Unit – V
Microprocessor: Introduction, Block diagram of a microprocessor system, Internal architecture of 8-bit microprocessor CPU, Microprocessor operation **Microcontroller:** Introduction, Block diagram of a microcontroller system.

I. Introduction:

Microprocessor systems are assembled on a single PCB consists of a microprocessor CPU together with a number of specialized support chips.

Typical applications for microprocessor systems include the control of complex industrial processes. Typical examples are based on families of chips such as the Z80CPU plus Z80PIO, Z80CTC, and Z80SIO.

1.Single-chip microcomputers

Putting a CPU, ROM, RAM, and data input/output circuitry into a single IC will make a single-chip microcomputer

A single-chip microcomputer is a complete computer system (comprising CPU, RAM and ROM etc.) in a single VLSI package.

A single-chip microcomputer requires very little external circuitry in order to provide all of the functions associated with a complete computer system

Single-chip microcomputers may be programmed using in-built programmable memories or via external memory chips.

Typical applications of single-chip microcomputers include computer printers, instrument controllers, and displays. A typical example is the Z84C.

2. Microcontrollers

A microcontroller is a single-chip microcomputer that is designed specifically for control rather than general-purpose applications.

They are often used to satisfy a particular control requirement, such as controlling a motor drive.. Typical applications include control of peripheral devices such as motors, drives, printers, and minor sub-system components.

Typical examples are the Z86E, 8051, 68705 and 89C51.

3. PIC microcontrollers (Peripheral Interface Controller)

A PIC microcontroller is a general-purpose microcontroller device that is normally used in a stand-alone application to perform simple logic, timing and input/output control.

PIC devices provide a flexible low-cost solution that very effectively bridges the gap between single-chip computers and the use of discrete logic and timer chips.

PIC microcontrollers are used in 'self-contained' applications involving logic, timing and simple analogue to digital and digital to analogue conversion. Typical examples are the PIC12C508 and PIC16C620.

4. Programmed logic devices

IT is not an example of a microprocessor device,

A programmed logic device (PLD) is a programmable chip that can carry out complex logical operations.

PLDs are capable of replacing a large number of conventional logic gates, thus minimising chip count and reducing printed circuit board sizes.

programmable logic array (PLA) is a kind of programmable logic device used to implement combinational logic circuits. The PLA has a set of programmable AND gate planes, which link to a set of programmable OR gate planes, which can then be conditionally complemented to produce an output.

Typical examples are the 16L8 and 22V10.

5. Programmable logic controllers

Programmable logic controllers (PLC) are microprocessor based systems that are used for controlling a wide variety of automatic processes, an operating an airport baggage handling system.

PLCs are rugged and modular and they are designed specifically for operation in the process control environment. The control program for a PLC is usually stored in one or more semiconductor memory devices.

The program can be entered (or modified) by means of a simple hand-held programmer, a laptop controller, or downloaded over a local area network (LAN). PLC manufacturers include Allen Bradley, Siemens and Mitsubishi.

