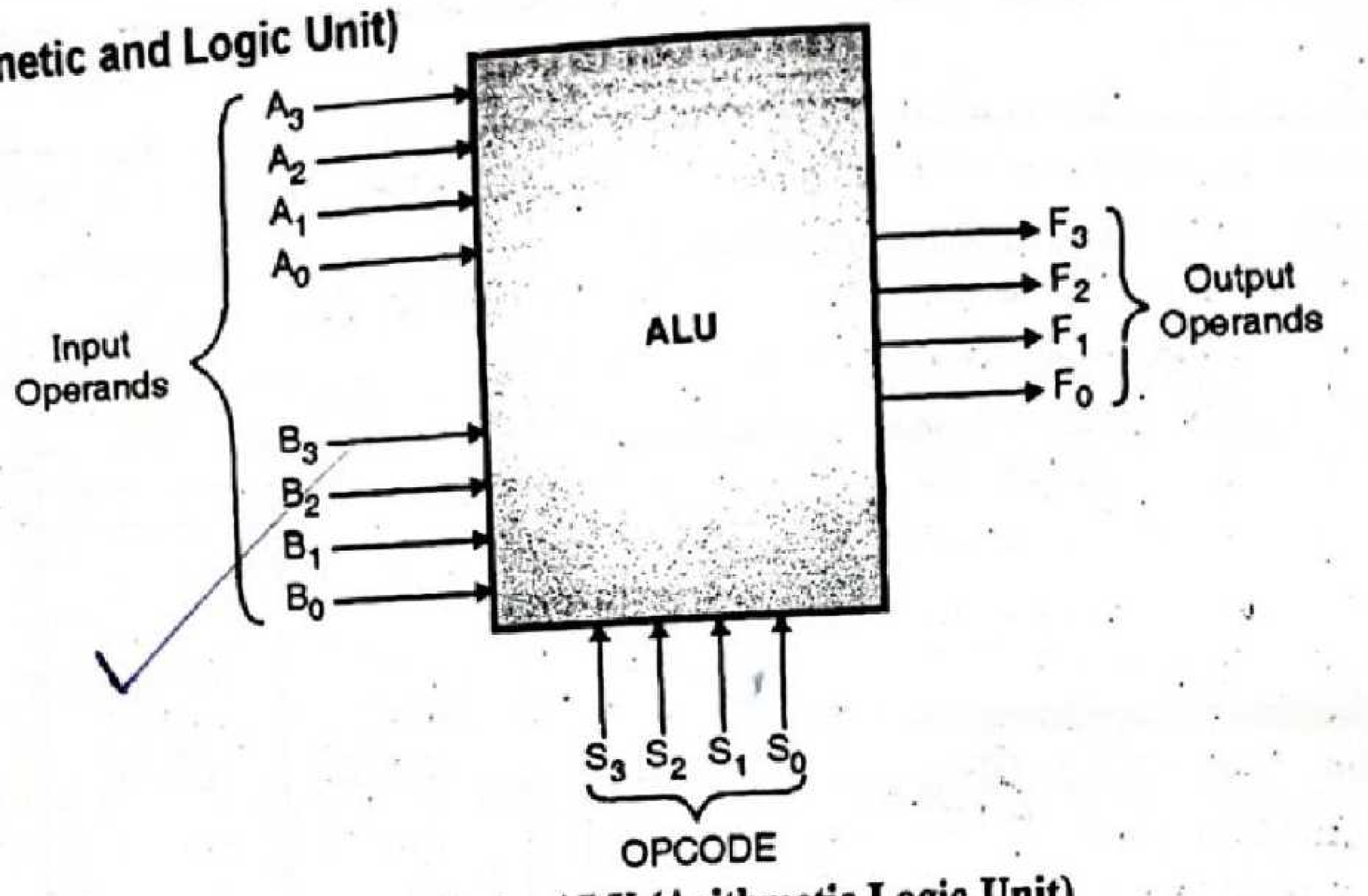


Fig. 1.4.1 : An ALU (Arithmetic Logic Unit)

This unit is used to perform arithmetic operations (like +, –, *, ÷ etc). and logical operations like (AND, OR, EX-OR etc). 'A' and 'B' are input operands while 'F' is output operand. 'S' are the select lines to select the operation (OPCODE).

Definitions

1. **Opcode :** A binary code, that indicates the operation to be performed is called as an Opcode.
2. **Operands :** The data on which the operation is to be performed (as well as the result of an operation) are termed as operands.
3. **Instruction :** The combination of opcode and an operand, that can be used to instruct a system, is called as an instruction.
4. **Instruction set :** A list of all the instructions that can be issued to a system, is called as instruction set of that system.
5. **Program/subroutine/routine :** A set of instructions written in a particular sequence, so as to implement a given task. A subroutine in assembly refers to function as in C/C++.
6. **Bus :** A group of lines, pins or signals having common function is termed as bus.

