Sem 1 - Digital Logic & Computer Organization

Module - 3 - Processing Unit Concepts

#Basic structure of computer

***Computer types:-**A computer is a device that transforms data into meaningful information.
-It processes the input according to the set of instructions provided to it by the user and gives the desired output.
-We can categorize computer in two ways: on the basis of data handling capabilities and size.

- on the basis of size, there are five types of computers;
  1. **Supercomputer:-**Supercomputers are one of the fastest computers currently available.
     -Supercomputers are very expensive
     -It is an extremely fast computer, which can execute hundreds of millions of instructions per second.
     -it is used in the areas of weather forecasting, scientific simulations, (animated) graphics, fluid dynamic calculations, nuclear energy research, electronic design, and analysis of geological data
  2. **Mainframe computer:-**Mainframe computers are designed in such a way that it can support hundreds or thousands of users at the same time.
     -It is also an expensive or costly computer.
     -It has high storage capacity and great performance.
     -It can process a huge amount of data (like data involved in the banking sector) very quickly.
     -It runs smoothly for a long time and has a long life.
  3. **Minicomputer:-**It is a midsize multi-processing system capable of supporting up to 250 users simultaneously.
     -It is light weight that makes it easy to carry and fit anywhere.
     -It is less expensive than mainframe computers.
     -It is fast.
  4. **Workstation:-**Workstation is designed for technical or scientific applications.
     -It consists of a fast microprocessor, with a large amount of RAM and high speed graphic adapter.
     -It is a single-user computer.
     -It generally used to perform a specific task.
     -It is expensive or high in cost.

-They are exclusively made for complex work purposes.
-It provides large storage capacity, with better graphics, and a more powerful CPU when compared to a PC.

5.PC (Personal Computer):-It is also known as a microcomputer.
-It is basically a general-purpose computer and designed for individual use.
-It consists of a microprocessor as a central processing unit(CPU), memory, input unit, and output unit.
-This kind of computer is suitable for personal work.
- In this limited number of software can be used.
-It is smallest in size.
-It is easy to use.
-For example, Laptops and desktop computers.

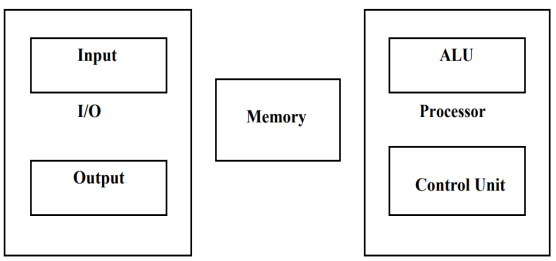
- on the basis of data handling capabilities, there are three types of computer:

1.Analogue Computer:-Analogue computers are designed to *process analogue data*.
-Analogue data is continuous data that changes continuously and cannot have discrete values.
-We can say that analogue computers are used where we don't need exact values always such as speed, temperature, pressure and current.

2.Digital Computer:-Digital computer is designed to perform calculations and logical operations at high speed.
-It accepts the raw data as input in the form of digits or binary numbers (0 and 1) and processes it with programs stored in its memory to produce the output.
-All modern computers like laptops, desktops including smartphones that we use at home or office are digital computers.
-It allows you to store a large amount of information and to retrieve it easily whenever you need it.

3.Hybrid Computer:-Hybrid computer has features of both analogue and digital computer.
-It is *fast like an analogue* computer and has memory and *accuracy like digital computers*

***Functional Units / Basic Functional Unit of a computer:-** A computer consists of five functionally independent main parts **input, memory, arithmetic logic unit (ALU), output and control unit.**



- **Input unit :-**The input unit consists of input devices that are attached to the computer.
-These devices take input and convert it into binary language that the computer understands.

-Some of the common input devices are keyboard, mouse, joystick, scanner etc.

- **Arithmetic and Logic Unit (ALU) :-** The ALU, as its name suggests performs mathematical calculations and takes logical decisions.
  -Arithmetic calculations include addition, subtraction, multiplication and division.
  -Logical decisions involve comparison of two data items to see which one is larger or smaller or equal.
- **Control Unit :-** The Control unit coordinates and controls the data flow in and out of CPU and also controls all the operations of ALU, memory registers and also input/output units.
  -This unit sends control-signals (read/write) to other units and senses their states.
  -Data transfers between processor and memory are also controlled by the control-unit through timing-signals.
  -Timing-signals are signals that determine when a given action is to take place.
- **Memory unit:-** This unit is used to store programs & data.
  -There are 2 classes of storage:
    1) **Primary-storage** is a fast-memory that operates at electronic-speed.
       -programs must be stored in the memory while they are being executed.
    2) **Secondary-storage** is used when large amounts of data & many programs have to be stored.
       Eg: magnetic disks and optical disks(CD-ROMs).
  -The memory contains a large number of semiconductor storage cells(i.e. flip-flops), each capable of storing one bit of information.
- **Output Unit :-** The output unit consists of output devices that are attached with the computer.
  -It converts the binary data coming from CPU to human understandable form.
  -The common output devices are monitor, printer, plotter etc.

#Performance:-the most important use of a computer is how quickly it can execute programs.
-Three factors affect performance they are;

- Hardware design
- Instruction set
- compiler

-The time to execute a program depends on the hardware involved in the execution of individual machine instructions.

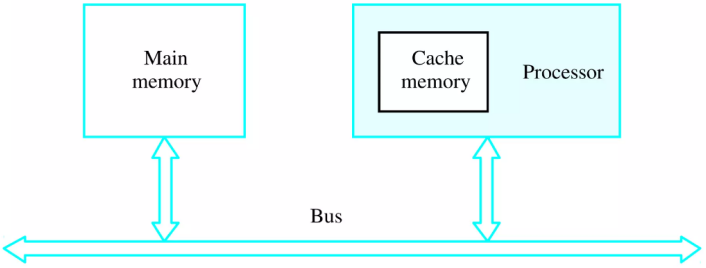


Figure 1.5. The processor cache.

-The processor and a relatively small cache memory can be fabricated on a small integrated circuit chip
-A program will be executed faster if the movement of instructions and data between the main memory and the processor is minimised ,which is achieved by using the cache.

***processor clock:-**processor circuits are controlled by a timing signal called a **clock**.
-The clock defines regular time intervals called **clock cycles**.
-The execution of each instruction is divided into several steps,each of which completes in one clock cycle.
-The length P of one clock cycle is an important parameter that affects processor performance.
-Its inverse is the **clock rate R=1/P**, which is measured in cycle per second.
-The term cycle per second is called **hertz (Hz).**

***Basic performance equation:- T:** - processor time required to execute a program that has been prepared in high-level language.
- **N:**- number of actual machine language instructions needed to complete the execution (note: loop)
- **S:**-average number of basic steps needed to execute one machine instruction. Each step completes in one clock cycle
- **R:**- clock rate.
-The program execution time is given by ⟶ $T = \frac{N \times S}{R}$

-This is often referred to as the basic performance equations.
-To achieve high performance , the computer designer must seek ways to reduce the value of T.
-which means reducing N and S ,and increasing R.
-the value of N is reduced if the source program is compiled into fewer machine instructions.
-The value of S is reduced if instructions have a smaller number of basic steps to perform or if the execution of instructions is overlapped.
-Using a higher frequency clock increases the value or R.
-which means that the time required to complete a basic execution step is reduced.

#Machine Instructions and Programs:-Machine Instructions are commands or programs written in machine code of a machine (computer) that it can recognize and execute.

***Memory Locations and addresses:-**The memory consists of many millions of storage cells, each of which can store a bit of information having the value 0 or 1.
-Because a single bit represents a very small amount of information.
- the memory is organised, so that a group of n bits can be stored or retrieved in a single, basic operation.
- Each group of n bits is referred to as a word of information, and n is called the word length. Or Data is usually accessed in n-bit groups. n is called word length.
- The memory of a computer can be schematically represented as a collection of words as shown in Figure 2.5—----👇
-If the word length of a computer is 32 bits, a single word can store a

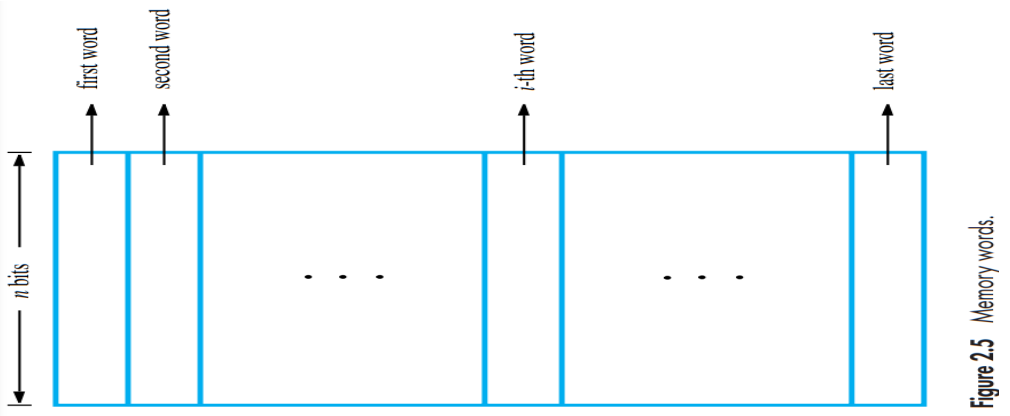


Figure 2.5 Memory words.

32-bit 2's-complement number or four ASCII characters, each occupying 8 bits, as shown in Figure 2.6

-A unit of 8 bits is called a byte.

-To retrieve information from memory ,ethiter for one word or one byte(8-bit),addresses for each location are needed.

-A k-bit address memory has 2^k memory locations, namely 0- 2^k -1 , called memory space.

-For example, a 24-bit address generates an address space of 224 (16,777,216) locations.

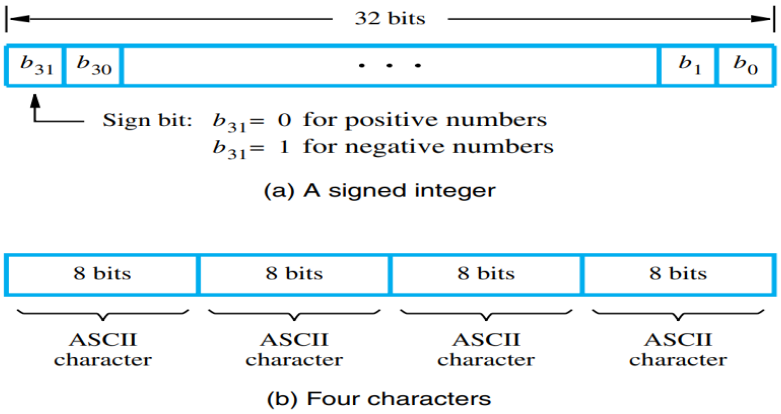


Figure 2.6 Examples of encoded information in a 32-bit word.

This number is usually written as 16M (16 mega), where 1M is the number 220 (1,048,576).

- 24-bit memory: 2^{24} = 16,777,216 = 16M (1M=2^{20})
- 32-bit memory: 2^{32} = 4G (1G=2^{30})
- 1K(kilo)=2^{10}
- 1T(tera)=2^{40}

***Byte Addressability:-** Bytes are always 8 bits.

-It is impossible to assign distinct addresses to individual bit locations in the memory.

-Successive byte locations in the memory have successive addresses.

-memory is byte -addressable

-thus ,successive byte locations have addressees 0,1,2,3,4…

-If the word length of the machine is 32 bits, successive words are located at the aligned addresses 0,4,8,12..

***Big- ending and little- ending assignments:-**There are two ways that byte addresses can be assigned across words, they are;

-> **Big -ending:**- higher byte addresses are used

For the less significant bytes of a word.

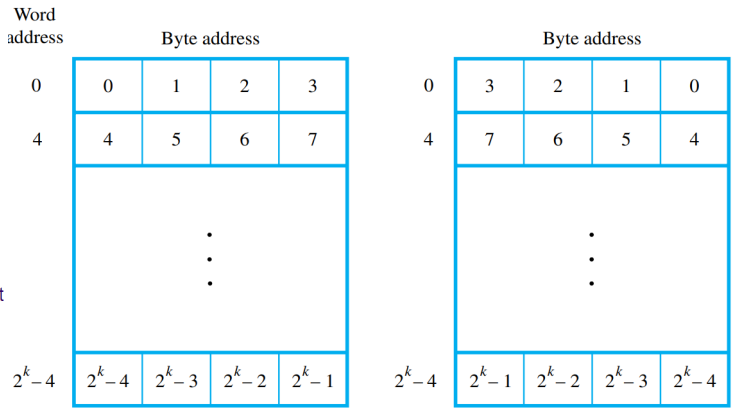
-> **Little -ending:**-lower byte addresses are used\For the less significant bytes of a word.

-Both little-endian and big-endian assignments are used in commercial machines.

or

Big-Endian: lower byte addresses are used for the most significant bytes of the word

Little-Endian: opposite ordering. lower byte addresses are used for the less significant bytes of the word



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

***Memory Operations:-** Both program instructions and data operands are stored in the memory.

-To execute an instruction, the processor control circuits must transfer the instruction from the memory to the processor.

-Two basic operations are needed for memory they are ; Load (or Read or Fetch) and Store (or Write).

- **Load:-**The Load operation transfers a copy of the content of a specific memory location to the processor.
  -The memory content remain unchanged.
  -To start a Load operation, the processor sends the address of the desired location to the memory and requests that its contents be read.
  -To start a Load operation, the processor sends the address of the desired location to the memory and requests that its contents be read.
  -The memory reads the data stored at that address and sends them to the processor.
- **Store:-**The Store operation transfers an item of information from the processor to a specific memory location,
  -And destroying the former contents of that location.
  -The processor sends the address of the desired location to the memory, together with the data to be written into that location.

-An information item of either one word or one byte can be transferred between the processor and the memory in a single operation.

-The processor contains a small number of registers, each capable of holding a word.

-These registers are either the source or the destination of a transfer to or from the memory.

-When a byte is transferred, it is usually located in the low-order (rightmost) byte position of the register.

***Instructions and Instruction sequencing:-**A computer must have instructions capable of performing four types of operations they are;

- Data transfers between the memory and the processor registers (MOV, PUSH, POP, XCHG).
- Arithmetic and logic operations on data (ADD, SUB, MUL, DIV, AND, OR, NOT).

- Program sequencing and control (CALL.RET, LOOP, INT).
- I/O transfers (IN, OUT).

-> **Register transfer notations(RTN):-** describe the transfer of information from one location in a computer to another.
-Possible locations that may be involved in such transfers are memory locations, processor registers, or registers in the I/O subsystem.
-We identify a location by a symbolic name standing for its hardware binary address (LOC,R0,..)
-Contents of a location are denoted by placing square brackets around the name of the location (R1←[LOC], R3←[R1]+[R2])
-The contents of a location are denoted by placing square brackets around the name of the location. Thus, the expression R1 ← [LOC] means that the contents of memory location LOC are transferred into processor register R1.

Eg; R3<–[R1]+[R2]

- Right hand side of RTN-denotes a value.
- Left hand side of RTN-name of a location. { e table important alle } Note that the right-hand side of an RTN expression always denotes a value, and the left-hand side is the name of a location where the value is to be placed

Location	Hardware Binary Address	Example	Description
Memory	LOC, PLACE, NUM	R1 ← [LOC]	Contents of memory-location LOC are transferred into register R1.
Processor	R0, R1 ,R2	[R3] ← [R1]+[R2]	Add the contents of register R1 &R2 and places their sum into R3.
I/O Registers	DATAIN, DATAOUT	R1 ← DATAIN	Contents of I/O register DATAIN are transferred into register R1.

->Assembly language notations(ALN):-Need another type of notation to represent machine instructions & pgms ,Use assembly language format.
Eg;

- Move LOC, R1 (contents of LOC unchanged & R1 changed)
- Add R1, R2, R3 (Adding contents of R1, R2 & place sum in R3).

Or

Assembly Language Format	Description
Move LOC, R1	Transfer data from memory-location LOC to register R1. The contents of LOC are unchanged by the execution of this instruction, but the old contents of register R1 are overwritten.
Add R1, R2, R3	Add the contents of registers R1 and R2, and places their sum into register R3.

->Basic instruction types:-Three address instructions– Add A,B,C.
-A, B-source operands
-C-destination operands

- Two address instructions-Add A,B
  B <–[A] + [B]
- One address instructions –Add A
  -Add contents of A to accumulator & store sum back to accumulator.

- Zero address instructions

  -Instruction store operands in a structure called push down stack.

Or

Instruction Type	Syntax	Example	Description	Instructions for Operation C<-[A]+[B]
Three Address	Opcode Source1,Source2,Destination	Add A,B,C	Add the contents of memory-locations A & B. Then, place the result into location C.	
Two Address	Opcode Source, Destination	Add A,B	Add the contents of memory-locations A & B. Then, place the result into location B, replacing the original contents of this location. Operand B is both a source and a destination.	Move B, C Add A, C
One Address	Opcode Source/Destination	Load A	Copy contents of memory-location A into accumulator.	Load A Add B Store C
		Add B	Add contents of memory-location B to contents of accumulator register & place sum back into accumulator.	
		Store C	Copy the contents of the accumulator into location C.	
Zero Address	Opcode [no Source/Destination]	Push	Locations of all operands are defined implicitly. The operands are stored in a pushdown stack.	Not possible

->Instruction execution & straight line sequencing:-

-Task C=A+B

- **In RTN representation**

-C←[A]+[B]

- **In Assembly Notation representation**

-MOV A,R0 (A ill ulla value R0 registerillakku move cheythu)

-ADD B,R0 (R0 ill ulla A um B um add cheythu. Result store cheyunnathu R0 register ill ayirikkum)

-MOV R0,C (R0 ill ulla content C illakku move cheyum)

-we consider a 32 bit word length memory (32 bit =4 bytes ,oru location illum 32 bit or 4 byte store cheyam)

-Also consider a byte addressable memory (oru location ill ulla byte ina nammuke seperate ayittu access cheyan pattum)

-Here memory address are i,i+1,i+2,i+3,i+4….

-1st location ill ulla 4 byte address are i,i+1,i+2,i+3.

-2nd location has 4 byte address they are i+4,i+5,i+6,i+7.

-here 1st instruction address is i, 2nd instruction address is i+4.

or

-The processor control circuits use information in PC to fetch & execute instructions one at a time in order of increasing address.

-This is called straight line sequencing.

-Executing an instruction-2 phase procedures.

-1st phase–"instruction fetch"-instruction is fetched from

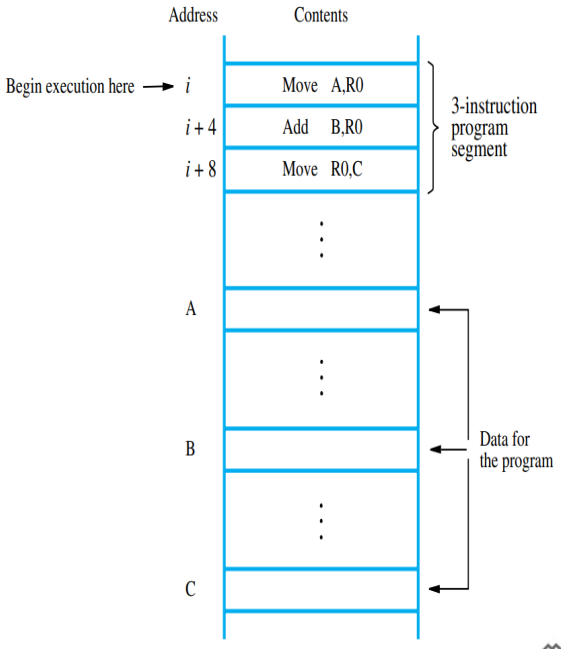


Figure 2.8 A program for C ← [A] + [B].

memory location whose address is in PC.
-This instruction is placed in instruction register in processor
-2nd phase-"instruction execute"-instruction in IR is examined
to determine which operation to be performed.

-> Branching:-instead of using a list of Add instructions, it is possible to place a single Add instruction in a program loop
-Loop is a straight –line sequence of instructions executed as many times as needed
-It starts at location LOOP and ends at the instruction Branch>0
-The no. of entries in the list n, is stored in the memory location N
-R1, is used as counter to determine the no. of times the loop is executed
-Contents of location N are loaded into R1 at the beginning of the program
-Within the body of the loop, the instruction Decrement R1 reduces the contents of R1 by 1 each time through the
loop
-Execution of the loop is repeated as long as the result of the decrement operation is > 0
-Branch-type of instruction loads a new value into program counter.
-So processor fetches & executes instruction at this new address called "branch target"
-Conditional branch-causes a branch if a specified condition is satisfied.
-E.g. Branch>0 LOOP –conditional branch instruction .it executes only if it satisfies condition.

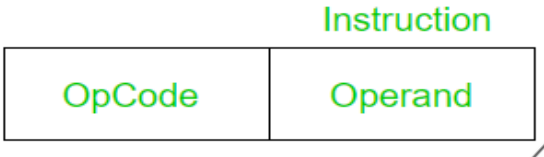
***Addressing Modes:**-The operation field of an instruction specifies the operation to be performed.
-This operation must be executed on some data stored in computer registers or memory words.
-The way the operands are chosen during program execution is dependent on the addressing mode of the instruction.
-"**The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually referenced".**
-Computers use addressing mode techniques for following purposes they are;

1. To give programming versatility (creativity) to the user by providing such facilities as pointers to memory, counters for loop control, indexing of data, and program relocation.
2. To reduce the number of bits in the addressing field of the instruction.

->Types of Addressing Modes

1. **Implied/ Implicit Addressing Mode:-**In the implied mode, the operands are implicitly specified in the definition of instruction.
   -eg:the instruction "complement accumulator.

2. **Immediate Addressing Mode:-**Operand is



Immediate Addressing Mode

specified in the instruction

-It contains an operand field rather than an address field

-The operand field contains the actual operand to be used in.

- They are useful for initializing registers to a constant value.

-Eg : ADD 10 will increment the value stored in the accumulator by 10.

: MOV R #20 initializes register R to a constant value 20.