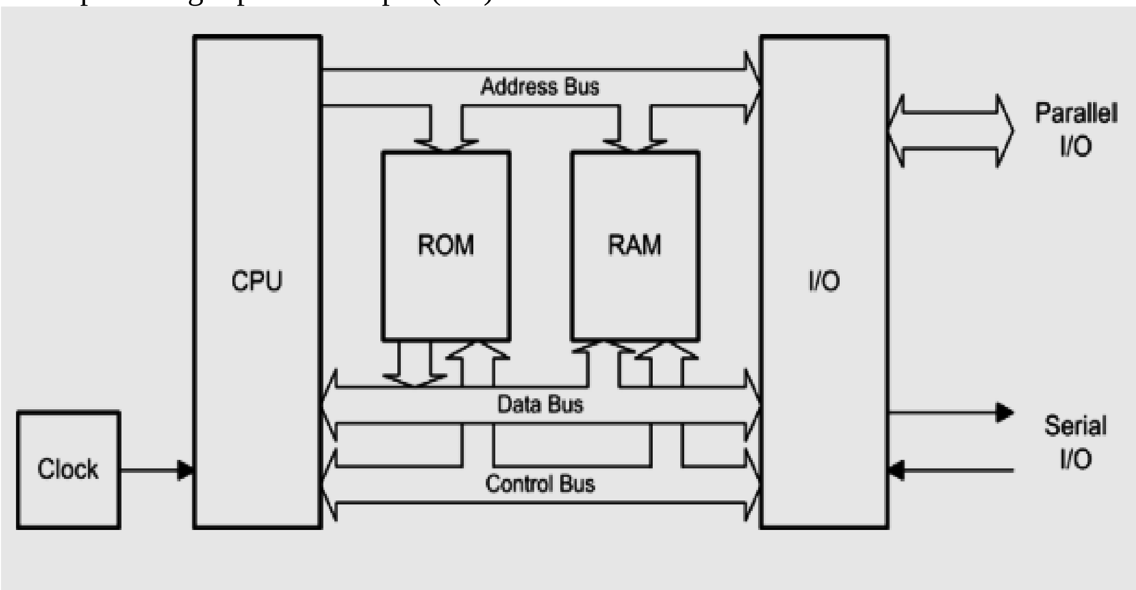
II. Block diagram of Computer System

The basic components of any computer system (see Fig. 11.1) are:

(a) a central processing unit (CPU)

(b) a memory, comprising both 'read/write' and 'read only' devices (commonly called RAM and ROM respectively)

(c) a means of providing input and output (I/O).



- In a microprocessor system the functions of the CPU are provided by a single very large scale integrated (VLSI) microprocessor chip. This chip is equivalent to many thousands of individual transistors.

1. The basic components of the system (CPU, RAM, ROM and I/O) are linked together using a multiple-wire connecting system known as a bus (see Fig. 11.1).

Central Processing Unit (CPU) consists of the following features −

- CPU is considered as the brain of the computer.
- CPU performs all types of data processing operations.
- It stores data, intermediate results, and instructions (program).
- It controls the operation of all parts of the computer.

- CU (short for Control Unit). It regulates the flow of input and output. It's the part that fetches and retrieves the instructions from main memory and later decodes them.
- ALU (short for Artithmetic Logic Unit). The part where all the processing happens. Here is where all mathematic calculations take place, such as addition, subtraction, multiplication, and division, as well as all the logical operations for decision making, such as comparing data.
- Registers. An extremely fast memory location. The data and instructions that are currenlty being processed during the fetch-execute cycle are stored there, for quick access by the processor.

2. **The Input Unit:**

Input devices are the devices which are used to feed programs and data to the computer. The input system connects the external environment with the computer system. The input devices are the means of communication between the user and the computer system. Typical input devices include the keyboard, floppy disks, mouse, microphone, light pen, joy stick, magnetic tapes etc. The way in which the data is fed into the computer through each of these devices is different. However, a computer can accept data only in a specific form. Therefore these input devices transform the data fed to them, into a form which can be accepted by the computer.

Thus the functions of the input unit are :

i. accept information (data) and programs.

ii. convert the data in a form which the computer can accept.
iii. provide this converted data to the computer for further processing.

3. **Output Unit:**

The output devices give the results of the process and computations to the outside world. The output units accept the results produced by the computer, convert them into a human readable form and supply them to the users. The more common output devices are printers, plotters, display screens, magnetic tape drives etc.

4. **Memory Unit**

RAM: Random-Access Memory. They can be read and written at run time by programs.
ROM: Read-Only Memory. They cannot be written by programer. Their contents can be modified only by plugging them into specialized hardware programmers**.**

5. **Buses**

Three different buses are present; these are:

(a) the address bus used to specify memory locations→Address bus
(b) the data bus on which data is transferred between devices→Data Bus
(c) the control bus which provides timing and control signals throughout the system→Control bus

- The number of individual lines present within the address bus and data bus depends upon the particular microprocessor used.
- Signals on all lines, no matter whether they are used for address, data, or control, can exist in only two basic states: logic 0 (low) or logic 1 (high).
- Data and addresses are represented by binary numbers (a sequence of 1s and 0s) that appear respectively on the data and address bus
- Therefore, the largest value of data that can be present on the bus at any instant of time is equivalent to the binary number 11111111 (or 255).
- Similarly, most the highest address that can appear on a 16-bit address bus is 1111111111111111 (or 65,535).

Data representation

Binary numbers – particularly large Data – are not very convenient. To make numbers easier to handle we often convert binary numbers to **hexadecimal (base 16).**

- To make numbers easier to handle we often convert binary numbers to hexadecimal (base 16)
- A group of **four bits** (or single hex. character) is sometimes called a **nibble**.