   The functional grouping of signals results in

   1. Data bus → to carry data
   2. Address bus → to select a memory or I/O location.
   3. Control bus → to issue and receive control signals.

Now in the Fig. 1.4.2, to give an instruction to the system (ALU), the programmer has to give OPCODE and OPERAND.

For e.g. if the operation is 5 + 7 i.e. $(0\ 1\ 0\ 1)_2 + (0\ 1\ 1\ 1)_2$ the data lines of A are to be $(0\ 1\ 0\ 1)_2$, while that of B are to be $(0\ 1\ 1\ 1)_2$. A binary code for addition is to be given on select lines. Suppose $(0\ 0\ 1\ 1)_2$ refers to add operation for this ALU. So the connections are to be made by the programmer as shown in Fig. 1.4.2.

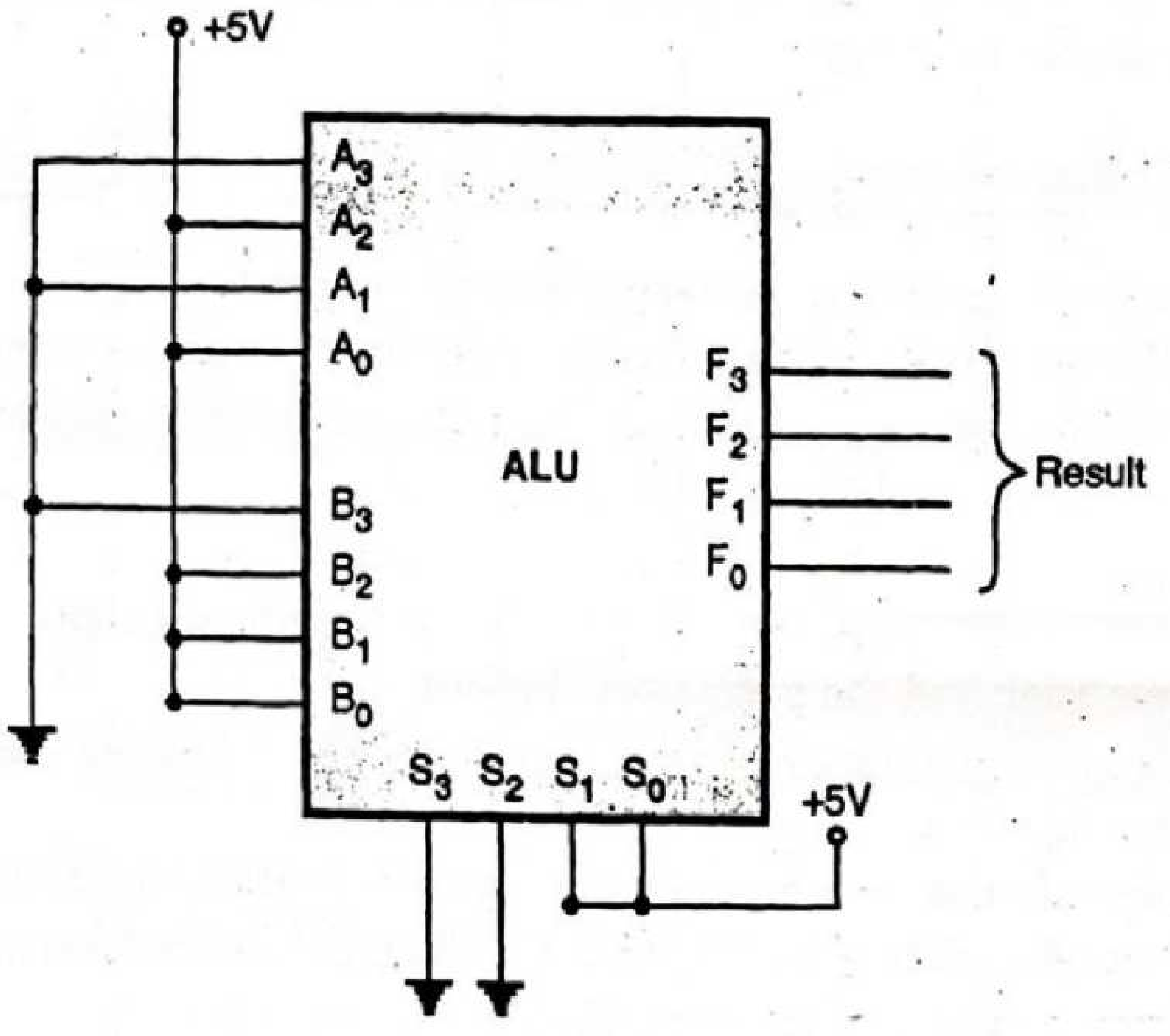


Fig. 1.4.2 : ALU connections to implement 5 + 7

To do these connections, the programmer may take few seconds, while the IC can perform the operation in microseconds. Hence, the time in which the ALU could have performed few million operations, it can perform only one operation. The reason for this delay is the slow speed of the programmer to enter the instruction. Hence if the instructions were stored in a memory and the processor accesses them, it would be much faster. Thus microprocessor should be a combination of ALU and control unit that can access memory.