-It is a fast method. But the disadvantage is that it has a limited range.

3. **Direct Addressing Mode:-**In direct addressing mode, the address field contains the address of the operand.

-EA = A A-content of an address field of the instruction

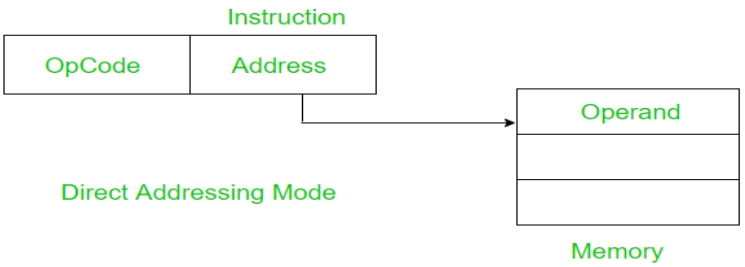
EA- (Effective Address)

-simple

-it also have limited address space.

-Eg:ADD X will increment the memory location X.

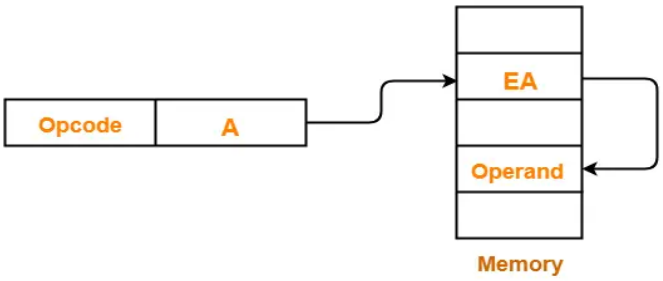
AC ←



4. **Indirect Addressing Mode:-**The address field of the instruction specifies the address of memory location that contains the effective address of the operand.

-Two references to memory are required to fetch the operand.

-Eg:



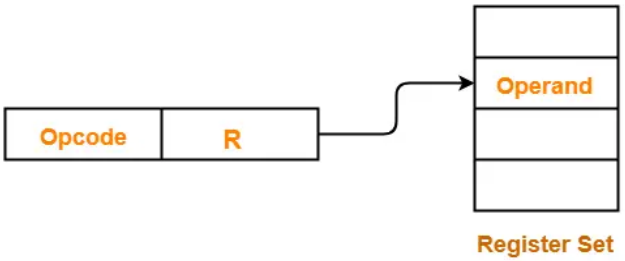
-Eg:ADD X will increment the value stored in the accumulator by the value stored at memory location specified by X.

AC ← AC + [[X]]

5. **Register Mode:-**The operand is contained in a register set.

-The address field of the instruction refers to a CPU register that contains the operand.

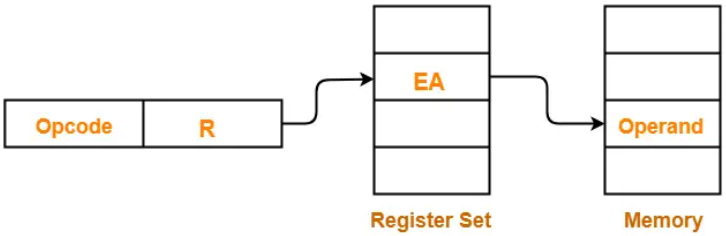
-No reference to memory is required to fetch the operand.



-Eg:ADD R will increment the value stored in the accumulator by the content of register R.

$$AC \leftarrow AC + [R]$$

6. **Register Indirect Addressing Mode:-**The address field of the instruction refers to a CPU register that contains the effective address of the operand.
   -Only one reference to memory is required to fetch the operand.



-Eg:ADD R will increment the value stored in the accumulator by the content of memory location specified in register R.

$$AC \leftarrow AC + [[R]]$$

7. **Auto decrement or the Auto increment Mode** :- The auto increment mode is similar to register indirect mode except that the register is incremented after the execution of the instruction.

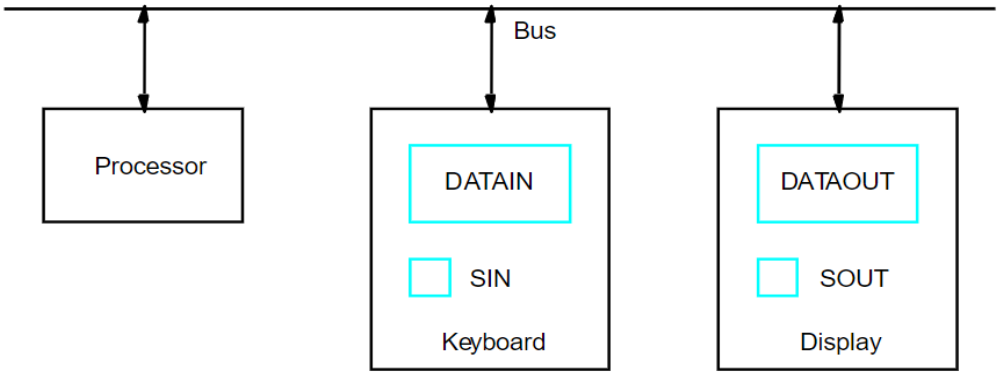
$$R=R+1$$

-The auto decrement mode,the content of register is decremented before the execution of instruction.

$$R=R-1$$

***Basic Input Output Operations:**-when we give input with is second it will show the output in the display.

-we didn't know what is the process behind it .



-Precessor who is incharge to do all the operations.
-along with processor two more things are connected to the bus.
-Here keyboard is the input device and Display is the output device.
-There are two register are associated with it ,they are; DATAIN and DATAOUT. And its size is 8 bit .
-DATAIN is a input register which is associated with keyboard.
-DATAOUT is a output register which is associated with display.
-DATAIN and DATAOUT are buffer registers it means that it can store data temporarily.
-It have two flags called SIN (status in flag) and SOUT (status out flag).

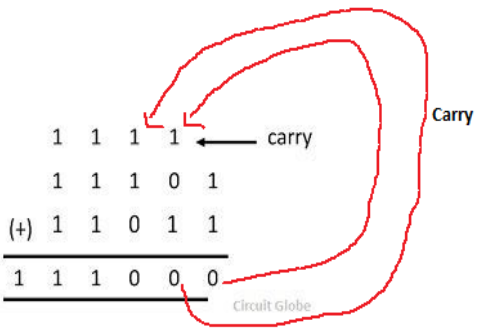
->working
-when ever a user press 'A' in the keyboard .
-Then 'A' will be stored in DATAIN register.
-The value of SIN can hold only 1bit either 0 or 1 ,default value is 0.
-When 'A' is stored in the DATAIN register then the value in SIN become 1

-When ever the SIN become 1 then the letter 'A' passes to the processor.
-And processor transmit the letter 'A' through the bus.
-And that letter 'A' will be stored in the DATAOUT buffer register.
-Know the letter 'A' have been erased (removed permanently) from the DATAIN buffer register.
-And the SIN value from 1 it again change to 0.

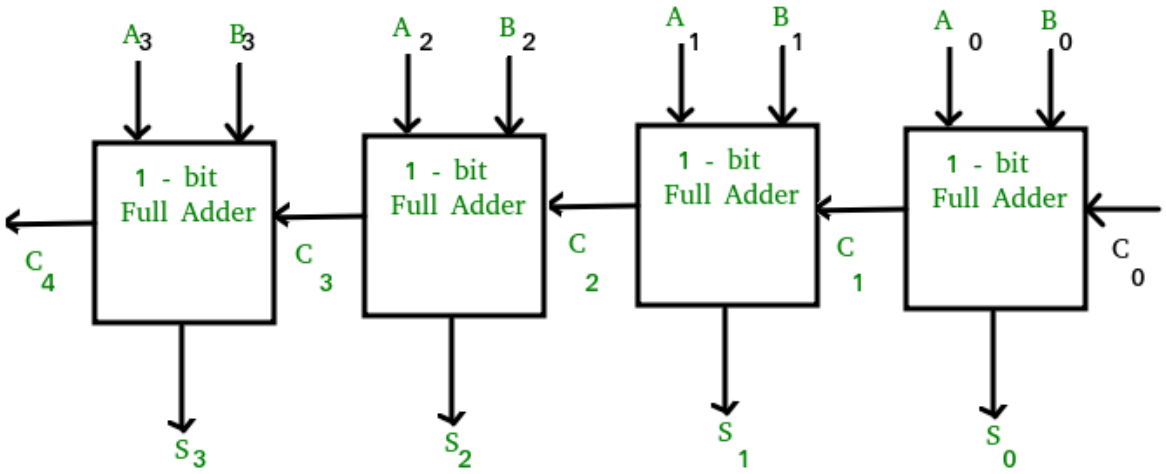
-Know the data is in the DATAOUT buffer register.
-And the value of SOUT from 0 to 1.
-At last the letter 'A' is removed from the DATAOUT buffer register by the processor and then the value is then displayed on the monitor.
-At that time the value in SOUT become 1 to 0.

#Computer Arithmetic

***Fast Adders:**-If an n-bit ripple-carry adder is used in the addition /subtraction unit may have too much delay to develop the output.
-waiting for carry makes delay



-because in n bit ripper carry adder cannot perform the addition operation without the carry from the previous stage (if any carry)



(fig of 4 bit adder)

-so it has to wait for the carry from the previous stage.

-it leads to an time delay in the addition process and it limits the operations speed.

-so the output is only after the carry is performed through each of the address that is the major draw back.

-Example to get S1(sum1) we need s0(sum 0)+ c1(carry 1) to get the output of S1.

-Eg:In a 4 -bit adder each full adder has 20 ns delay.

-Then the total time required for the final output

=4x20ns => 80ns

:If n=16 bit

-Then the total time required for the final output

=16x20ns=> 320ns

-In order to speed up the adder we use a fast adder.

-so we use carry look ahead adder to speed up the adder.

-carry look ahead adder is a type of fast adder.

→**Carry look ahead adder:**-it is a fast adder.

-Here two carry functions are used they are

- Carry propagation (Pi) $P_i = A_i \oplus B_i$
- Carry generation (Gi) $G_i = A_i B_i$

-ethinta problem net ill nokkikonam

***Signed Addition and Subtraction:-** difference between unsigned data and signed data are;

-in unsigned data we didn't considered the sign (+ or -)

-In signed data we considered the sign .and +ve represented as 0 and -ve represented as 1

-Eg: +3 = 0 11

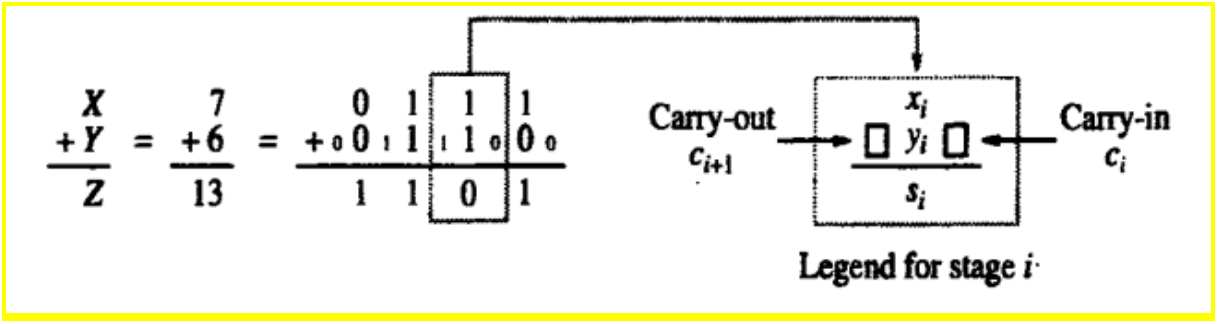
: -3 = 1 11

- Figure shows the logic truth table for the sum and carry-out functions for adding bits xi and yi in two numbers X and Y.

x_i	y_i	Carry-in c_i	Sum s_i	Carry-out c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$s_i = \bar{x}_i\bar{y}_i c_i + \bar{x}_i y_i \bar{c}_i + x_i\bar{y}_i\bar{c}_i + x_i y_i c_i = x_i \oplus y_i \oplus c_i$$
$$c_{i+1} = y_i c_i + x_i c_i + x_i y_i$$

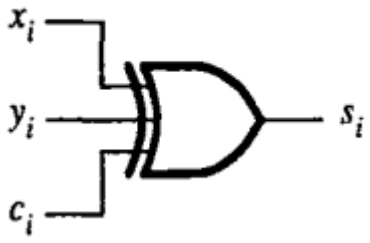
-The below figure also shows logic expressions for these functions, along with an example of addition of the 4-bit unsigned numbers 7 and 6.



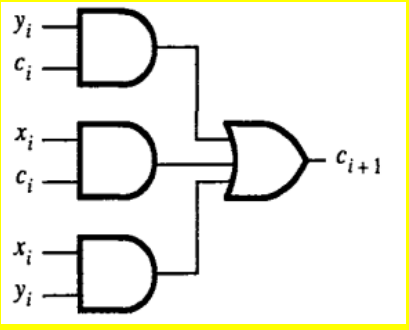
-Note that each stage of the addition process hold a carry-in bit.

-We use ci to represent the carry-in to the ith stage, which is the same as the carry-out from the (i-1)st stage.

-The logic expression for si can be implemented with a 3-input gives to a XOR gate,

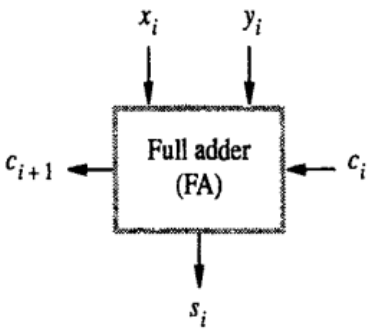


- The carry-out function, C_1+1, is implemented with a two-level AND-OR logic circuit.

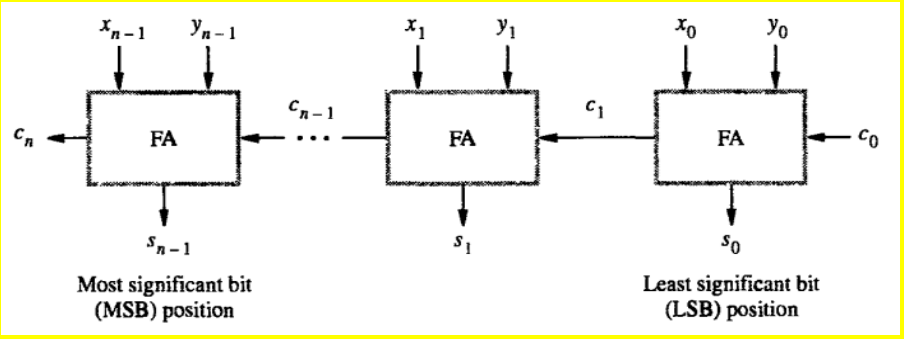


-The hardware circuit which executes this addition is called Adder.

-A convenient symbol for the complete circuit for a single stage of addition, called a full adder (FA), is also shown in the figure.



-A cascaded connection of n full adder blocks, can be used to add two n-bit numbers. And it is called an n-bit ripple-carry adder.



-The n-bit adder can be used to add 2's-complement numbers X and Y, where the xn-1 and yn-1 bits are the sign bits.

- In this case, the carry-out bit, (Cn) is not part of the answer.

- Overflow can only occur when the signs of the two operands are the same.

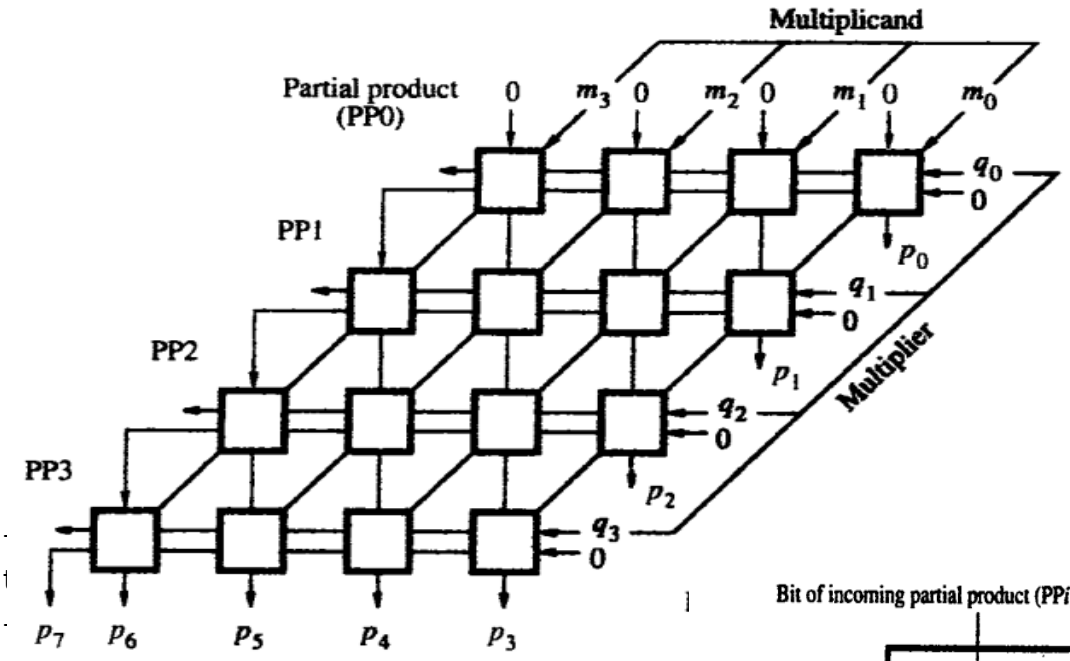
-In this case, overflow obviously occurs if the sign of the result is different.

-Therefore, a circuit to detect overflow can be added to the n-bit adder .

***Multiplication of positive numbers:**-The usual algorithm for multiplying integers by hand is

```
          1 1 0 1      (13) Multiplicand M
        × 1 0 1 1      (11) Multiplier Q
        ---------
          1 1 0 1
        1 1 0 1
      0 0 0 0
    1 1 0 1
    ---------------
  1 0 0 0 1 1 1 1      (143) Product P
```

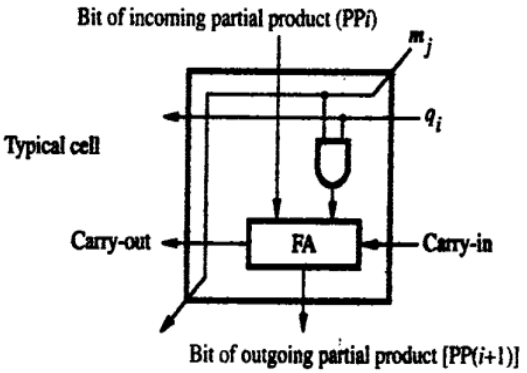
-This algorithm applies to unsigned numbers and to positive signed numbers.
-The product of two n-digit numbers can be accommodated in 2n digits,
-so the product of the two 4-bit numbers in this example fits into 8 bits.



ional,

multiplicand bit (mj) is added to the incoming partial-product bit based on the value of the multiplier bit (qi).

- Google ill nokkikkonam



***signed operand multiplication:-** considering 2's complement signed operand to multiple two signed operand.
-if the operand have n-bit then the product that generated is double the length is 2n -bit.
-if we need to multiple two numbers 45 and 13.
-here 45 is multiplicand and 13 is multiplier.
-in multiplication the multiplicand (45) is added based on the multiplier bit (13).
->Example suppose we have a -ve multiplicand (-13) and a +ve multiplier (+11).
-here based on multiplier (+11) the multiplicand (-13) put in as a partial product.

```
  10011      (-13)
x01011       (+11)
11111100111← partial product
11111001 1
00000000
1110011
000000
1101110001   (-143)
```

-it is a 5 bit number have generated of 10 bit number

-for the remaining bits we do a signed extension.

->steps

-The multiplicand is 10011 and the 1st bit needs to multiply with is the LSB of the multiplier is 1.----------> 10011 (-13) x01011 (+11) 10011 LSB

-and we multiply with 1 and we know that if we multiple with any number with 1 we get the same number(10011)---------->

-And we extended the sign bit (here 1 is the sign bit) with the Remaining 5 bit —----------> 10011 sign bit x01011 11111 10011

-next we take the next bit of multiplier (1).

-we shifted one place to the left and then place the product (10011)

-and for the remaining bits we add the sign bit.

- —----------> extended bit signed bit 10011 x01011 1111110011 11111 10011 shifted bit

-next multiplier is 0 ,so there is no addition of the Multiplicand in this case.---👇

10011 x01011 1111110011 11111 10011 00000000

-when ever tl ill add a shifted version of the multiplicand.

-and since the multiplicand was -ve ,we put the extension of the sign bit in the remaining bits.

-once we have the partial product we can add them.(1101110001)

10011 (-13) Signed bit x01011 (+11) 1111110011 11111 10011 partial product 0000000000 Shifted bit 11 10011 0000000000 1101110001 (-143)

-What if we have the multiplier is -ve.

-suppose 13 x -11 we cannot perform normal multiplication but can multiply -13 x 11 Both are same .

-(-13 x 11) it makes the multiplier positive.

-and we can proceed it in the normal multiplication method.

-we use 2's complement at both multiplicand side and multiplier side that way we can change the sign 13 into -13 and -11 into 11.

-Know we have a algorithm called booth's algorithm which work for both +ve and -ve multiplier.

→**Booth's algorithm:**-It is used when multiplier have +ve or -ve value.

-The booth algorithm says that the multiplier can be re-coded and it can be represented as a differences of two number

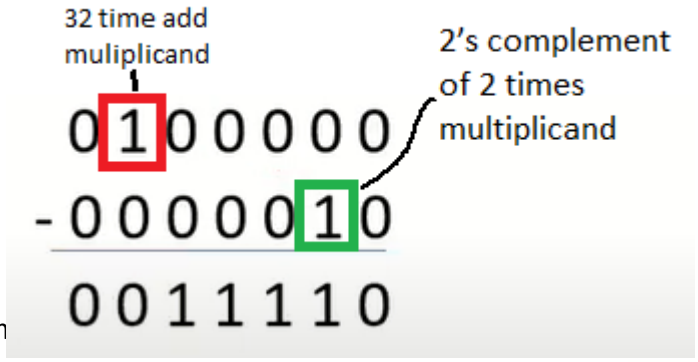
-suppose the multiplier is 30 and we can represent 30 as 32 - 2 (32-2=30).