Table 11.1 Binary, denary and hexadecimal

Binary (base 2)	*Denary (base 10)*	*Hexadecimal (base 16)*
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Example:

Ex: 11.1 Convert hexadecimal A3 into binary.
Solution
,A = 1010 and 3 = 0101. Thus A3 in hexadecimal is equivalent to 10100101 in binary.
Ex 11.2 Convert binary 11101000 binary to hexadecimal.
Solution
1110 = E and 1000 = 8. Thus 11101000 in binary is equivalent to E8 in hexadecimal

- A **byte** of data comprises a group **of eight bits**. Thus a byte can be represented by just two hexadecimal (hex) characters.
- A group of **sixteen bits (a word)** can be represented by four hex characters,
- A group of **thirty-two bits (a double word** by eight hex. characters, and so on).

Data types

- A byte of data can be stored at each address within the total memory space of a microprocessor system.
- Hence one byte can be stored at each of the 65,536 memory locations within a microprocessor system
  having a 16-bit address bus.
- Individual bits within a byte are numbered from 0 (least significant bit) to 7 (most significant bit).
- In the case of 16-bit words, the bits are numbered from 0 (least significant bit) to 15 (most significant bit).
- Negative (or signed) numbers can be represented using two's complement notation where the leading (most significant) bit indicates the sign of the number (1 = negative, 0 = positive).

  For example, the signed 8-bit number 10000001 represents the decimal number -1.

- The range of integer data values that can be represented as bytes, words and long words are shown in Table 11.2.

Table 11.2 Data types

Data type	*Bits*	*Range of values*
Unsigned byte	8	0 to 255
Signed byte	8	−128 to +127
Unsigned word	16	0 to 65,535
Signed word	16	−32,768 to +32,767

Data storage
ROM:

- The semiconductor ROM within a microprocessor system provides storage for the program code and all permanent data that requires storage.
- All of this data is referred to as non-volatile because it can retain stored information even after power is removed

RAM:

- The semiconductor RAM within a microprocessor system provides storage for the transient data and variables that are used by programs.
- Part of the RAM is used by the microprocessor as a temporary store for data while carrying out its normal processing tasks.

➢ Any program or data stored in RAM will be lost when the power supply is switched off or disconnected. The only exception to this is CMOS RAM that is kept alive by means of a small battery. This battery-backed memory is used to retain important data, such as the time and date.

Storage Capacity

The amount of storage provided by a memory device is expressed Kilobytes (Kbyte, Mbyte, Gbyte, Tbyte).

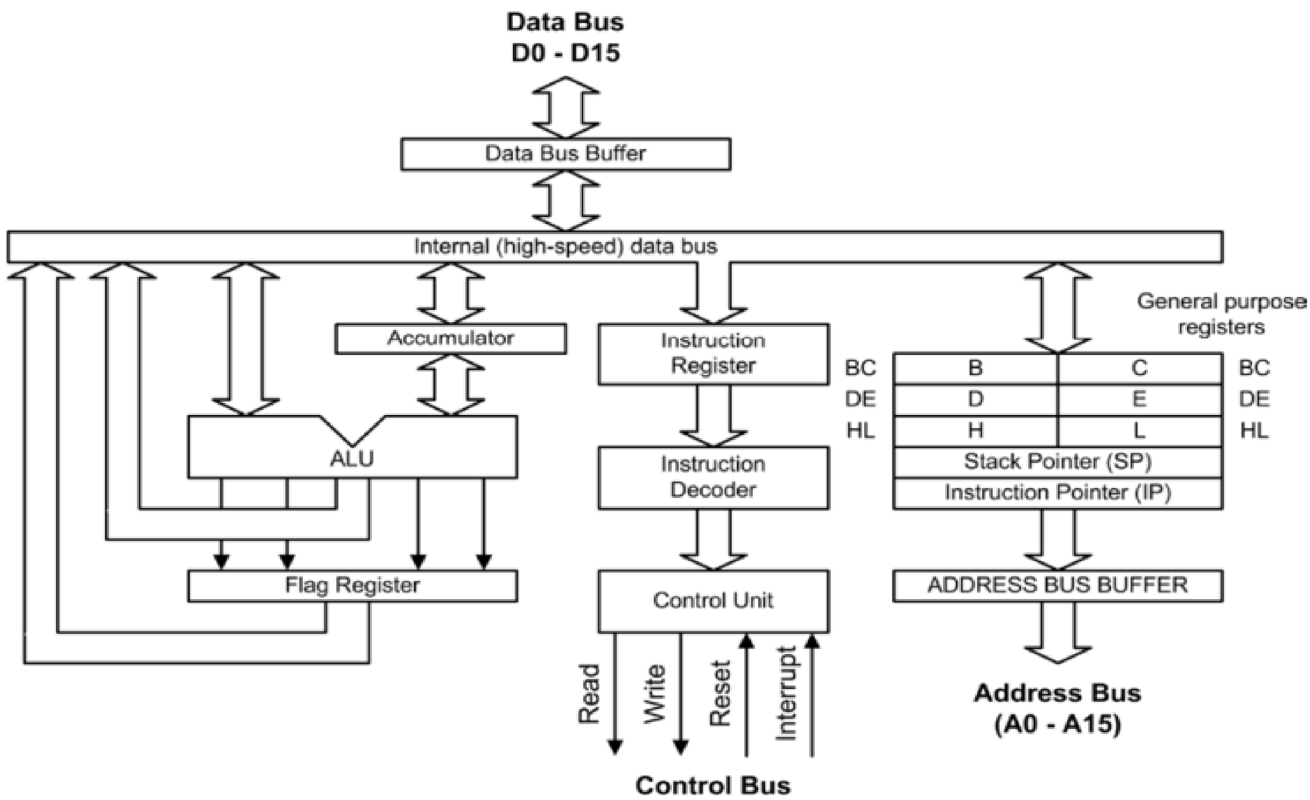
Kilobyte of memory is actually 1,024 bytes (not 1,**000 bytes**). i.e 1,024 happens to be the nearest power of 2 (note that $2^{10} = 1,024$).