- Another major feature of a microprocessor is interrupts.
- When many I/O devices are connected to a microprocessor based system, one or more than one of the I/O devices may request for service at any time. The microprocessor stops the execution of the current program and gives service to the I/O devices. This feature is called as **interrupt**.
- It is an external asynchronous input or an instruction that informs the microprocessor to complete the instruction that it is currently executing and fetch a new routine in order to offer service to the I/O device. Once the I/O device is serviced, the microprocessor will continue with the execution of its normal program.

Definitions

1. **Interrupt** : It is a mechanism by which an I/O device (Hardware interrupt) or an instruction (Software interrupt) can suspend the normal execution of the processor and get itself serviced.
2. **Interrupt service routine (ISR)** : A small program or a routine that when executed services the corresponding interrupting source is called as an ISR.
3. **Vectored/Non-vectored interrupt** :If the ISR address of an interrupt is to be taken from the interrupting source itself, it is called as a non-vectored interrupt; else it is a vectored interrupt.

4. **Maskable/Non-maskable interrupt :** Interrupt that can be masked (disabled) or unmasked (enabled) by the programmer is called as maskable interrupt; else it is a non maskable interrupt.

1.5 Microprocessor Systems with Bus Organization

The microprocessor unit performs four operations :

(1) Memory read : reads data (or instructions) from memory.

(2) Memory write : writes data (or instructions) into memory.

(3) I/O read : reads data from input devices

(4) I/O write : writes data to output devices.

- All these operations are part of the communication process between the microprocessor and the peripheral devices.
- In order to communicate with a peripheral the microprocessor requires to perform the following steps :

(1) Identify the peripheral or the memory location (with its address).

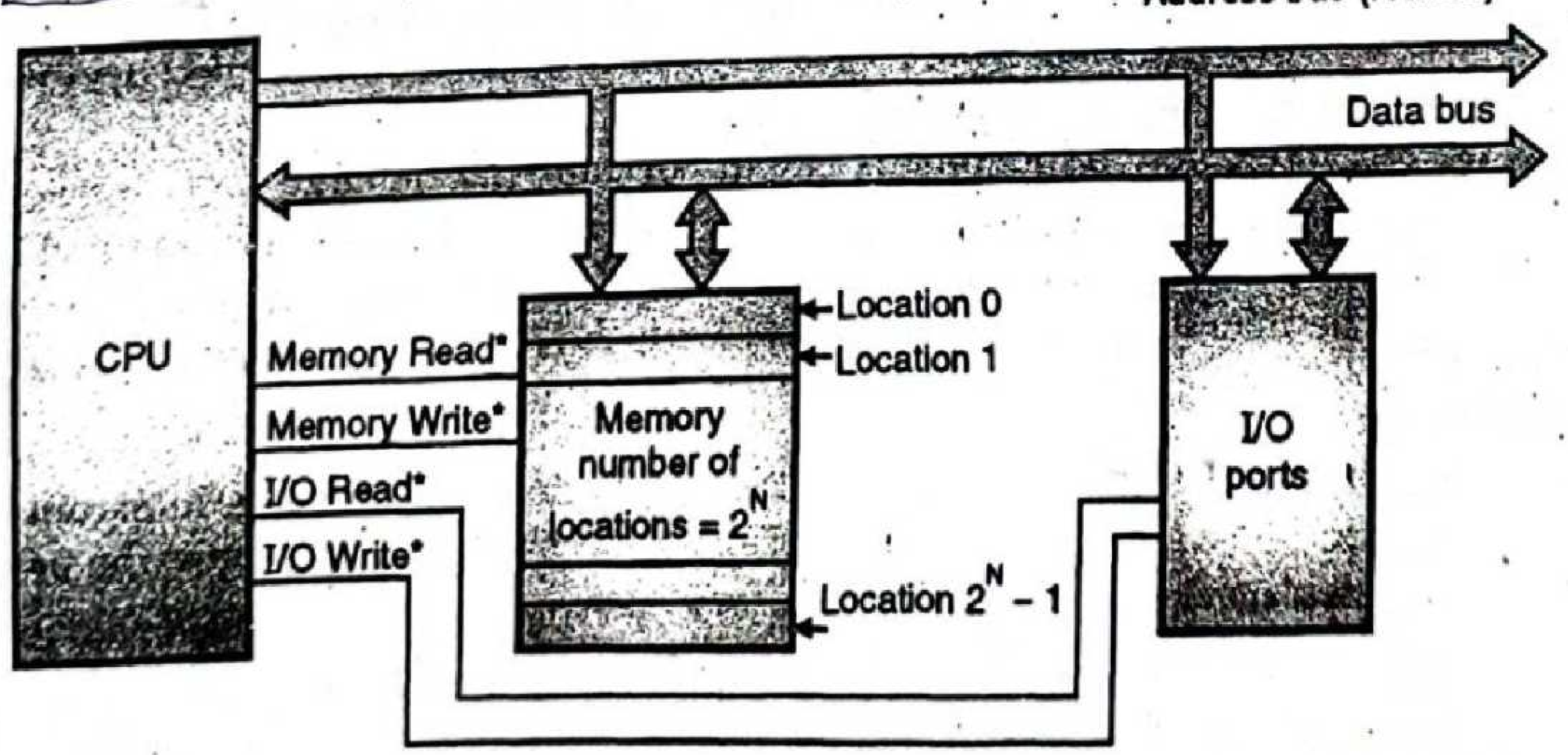
(2) Transfer binary information (i.e. data and instructions).