-if we recode 30 as 0+1000-10 where +1 is representing 32 and -1 is representing 2.

-so we can re-coded or represented 30 as 0+1000-10

-it means that it add 2^5 (32) time multiplicand to 2's complement of 2^1 times multiplicand.

```
  0 1 0 0 0 0 0   (32)
- 0 0 0 0 0 1 0   (2)
  0 0 1 1 1 1 0   (30)
```



-Exam

-we have multiplier (0011110) ,it have 4 number of 1's (1,1,1,1).

-it means that if we have 4 1'st at the multiplier and We have 4 time shifted bit is added.

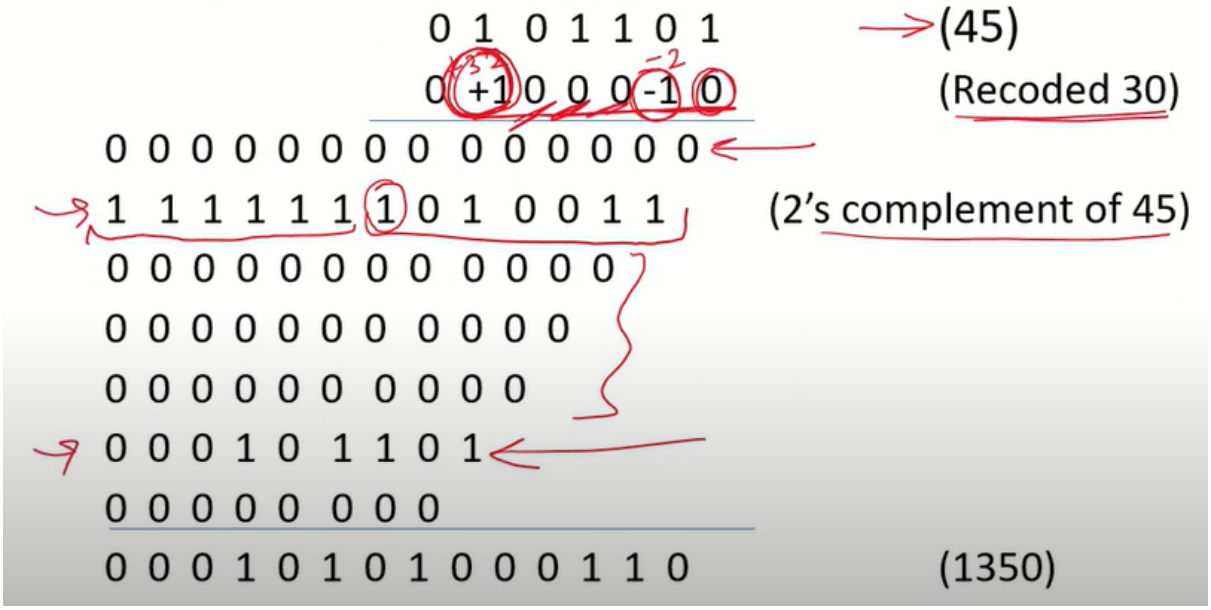
45*30: Normal multiplication

```
                0 1 0 1 1 0 1
                0 0 +1+1+1+1 0
                0 0 0 0 0 0 0
              0 1 0 1 1 0 1   ----(1)
            0 1 0 1 1 0 1     ----(2)
          0 1 0 1 1 0 1       ----(3)
        0 1 0 1 1 0 1         ----(4)
      0 0 0 0 0 0 0
    0 0 0 0 0 0 0
0 0 0 1 0 1 0 1 0 0 0 1 1 0
```

->Example of booth algorithm

-here 43 x 30 using recoded multiplier

-here we write 30 as recoded 30 ,means (32-2) form



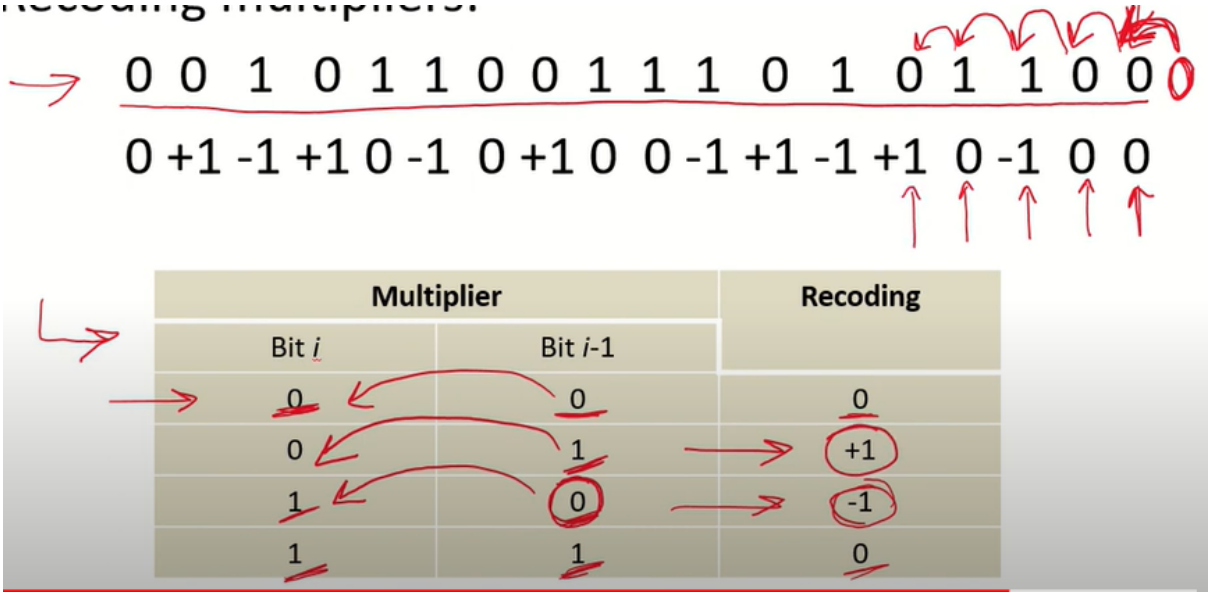
-In normal method we added shifted bit 4 time but in booth method we only add 2 items.

- re-code done by following method

Multiplier		Recoding
Bit *i*	Bit *i*-1	
0	0	0
0	1	+1
1	0	-1
1	1	0

-Example we have 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 1 0 0

-1st a imaginary 0 to the right of the actual multiplier 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 1 0 0 0.



-the recode multiplier is 0 +1 -1 +1 0 -1 0 +1 0 0 -1 +1 -1 +1 0- 1 0 0

-There are 3 type of multiplier cases

- Worst -case multiplier:-constantly changing 1 and 0
  -eg;101010101

-its recode is -1 +1 -1 +1 -1 +1..

- Ordinary multiplier:-a fair mix or 1's and the a fair mix of 0's
  Eg;11000111
  -its recode is 0 -1 0 0 +1 0 0 -1..
- Good multiplier:-it have a long run of 1's and 0's
  Eg:1111100000
  -its recode is 0 0 0 0 -1 0 0 0 0..

Worst-case multiplier:

0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
+1 -1 +1 -1 +1 -1 +1 -1 +1 -1 +1 -1 +1 -1 +1 -1

Ordinary multiplier:

1 1 0 0 0 1 1 1 0 1 1 1 0 1 1 1
0 -1 0 0 +1 0 0 -1 +1 0 0 -1 +1 0 0 -1

Good multiplier:

1 1 1 1 1 0 0 0 0 0 1 1 1 1 0 0
0 0 0 0 -1 0 0 0 0 +1 0 0 0 -1 0 0

***Fast Multiplication:**-There are two techniques for speeding the Multiplication Operation they are;

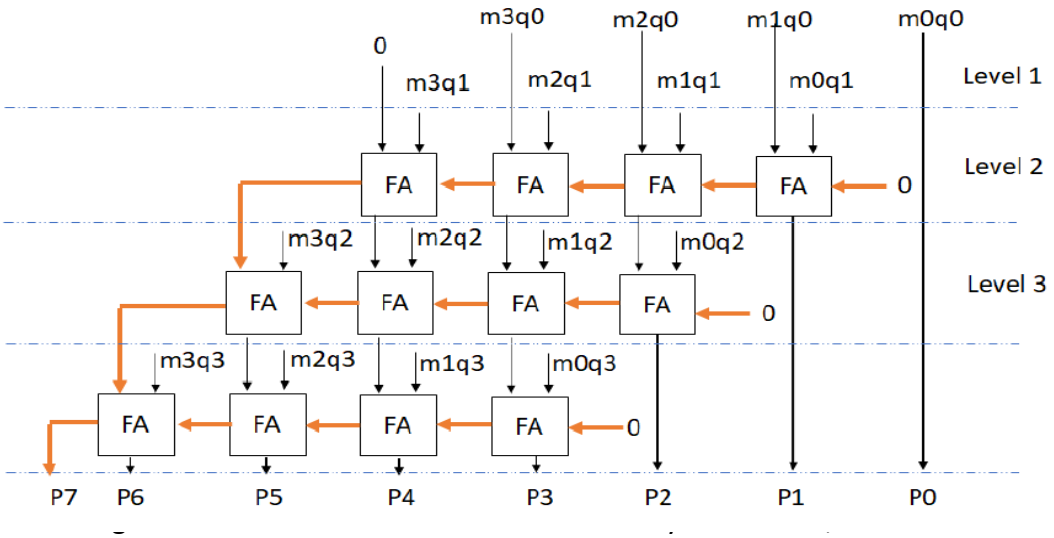
1. **Reducing Maximum number of Summands using Bit Pair Recoding of Multipliers/Bit Pair Recoding of Multipliers**:-It is a modified Booth Algorithm,
   -In this it uses one summand for each pair of booth re-coded bits of the multiplier.
   Step 1: Convert the given Multiplier into a Booth Recode the Multiplier.
   Step 2: Group the recoded Multiplier bits in pairs and observe the following
   ->For example - If the pair is (+1 -1) - It is equivalent to the pair (0 +1).
   -Reason: A pair (+1 -1) means adding (-1) time the Multiplicand M at shifted position i with (+1) times Multiplicand at shifted position (i+1) is equivalent to a pair (0 +1) which adds +1 times Multiplicand at position i.
   -A pair (+1 -1) = (21 x M - 20 x M) = (2M – 1M) = +1M Let us say Multiplicand = (1 1 0 1) and Bit pair (+1 -1)

   ->In this example the given Multiplier is (0 1 0 1 0 1) and Multiplicand is (0 0 1 1 1 0).
   -The recoding of Normal Multiplier using Booth Recoding and Bit pair is as shown below

   [0 1 0 1 0 1]--->[+1 -1 +1 -1 +1 -1 +1]--- >[(+1,-1) (+1,-1) (+1,-1)]-->[+1 +1 +1]

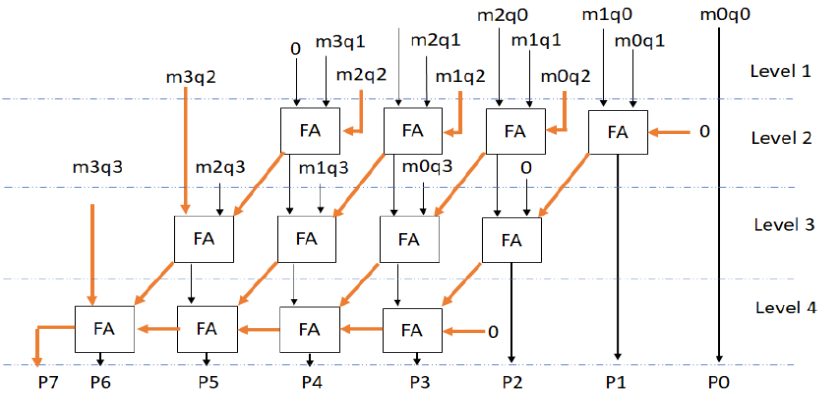
   -The worst case of booth recoded Multiplier is also can reduced to n/2 summands in Bit pair Booth algorithm .
   -Advantages: This reduces the maximum number of summands (versions of multiplicand) that must be added to n/2 for n bit operands

2. **Carry Save Addition of Summands:**-As we know Multiplication involves the addition of several summands.
   -A technique called carry-save addition (CSA) can be used to speed up the process.
   -Let us consider the 4 × 4 multiplication array (m3 m2 m1 m0 multiplied with q3 q2 q1 q0) Use of Full Adders in first row is not needed as there is No addition of summands involved, therefore in first row (Level1) is made to consists of just the AND gates that produce the partial products (m3q0, m2q0, m1q0 and m0q0).
   -From second row onwards n-bit full adders are use and it can be seen that in Second row each full adder takes summands of first row as one input and summands of second row as second input with carry rippling from in row.
   -This basically reduces the number of Full Adder by n.



-In this, I n that the Carry is getting rippled from an adder to the other adder in each row (of same level) which basically delays the summands addition.

-Instead of letting the carries ripple along the rows (ith row), they can be "saved" and introduced into the next row i.e., (i+1)th row, at the correct weighted positions.
-As carry in ith row propagates to (i+1)th row, it frees up Cin input of three full adders in the Second row (Full adder at LSB takes carry input as Zero).
-Now these inputs are now used to introduce the third summand bits m2q2, m1q2, and m0q2.
-The summand m3q2 goes as input to the FA at the left end of next successive row [1].
-The Carry Save Addition of summands for n=m=4 is as shown below.

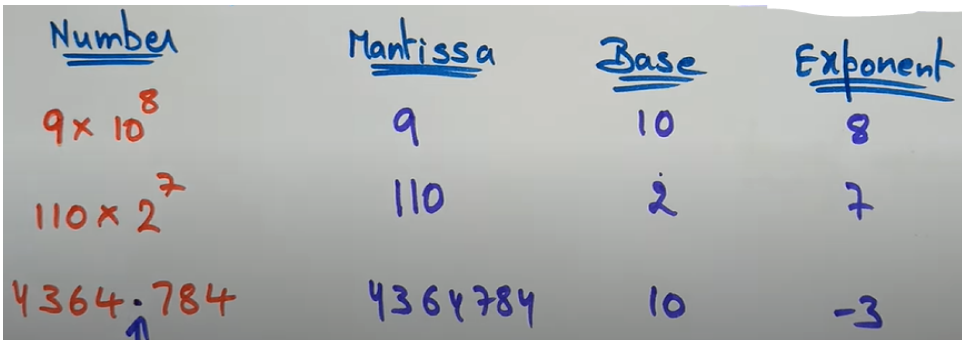


-Now, two inputs of each of three full adders in the third row (level 3) are fed by the sum and carry outputs from the Second row (Level 2).
-The third input is used to introduce the bits m2q3, m1q3, and m0q3 of the fourth summand.
-The high-order bit m3q3 goes input to the Full Adder at the left end of next successive row.
-The saved carry bits and the sum bits from the third row are now added in the fourth row (Level 4), which is a ripple-carry adder, to produce the final product bits [1].
-The delay through the carry-save array is somewhat less than the delay through the ripple-carry array.
-This is because the S and C vector outputs from each row are produced in parallel in one full-adder delay [1] .(manasiyali illangill youtube nokkikonam)

***Floating point Numbers:-**In floating point representation where the binary point floats to the right of most significant bit (MSB).
-And a exponent is used
-nowaday whatever the technology that we are using in the technology the number system are following that number system is represented in the floating point.
-to observe the speed of light ,charge of electrons etc are used.
-The floating point representation can be used for very large numbers as well as very small numbers
-It has 3 parts

- **Mantissa**
- **Base**
- **Exponen**t

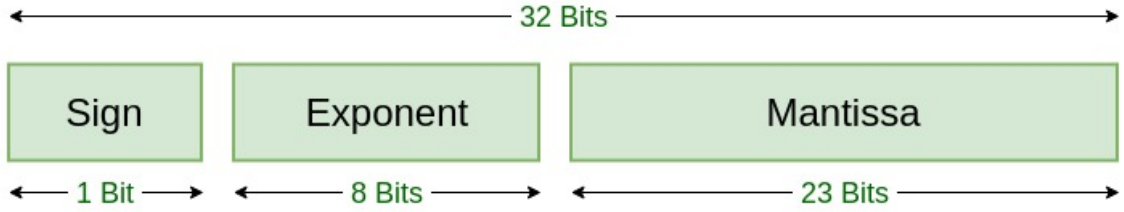
-so whatever binary number we take we represent the number in this format (Mantissa , Base ,Exponent).
-The scientific notation of floating point is **+- M x B^E**. [M-Mantissa ,B-Base ,E-Exponent]
->Example



-4364.784, its Mantissa is the whole number without point is 4364784
and its Base is 10 (.784 is converted into 10),
And Exponent is -3 (because 4364784 can be represented as 10 power -3)

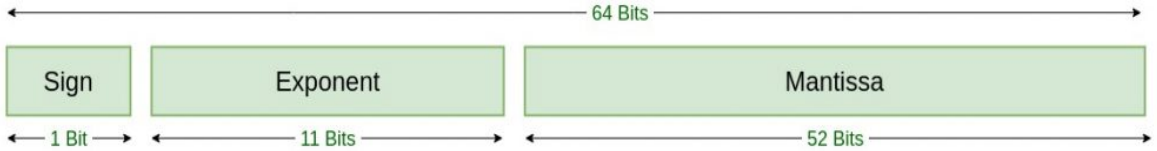
→**IEEE Representation of Floating point Numbers:-** it can be represented in two form they are ;

1. **single precision**:- here we taking the bit size as 32 bit .
   -The size of single precision (32 bit) is divided into the complete number is represented in the the form of sign ,exponent,mantissa.



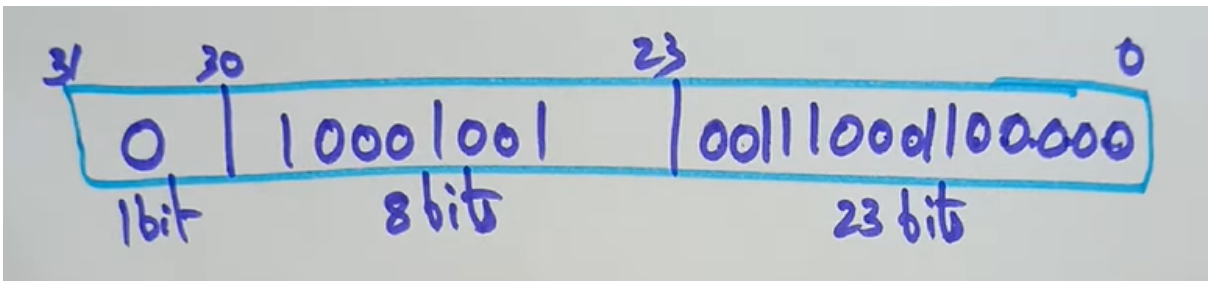
   -in sign bit 0 means it is a +ve number and 1 means it is a -ve number.

2. **double precision**:- here we taking the bit size as 64 bit



->Example represent $(1259.125)_{10}$ in single and double precision format.

- Step 1: we have to convert the decimal to binary.
  $(1259.125)_{10} = (10011100011.001)_2$
- Step 2: normalize the number
  -formula for single precision is $(1 \text{ X } N)\ 2^{E-127}$.
  -formula for single precision is $(1 \text{ X } N)\ 2^{E-1023}$.
  $(10011100011.001)_2 = 1.0011100011001 \text{ x } 2^{10}$.
- Step 3:single precision formula
  $(1 \text{ X } N)\ 2^{E-127} = 1.0011100011001 \text{ x } 2^{10}$. [0011100011001 is N] ,
  E-127 =10
  therefore E=137
  -E=137 it is a decimal number so we need to convert it into binary number.
  $E=(10001001)_2$.
  -Here N have the sign bit and mantissa (0011100011001).



  -empty set in mantissa must be filled , so add 0's to complete the 23 bits .

- Step 4:double precision format

  (1 X N) 2^{E-1023}= 1.0011100011001 x 2^{10}. [0011100011001 is N] ,

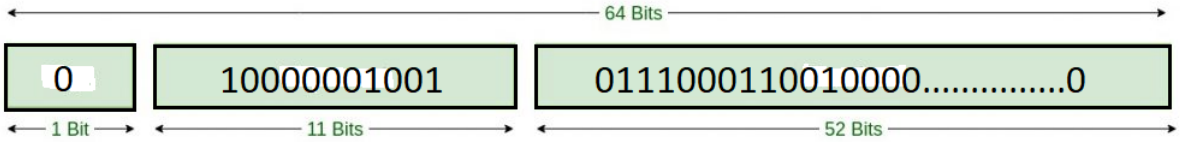
  E-1023 =10

  therefore E=1033

  -E=1033 it is a decimal number so we need to convert it into binary number.

  $E=(10000001001)_2$.

  -Here N have the sign bit and mantissa (0011100011001).



-empty set in mantissa must be filled , so add 0's to complete the 52 bits .

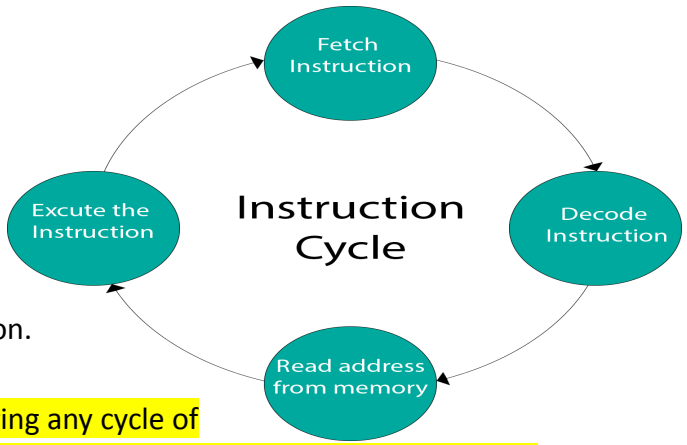
Processing Unit

***Instruction execution cycle:**-A program consisting of the memory unit of the computer includes a series of instructions.

