① Illustrate the function of various register in Processor.

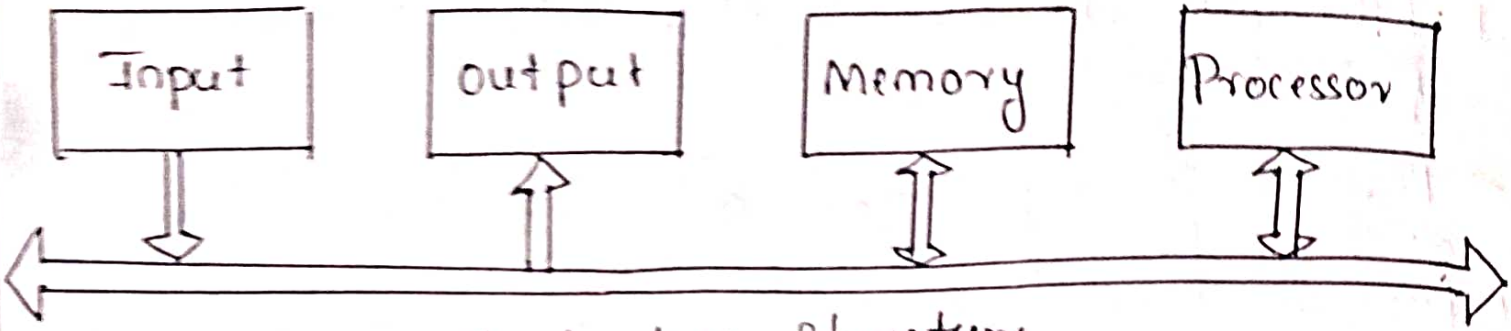
⇒ Registers in a processors are like small storage areas that hold data for quick access.

* Program Counter (PC): The program counter keeps track of the memory address of the next instruction to be executed. It helps the processor fetch the next instruction from memory.
* Instruction Register (IR): The instruction register holds the current instruction being executed. It stores the operation code & data of the instruction.
* Memory address register (MAR): The MAR holds the address of the location to be accessed. It is used during read and write operations to specify the location in memory.
* memory data register (MDR): The MDR contains the data to be written into or read out of the addressed location. It temporarily stores the data during memory operations.
* Index Register: Index registers are used for indexing or addressing memory locations.
* Input / output Register (I/O): I/O registers are used for communication between processor & external devices.

2] What is bus? Explain single bus and multiple bus structure used to interconnect functional units in a Computer.

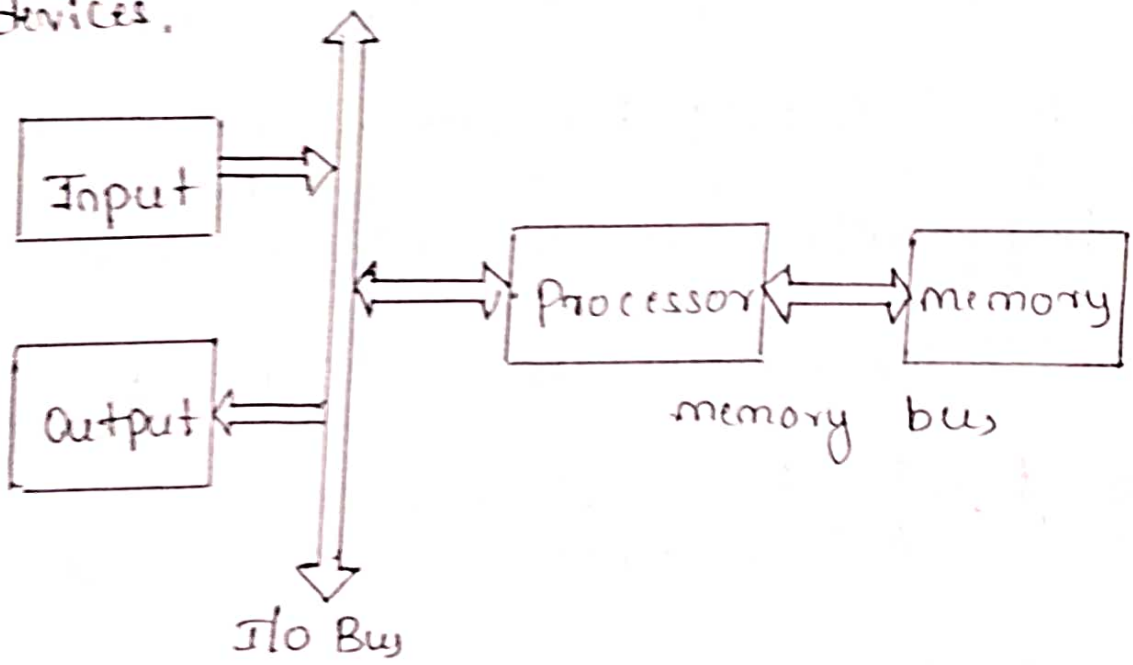
⇒ A group of lines that serves as a connecting path for several devices is called a bus. In addition to the lines that carry the data, the bus must have lines for address and control purpose