(3) Provide the timing or synchronization signals.

- The microprocessor unit performs these functions using three sets of communication line called buses; the address bus, the data bus and the control bus.

1.5.1 Address Bus

- The bus over which the CPU sends out the address of the memory location is called as the **address bus**.
- The address bus carries the address of the memory location to be written to or read from.
- The address bus may consist of 16, 20, 24 or 32 parallel signal lines. If there are N address lines, then it can address 2^N memory locations.



*** Read and write signals are a part of control bus**

Fig. 1.5.1 : The three types of buses and their utility

- As shown in Fig. 1.5.1, the address bus is a unidirectional bus (indicated by the arrow).
- The address bus is also used to send the I/O port address on the address bus. When the CPU reads data from or writes data to a port, then it sends the port address out on the address bus.
- The internal structure of memory is shown in the Fig. 1.5.2. The address issued to the memory is decoded by an internal decoder to select one of the location and then this location is accessed via the data bus.

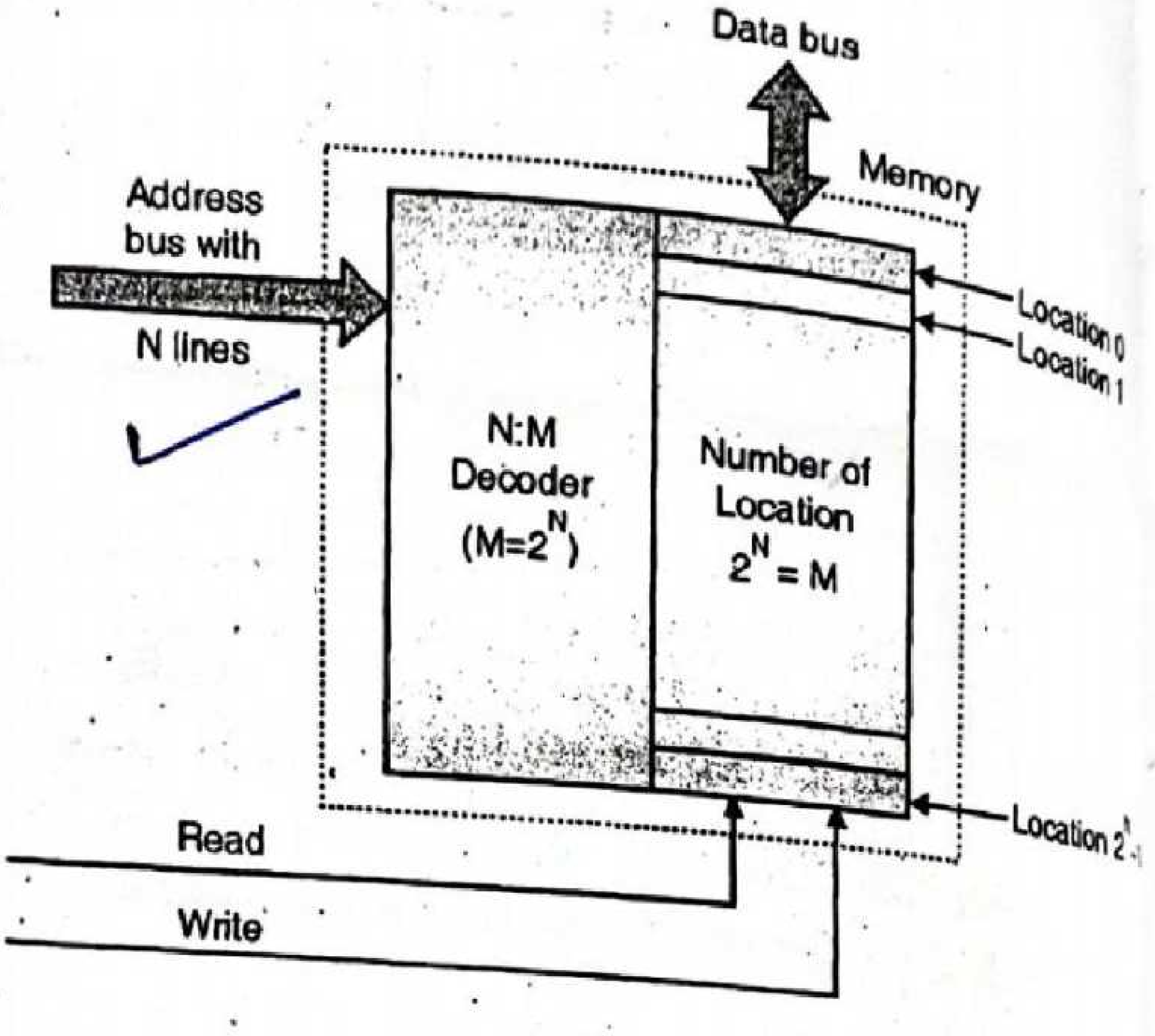


Fig. 1.5.2 : Structure of memory

- If the number of address lines N = 16 then it can address 2^{16} = 65,536 or 64 K memory locations.

$2^0 = 1$	$2^1 = 1$
$2^2 = 4$	$2^3 = 8$
$2^4 = 16$	$2^5 = 32$
$2^6 = 64$	$2^7 = 128$
$2^8 = 256$	$2^9 = 512$
2^{10} = (1024) 1 K (1 Kilo)	2^{20} = 1 K * 1 K = 1 M (1 Mega)
2^{30} = 1 G (1 Giga)	2^{40} = 1 T (1 Tera)

e.g. 1) A processor has 24 address lines, how many memory locations it can access ?

$$2^{24} = 2^4 \times 2^{20} = 16 \times 1\text{ M} = 16\text{ MB}$$

2) A processor has 4 GB memory, how many address lines are required to access this memory.

$$4\text{ GB} = 4 \times 1\text{ G} = 2^2 \times 2^{30} = 2^{32}$$

Hence, 32 address lines are required.

1.5.2 Data Bus

Q. Why is data bus bidirectional ?

- The data bus consists of 8, 16 or 32 parallel lines.