The capacity of a **semiconductor ROM** is usually specified in terms of an address range and the number of bits stored at each address.

For example, 2 K × 8 bits (capacity 2 Kbytes), 4 K × 8 bits (capacity 4 Kbytes), and so on.

III. Internal architecture of a typical 8-bit microprocessor CPU

i. The microprocessor central processing unit(CPU) forms the heart of any microprocessor or microcomputer system computer .
ii. The primary function of the microprocessor is that of **fetching, decoding, and executing instructions stored in memory.**As such, it must be able to transfer data **from external memory into its own internal registers** and vice versa.



The main parts of a microprocessor CPU are:

(a) registers for temporary storage of addresses and data;
(b) an arithmetic logic unit (ALU) that performs arithmetic and logic operations;
(c) a unit that receives and decodes instructions; and
(d) a means of controlling and timing operations within the system.

1) Accumulator

➢ The accumulator functions as a source and destination register for microprocessor operations.
➢ As a source register it contains the data that will be used in a particularoperation.
➢ As a destination register it will be used to hold the result of a particular operation. The accumulator **(or A-register)**

2) Arithmetic logic unit (ALU)

- ALU performs arithmetic and logic operations.
- The ALU has two inputs (in this case these are both 8-bits wide).
- One of these inputs is from the Accumulator and the other is taken from the internal data bus via a temporary register (not shown in Fig. 11.3).
- The operations provided by the ALU usually include addition, subtraction, logical AND, logical OR, logical exclusive-OR, shift left, shift right, etc. The result of most ALU operations appears in the accumulator.

3) Flag register (or status register)

- Depending upon the value of the result after any arithmetic and logical operation, the flag bits become set (1) or reset (0).
- The flag register contains a number of individual bits that are set or reset according to the result of an ALU operation.
- **These bits are referred to as flags**.

- The following flags are available in most microprocessors:

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

i. **Zero Flag**:
The zero flag is set when the result of an ALU operation is zero (i.e. a byte value of 00000000)

ii. **Carry Flag:**
The carry flag is set whenever the result of an ALU operation (such as addition or subtraction) generates a carry bit
- During subtraction (A-B),
- if A>B it becomes reset Carry CY=0, and
- if (A<B) it becomes set CY=1.
- Carry flag is also called the borrow flag. bit.