* The simplest way to interconnect functional units is to use a single bus



Single bus structure

* All units are connected to this bus. Because the bus can be used for only one transfer at a time, only two units can actively use the bus at any given time.
* Bus control lines are used to arbitrate multiple requests for use of the bus.
* The main virtue of the single-bus structure is its low cost and its flexibility for attaching peripheral devices.



* In a multiple bus structure, one bus is used to fetch instructions while other is used to fetch data, required for execution.
* This leads to better performance but at an increased cost.

3) Explain how the performance of a computer can be increased. What are the measures to improve the performance.

⇒

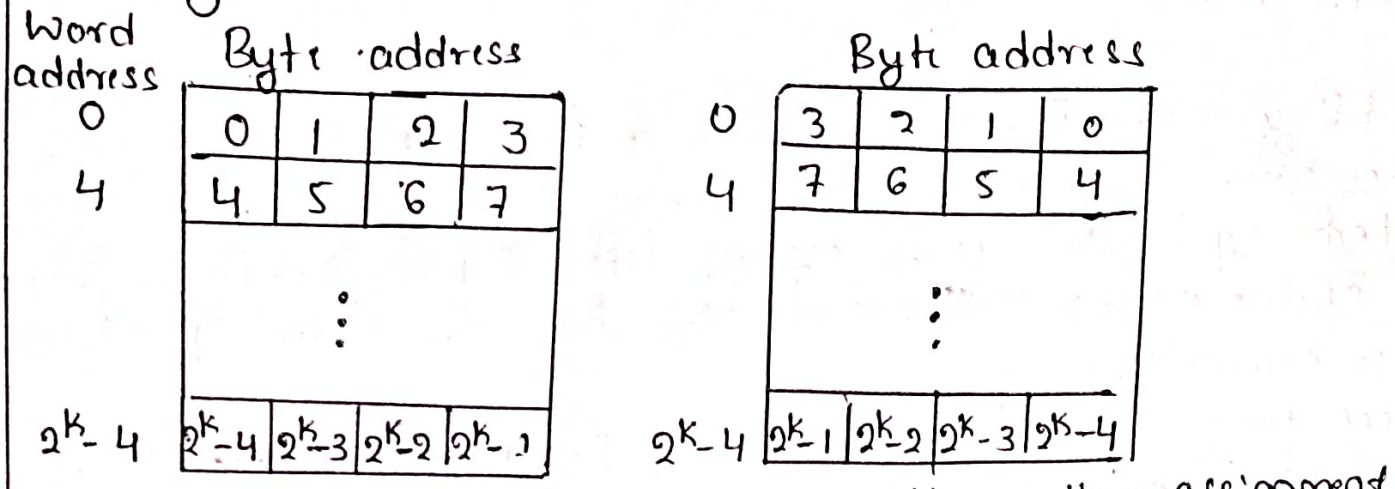
* Adding more RAM or getting a faster processor can make your computer run faster.
* Clean up! Delete unnecessary files & programs to free up space on your hard drive. This can help speed up your computer.
* Keep your operating system & programs up to date. Updates often include performance improvements & bug fixes.
* On laptops, we can change the power settings to prioritize performance over energy saving. This can give our computer a performance boost.