-These instructions are executed by the processor by going through a cycle for each instruction.

-In a basic computer, each instruction cycle consists of the following phases:

- Fetch instruction from memory.
- Decode the instruction.
- Read the effective address from memory.
- Execute the instruction.



-After the following four procedures are done, the control switches back to the first step and repeats the similar process for the next instruction.

-until an interrupt occur .

-In some CPUs, interrupt handling may occur during any cycle of the instruction cycle. An interrupt is a signal that the CPU receives from an external device or software that requires immediate attention. When an interrupt occurs, the CPU suspends the current instruction and executes an interrupt handler to service the interrupt.

***Sequencing of control signals:**-the control unit is responsible for directing the flow of data and instructions within the CPU.

-There are two main approaches to implementing a control unit:

- hardwired:-A hardwired control unit is a control unit that uses a fixed set of logic gates and circuits to execute instructions.

-The control signals for each instruction are hardwired into the control unit, so the control unit has a dedicated circuit for each possible instruction.
-Hardwired control units are simple and fast, but they can be inflexible and difficult to modify.

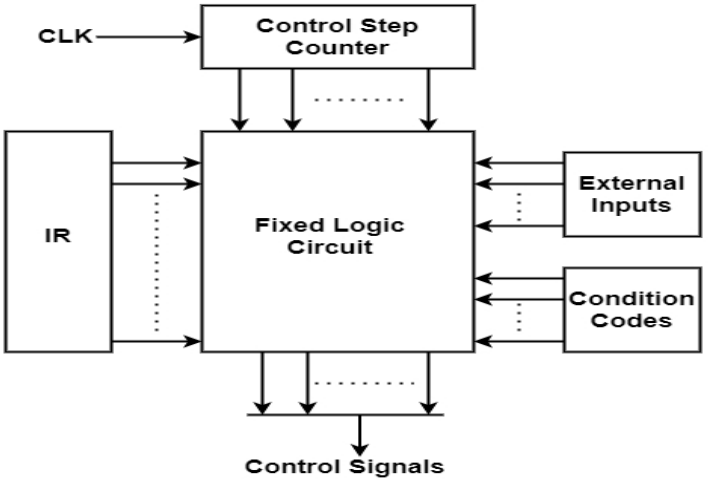
- micro-programmed:-a micro-programmed control unit is a control unit that uses a microcode to execute instructions.
-The microcode is a set of instructions that can be modified or updated, allowing for greater flexibility and ease of modification.
-The control signals for each instruction are generated by a microprogram that is stored in memory, rather than being hardwired into the control unit.

-Micro-programmed control units are slower than hardwired control units because they require an extra step of decoding the microcode to generate control signals, but they are more flexible and easier to modify.
-They are commonly used in modern CPUs because they allow for easier implementation of complex instruction sets and better support for instruction set extensions.
-To execute an instruction, the control unit of the CPU must generate the required control signal in the proper sequence.

-There are two approaches used for generating the control signals in proper sequence as

1. Hardwired Control unit
2. Micro-programmed control unit

***Hardwired Control**:-it is implemented as a sequential logic circuit that generates a specific sequence of control signals.
-The final circuit is constructed by physically connecting the components such as gates, flip flops, and drums.
-The figure shows a 2-bit sequence counter, which is used to develop control signals.
-The output obtained from these signals is decoded to generate the required signals in sequential order.



-The instruction that is loaded in the IR is decoded by the instruction decoder.
-If the IR is an 8-bit register, then the instruction decoder generates 28 (256) lines.
-Inputs to the encoder are given from the instruction step decoder, external inputs, and condition codes.
-All these inputs are used and individual control signals are generated.
-The end signal is generated after all the instructions get executed.
-The major goal of implementing the hardwired control is to minimize the cost of the circuit and to achieve greater efficiency in the operation speed.
-Techniques for design of hardwired control unit are;

- **State Table Method**:-classical method of sequential design.
  -it consist of minimize of hardware.
  -it contruts a state transition table.
  -every generation of states has a set of control signals.
  Or
  -This method involves the traditional algorithmic approach to design the Notes controller using the classical state table method.
- **Delay Element Method**: − This method is dependent on the use of clocked delay elements for generating the sequence of control signals.
  -A specific time delay between two groups of control signals.
- **Sequence Counter Method** − This is the most convenient method employed to design the controller of moderate complexity.
  -and it uses counter for timing purpose.

***Microprogrammed Control**:-Micro-instructions are stored in a special memory called control memory.
-it is implemented using a programming approach.
-In Microprogrammed Control, the micro-operations are performed by executing a program consisting of micro-instructions.
→Advantages of Microprogrammed Control are
-It is simpler to debug and change.
-It can more systematic design of the control unit.
-It can make the design of the control unit much simpler. Hence, it is inexpensive and less error-prone.
-It is more flexible.
-It is used to complex function is carried out easily.
-Once the hardware configuration is established, there is no need for further hardware or wiring changes.
→Disadvantages of Microprogrammed Control.
-It is slower than a hardwired control unit.

->difference

Hardwired Control Unit	Micro-programmed Control Unit
Fixed set of logic gates and circuits	Microcode stored in memory
Less flexible, difficult to modify	More flexible, easier to modify
Supports limited instruction sets	Supports complex instruction sets
Simple design, easy to implement	Complex design, more difficult to implement
Speed Fast operation	Slower operation due to microcode decoding
Difficult to debug and test	Easier to debug and test
Smaller size, lower cost	Larger size, higher cost
Difficult to upgrade and maintain	Easier to upgrade and maintain

Or

- The control signals needed by the CPU are generated by the hardwired control unit, whereas the microprogrammed control unit generates the signals through microinstructions.
- A microprogrammed control unit is slower than a hardwired control unit.
- Microprogrammed control units are easier to alter than hardwired control units.
- The cost of a hardwired control unit is higher than that of a microprogrammed control unit.
- Due to the complexity of the circuit design, the hardwired control unit has trouble handling complicated instructions.
- Limited instructions can be used by a hardwired control device.

*** Micro-instructions:**-A microinstruction format includes 20 bits in total.

-They are divided into four elements as displayed in the figure.

Microinstruction Code Format

3	3	3	2	2	7
F1	F2	F3	CD	BR	AD

-Each step in a sequence of steps in executing a certain machine instruction is considered a microinstruction, and a control word represents it.
-F1, F2, F3 are the micro-operation fields. They determine micro-operations for the computer.
-CD is the condition for branching. They choose the status bit conditions.
-BR is the branch field. It determines the type of branch.
-AD is the address field. It includes the address field whose length is 7 bits.
-The micro-operations are divided into three fields of three bits each (F1,F2,F3). These three bits can define seven different micro-operations.
-In total there are 21 operations.
→Symbols with their Binary Code for Microinstruction Fields

	Name:	Code	Symbol
F1	000	None	NOP
	001	AC ← AC + DR	ADD
	010	AC ← 0	CLRAC
	011	AC ← AC + 1	INCAC
	100	AC ← DR	DRTAC
	101	AR ← DR(0 – 10)	DRTAR
	110	AR ← PC	PCTAR
	111	AC ← AC + DR	WRITE
F2	000	None	NOP
	001	AC ← AC + DR	SUB
	010	AC ← AC v DR	OR
	011	AC ← AC ∧ DR	AND
	100	DR ← M[AR]	READ
	101	DR ← AC	ACTDR
	110	DR ← DR + 1	INCDR
	111	DR(0 – 10) ← PC	PCTDR
F3	000	None	NOP
	001	AC ← AC ⊕ DR	XOR
	010	AC ← AC'	COM
	011	AC ← shl AC	SHL
	100	AC ← shr AC	SHR
	101	PC ← PC + 1	INCPC
	110	PC ← AR	ARTPC
	111	DR(0 – 10) ← PC	Reserved

-As shown in the table, each microinstruction can have only three micro-operations, one from each field. If it uses less than three, it will result in more than one operation using the no operation binary code.
OR
If the microinstruction needs microoperation less than three,one or more of the microoperation fields will be filled by a binary code 000 for no operations.

***Microprogram Sequencing:**-A microprogram sequencer uses its address to determine the next microinstruction that needs to be executed.
- This overall process is known as microprogram sequencing.
-The next-address is loaded into the control address register(CAR).
-Two important factor that must be considered while designing the micro-instruction sequential are;

1. Size of the microinstruction
2. Time of address generation

-The main purpose of microprogram sequencer is to provide address to the control memory.
-The microinstruction's size should be in the least, therefore that the control memory necessary is less and the cost is decreased.
-Microinstructions can be implemented at a quicker rate if the time to create an address is less.

→The advantages of microprogram sequencing are the following:

- The execution instruction can be easily controlled by microprogram sequencing.
- It is easily modifiable due to how simple it is to update the code.
- It can also handle complicated instructions with ease.
- A microprogram sequencing implementation is less expensive.

→The disadvantages of microprogram sequencing are the following:

- It is a bit slower.
- Larger storage spaces are made possible with the aid of distinct micro routines for each device command.
- More time was required for the branching's implementation.

***RISC characteristics and CISC characteristics:**-RISC is Reduced Instruction Set Computer and CISC is Complex Instruction Set Computer.
-the RISC processors have a comparatively smaller set of instructions along with few addressing nodes.
-On the other hand, the CISC processors consist of a larger set of instructions along with multiple addressing nodes.

→Reduced Instruction Set Computer or RISC
-The fundamental goal of RISC is to make hardware simpler by employing an instruction set that consists of only a few basic steps.
-The main idea behind this is to make hardware simpler by using an instruction set composed of a few basic steps for loading, evaluating, and storing operations just like a load command will load data, a store command will store the data.
-Characteristics of RISC are

- It has simpler instructions and simple instruction decoding.
- More general-purpose registers.
- The instruction takes one clock cycle in order to get executed.
- The instruction comes under the size of a single word.
- Pipeline can be easily achieved.
- Few data types.
- Simpler addressing modes.

→Complex Instruction Set Computer or CISC

-The main idea is that a single instruction will do all loading, evaluating, and storing operations just like a multiplication command will do stuff like loading data, evaluating, and storing it, hence it's complex.

-Characteristics of CISC are;

- Instructions are complex, and thus it has complex instruction decoding.
- The instructions may take more than one clock cycle in order to get executed.
- The instruction is larger than one-word size.
- Lesser general-purpose registers since the operations get performed only in the memory.
- More data types.
- Complex addressing modes.

-CISC and RISC approaches primarily try to increase the performance of a CPU.

- RISC: Reduce the cycles per instruction at the cost of the number of instructions per program.
- CISC: The CISC approach attempts to minimize the number of instructions per program but at the cost of an increase in the number of cycles per instruction.

$$CPU\ Time = \frac{Seconds}{Program} = \frac{Instructions}{Program} \times \frac{Cycles}{Instructions} \times \frac{Seconds}{Cycle}$$

->Example: Suppose we have to add two 8-bit numbers:

-CISC approach: There will be a single command or instruction for this like ADD which will perform the task.

-RISC approach: Here programmer will write the first load command to load data in registers then it will use a suitable operator and then it will store the result in the desired location.

->Advantages of RISC:

- Simpler instructions
- Faster execution
- Lower power consumption:

->Disadvantages of RISC:

- More instructions required
- Increased memory usage
- Higher cost

->Advantages of CISC:

- Reduced code size
- More memory efficient
- Widely used

->Disadvantages of CISC:

- Slower execution
- More complex design
- Higher power consumption

->Difference

RISC	CISC
Focus on software	Focus on hardware
Uses only Hardwired control unit	Uses both hardwired and microprogrammed control unit
Transistors are used for more registers	Transistors are used for storing complex Instructions
Fixed sized instructions	Variable sized instructions
Requires more number of registers	Requires less number of registers
Code size is large	Code size is small
An instruction executed in a single clock cycle	Instruction takes more than one clock cycle
An instruction fit in one word.	Instructions are larger than the size of one word
Simple and limited addressing modes.	Complex and more addressing modes.
RISC is Reduced Instruction Cycle.	CISC is Complex Instruction Cycle.
The number of instructions are less as compared to CISC.	The number of instructions are more as compared to RISC.
It consumes the low power.	It consumes more/high power.
RISC is highly pipelined.	CISC is less pipelined.
RISC required more RAM.	CISC required less RAM.
Here, Addressing modes are less.	Here, Addressing modes are more.

Module -4

#Main Memory And I/O Organization

*** Memory Hierarchy:-**A memory unit is an essential component in any digital computer since it is needed for storing programs and data.

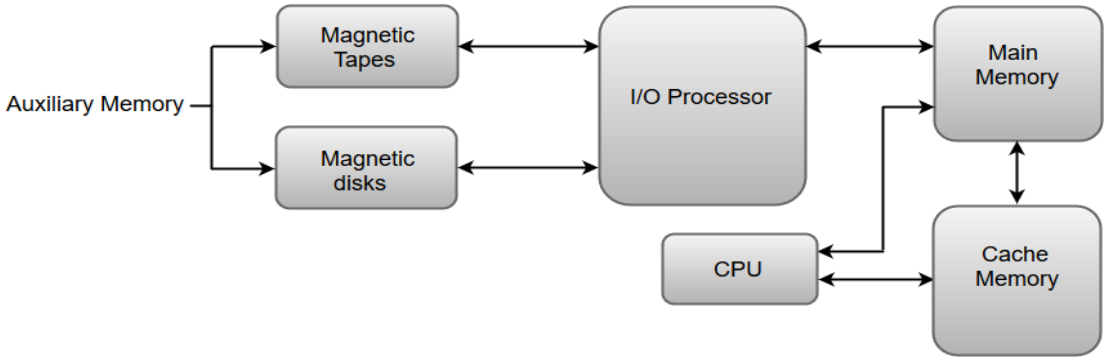
-Typically, a memory unit can be classified into two categories:

- The memory unit that establishes direct communication with the CPU is called **Main Memory**. The main memory is often referred to as RAM (Random Access Memory).
- The memory units that provide backup storage are called **Auxiliary Memory.** For instance, magnetic disks and magnetic tapes are the most commonly used auxiliary memories.

-Apart from the basic classifications of a memory unit, the memory hierarchy consists all of the storage devices available in a computer system from the slow but high-capacity auxiliary memory to relatively faster main memory.

-The following image illustrates the components in a typical memory hierarchy.

Memory Hierarchy in a Computer System:



→**Auxiliary memory:-**Auxiliary memory is known as the lowest-cost, highest-capacity and slowest-access storage in a computer system.

-Auxiliary memory provides storage for programs and data that are kept for long-term storage or when not in immediate use.

-The most common examples of auxiliary memories are magnetic tapes and magnetic disks.

- A magnetic disk is a digital computer memory that uses a magnetization process to write, rewrite and access data. For example, hard drives, zip disks, and floppy disks.
- Magnetic tape is a storage medium that allows for data archiving, collection, and backup for different kinds of data.

→**Main memory:-**The main memory in a computer system is often referred to as Random Access Memory (RAM).

-This memory unit communicates directly with the CPU and with auxiliary memory devices through an I/O processor.

-The programs that are not currently required in the main memory are transferred into auxiliary memory to provide space for currently used programs and data.

→**I/O processor:-**The primary function of an I/O Processor is to manage the data transfers between auxiliary memories and the main memory.

→**Cache memory:-**The data or contents of the main memory that are used frequently by CPU are stored in the cache memory so that the processor can easily access that data in a shorter time.

-Whenever the CPU requires data for memory, it first checks the required data into the cache memory.

-If the data is found in the cache memory, it is read from the fast memory. -Otherwise, the CPU moves onto the main memory for the required data.

***Main Memory:-**The main memory acts as the central storage unit in a computer system.

-It is a relatively large and fast memory which is used to store programs and data during the run time operations.

-The primary technology used for the main memory is based on semiconductor integrated circuits.

-The integrated circuits for the main memory are classified into two major units.

1. **RAM (Random Access Memory) integrated circuit chips:-**The RAM integrated circuit chips are further classified into two possible operating modes, static and dynamic.
    - **static RAM:-** are made of flipflops.

      -The nature of the stored information is volatile, i.e. it remains valid as long as power is applied to the system.

      -static RAM is easy to use and takes less time performing read and write operations as compared to dynamic RAM. or SRAM has lower access time , so it is faster compared to DRAM.

      -it required constant power supply, which means this type of memory consumes more power.

      -it has low storage capacity.

      -its structure is complex than DRAM.

      -it is faster than DRAM.

      -The static RAM is easier to use and has shorter read and write cycles.
    - **dynamic RAM:-**Data is stored in capacitors. Or made up of capacitors.

-Slow access speed and high power consumption.

-it needs periodic refreshment to maintain the charge in the capacitors for data.

-its structure is simplex than SRAM.

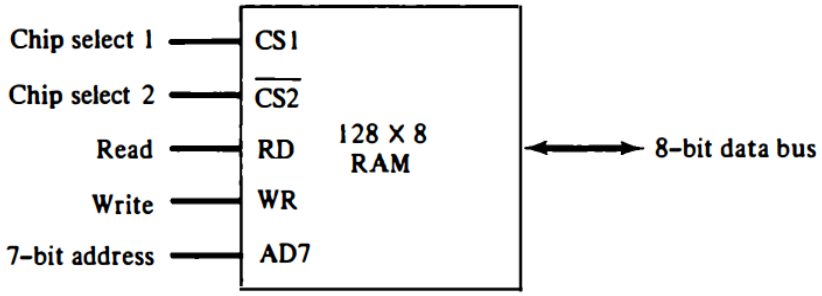
-it is less expensive as compare to SRAM.

-high storage capacity.

-consumes less power.

-data loses with time,so need refreshing circuit

Figure 12-2 Typical RAM chip.



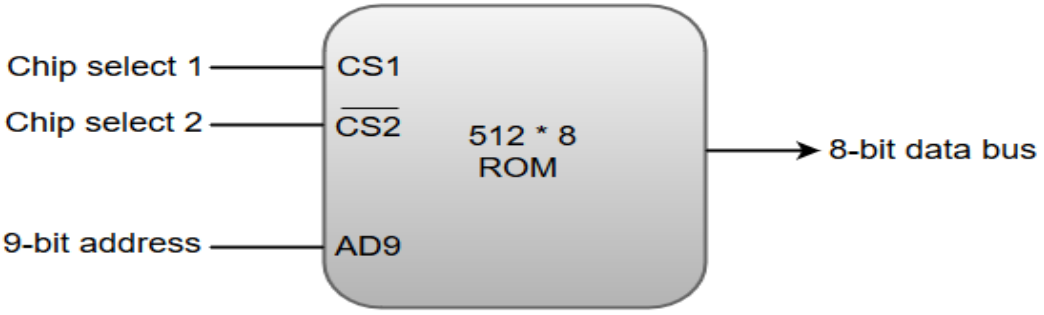
2. **ROM (Read Only Memory) integrated circuit chips**:-The primary component of the main memory is RAM integrated circuit chips, but a portion of memory may be constructed with ROM chips.

   -A ROM memory is used for keeping programs and data that are permanently in the computer.

   -Apart from the permanent storage of data, the ROM portion of main memory is needed for storing an initial program called a **bootstrap loader**. The primary function of the **bootstrap loader** program is to start the computer software operating when power is turned on.

   -ROM chips are also available in a variety of sizes and are also used as per the system requirement.

   Typical ROM chip:



***RAM:-**which stands for Random Access Memory.
-it is a hardware device generally located on the motherboard of a computer and acts as an internal memory of the CPU.
-It is a read/write memory which stores data until the machine is working. As soon as the machine is switched off, data is erased.
-RAM is volatile in nature, it means if the power goes off, the stored information is lost.
-Integrated RAM chips can be of two types:

1. Static RAM (SRAM):-mukalill unde
2. Dynamic RAM (DRAM):mukalill unde

-Both types of RAM are volatile, as both lose their content when the power is turned off.

***Semiconductor RAM memories:-**semiconductor memories are available in a wide range of speeds.
-their cycle times range from 100 ns(nano second to 10 ns.
-it was first introduced in the late 1960's
-they were much more expensive than magnetic-core memories they replaced.
-because of rapid advances in VLSI (very large scale integration) technology,the cost of semiconductor memories has dropped .
-As a result they were used more.

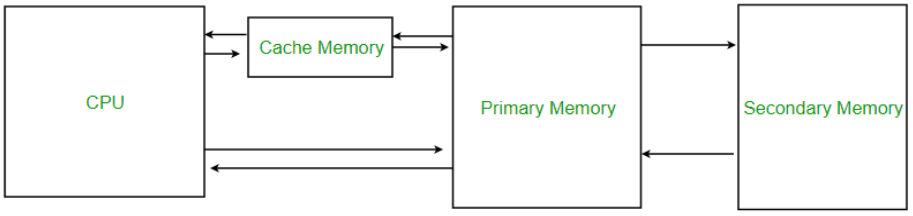
***ROM:-**ROM stands for Read Only Memory.
-it is a Non-volatile Memory.
-*Non-volatile memory* is a type of computer memory that is used to retain stored information during power is removed.
-t is less expensive than volatile memory.
-It has a large storage capacity.
-The information stored in the ROM in binary format.
-It is also known as permanent memory.
-Information and programs stored on it, we can only read.
-It is used in the start-up process of the computer.
→types of ROM

1. **MROM (Masked read-only memory):-**It is the oldest type of read only memory (ROM)
   -It has become old fashion so it is not used anywhere in today's world.
   -It is a hardware memory device in which programs and instructions are stored at the time of manufacturing by the manufacturer.
   - it is programmed during the manufacturing process and can't be modified, reprogrammed, or erased later.
   -The MROM chips are made of integrated circuits
2. **PROM (Programmable Read Only Memory):-**PROM is read-only memory that can be modified only once by a user.
   -The user buys a blank PROM and enters the desired contents.
   -To write data onto a PROM chip; a device called PROM programmer or PROM burner is used.
   - The process or programming a PROM is known as burning the PROM.

-Once it is programmed, the data cannot be modified later, so it is also called as one-time programmable device.
-It is used in cell phones, video game consoles, medical devices, RFID tags, and more.