- The data bus is a bi-directional bus. That means the data can get transferred from CPU to memory and vice versa.
- The data bus also connects the I/O ports and CPU. So the CPU can write data to or read data from the memory or I/O ports.
- The number of data lines used in the data bus is equal to the size of data word that can be written or read simultaneously.
- Many devices in the system will connect their outputs to the same data bus. But only one device at a time will have its output enabled, so that there is no data contention (collision of information).
- In order to completely eliminate the possibility of more than one devices putting their outputs simultaneously on the data bus, all the devices getting connected to data bus should have **three state outputs**. viz. logic '0', logic '1' and high impedance state.
- This will make it possible to disable the outputs of devices which should be stopped from connecting their outputs to the data bus.

1.5.3 Control Bus

- The control bus is used for sending control signals to the memory and input/output devices, as shown in Fig. 1.5.1.
- The CPU sends signals on the control bus to enable the outputs of addressed memory devices or I/O port devices.
- Some of the control bus signals are as follows :

1. Memory read	2. Memory write
3. I/O read	4. I/O write

Combined utilization of the three buses

- In practice the CPU needs to make use of all the buses in order to perform a desired operation.
- For example if a byte (8 bits) is to be read from the memory location 18F8 H, then the sequence of operations followed by the CPU is as follows :

Task : Read data from memory location 18F8 H

Step 1 :	CPU sends out "18F8" H i.e. the address of the desired memory location, on the 16 bit address bus.
Step 2 :	CPU sends the memory read signal on the control bus.
Step 3 :	The memory read signal enables the memory device to put the data stored in the location 18F8H onto the data bus. This data travels on the data bus from memory to CPU.