Measures to improve the performance

* The most important measure of the performance of a computer is how quickly it can execute programs.
* For best performance, it is necessary to design the compiler, the machine instruction set, & the hardware in a coordinated way.
* The speed with which a computer executes programs is affected by the design of its hardware and its machine language instructions. Because programs are usually written in a high-level language, performance is also affected by the compiler that translates programs into machine language.
* time is a measure of the performance of the entire computer system.

4) Differentiate big-endian and little endian assignment.

⇒ There are two ways that byte addresses can be assigned across word



(a) Big-endian assignment (b) Little-endian assignment

(a) Big-endian assignment

* Big-endian is used when lower byte addresses are used for the more significant bytes of the word.
* Big-endian are used in commercial machines.

(b) Little-endian assignment

* Little-endian is used for the opposite ordering, where the lower byte addresses are used for the less significant bytes of the word.
* Little-endian are used in commercial machines.

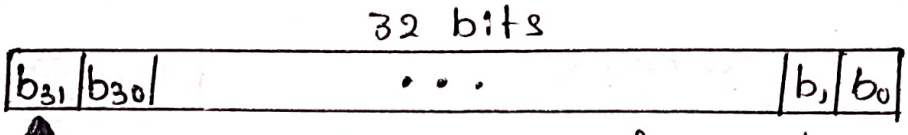
⇒ In both cases, byte addresses 0, 4, 8, ..., are taken as the addresses of successive words in the memory and are the addresses used when specifying memory read and write operations for words.

5) What is byte addresable memory - illustrate with an example.

⇒ * Three basic information quantities: bit, byte & word
* A byte is always 8 bits, but the word length typically ranges from 16 to 64 bits.

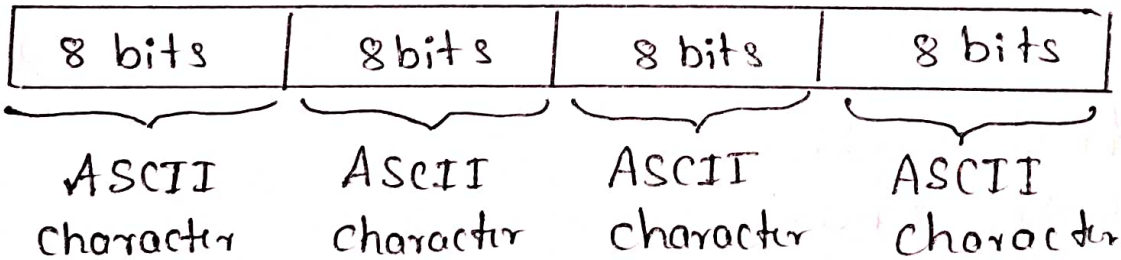
* It is impractical to assign distinct addresses to individual bit locations in the memory.
* The most practical assignment is to have successive addresses refer to successive byte locations in the memory.
* This is the assignment used in most modern computers, and is the one we will normally use..
* The term byte-addressable memory is used for this assignment.
* Byte locations have addresses 0, 1, 2, ... Thus, if the word length of the machine is 32 bits, successive words are located at addresses 0, 4, 8, ..., with each word consisting of four bytes.

Example of encoded information in a 32-bit word.



Sign bit : $b_{31} = 0$ for positive numbers
$b_{31} = 1$ for negative numbers

(a) A signed integer.



(b) Four characters

6] What is addressing mode? Explain various addressing modes with example.

⇒ The different ways in which the location of an operand is specified in an instruction are referred to as addressing mode.