3. **EPROM (Erasable and Programmable Read Only Memory):-**EPROM is a type of ROM that can be reprogramed and erased many times
   -EPROM can be erased by exposing it to ultra-violet light for a duration of up to 40 minutes.
   -it is a type of PROM but it can be reprogrammed.
   -we need a special device called a PROM programmer or PROM burner to reprogram the EPROM.
4. **EEPROM (Electrically Erasable and Programmable Read OnlyMemory):-**ROM is a type of read only memory that can be erased and reprogrammed repeatedly, up to 10000 times.
   -It is also known as Flash EEPROM as it is similar to flash memory.
   -It is erased and reprogrammed electrically without using ultraviolet light. -Access time is between 45 and 200 nanoseconds.
   -The data in this memory is written or erased one byte at a time; byte per byte.
5. **FLASH ROM/flash memory:-It is an advanced version of EEPROM.**
   -in flash memory data is written and erased in blocks. So, it is faster than EEPROM.
   -It can be reprogrammed without removing it from the computer.
   -it is cheaper.
   -its life cycle is more than EEPROM.

***Cache Memory**:-Cache Memory is a special very high-speed memory.
-Cache memory is costlier than main memory.
-It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.
-Cache memory is used to reduce the average time to access data from the Main memory.
-The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations.



- **Cache memory** – It is the fastest memory which has faster access time where data is temporarily stored for faster access.
- **Main Memory** – It is memory on which computer works currently. It is small in size and once power is off data no longer stays in this memory.
- **Secondary Memory** – It is external memory which is not as fast as main memory but data stays permanently in this memory.

→**Cache Mapping:-** There are three different types of mapping used for the purpose of cache memory which is as follows:

1. **Direct mapping:-**The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line.

-or In Direct mapping, assign each memory block to a specific line in the cache.
-If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed.
-An address space is split into two parts index field and a tag field.
-The cache is used to store the tag field whereas the rest is stored in the main memory.

2. **Associative mapping:-**In this type of mapping, the associative memory is used to store content and addresses of the memory word.
   -Any block can go into any line of the cache. This means that the word id bits are used to identify which word in the block is needed, but the tag becomes all of the remaining bits.
   -This enables the placement of any word at any place in the cache memory.
   -It is considered to be the fastest and the most flexible mapping form.
   -In associative mapping the index bits are zero.
3. **Set-Associative mapping:-**This form of mapping is an enhanced form of direct mapping where the drawbacks of direct mapping are removed.
   -Set associative addresses the problem of possible thrashing in the direct mapping method.
   -It does this by saying that instead of having exactly one line that a block can map to in the cache, we will group a few lines together creating a set.
   -Then a block in memory can map to any one of the lines of a specific set.
   -Set-associative mapping allows that each word that is present in the cache can have two or more words in the main memory for the same index address.
   -Set associative cache mapping combines the best of direct and associative cache mapping techniques.
   -In set associative mapping the index bits are given by the set offset bits.
   -In this case, the cache consists of a number of sets, each of which consists of a number of lines.
   -In set associative mapping a cache is divided into a set of blocks.
   -The number of blocks in a set is known as associativity or set size.
   -Each block in each set has a stored tag.

***Secondary memory:**-Memory are of two types

- primary memory:-Primary memory is made up of semiconductors,
  -It is also divided into two types, Read-Only Memory (ROM) and Random Access Memory (RAM).
- secondary memory:-Secondary memory is a physical device for the permanent storage of programs and data(Hard disk, Compact disc, Flash drive, etc.).

-Secondary memory is computer memory that is non-volatile and not immediately accessible by a computer or processor.
-Which data and programs are not lost when the computer is turned off.
-It allows users to store data and information that can be retrieved, transmitted and services quickly and easily.
-Secondary storage is another name for secondary memory.
-Secondary or external storage devices have a much larger storage capacity and the cost of secondary memory is less as compared to primary memory.

->Use of Secondary memory

- **Permanent storage:** As we know that primary memory stores data only when the power supply is on, it loses data when the power is off.
  -So we need a secondary memory to stores data permanently even if the power supply is off.
- **Large Storage:** Secondary memory provides large storage space so that we can store large data like videos, images, audios, files, etc permanently.
- **Portable:** Some secondary devices are removable.
  -So, we can easily store or transfer data from one computer or device to another.

->Types of Secondary memory

-Secondary memory is of two types:

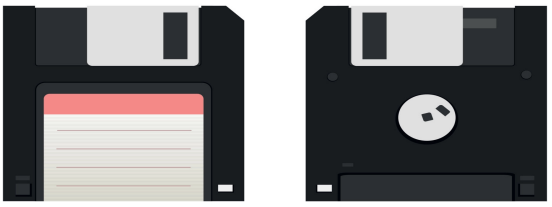
1. **Fixed storage:-**In secondary memory, a fixed storage is an internal media device that is used to store data in a computer system.
   -Fixed storage is generally known as fixed disk drives or hard drives.
   -Generally, the data of the computer system is stored in a built-in fixed storage device.
   -Fixed storage does not mean that you can not remove them from the computer system, you can remove the fixed storage device for repairing, for the upgrade, or for maintenance, etc. with the help of an expert or engineer.
   -Eg;
   - Internal flash memory (rare)
   - SSD (solid-state disk)
   - Hard disk drives (HDD)

2. **Removable storage:-**In secondary memory, removable storage is an external media device that is used to store data in a computer system.
   -Removable storage is generally known as disks drives or external drives.
   -It is a storage device that can be inserted or removed from the computer according to our requirements.
   -We can easily remove them from the computer system while the computer system is running.
   -Removable storage devices are portable so we can easily transfer data from one computer to another.
   - Also, removable storage devices provide the fast data transfer rates associated with storage area networks (SANs).
   -Eg;
   - Optical discs (like CDs, DVDs, Blu-ray discs, etc.)
   - Memory cards
   - Floppy disks
   - Magnetic tapes
   - Disk packs
   - Paper storage (like punched tapes, punched cards, etc.)

→Secondary memory devices

-Following are the commonly used secondary memory devices are:

1. **Floppy Disk:** A floppy disk consists of a magnetic disc in a square plastic case.
   -It is used to store data and to transfer data from one device to another device.
   -To use a floppy disk, our computer needs to have a floppy disk drive.
   -This storage device becomes obsolete now and has been replaced by CDs, DVDs, and flash drives.



2. **Compact Disc:** A Compact Disc (CD) is a commonly used secondary storage device.
   -It contains tracks and sectors on its surface.
   -Its shape is circular and is made up of polycarbonate plastic.
   -A CD may also be called a CD-ROM (Compact Disc Read-Only Memory), in this computers can read the data present in a CD-ROM, but cannot write new data onto it.
   -CD is of two types:

   - CD-R (compact disc recordable): Once the data has been written onto it cannot be erased, it can only be read.
   - CD-RW (compact disc rewritable): It is a special type of CD in which data can be erased and rewritten as many times as we want.
     -It is also called an erasable CD.

3. **Hard Disk**: A hard disk is a part of a unit called a hard disk drive.
   -It is used to storing a large amount of data.
   -Hard disks or hard disk drives come in different storage capacities.(like 256 GB, 500 GB, 1 TB, and 2 TB, etc.).
   -It is created using the collection of discs known as platters.
   - The platters are placed one below the other.
   -They are coated with magnetic material.
   -Each platter consists of a number of invisible circles and each circle having the same centre called tracks.
   - Hard disk is of two types (i) Internal hard disk (ii) External hard disk.

4. **Pen drive:**-Pen drive is a compact secondary storage device.
   -It is also known as a USB flash drive, thumb drive or a jump drive.
   -It connects to a computer via a USB port.
   - It is commonly used to store and transfer data between computers.
   - For example, you can write a report using a computer and then copy or transfer it in the pen drive.
   -Later, you can connect this pen drive to a computer to see or edit your report.
   -You can also store your important documents and pictures, music, videos in the pen drive and keep it at a safe place.
   -Pen drive does not have movable parts; it comprises an integrated circuit memory chip that stores the data.
   - This chip is housed inside a plastic or aluminium casing.

-The data storage capacity of the pen drive generally ranges from 2 GB to 128 GB.
-Furthermore, it is a plug and play device as you don't need additional drives, software, or hardware to use it.

5. **Sd card:-**It is known as a Secure Digital Card. It is generally used in portable devices like mobile phones, cameras, etc., to store data.
   -bakki thanne ezhuthhukaa.

***Performance Considerations**:- A key design objective of a computer system is to achieve the best possible performance at the lowest possible cost.
-Price/performance ratio is a common measure of success.
-Performance of a processor depends on:

- How fast machine instructions can be brought into the processor for execution.
- How fast the instructions can be executed.

-There are 3 parameters,based on those 3 parameters we are able to design a computer with high performance and with low price.

1. **Interleaving:**-suppose two person are talking through a walkie talkie .and if one person is communicating the another person should be in silent.
   -when the other person is communicating then this person should be in silence.
   -only one way communication is possible.
   -the problem of the method is that the speed is very very slow.
   -similarly we can use this concept to enhances the performance of the system.
   -we can enhances the system by making two registers
   - **DAR [data address register] /ABR [address buffer register]**
     -note text book ill oka ABR enna ullathu .
     -DAR inta place ill ABR ennu ezhuthiyal mathii
   - **DBR [data buffer register]**

   -This two are the temporary registers that can be used to store the address of the data which we want to access in DAR and the actual data can be stored in the DBR
   -By this two registers we can simultaneously we can access the data ,if multiple process can also access the data from the memory.
   -The big advantage of this is that the performance of the computer will increases.
   -It can simultaneously access or parallelly access resources from the memory.

2. **Hit rate and miss penalty:**-If our computer want to access the data .it will search in the cache memory first, if the data is not found in the cache memory then it will search in the main memory.
   -If data is not found in the main memory then last it will search in the secondary memory.
   -The meaning of **Hit** is that nothing but if our processor found the required data in the cache memory itself then it is called **Hit .**
   -The meaning of **Miss** is that if our processor did not able to found the required data in the cache memory then it is a **Miss**.
   -Hit rate means that how many times our processor can able to find the data in the cache memory itself.and it is called as **Hit rate.**
   -Eg;

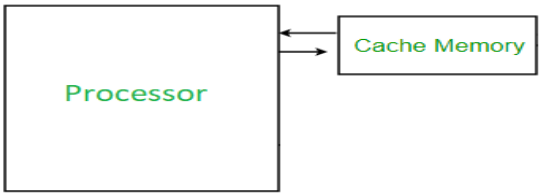
-10 times our processor has search the data in the cache memory and out of 10 time 9 times our processor has find successfully
-only one time it did not find the data in the cache memory.
-then we can say that our computer have a 90% Hit rate .

-Miss penalty means that if our processor not found the required data in the cache memory processor have to check either in main memory or not in the secondary memory is called **Miss penalty**

-if our computer have a good performance then we have a high rate of Hit rate compare to Miss penalty then we can say that our computer have a very good performance.

3. **Caches on the processor chip:**-In added to access the data from the cache memory processor should take the data from the cache memory .

-cache memory is located separately .it is not on the processor know.



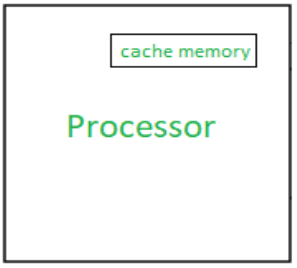
-there will be come delay because processor should memory and cache memory should transfer the data to the processor

-And there will be a small delay that happen during the transfer rate of data.

-To avoid this type of delay we place the cache memory on the processor

-then the delay of data reduced totally by placing the cache memory in the processor.

-so there is no transfer rate now.



4.Other performance enhancements:-There are several other performances enhancements are there they are;

- Write buffer:-
- Prefetching:-
- Lockup-free cache:- google ill nokkikkonam

***Virtual Memory:**-in over our concept processor need any memory it will check in cache Memory and then check in main memory if data is not found in cache memory and then check in secondary Memory if required data is not found in main memory .

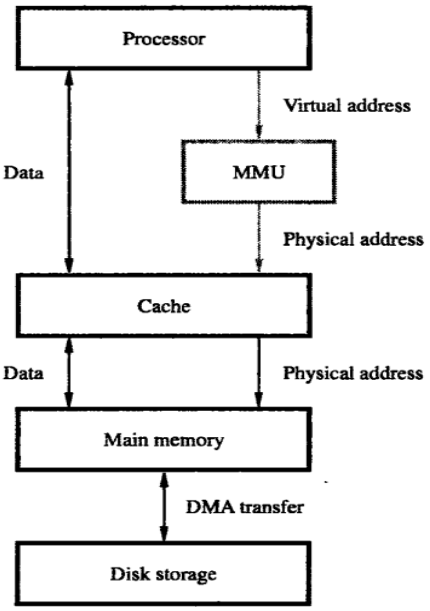
-it may consume time.

-the size of memory is always secondary memory is larger Than main memory and main memory is larger than cache Memory.

-The size of the main memory is very small compare to The size of the second memory.

-so our main memory can't able to store more number of Programs at a time.

-the solution for that is just increase the size of main memory.



-we can’t simply increase the main memory size because
It is very very costly to Implement.
-So we need a technique to increase the size of main memory
So, we take some space from the secondary memory and
Treat it as a main memory.
-Eg; our main memory size is 8 gb and our disk storage
(secondary memory) size is 500gb ,once the 8gb is not sufficient for as.
-out of 500 gb it can take 4 gb from the secondary memory as a main memory ,now we have 8+4 gb as main memory
-Than extra added 4gb is become our **virtual memory.**
-This 4gb is virtual memory because it is not really present but virtually we make it as a main memory.
-so the advantage is that we can increase the size of main memory and it can able to store more programs.
-how will data transfer take place from main memory and secondary memory
,How the data transfer take place between this 4 gb and 8gb memory ,it uses a technique called **DMA transfer.(direct memory access transfer)**
-if we are using some part of a secondary memory as a main memory from that the data will transfer to the main memory .it is called **DMA transfer (direct memory access transfer) .**
-directly to implement virtual memory technique our computer need a hardware called **MMU (memory management unit).**
-MMU is a hardware part not a software.
-if we need to access the virtual memory then the processor will send an address which is an virtual address and this virtual address will translate by MMU into a actual physical address.
-Through this physical address the required data from the secondary memory which is assumed as a main memory will be accessed.
-once the physical address is passed through the secondary memory then the secondary storage device will communicate with the main memory with the help of DMA transfer method.
- and then the data will be communicated to the processor. (main memory to cache and cache memory to processor)
-we use virtual memory when our main memory cannot able to store more amount of data ,if you need to increase the size of our main memory then we take some space from the secondary memory and you can use it as main memory this concept is called **virtual memory.**

***Memory Management Requirements:**-Memory management helps in keeping track of each memory location’s status – whether it is free or allocated.
-Memory management is meant to satisfy some requirements that we should keep in mind.
-These Requirements of memory management are:

1. Relocation:-The programmer does not know where the program will be placed in memory when it is executed while the program is executing, it may be swapped to disk and returned to main memory at a different location (relocated).
   -Memory references must be translated into the code to the actual physical memory addresses.
   -The address generated by the CPU is said to be a logical address.
   -An address generated by MMU is called a physical address.

2. Protection:-Processes should not be able to reference memory locations in another process without permission.
   -It must be checked at run time.
   -The memory protection requirement must be satisfied by the processor (hardware) rather than the operating system (software).
   -The word protection means provide security from unauthorized usage of memory.
   -The operating can protect the memory with the help of base and limit register.
   -Base registers consisting of the starting address of the next process.
   -The limit specifies the boundary of that job, so the limit register is also said to be a fencing register.

3. Sharing:-Allow several processes to access the same portion of memory.
   -It is better to allow each process access to the same copy of the program rather than have their own separate copy.

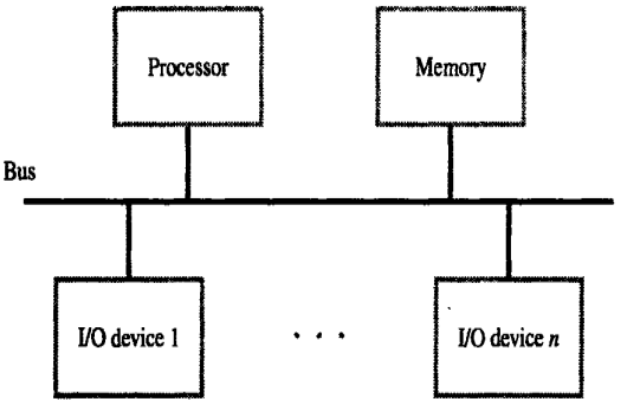
4. Logical organization:-Programs are written in modules. Modules can be written and compiled independently.
   -Different degrees of protection given to modules (read-only, execute-only). Modules are shared among processes.

5. Physical organization:-Memory available for a program plus its data may be insufficient.
   -Overlaying allows various modules to be assigned to the same region of memory.

#Input / Output Organization :-One of the basic features of a computer is its ability to exchange data with other devices.
-This communication is enabled by a human operator,
-for example, to use a keyboard and a display screen to process text and graphics.
-We make extensive use of computers to communicate with other computers over the Internet and access information around the globe.
Or
-We frequently use computers to access information all over the globe and communicate with other computers over the Internet.
-In other applications, computers are less visible but equally important.
-They are an integral part of home appliances, manufacturing equipment, transportation systems, banking and point-of-sale terminals.
-In such applications, input to a computer may come from a sensor switch, a digital camera, a microphone, or a fire alarm.
-Output may be a sound signal to be sent to a speaker or a digitally coded command to change the speed of a motor, open a valve, or cause a robot to move in a specified manner.
-In short, a general-purpose computer should have the ability to exchange information with a wide range of devices in varying environments.

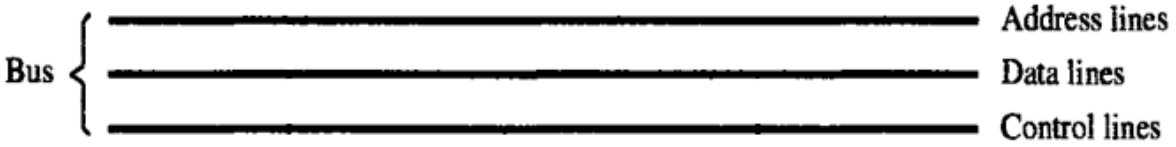
***Accessing I/O devices:-** input /output devices are one of the essential components of any computer .

-Without input and output devices there is no use of computer at all.
-The internal connection between the input and output devices which are in the outside world and to the processor and main memory communicate with the help of bus.
-Bus is a set of words which is used to connect the internal compounded of a computer.
-when ever a user use mouse the signal will be passed from input device to the processor first .
-And data will be passed to the processor from input device with the help of a bus.
-Then the processor grand the permission To the input device, to open come particure
-Program which is available in the main memory.
-Then the program will be opened to us with The help of double clicking the mouse.



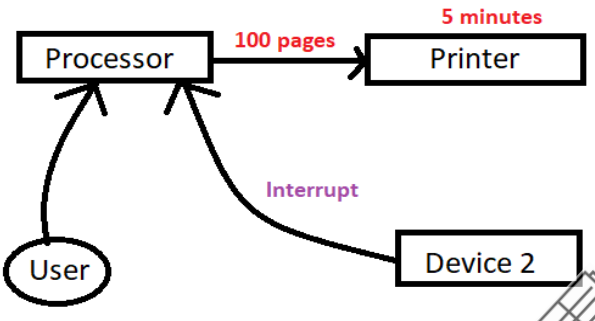
-Example: when ever the user is typing in the keyboard that data will be passed to the processor with the help of bus
-And again the processor will interact with the main memory to store the data temporary .
-And then if user what to display the data on the output devices such as monitor .
-Then again from main memory the data will be passed through bus to the output device.
-Here the main role played by a mediator is bus.
-In bus there are 3 types of connection that are available that help to achieve this communication they are;

- **Address lines:-** it is used for specify where the data should be accessed or where should be the data to be taken or where should be the data to be stored.
  -The actual address of the location will be determined in the address lines
- **Data lines:**-it is used to store the data
  -Eg; when ever you type on the keyboard the characters you typed are stored in the data line and transferred to the bus.
- **Control lines:-**it is used to control the activity/operation of the bus.
  -it is responsible for that the user is requesting the read operation ,write operation .



-every bus is a combination of these 3 things.
-which are responsible for transfer of information .

***Interrupts:-** interrupts simply means disturbance or abnormal termination.
-something which is not properly terminated is an interrupt.
-Example: we have a user, processor and output services as printer
-user want to take 100 pages of print .
-then first the request go the processor from the user.
-then with the permission of the processor only the control will go to the printer.
-and the 100 pages command go to the printer.



-And the printer cannot print 100 pages within
a minute. suppose it takes 5 minutes .
-when the printer is printing the 100 pages in the
duration of 5 minutes ,during this 5 minutes our
processor will be idle (free).
-which means the processor doesn't have any work to do.
-And the processor simply concentrating when the 100 pages is going to print it out.
-meanwhile there are some other device (device 2) which want the request of the processor
-then the device 2 has to wait for 5 minute and it is very time consuming .
-the device 2 is not waiting for it. so it send a signal which is known as **interrupt**
- interrupt is a hardware generated signal.
-And the device 2 send interrupt signal directly to the processor that the device 2 is ready for do this work or that work.
-once the processor gets the signal it give the permission to device 2 to perform the operation.

*Types of interrupts

→**Interrupts Hardware:-**interrupt generated by the hardware of the computer system is called hardware interrupts.
-Interrupt uses the interrupt line bus.
-All devices that uses an interrupts are connected to interrupt - request line through switches ground.
-To request an interrupt ,the device closes its switch.
-It all switches are open then the line is inactive at that time the voltage on device is vdd.
-If all switches are closed then the line voltage is dropped to 0.