1.6 Microprocessor Architecture and its Operations

Fig 1.6.1 shows the architecture of microprocessor. This architecture is divided in different groups as follows :

• Registers	• Arithmetic and logic unit
• Interrupt control	• Timing and control circuitry

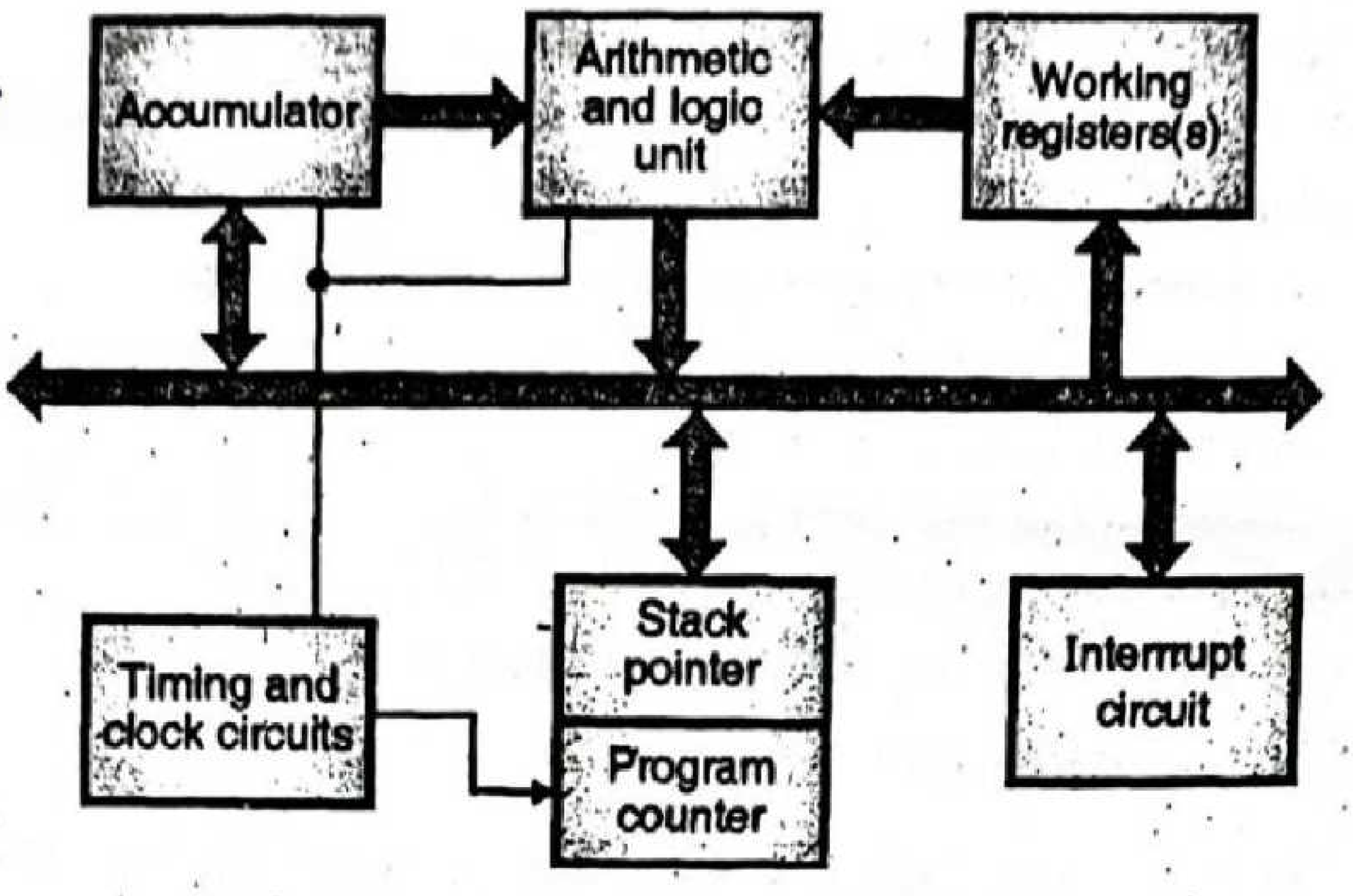


Fig. 1.6.1 : General architecture of a microprocessor

1.6.1 Register Section

- It consists of PIPO (Parallel in parallel out) register as shown in Fig. 1.6.2.
- The register organization affects the length of program, the execution time of program and simplification of the program. To achieve better performance, the number of registers should be large.
- The architecture of microcomputer depends upon the number and type of the registers used in microprocessor. It consists 8-bit registers or 16 bit registers.
- The register section varies from microprocessor to microprocessor.
- The registers are used to store the data and address.