Various addressing modes

Name	Assembler Syntax	Addressing function
Immediate	#value	operand = value
Register	Ri	EA = Ri
Absolute (Direct)	LOC	EA = LOC
Indirect	(Ri) (LOC)	EA = [Ri] EA = [LOC]
Index	X(Ri)	EA = [Ri] + X
Base with index	(Ri, Rj)	EA = [Ri] + [Rj]
Base with index and offset	X(Ri, Rj)	EA = [Ri] + [Rj] + X
Relative	X(PC)	EA = [PC] + X
Auto increment	(Ri)+	EA = [Ri]; increment Ri
Auto decrement	-(Ri)	Decrement Ri; EA = [Ri]

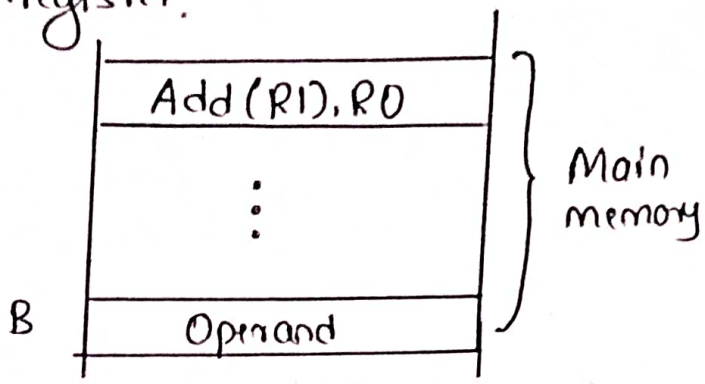
* EA = effective address
* Value = a signed number

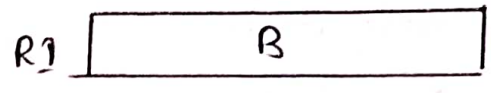
① Immediate mode ⇒ Move #200, R0

② Absolute mode ⇒ Assuming that A & B have been declared earlier as variables and may be accessed using the Absolute mode, this statement may be compiled as follows:

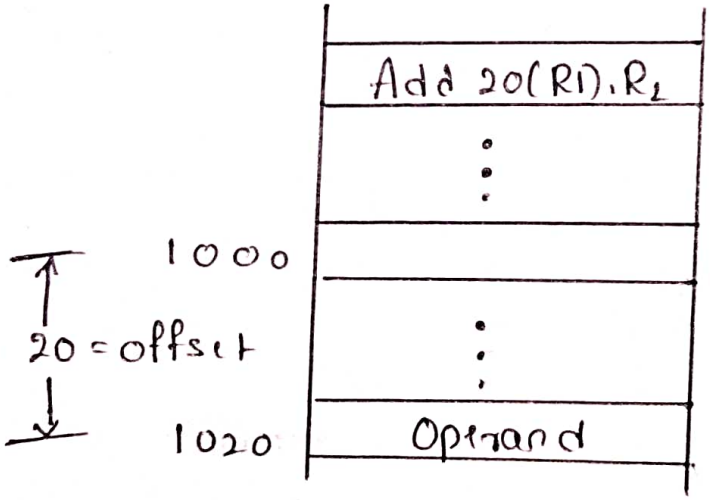
```
Move  B,R1
Add   #6,R1
Move  R1,A
```

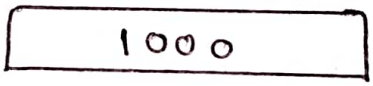
③ Indirect addressing mode - the effective address of the operand is the contents of a register or memory location whose address appears in the instruction.
Example: Indirect addressing through a general-purpose register.





④ Indexing: The effective address of the operand is generated by adding a constant value to the contents of a register.
Example: the index register, R1, contains the address of a memory location, & the value X defines an offset from this address to the location where the operand is found.





Coffset is given as a constant

⑤ Relative Addressing mode: The effective address is determined by the index mode using the PC in place of the general purpose register Ri:

Branch > 0 LOOP

7) Identify the addressing mode used by the following instructions

a) R_0

b) Branch>0 LOOP

c) MOVE #83, R1

d) Add (R_0), R_1

e) MOVE (LOC), R_3

⇒

a) R_0

b) Branch >0 LOOP ⇒ Relative Addressing mode

c) MOVE #83, R1 ⇒ Immediate mode

d) Add (R_0), R_1 ⇒ Indirect addressing mode

e) MOVE (LOC), R_3