-Interrupt signal ($\overline{INTR}$) is the combination of all interrupt signal generated by the switches (INTR1+INTR2+.....INTRn).

$$\overline{INTR} = INTR1 + INTR2 + ... + INTRn$$

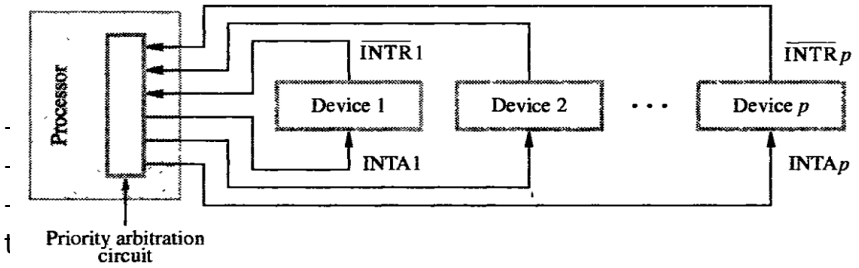
→**Enabling and Disabling Interrupts:**-When an interrupt from an external device causes the processor to suspend the execution of one program and starts the another.
-Computer has the ability to enable and disable interrupts as desired.
-When a device request the interrupt during the processor service for another interrupt, the result cause the processor enter into the infinite loop.
-when computer is processing a device1 and devices 2 send a interrupt signal and then the the computer accepting the device2 it is called enabling interrupt.
-When computer is processing a device1 and device 2 send a interrupt signal ,at some cases the computer ignore the request from the device2 and continue the process with device1 is called disabling the interrupt.

→ **Handling Multiple Devices:**-several devices are connected to the processor which can generate interrupt signals.
-As the device are independent they can generate the interrupt signal independently without considering the other devices.
-the possible solution are;

- If 2 devices generate interrupt signals it can cause a tie. Processor select one and breaks the tie.
- Once it is selected interrupt is completed and other interrupt signals which is treated as in a queue

-|There are some approaches to handling multiple devices they are;

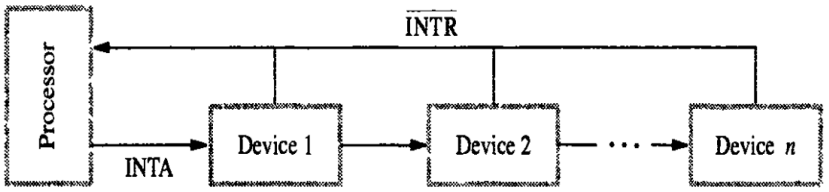
→**vectored interrupt:-**The device which is generated the interrupt signal it has to make itself visible to the processor by sending a special code .
-so this type of approach is called vectored interrupts approach
-So a device requesting an interrupt can identify itself by sending a special code to the processor over the bus
→ **Interrupt nesting:-**



n once

-The processor sending an acknowledgement to the device back.
-This is how the processor is nesting the devices.

→**Simultaneous Requests:**-suppose if n number of devices are there ,they all raised the interrupt request at the same time .
-then the processor is selecting one .



-based on the priority it make a queue.
-highest priority will selected first and then so on
-This is called simultaneous request.

***Controlling Device Requests:**-There are two types of devices are there, one which use programs and the device which not use programs those devices are called Idle devices.
-it is important that only devices with program can only generate interrupt request.
-Idle devices must not allowed to generate interrupt request.
-There should be a mechanism which can ensure that those devices which use programs which are not idle ,they can rise the interrupt signals.
-so that the controlling mechanism is provide a bit .that bit make sure that are communicate the devices with the processor .
-that those devices send an interrupt request.

-The keyboard interrupt enable(KEN) it is a input interrupt enable and display interrupt enable (DEN) it is a output interrupt enable ,The flags in register CONTROL are used to perform the function .
-Here KEN and DEN are flags.
-The bits that provided by this mechanism are KIRQ and DIRQ .
-KIRQ is the bit that is related to the keyboard and it is the input unit .
-And DIRQ is the bit that is related to the display unit and it is the output unit.

***Exception:**-Any event that causes an interruption is called as Exception.
-I/o interrupts are one of the example of exception.
-There are few kinds of Exception they are ;

1. **Recovery from errors:**-The computers use error checking code (ECC) to identify the errors in the stored data.
   -once the error occurs, the controlled hardware detect this error and informs the processor by raising an interrupt.
   -When the error is detected the processor suspends the program being executed and then immediately it will take an action to recover that error .
   -so that recovery is done by a routine called exception service routine.

2. **Debugging:**-System software usually includes a program called debugging.
   -Which helps the programmer to find errors in the program.
   -It provides two types of facilities they are, **trace** and **breakpoints.**
   -Trace: If an exception is raised after each and every instruction is called Trace.
   -And it enables the user to examine the contents of registers,memory location ,and so on.
   -Breakpoint :The programmer selects specific points in the program and after that specific points only the exception will be raised.
   -And the breakpoints also provide the same kind of facilities as provided by the trace.

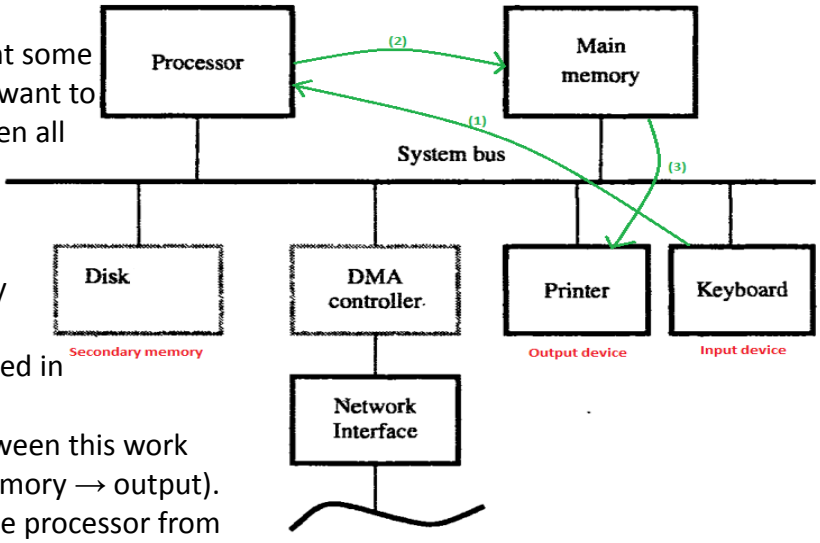
3. **Privilege Exception:**-Normally in a computer there is an operating system and there may be a chances that the operating system gets corrupted due to many reasons.
   -so, it is operating system responsibility to protect to do something to protect that being corruption.
   -so operating system run some of the OS routine to protect the operating system from being corrupted .
   -so those instruction programs are called privileged instructions.
   -The exception created by this privilege instruction is called a privileged Exception.
   -This are the features done by the privilege exception.
   - Switch from one user program to another user program .
   - Implement the security and protection feature.
   - Coordinate i/o activity.

->Interrupt v/s Exception

Interrupt	Exception
Interrupts are unexpected events that put the normal flow of execution of instructions to a halt which prompts the OS to take immediate action.	Exceptions are unexpected events that exist somewhere in the system, the processor, or within a program that requires attention of the CPU

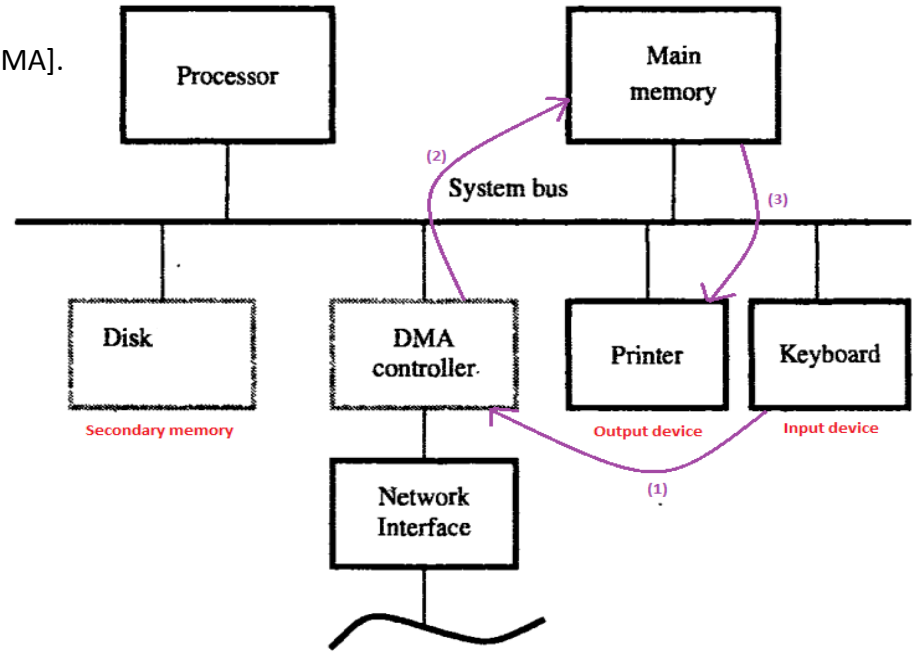
Interrupt is one of the many classes of exception.	Exception is divided mainly into four classes: interrupt, fault, trap and abort.
Interrupts occur at random times during a program execution in response to hardware signals.	Exceptions occur when the CPU detect an anomaly during execution of an instruction, such as divide by zero exception.
Interrupts can be generally classified as synchronous and asynchronous interrupts.	Exceptions can be generally classified as processor-detected exceptions and programmed exceptions.

#Direct memory access (DMA)/DMA operations:- If the user want to type something the request from keyboard to processor will go.
-once the processor give the permission then only the control go to the main memory.
-and it pass the output to the respective output device.
-And the bus is responsible transfer the information.
-there is a major drawback in this method.
-when ever the user input some character,when ever the user want some input device ,when ever the user want to communicate with the system ,then all the request should go to the processor .
-only with the permission of processor only the user can do any operations.
-every time the processor is involved in the work .
-if we remove processor from between this work
(input -> processor -> main memory → output).
-the big advantage by removing the processor from the work is that we can save lot of time.

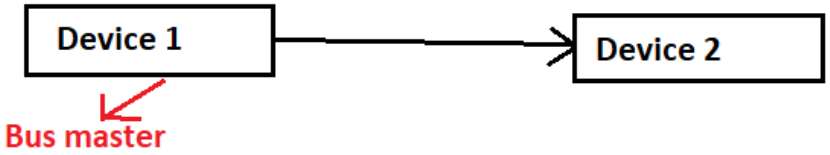


-But the problem is the without processor there is no computer.
-Our aim is to speed up the communication process between input and output devices to main memory and we can't remove processor.
-There is a solution for this problem,that there is not need the involvement of the processor in the work . by introducing the DMA controller.
-The hardware module is known as DMA controller .which we can place it and connect it to the bus .
-Then whenever the user requests it go to the DMA controller instead of going to processor
-And DMA controller directly communicate with the main memory .
-By using DMA controller we can

speed up the process .
-This method is called
Direct memory access [DMA].



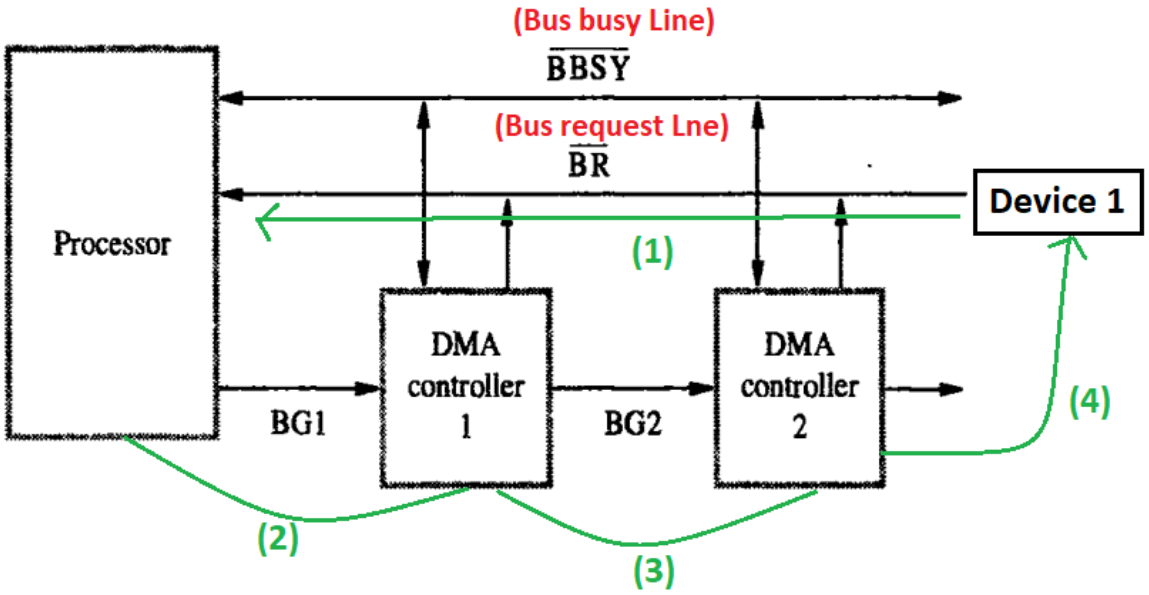
*** Bus Arbitration:**-Bus is used to connect all the internal components of our computer.
-if one device wants to communicate with another device, here the bus acts as a mediator..
-Example : If we have two devices (device1 and device2) to communicate.
-And here one device needs to start the communication. (communication means transferring of data)
-suppose device1 sends the data first to device2 .Then the device1 will become the bus master .



-we call device1 as bus master because it started the communication.
-the problem here is how to select the bus master or how we know which device will start the communication.
-The process of granding the bus mastership to a device based on its capability is know as **Bus Arbitration .**
-Bus Arbitration means it is a process of deciding who is the bus master.
-Bus Arbitration can be done on our computer by using 2 methods.
-The main purpose of this 2 methods to find out the bus master.

1. **Centralized arbitration**:-In the fig we have 2 DMA controller .
   -our aim is to find out which device is a bus master.
   -BBSY it indicates that Bus busy line.
   -Which means sometimes more than one device is communicating with the bus then the bus becomes busy, so it may not fulfill the request also.

-Here device 1 want to communicate with the another device ,then the device 1 should send one bus request to the processor .



-BR stands for bus request line.
-device 1 send the bus request to the processor with the help of DMA controller .
-DMA controller will look after all the crosses checking activities here ,whether the requested device is a trusted one or the requested device is a genuine one or not.
-so this DMA controller will look after the request and thy will pass to the processor .
-once the processor feels ok for the particular device to grand the bus mastership .
-then the processor is going to generate two signals BG1 (bus grand 1) and BG2 (bus grand 2).
-Device has request the bus with the help of BR signal and processor give back the permission to the device with the help of BG signal.
-The BG1 will given to the DMA controller 1 and DMA controller 1 convert the signal to BG2 ,then BG2 will pass to DMA controller 2.
-then the DMA controller 2 give back the bus mastership to the device1.
-with the help of centralized activity bus mastership will given to the particular device,that's why we call this method as **Centralized arbitration.**
-here processor is the decision maker here and the processor will take the request and processor will grand the request tha's why we call this method as **Centralized arbitration.**

2. **Distributed arbitration**:-Here there is no particular involvement of the processor here.
   -All the devices they can communicate with each other and the device which is in need of bus in a urgent way for the device bus mastership will be provided.

-Centralized arbitration is always better than Distributed arbitration because the processor is not completely involved in decision making ,that there is a chances that there are some devices which may not get the bus mastership at all.
-for those devices it is a drawback .
-so that's why our computer prefers Centralized arbitration method.

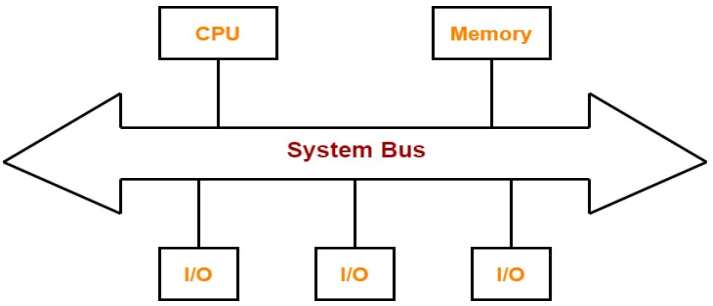
*** Buses:**-The processor, main memory, and I/O devices can be interconnected by a common bus whose primary function is to provide a communications path for the transfer of data.

-Bus can transfer data from left to right and right to left.

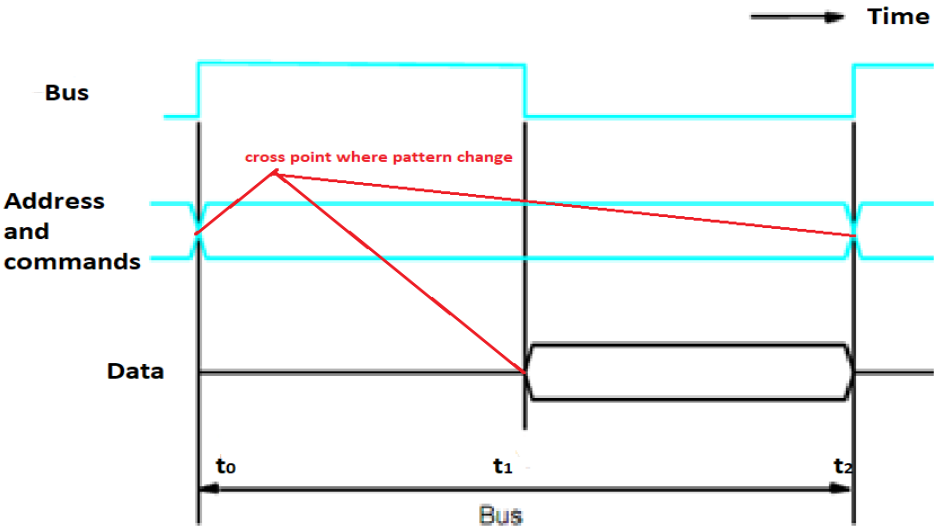
-Mukalill bus ina kurichu ezhuthitunde athum Evida ezhutham.

-Bus master oka parayanam

-Types of buses are;



1. **Synchronous bus:**-In a Synchronous bus all the devices are connected to the clock line.
   -clock line is having equally spaced time interval.
   -each time intervals can create one bus cycle.
   -in one bus cycle we can transfer one data.
   -cross section point is the point at which the pattern change.



   or
   The transactions between devices is under the control of a synchronizing clock signal.
   -A transmission error will take place if a connected device fails to reply to a command within the time frame dictated by the frequency of the clock signal.
   -Every device on the bus must run at the same speed.
2. **Asynchronous bus**:-The transfers between devices are self-timed rather than controlled by a synchronizing clock signal.
   -Transmitters and receivers are not synchronized by clock.
   -it can accommodate more devices.

→Difference between Synchronous and Asynchronous bus are;

Sr. No.	Topic	Synchronous Bus	Asynchronous Bus
1.	**Clock Rate**	A synchronous bus works at a fixed clock rate.	An asynchronous bus is not dependent on a fixed clock rate.
2.	**Clock Synchronization**	Transmitter and receivers both are synchronized with the clock.	Transmitters and receivers are not synchronized with the clock.
3.	**Clock Skew**	Synchronous Bus affected by clock skew.	Asynchronous Bus not affected by clock skew.
4.	**Bus Length**	The length of a synchronous bus could be limited to avoid clock-skewing problems.	The length of the asynchronous bus could not be limited.
5.	**Bus Protocol**	Bus protocol is predetermined in Synchronous Bus.	Bus protocol is not predetermined in Asynchronous Bus.
6.	**Physical Distance**	Synchronous buses cannot handle longer physical distances.	Asynchronous buses can handle longer physical distances.
7.	**Number of Devices**	Synchronous buses cannot handle a higher number of devices.	Asynchronous buses can handle a higher number of devices.
8.	**Data Transfer**	Data transfer takes place in the block.	Data transfer is character-oriented.
9.	**Data Bits Transmission**	Bits of data are transmitted with the synchronization of the clock.	Bits of data are transmitted at a constant rate.
10.	**Character Rate**	Character is received at a constant Rate.	Character may arrive at any rate at the receiver.
11.	**Speed of Buses**	Synchronous Buses are faster.	Asynchronous Buses
12.	**Speed of Data Transmission**	Used for high-speed data transmission.	Used for low-speed data transmission.
13.	**Overhead**	No overhead is present to establish a time reference for each transaction.	Overhead is present to establish a time reference for each transaction.
14.	**Finite State machine**	Require very less logic to implement Finite State machine.	Require more logic to implement Finite State machine.
15.	**Type of Buses**	Processor-memory buses are typically synchronous because the devices connected to the bus are fast, are small in number, and are located in close proximity.	I/O buses are typically asynchronous because many peripherals need only slow data rates and are physically situated far away.

***Interface Circuits :**-The I/O interface circuit is a mediator between the I/O device and the system to which this I/O has to be connected.

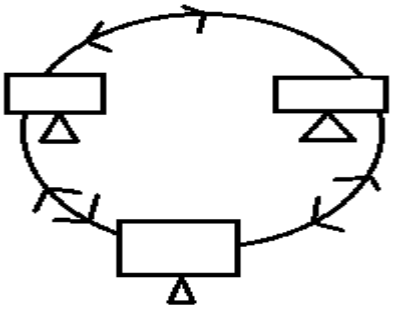
-The I/O interface circuit is circuitry that is designed to link the I/O devices to the processor.

-it is also known as end point connectivity.

-The end to end connection between various systems can be done with the help of this two ports.

-there are two ways in which we can connect the interfaces they are;

1. **Parallel ports:**-from one system to another system the transfer of data can be done **simultaneously** by using **more number of bits at a time** .

   -which means one computer to another computer which are making of parallel ports they can transfer the data many bits at a time

   -And the transfer of data can be take place simultaneously .

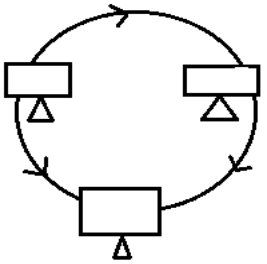
   -and the transmission of data can be take place very fast.

   -Here two way communication is possible.

   -only possible when all the system at a closed distance.

   -This method is very costly .

2. **Serial ports:**-it can transfer the data only one bit at a time.

   -there is no simultaneously transfer of data .

   -it is a slow transmission.