iii. **Auxiliary Carry Flag (AC)**
In an Arithmetic operation when carry is generated from bit D3 to D4 the auxiliary carry is SET to 1, otherwise it is reset to 0

iv. **Parity Flag**
Parity flag is set to 1, if the result stored in accumulator contains even number of 1's. it is even parity
Parity flag is reset to 0, If accumulator contains odd number of 1'S , then it is called as odd parity

v. **Sign Flag (S)** –
After any operation if the MSB (D(7)) of the result is 1, it indicates the number is **negative** and the sign **flag becomes set, i.e. 1. from 80H to FF, sign flag is 1**
If the MSB is 0, it indicates the number is positive and the sign flag becomes **reset i.e. 0**. from **00H to 7F,**

4) Instruction register

i. It provides a **temporary storage location** in which the current microprocessor instruction is held while it is being decoded.
ii. Program instructions are passed into the microprocessor, one at time, through the data bus.
iii. On the first part of each machine cycle, the instruction is **fetched and decoded**.
iv. The instruction is executed on the second (and subsequent) machine cycles.

5) Instruction decoder

i. The instruction decoder is an arrangement of logic gates that acts on the bits stored in the instruction register and determines which instruction is currently being referenced.
ii. The instruction decoder provides output signals for the microprocessor's control unit.

6) Control unit

i. The control unit is responsible for organizing the orderly flow of data within the microprocessor and generating, and responding to, signals on the control bus.
ii. The control unit is also responsible for the timing of all data transfers. This process is synchronized using an internal or external clock signal

7) Control bus

The control bus is a collection of signal lines that are both used to control the transfer of data around the system and also to interact with external devices.

The following are commonly found:

i. **READ** an output signal from the microprocessor that indicates that the current operation is a read operation from memory or I/O device (data enters **into the processor during a read operation.)**
ii. **WRITE** an output signal from the microprocessor that indicates that the current operation is a write operation to memory or I/O device (data goes **out of the microprocessor on a write operation)**
iii. **RESET** a signal that resets the internal registers and initializes the program counter so that the program can be re-started from the beginning
v. **IRQ interrupt request**: Interrupt is signals send by an external device to the processor, to request the processor toperform a particular task or work

8) General-purpose registers

i. The six general-purpose registers to store 8-bit data; these are identified as- B, C, D, E, H, and L.
ii. These can be combined as register pairs – BC, DE, and HL, to perform some 16-bit operation.
iii. These registers are used to store or copy temporary data, by using instructions, during the execution of the program.

9) Stack pointer

i. Microprocessors make use of a region of external random access memory (RAM) known as a stack.
ii. The microprocessor has a 16-bit register known as the **Stack Pointer**.
iii. The function of the stack pointer is to hold the starting address of the stack.
iv. This address can be decided by the programmer.
v. The stack operates on the Last In, First Out (LIFO) principle.
vi. When the main program is interrupted, the microprocessor temporarily places in the stack the contents of its internal registers together with the address of the next instruction in the main program.
vii. When the interrupt has been attended to, the microprocessor recovers the data that has been stored temporarily in the stack together with the address of the next instruction within the main program.
viii. It is thus able to return to the main program exactly where it left off and with all the data preserved in its registers.
ix. The stack pointer is simply a register that contains the address of the last used stack location.

10) Program counter

i. The program counter is a register always it will hold the address of the memory location from where the next instruction for execution will have to be fetched.
ii. Program counter stores the address of the next instruction to be fetched. Thus itis used as pointer to the instruction

11) Data bus (D0 to D7)

i. The external data bus provides a highway for data that links all of the system components (such as random access memory, read-only memory, and input/output devices) together.
ii. In an 8-bit system, the data bus has eight data lines, labelled D0 (the least significant bit) to D7 (the most significant bit) and data is moved around in groups of eight bits, or bytes.
iii. With a sixteen-bit data bus the data lines are labelled D0 to D15, and so on.

12) Data bus buffer

i. The data bus buffer is a temporary register through which bytes of data pass on their way into, and out of, the microprocessor.

ii. The buffer is thus referred to as bi-directional with data passing **out of the microprocessor on a write operation and into the processor during a read operation.**

iii. The direction of data transfer is determined by the control unit as it responds to each individual program instruction.

13) Address bus (A0 to A15)

i. The address bus provides a highway for addresses that links with all of the system components (such as random access memory, read-only memory, and input/output devices).

ii. In a system with a 16-bit address bus, there are sixteen address lines, labelled A0 (the least significant bit) to A15 (the most significant bit).

iii. In a system with a 20-bit address bus there are 20 address lines, labelled A0 to A31, and so on.