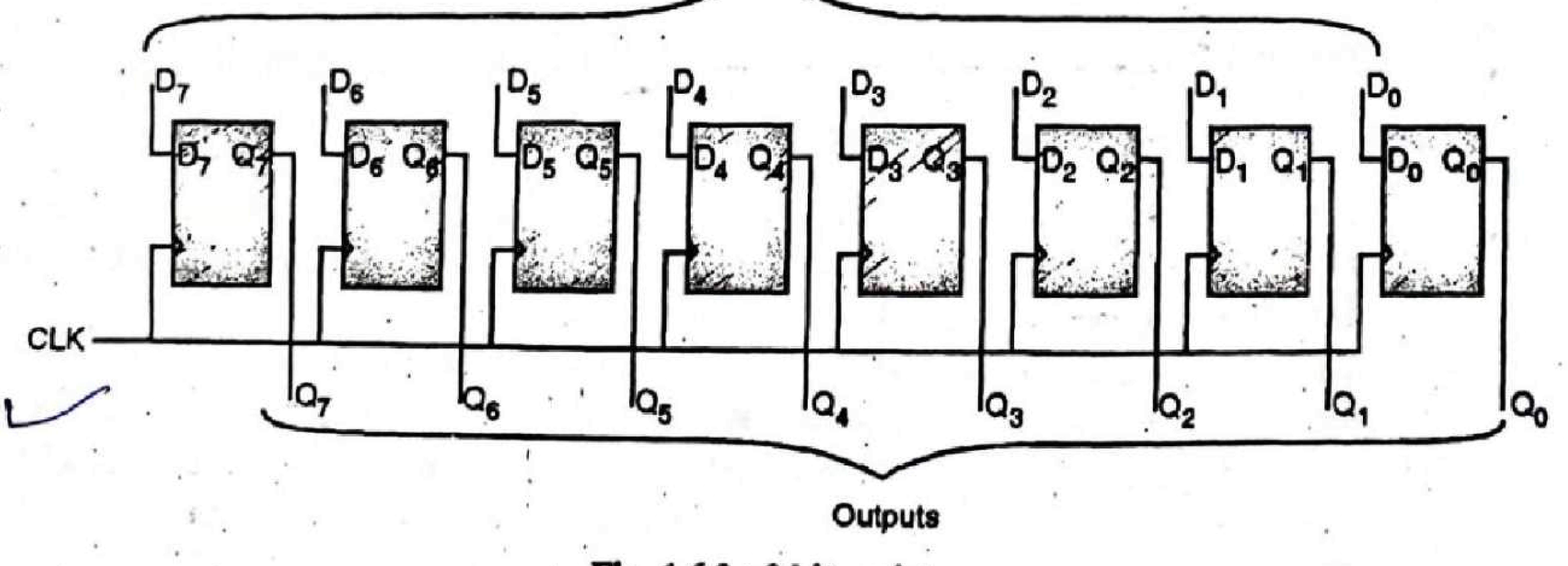


Fig. 1.6.2 : 8 bit register

- These registers are classified as :
  - Temporary registers
  - General purpose registers
  - Special purpose registers.

1.6.2 Arithmetic and Logical Unit

- This section processes data i.e. it performs arithmetic and logical operations.
- It performs arithmetic operations like addition, subtraction and logical operations like ANDing, ORing, EX-ORing, etc.
- The ALU is not available to the user. Its word length depends upon the width of an internal data bus.
- The ALU is controlled by timing and control circuits.
- It accepts operands from memory or register. It stores result of arithmetic and logic operations in register or memory.
- Flag register shows the status of result like, if the result is +ve or –ve, zero or non-zero etc. This flag register is updated by the ALU.
- ALU also looks after the branching decisions.

1.6.3 Interrupt Control

This block accepts different interrupt request inputs. When a valid interrupt request is present it informs control logic to take action in response to each signal.

1.6.4 Timing and Control Unit

- This is a control section of microprocessor made up of synchronous sequential logic circuit.
- It controls all internal and external circuits.
- It operates with reference to clock signal.
- This accepts information from instruction decoder and generates microsteps to execute it. In addition to this, the block accepts clock inputs, performs sequencing and synchronising operations. The synchronization is required for communication between microprocessor and peripheral devices. To implement this, it uses different status and control signals.
- The basic operation of a microprocessor is regulated by this unit.
- It synchronizes all the data transfers.
- This unit takes appropriate actions in response to external control signals.

1.6.5 Microprocessor Operations

- The microprocessor can be programmed to perform different functions like addition, subtraction, multiplication, division etc. All these operations are done on the given data from the instruction set of the microprocessor.
- The instructions are sent to the microprocessor by writing the instructions to the memory. The instructions are entered through an input device like keyboard.
- The microprocessor reads or transfers one instruction at a time, matches it with the instruction set and performs the data manipulation as indicated by the instruction.
- The result can be stored in the memory or sent to output.
- The microprocessor can also respond to external signals like interrupts, DMA (wait for processor to synchronize with slower signals), programmable parallel interfaces etc.