   -only one way communication is possible.

   -it is applicable for long distance.

   -less cost .

-The parallel port have many advantages than serial port but there is some problems also.

***Standard I/O Interface:**-A standard I/O Interface is expected to fit the I/O gadget with an Interface circuit.

-Types of Standard i/o interfaces are;

1. PCI (Peripheral Component Inter Connect):-explain cheyandaa
2. SCSI (Small Computer System Interface) : - explain cheyandaa
3. USB (Universal Serial Bus):-The main feature of usb is that the plug and play feature.

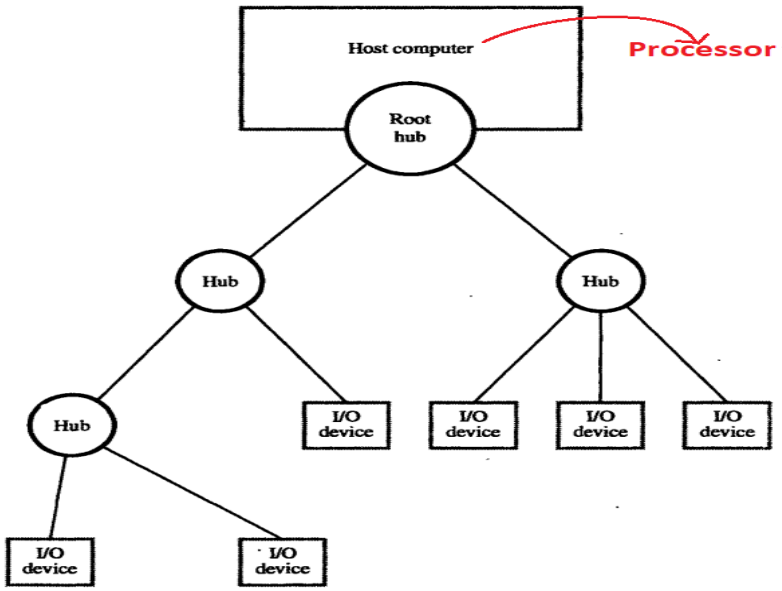
   -This feature made the usb unique and different from others.

   -no need of installing any software ,just simply connect the bus ,and as soon as the bus connected the transfer of data can take place.

   -it will support dual speed of operation which means it can support the lowest speed 1.5 megabits/s or with a moderate speed of 12 megabits/s.

   -the recent version of usb 2.0 can support speed of 480 megabits/s.

->fig



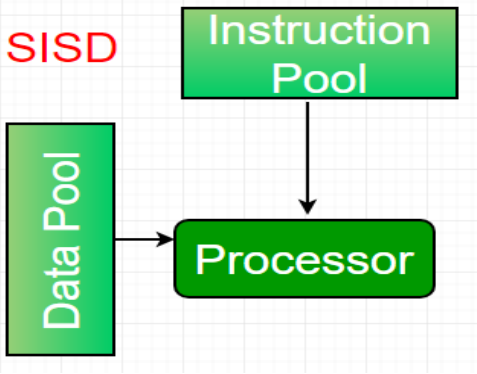
-host computer means the processor.
- hub: it used to connect multiple devices.
-Root hub is connected to the host computer and other hubs are connected to the root hub.
-A hub can used to connect other hubs and input/output devices.
-or we can connect hub only with the input/output devices.
-we can form tree structure thats why it also called **usb tree structure.**

Module-5 -Advanced Computer Architecture

***Parallel Processing:**-The purpose of parallel processing is to enhance performance of our computer.
-Concurrent data processing is possible in a parallel processing system, which leads to quicker execution with less time .
-As an example, the next instruction can be read from memory, while an instruction is being executed in ALU.
-According to M.J. Flynn there are 4 major classification of parallel processing.
-M.J. Flynn classification based on the number of instructions as well as data items that are changed at the same time.

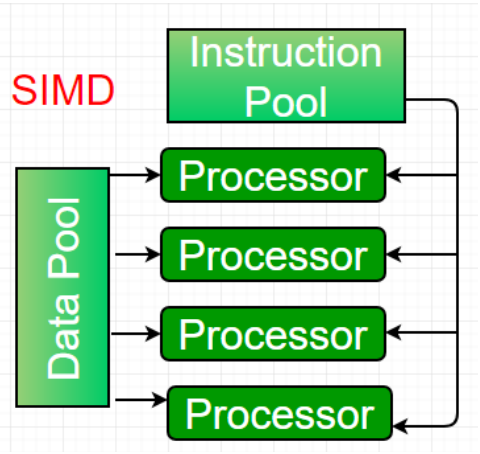
→**Flynn's Classification are**;

- **Single Instruction and Single Data stream (SISD) :**-:-An SISD computing system is a uniprocessor machine which is capable of executing a single instruction.
  -it operates on a single data stream.
  -In SISD, machine instructions are processed in a sequential manner and computers in this model are popularly called sequential computers.
  -Most conventional computers have SISD architecture.

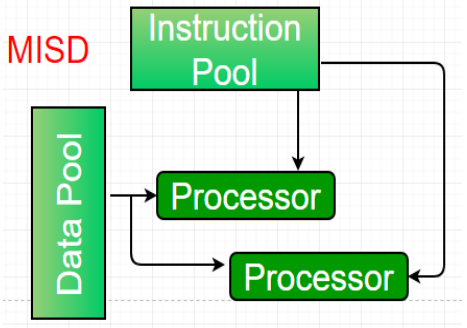


  -All the instructions and data to be processed have to be stored in primary memory.
  -The speed of the processing element in the SISD model is limited(dependent) on the computer it used.

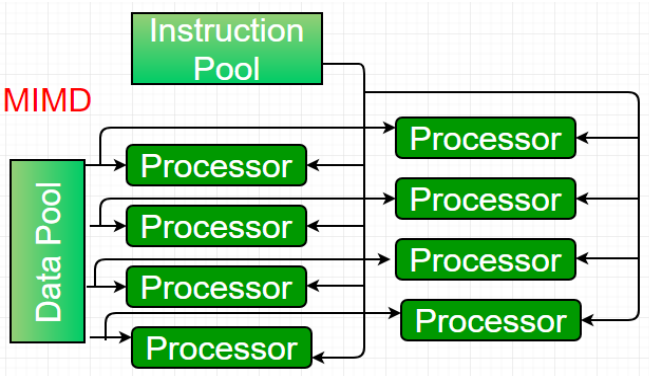
- **Single Instruction and Multiple Data stream (SIMD)**:-
  -An SIMD system is a multiprocessor machine capable of executing the same instruction on all the CPUs.
  -It operates on different data streams.
  -Machines based on an SIMD model are well suited to scientific computing since they involve lots of vector and matrix operations.



- **Multiple Instruction and Single Data stream (MISD)** :-An MISD computing system is a multiprocessor machine capable of executing different instructions on different PEs(Processing Elements).
  -All the operates are on the same dataset.
  -The system performs different operations on the same data set.
  -Machines built using the MISD model are not useful in most of the application, a few machines are built, but none of them are available commercially.

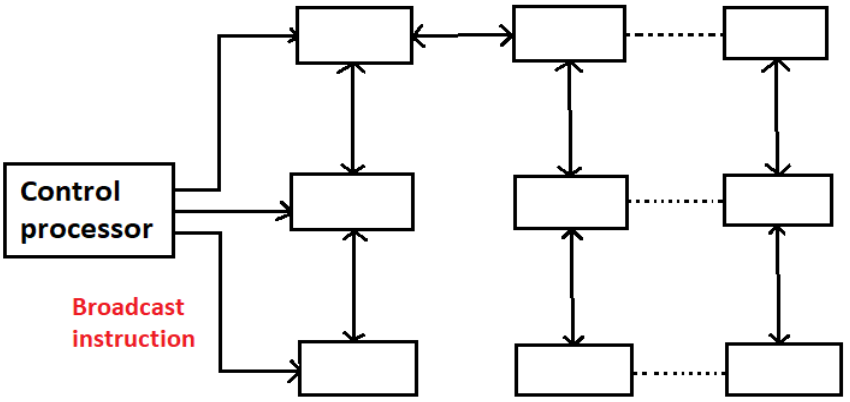


- **Multiple Instruction and Multiple Data stream (MIMD)**:-An MIMD system is a multiprocessor machine which is capable of executing multiple instructions on multiple data sets.
  -Each PE (Processing Elements) in the MIMD model has separate instruction and data streams.



-Therefore machines built using this model are capable to any kind of application.
-Unlike SIMD and MISD machines, PEs in MIMD machines work asynchronously.

***Array Processors**:-is one of the important method used to enhance the performance of a computer.
-Array means that it is a collection of similar data.
-In Array processor we are not using one processor instead we use a group processor of similar type.
-Array processor is an example that belongs to the category of SIMD.
-the number of processor in array processor is depend on our complexity of the computer.



Grid of processing elements

-In the figure we take n number of processor.
-All this processor are connected with one another .
-To control all this processors array there is one more processor called control processor .
-The control processor is responsible to handle the remaining processor here.
-The control processor will give the set of instruction to all the processors to handle the task.
-Broadcast instruction are used to sending the instruction or forwarding the instruction to multiple resources or multiple sources.
-Then the processor will take the data and the will do the work.
-we call this entire set of array of processor as grid of processing elements.

***Vector processors:**-Vector processor is basically a central processing unit that has the ability to execute the complete vector input in a single instruction.
-More specifically we can say, it is a complete unit of hardware resources that executes a sequential set of similar data items in the memory using a single instruction.
-The scientific and research computations involve many computations which require extensive and high-power computers. These computations when run in a conventional computer may take days or weeks to complete.
-The science and engineering problems can be specified in methods of vectors and matrices using vector processing.

-computers with vector processing capable for;

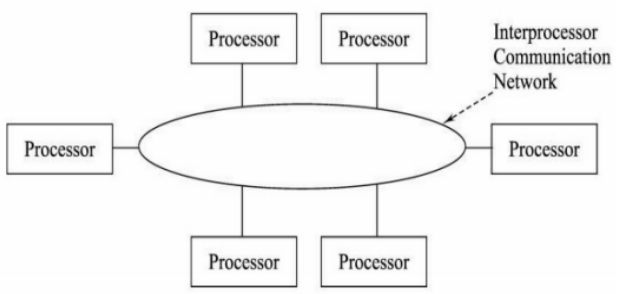
- Long-range weather forecasting.
- Petroleum explorations.
- Medical diadnosis.
- Image processing

- Vector Processor are Classification into two types they are;

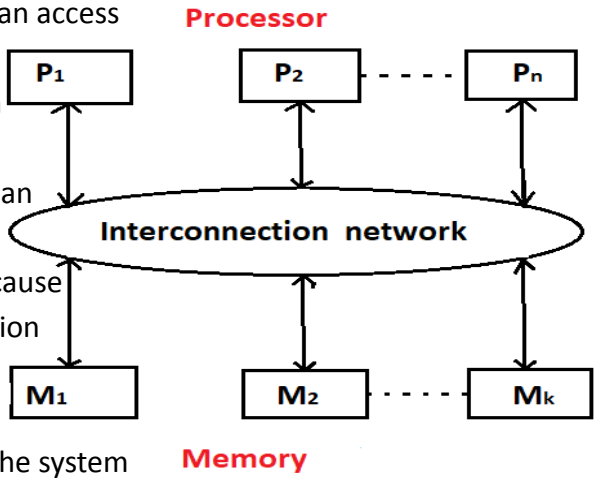
1. Register to Register Architecture:-In register to register architecture, operands and results are retrieved indirectly from the main memory through the use of large number of vector registers or scalar registers.
   -It has limited size.
   -Speed is very high as compared to the memory to memory architecture.
   -The hardware cost is high in this architecture.
2. Memory to Memory Architecture:-In memory to memory architecture, source operands, intermediate and final results are retrieved (read) directly from the main memory.
   -There is no limitation of size
   -Speed is comparatively slow in this architecture.

*Structure of Multiprocessors/structure of general purpose multiprocessor:-

In general purpose multiprocessor data parallelism will not present.
-But there will be processors are there.
-This multi processors can execute different routines parallelly.
-There are 3 types of multiprocessor systems they are;

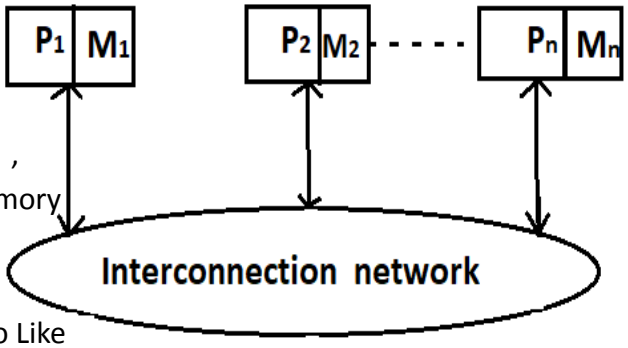


1. **UMA multiprocessor**:- UMA stands for uniform memory access.
   -There are n processor and each processor can access k number of memory.
   -They all are connected with communication lines.
   -In between processor and memory there is an interconnection between them.
   -This network can experience time delay because of the processor location and memory location apart.
   -Because of it there will be some delay.
   -If we eliminate this short time delays then the system will become very costly and complex to implement.
   or



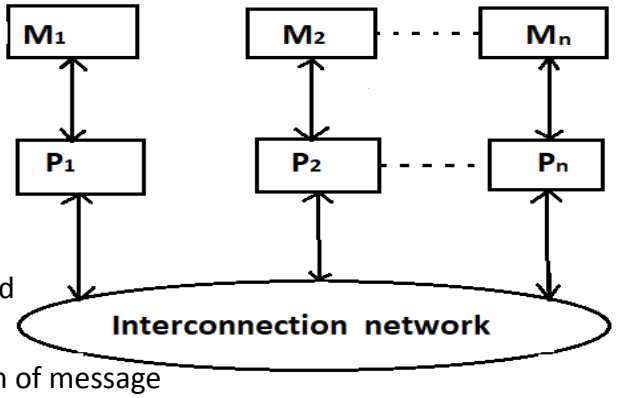
-interconnection network with very short delays are costly and complex to implement.

2. **NUMA multiprocessor**:- NUMA stands for non uniform memory access.
   -attaching memory units to processor instead of away from processor in UMA.
   -by attaching memory and processor we can remove the delay because they are located in same position.
   -In addition to accessing its local memory , each processor can also access other memory over the network .
   -Eg processor P1 not only can access m1 but also it can access other memories also Like M2,M3…Mn in the network.



   -It take time to access P1to M2 or M3 or Mn than M1.
   -The time is not uniform when P1 is accessing M1 it take only less time but P1 to M2 take more time than M1 because of the location
   -And the time taken to access M3 by P1 is more than M2 and M1 .that's why it take different time to access the memory
   -That's why it is called NUMA (non uniform memory access.

3. **Distributed memory access:**-It is the combination of UMA and NUMA.
   -by combining UMA and NUMA we get a global memory where any process can access any memory module without intervention of any processor.
   -Here all memory modules act as private memories for the processors that are directly connected to them.
   -A processor cannot access a remote memory without the co-operation of the remote processor.
   -Example : if P1 want to access Mn it need the co-operation of Pn.



   -The co-operation taken place in the form of message exchange by the processor.
   -Eg: if Pn want to access the M1 ,then the Pn will pass the message to the P1 .Then P1 gives the privilege to access the memory M1.
   -This system are called distribution memory system with a message passing protocol.

***Interconnection Networks:**-Is a network between the processor and the memory system.

-Interconnection network is a combined network which where used to pass the messages or information.

-Any processor can access Any memory unit through only interconnection network.

-A network can judge in terms by using the cost,bandwidth,effective throughput and ease of implementation.

-This are the types of interconnection networks;

1. **<u>Single bus:</u>**-Bus is a connecting line between one or more devices which can be used to transfer the data instruction and address

   -Single bus have only one bus.

   -It can connect to the number of memory modules .Memory modules means when we divide the main memory into the number of parts and each part is called as memory module.

   -Since several modules connected to the bus and any module can request a data transfer at any time.

   -This bus is dedicated to a particular source to destination pair for the full duration ,It means that no other device can use the bus when it is serving a particular purpose.

   -The bus will be idle during memory access, and when transfer is completed the bus can be assigned to handle another request.

   -The main limitation is that the number of modules that can be connected to the bus is not large.

2. **<u>Crossbar network</u>**:-It is look like a array network.

   -there will be many module and each module is connected to through a link and switches are present .

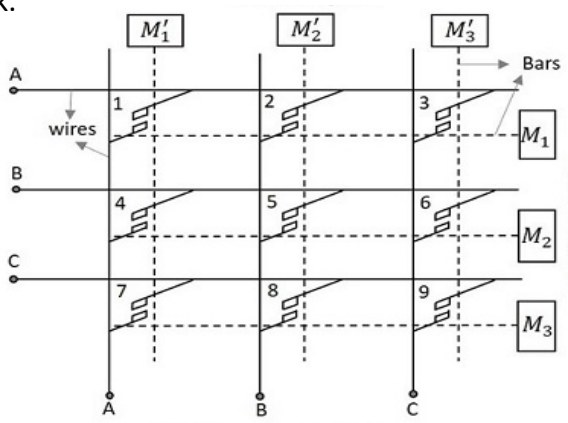
   -any module can communicate with any other module by closing the appropriate switches.

   -If there is a link between the module and the nodes or module to module then that network is called fully connected network.

   -we can send all the message concurrently to n number of destination from n number of sources.

   -so no there is delay of time to waiting for bus.

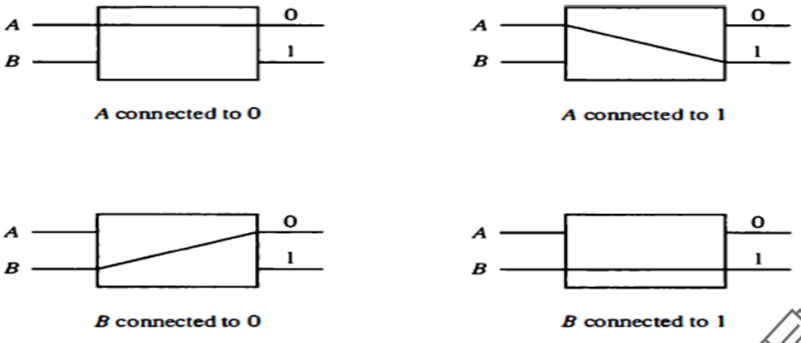
   -



3. **<u>Multistage network</u>**:-It is the ability of the switch to produce multiple outputs.

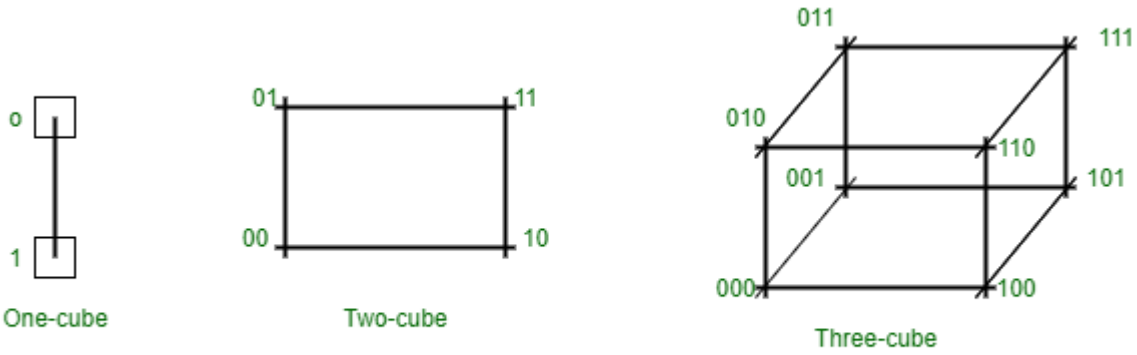
   -EXample:In the figure A and B Are two inputs and 0 and 1 Are the two outputs .

   -Here we have 4 types of figure In fig 1 Input A is connected to Output 0 and in fig 2 A is



Connected to output 1.
-here either we can get the output 0 or 1 depending on the processor which are connected to it.

4. **Hypercube Network:**- we can form the hypercube network with the help of this formula $N = 2^n$ Processors.
-Hypercube structure for n=1,2,3



-if we take n=1 then we get 2 processor (0 indicate processor1 and 1 indicate processor2 in the figure) . ($N = 2^n$)
-if we take n=2 then we will get 4 processor,(00,01,10,11) and we can form two cube structure.

*** Memory Organization:**-The memory is organized in the form of a cell, each cell is able to be identified with a unique number called address.
-Each cell is able to recognize control signals such as "read" and "write", generated by CPU when it wants to read or write address.
-Whenever CPU executes the program there is a need to transfer the instruction from the memory to CPU because the program is available in memory.
-To access the instruction CPU generates the memory request.
-Memory request contains the address along with the control signals.

#Pipelining:- To enhance the performance of the computer we can use the concept pipeline by performing multiple executions of programs at the same time.
-suppose our program has 3 instruction (I1,I2,I3) ,if all the 3 instruction are executed then only we can say that our program is successfully executed.
(program is a set of instruction)
- The executions are in a sequence order because instruction 2(I2) only start after the completion of instruction 1 (I1) so it is a sequential execution.
-The draw back of this method is that the speed of execution is low.
-we can improve the performance of a computer by pipelining the concept that we can perform multiple instructions at the same time.

-concurrent execution is possible in the pipeline .

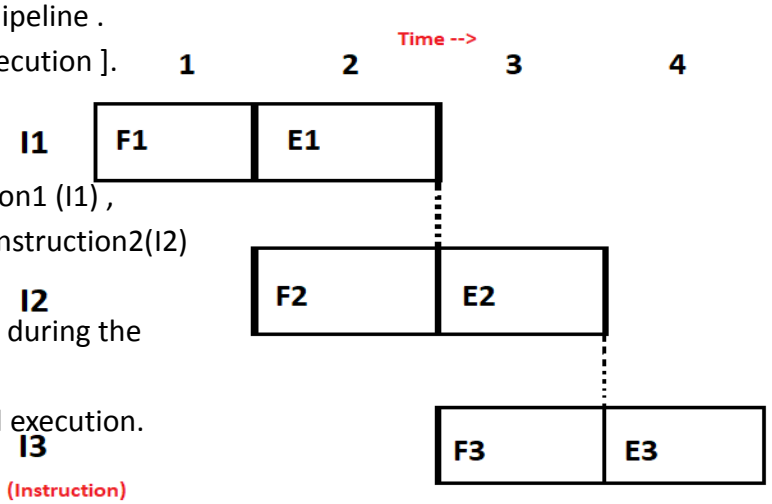
-Flg [F=fetch , E=execution].