IV) Microprocessor operation

i. The majority of operations performed by a microprocessor involve the movement of data.

ii. The program code (a set of instructions stored in ROM or RAM) must itself be fetched from memory prior to execution.

iii. The microprocessor thus performs a continuous sequence of instruction fetch and execute cycles.

iv. The act of fetching an instruction code (or operand or data value) from memory involves a read operation while the act of moving data from the microprocessor to a memory location involves a write operation – see Fig. 11.6.

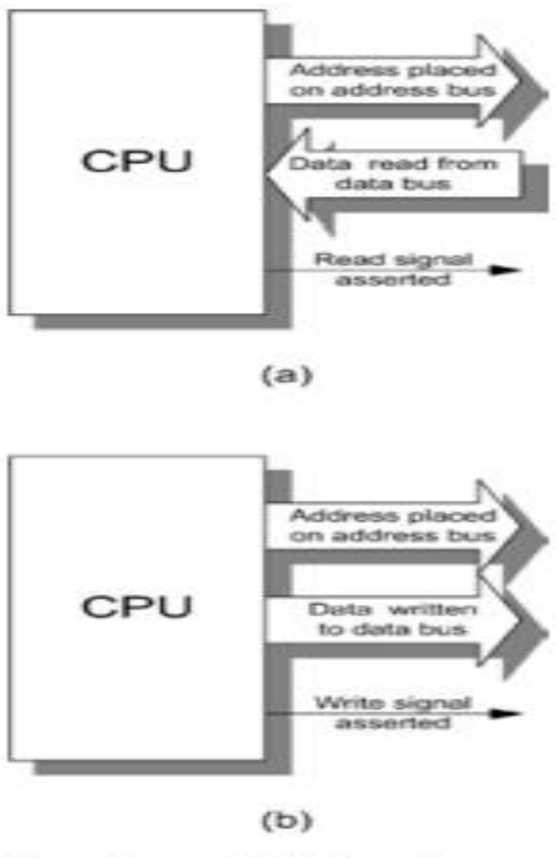


Figure 11.6 (a) Read, and (b) write operations

i. Each cycle of CPU operation is known as a machine cycle.

ii. Program instructions may require several machine cycles (typically between two and five).

iii. The **first machine cycle** in any cycle consists of an instruction fetch (the instruction code is read from the memory) and it is known as the M1 cycle.

iv. Subsequent cycles M2, M3, and so on, depend on the type of instruction that is being executed.

v. This fetch-execute sequence is shown in Fig. 11.7.

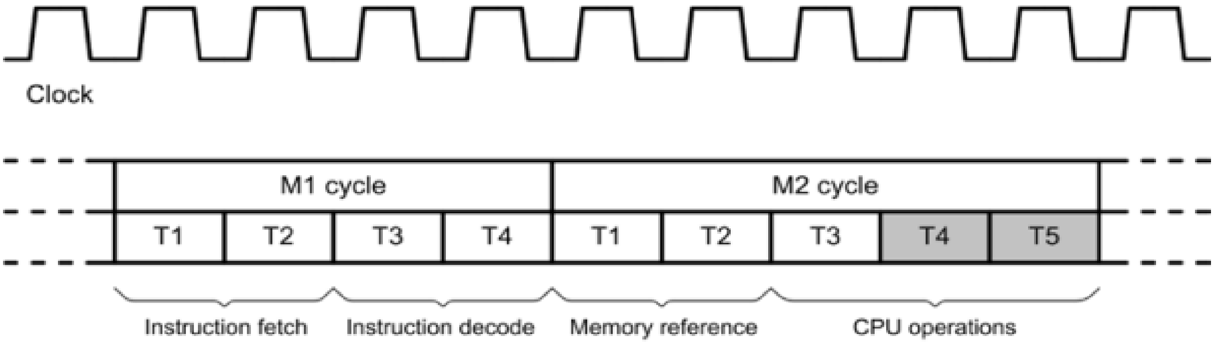


Figure 11.7 A typical timing diagram for a microprocessor's fetch-execute cycle

vi. Microprocessors determine the source of data (when it is being read) and the destination of data (when it is being written) by placing a unique address on the address bus.

vii. The address at which the data is to be placed (during a write operation) or from which it is to be fetched (during a read operation) can either constitute part of the memory of the system or it can be considered to be associated with input/output (I/O).