- All the above operations can be classified as
  - Microprocessor initiated operations
  - Internal operations
  - Peripheral (or externally initiated) operations.
- Inorder to process these operations, the microprocessor needs a group of log... circuits called as control signals.

(i) Microprocessor Initiated Operations

The microprocessing unit as responsible for performing four operations.

(i) memory read : it reads the data or instructions from memory.

(ii) memory write : it writes data or instructions into memory.

(iii) I/O read : accepts data from input devices.

(iv) I/O write : sends data to output devices.

All these operations are part of the communication process between th... microprocessing unit and the peripheral devices.

- Inorder to communicate with a peripheral the following steps need to b... performed.

  Step I : Identify the peripheral or the memory location.

  Step II : Transfer the binary information (i.e. data and instructions)... microprocessor.

  Step III : To provide the timing or synchronization signals.

- The microprocessor performs these functions using three sets ... communication buses; the address bus, data bus and control bus as seen... section 1.5.

(ii) Internal Data Operation

The internal architecture of a microprocessor determines how and what operatio... need to be performed with the data. The operations are :

(i) Storing 8-bit data

(ii) Perform arithmetic and logic operations.

(iii) Test for conditions

(iv) Sequence the execution of instructions.

(v) Store data temporarily during execution is the stack.

In order to complete these operations, the microprocessor needs registers, an AL... (arithmetic logic unit), control logic and internal buses for data flow.

(iii) Peripheral or Externally Initiated Operations

The external devices or signals can initiate the following operations for individu... pins on the microprocessor chip are assigned, Reset, Ready, Interrupt, Hold. ... these signals will be discussed in chapter 2.

1.7 Input and Output Devices

- Input/Output devices are means through which the microprocessor u... communicates with the outside world.

- The microprocessor unit accepts the data/instructions from the input devices like keyboard, A/D converter etc. and sends the result to output devices like LEDs, printers, CRTs.
- There are different methods by which the I/O devices can be identified. One method uses 8 bit address while other uses 16 bit address. They are :
  - I/O mapped I/O or peripheral mapped I/O.
  - Memory mapped I/O.

1.7.1 I/O Mapped I/O (Peripheral Mapped I/O)

- In this type of I/O, the microprocessor unit uses eight address lines to identify an input or output device.
- It is an 8 bit numbering system for I/Os used in conjunction with the input and output instructions.
- It is also called as I/O space. It is different from memory space i.e. a 16 bit numbering system.
- The eight address lines have $2^8 = 256$ combinations of addresses. Thus, the microprocessing unit can identify 256 I/O devices with address ranges ranging from 00H to FFH.

I/O Mapped I/O [I/O Port Addressing]

- In this case the I/O device is treated as I/O device only.
- Each I/O device uses eight bits of address lines and control signals $\overline{IOR}$ (Input output read) and $\overline{IOW}$ (Input output write).
- The address bus of 8085 microprocessor is of 16 bits, but for I/O devices only 8 bits address is used. So to implement this the 8 bit address is transferred on both address groups ie A_0 to A_7 and A_8 to A_{16}.
- The contents available are same, so one can use any one group. i.e. A_{16} to A_7 or A_7 to A_0 or combination of the two groups.
- Let us consider an example if address of I/O device is 50 H.

  The contents transferred on A_0 to A_{16} will be as follow :

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0
				50 H										50 H	

So we can use 50 H address given by A_7 to A_0 or A_{16} to A_7 or we can use A_0 or A_8, A_1 or A_9 and so on A_7 or A_{15}. This will be represented as follows :

A_7/A_{15}	A_6/A_{14}	A_4/A_{12}	A_3/A_{12}	A_2/A_{10}	A_2/A_{10}	A_1/A_9	A_0/A_8

- The above representation specifies that and line contents are same. So to address an I/O device we can use $A_0 - A_7$ or $A_8 - A_{15}$.

The steps in communicating with an I/O device are :

(i) The microprocessing unit places an 8 bit address on the address bus, that is decoded by the external logic.

(ii) The microprocessing unit places an 8-bit address on the address bus, which is decoded by external decode logic.

(iii) The microprocessing unit sends a control signal (I/O Read or I/O write) and enables the I/O device.

(iv) Data are transferred using the data bus.

1.7.2 Memory Mapped I/O

- In this type of I/O, the microprocessing unit uses 16 address lines to identify an I/O device. Each I/O device will have 16 bit address.
- The microprocessing unit uses the same control signal ($\overline{MEMR}$ (memory read) or $\overline{MEMW}$ (memory write) and instructions as those of memory.
- The number of I/O device will be 64 KB shared by I/O and memory.

❑❑❑