-Instruction 1(I1) will fetch(F1) and execute(E1) normally.

-but during the execution(E1) of instruction1 (I1) , instruction 2 (I2)will be fetched(F2) and instruction2(I2) will be executed (E2).

-In next step Instruction 3 will be fetched during the execution of instruction 2.

-This type of execution is called pipelined execution.



→**There are mainly two types of pipeline**;

1. Arithmetic Pipeline
2. Instruction Pipeline

***Arithmetic pipelining:**-Arithmetic Pipelines are commonly used in various high-performance computers.

-It is used to speed up the processor.

-They are used in order to implement floating-point operations, fixed-point multiplication, and other similar kinds of calculations that come up in scientific situations.

-An arithmetic pipeline divides an arithmetic problem into various sub problems for execution in various pipeline segments.

-Eg: let us consider an example of a pipeline unit for floating-point addition and subtraction.

-The combined operation of floating-point addition and subtraction is divided into four segments

- Compare the exponents by subtraction.
- Align the mantissas.
- Add or subtract the mantissas.
- Normalize the result.

$$x = 0.90 * 10^3$$

$$Y = 0.82 * 10^2$$

-> step 1 is Compare the exponents by subtraction

-here exponent are 3 and 2.

-by comparing 3 and 2 ,and 3 is the greatest number ,so the final result exponent will be 3 .

->step 2 is Align the mantissas

-assign the equation 2 exponent as 2 ($Y = 0.82 * 10^2$) to 3 ($Y = 0.82 * 10^3$)

->step 3 is Add or subtract the mantissas

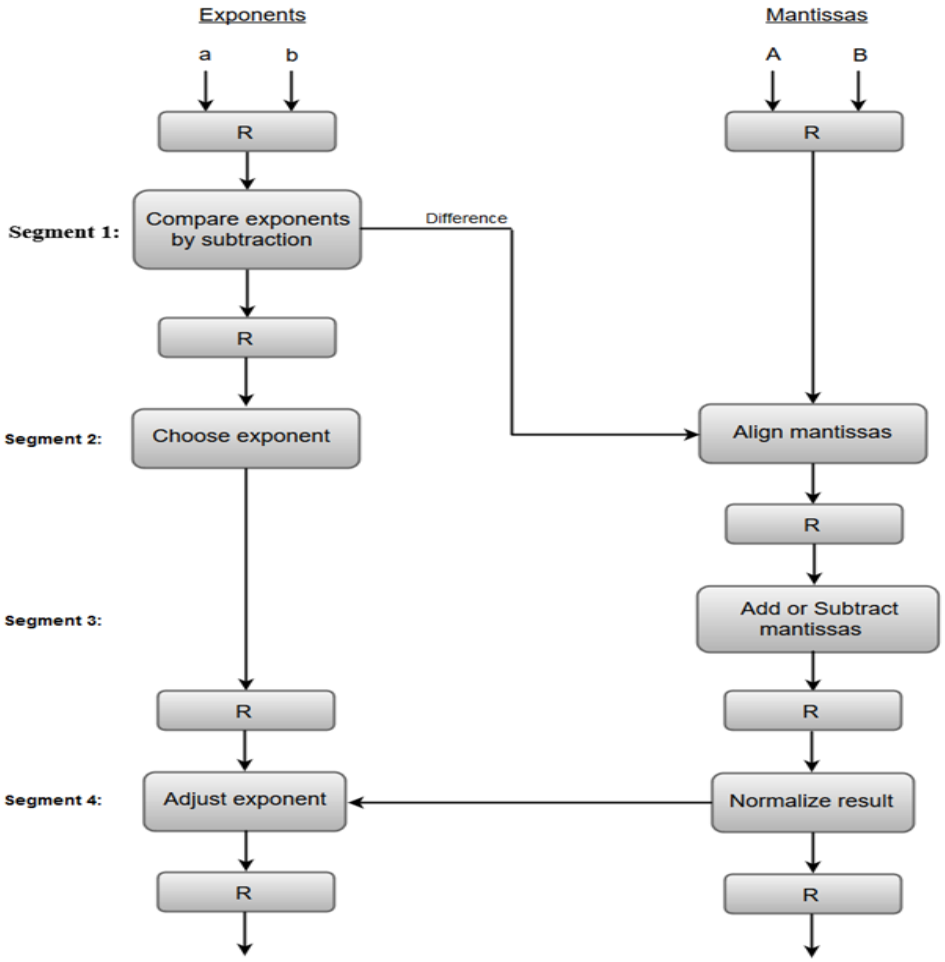
- **x= 0.90 * 10^3** + **Y = 0.82 * 10^3** = Z = 1.72 * **10^3**.

->step 4 is Normalize the result

-we don't keep any value before decimal point ,so replace Z = 1.72 * **10^3** to Z = 0.172 * **10^4**.

-fig

Pipeline organization for floating point addition and subtraction:



***Instruction Pipelining:**-In this a stream of instructions can be executed by overlapping fetch, decode and execute phases of an instruction cycle.

-This type of technique is used to increase the throughput of the computer system.

-An instruction pipeline reads instruction from the memory while previous instructions are being executed in other segments of the pipeline.

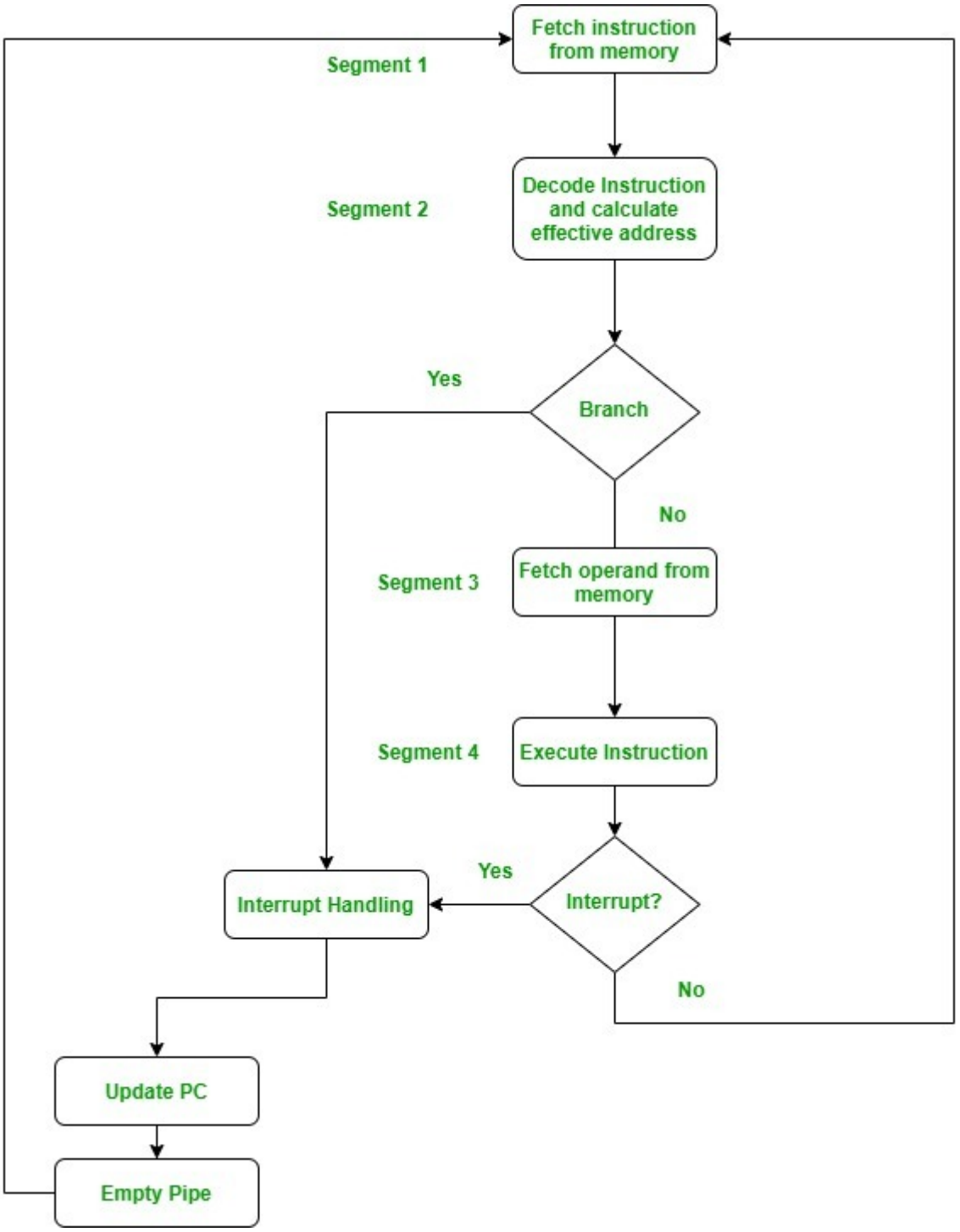
-Thus we can execute multiple instructions simultaneously.

-The pipeline will be more efficient if the instruction cycle is divided into segments of equal duration.

-step in instruction pipeline are;

- Fetch the instruction from memory (FI)
- Decode the instruction (DA)

- Calculate the effective address
- Fetch the operands from memory (FO)
- Execute the instruction (EX)
- Store the result in the proper place



-eg:

Instruction Branch

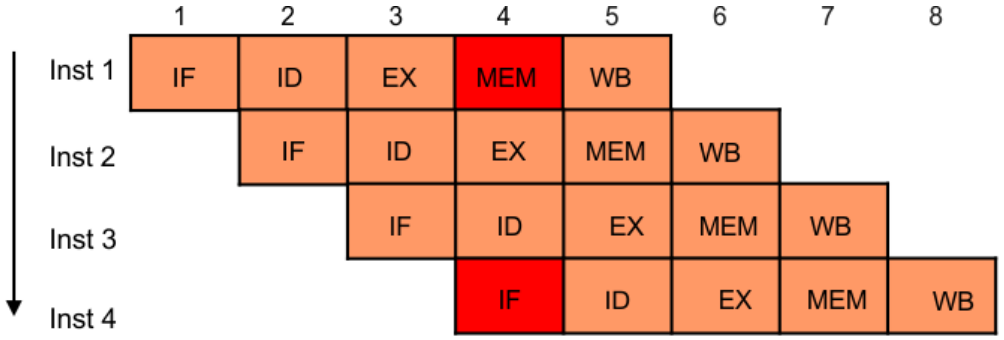
Stage	1	2	3	4	5	6	7	8	9	10	11	12	13
1	FI	DA	FO	EX									
2		FI	DA	FO	EX								
3			FI	DA	FO	EX							
4				FI	---	---	FI	DA	FO	EX			
5								FI	DA	FO	EX		
6									FI	DA	FO	EX	
7										FI	DA	FO	EX

-Here the instruction is fetched on first clock cycle in segment 1.
-Now it is decoded in next clock cycle, then operands are fetched and finally the instruction is executed.
-We can see that here the fetch and decode phase overlap due to pipelining.
-By the time the first instruction is being decoded, next instruction is fetched by the pipeline.

-In case of third instruction we see that it is a branched instruction.
-Here when it is being decoded 4th instruction is fetched simultaneously.
-But as it is a branched instruction it may point to some other instruction when it is decoded. -Thus fourth instruction is kept on hold until the branched instruction is executed.
-When it gets executed then the fourth instruction is copied back and the other phases continue as usual.

***Hazards:**-Pipeline Hazards are situations that prevent the next instruction in the instruction stream from executing.
-Hazards reduce the performance from the ideal speedup gained by pipelining
-Three types of hazards they are;

- **Structural Hazards:-**Structural Hazard occurs when multiple instructions need the same resource.
  -For example, the case when multiple instructions are fetching and writing the data to the main memory at a time cause the structural hazard.
  -Resources may be Memory, cache, registers, Alu, Common bus.



  -Eg:Structural Hazard happens due to conflict miss. If no of resources are limited and higher the no of instructions then conflict miss will occur.

-For example, if 20 resources and 100 instructions then conflict miss will occur. We can never equalize resources and instructions because resources are too costly. In this way, system will become too costly.

→Real-Life Example: If there are 500 students in the block then there may be 20 to 30 washrooms. Washrooms never are equal to no. of students. So, two students cannot use the same washroom at the same time due to which conflict miss will occur.

- **Data Hazards:-**The Data hazards occur when one instruction depends on a data value, which will be produced by a preceding instruction which is still in the pipeline.

-Data hazards are of three types

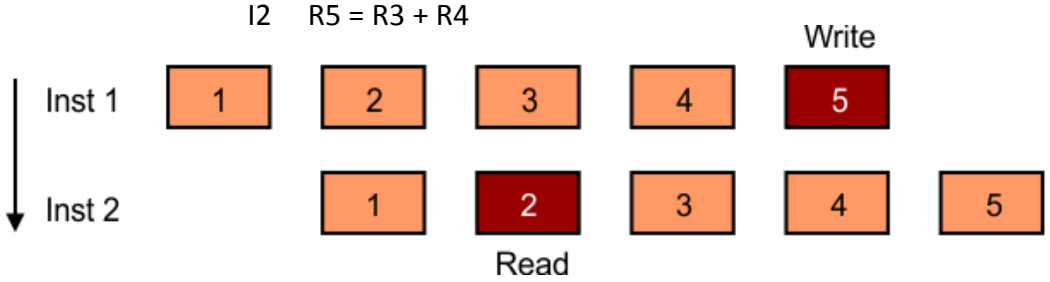
1. Read After Write (RAW):-A Read-After-Write hazard occurs when an instruction needs the result of some issued instruction, but that instruction is till uncompleted.

   -In the following example of RAW, the second instruction requires the result of R3 to ADD with R4 But it has not yet been produced by the first instruction.

   -Example

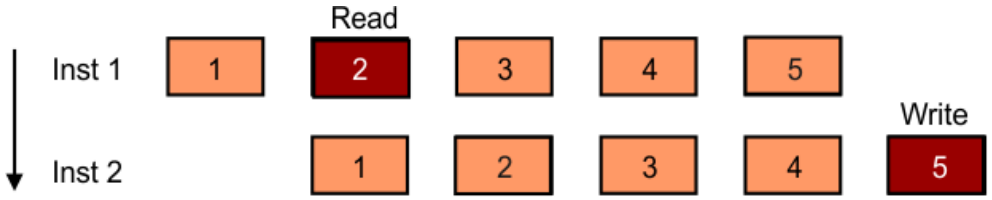
   I1 R3 = R1 + R2 [I1=instruction 1]

   I2 R5 = R3 + R4



2. Write After Read (WAR):- A WAR hazard occurs when an instruction needs to write to a register which is not still been read by a previously used instruction.

   -This problem does not occur generally in normal pipelining.

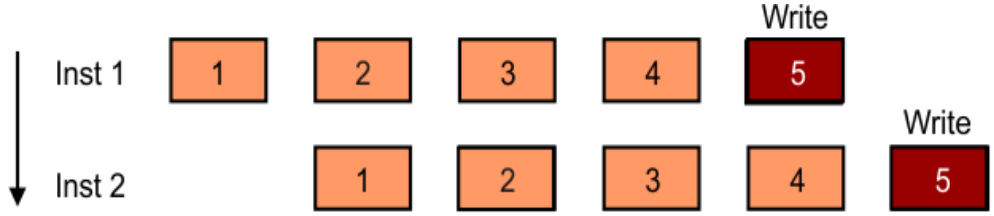


3. Write After Write (WAW):-A WAW hazard occurs when an instruction requires writing its result into the same register as a previously used instruction.

   -In the following example of WAW, both instructions write their results to R3.

   -eg: Inst I1 R3 = R1 * R2

   Inst I2 R3 = R4 + R5



- **Control Hazards:**-Control hazards are Occur Due to branches.
  -It can cause greater performance loss for pipeline.

***Instruction-level parallelism**:-Instruction-level parallelism (ILP) is a measure of how many operations in a system are simultaneously executable.

-Eg; s = a + b;
t = c + d;
u = s + t;

-Here, the first two lines can be executed entirely in parallel.
-There are no shared data dependencies between the first line and the second line.
-Thus, a processor may be able to execute the instructions in parallel.
-The third line is depends on the first two.
-It cannot be executed until the first two complete, as the stores into s and t are needed.
-If each line took one clock cycle and we were able to execute the first two lines in parallel, we might say we had an ILP ratio of 3/2.

***Superscalar:**- In a superscalar computer, the central processing unit (CPU) manages multiple instruction pipelines to execute several instructions concurrently during a clock cycle.
-Superscalar processors issue more than one instruction per clock cycle.
Or
-A scalar processor executes single instruction for each clock cycle; a superscalar processor can execute more than one instruction during a clock cycle.
-Superscalar was designed to improve the performance of these operations by executing them concurrently in multiple pipelines.
-If the input to one instruction depends on the output of a preceding instruction, then the latter instruction cannot complete execution at the same time or before the former instruction.

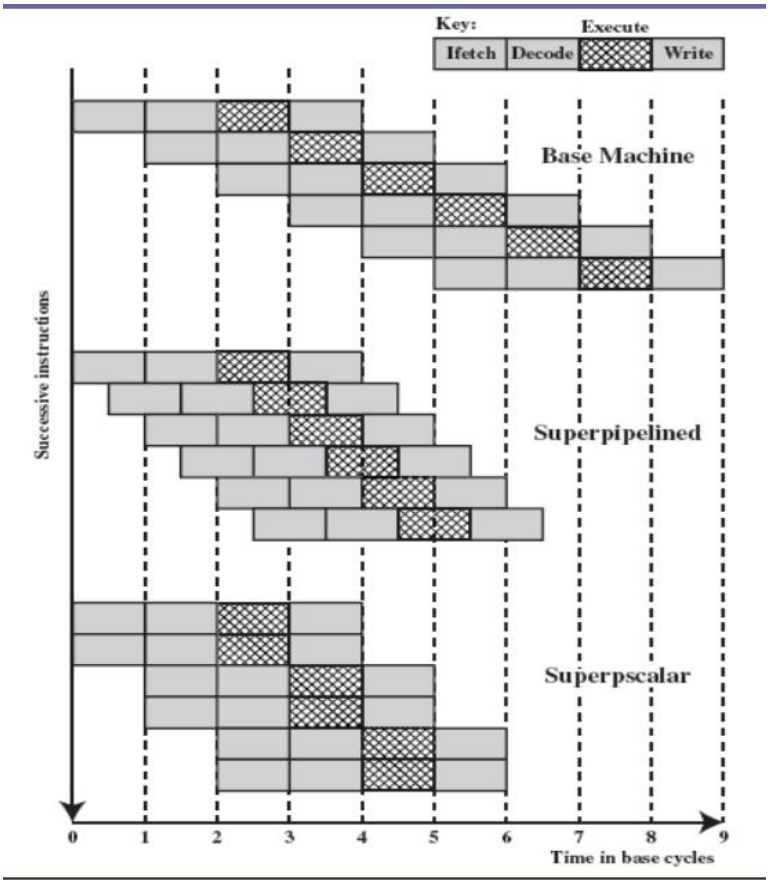
***Super pipelined:**-Super-pipeline is an alternative approach to achieve greater performance.
-Many pipeline stages need less than half a clock cycle.
-Super-pipelining is the breaking of stages of a given pipeline into smaller stages (thus making the pipeline deeper) in an attempt to shorten the clock period and thus enhancing the instruction throughput by keeping more and more instructions in flight at a time.
->Benefits:
-The major benefit of super-pipelining is the increase in the number of instructions which can be in the pipeline at one time and hence the level of parallelism.
->Drawbacks:
-The larger number of instructions "in flight" (i.e., in some part of the pipeline) at any time, increases the potential for data dependencies to introduce stalls.

→ **Superscalar v/s Superpipelined**

-Simple pipeline system performs only one pipeline stage per clock cycle

-Super-pipeline system is capable of performing two pipeline stages per clock cycle

-Superscalar performs only one pipeline stage per clock cycle in each parallel Pipeline.



*Multicore Systems:-A single computing component with multiple cores (independent processing units) is known as a multicore processor.

- It denotes the presence of a single CPU with several cores in the system.

-Individually, these cores may read and run computer instructions.

-They work in such a way that the computer system appears to have several processors, although they are cores, not processors.

-These cores may execute normal processors instructions, including add, move data, and branch.

-Multicore processors are used in various applications, including general-purpose, embedded, network, and graphics processing (GPU).

- It decreases the amount of heat generated by the CPU while enhancing the speed with which instructions are executed.

-The software techniques used to implement the cores in a multicore system are responsible for the system's performance.

->Advantages

- Multicore processors may execute more data than single-core processors.
- It will have less traffic.

- These systems are energy efficient because they provide increased performance while using less energy.

->Disadvantages

- These are very difficult to manage than single-core processors.
- These systems use huge electricity.
- Multicore systems become hot while doing the work.
- These are much expensive than single-core processors.

→difference between multicore and multiprocessor

MultiCore	MultiProcessor
A single CPU or processor with two or more independent processing units called cores that are capable of reading and executing program instructions.	A system with two or more CPU's that allows simultaneous processing of programs.
It executes single program faster.	It executes multiple programs Faster.
Not as reliable as multiprocessor.	More reliable since failure in one CPU will not affect other.
It has less traffic.	It has more traffic.
It does not need to be configured.	It needs little complex configuration.
It's very cheaper (single CPU that does not require multiple CPU support system).	It is Expensive (Multiple separate CPU's that require system that supports multiple processors) as compared to MultiCore.

Join for more MCA short note : https://t.me/mgu_mca_shortnote

@MGU_MCA_SHORTNOTE