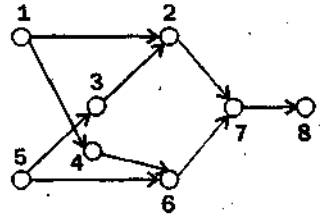
UNIT – I

INTRODUCTION TO DESIGN METHODOLOGY

Q.1. What is a system and how it is represented what are the main properties of any system.

Ans. System is defined informally as a collection often a large and a complex one of objects (components) that are convected to form a coherent entity with a specific function or purpose e.g. computer.



—System is generally modelled by a graph. The below graph consists of set of objects $V = \{V_1, V_2, V_3, V_n\}$ called nodes vertices and a set of edges E whose members are pairs of notes taken from the set $((V_1, V_2), (V_1, V_3), ..., (V_{n-1}, V_n))$ of all such pairs. The node $e = (V_i, V_j)$ joins or connects node V_i to node V_j.

The central properties of any system:

(*i*) Structure: It is the abstract graph consisting of its block diagram with nu functional information. The structural description names components and defines their interconnection.

(*ii*) Behaviour: It enables out to determine for any given input signal to the system, the corresponding output. e.g.

i/p	*(a)*	*o/p*
x_1	x_2	$f(a)$
0	0	0
0	1	1
1	0	1
1	1	0

← Behavioural description of X-OR gate

Q.2. What is HDL? What are main parts of HDL description. List advantages and disadvantages of HDL by giving an example.

Ans. HDL is hardware description language, a format that resembles a high level programming language. e.g. VHD< Verilog etc.

The main parts of HDL taking VHDL as e.g. is

(*i*) Entity: It is a formal statement of the system's structure at the highest level i.e. as a single component. It describes system's interface but says nothing about system's behaviour or its internal structure.

(*ii*) Architecture: It specifies behaviour and/or internal structure.

Advantages:

(*i*) Provides precise, technology independent descriptions of digital circuits of various levels of abstraction.

(*ii*) Used for documentation purposes.

(*iii*) Suitable for use with CAD.

Disadvantages:

(*i*) Long and very descriptive.

(*ii*) They lack interective appeal and rapid insights that circuit diagrams and less formal descriptive methods.

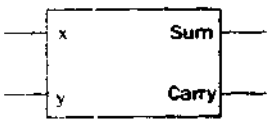
Q.3. Give the following VHDL description of half adder with block diagram.

(*i*) Behavioural description

(*ii*) Structure description

Ans. (*i*) Behavioural description: Entity half_adder is part (*x*, *y* : in bit; sum, carry : one bit); end half_adder;

www.btechsamplepapers.in



I/P		O/P	
x	y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

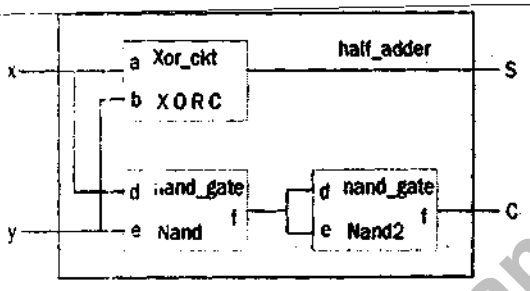
architecture behaviour of half_adder is begin

Sum <= x Xor y;

Carry <= x and y;

end behaviour;

(ii) **Structural description:** Entity half_adder is port (x, y : in bit; sum, carry : out bit) end half_adder;



Architecture structure of half_adder is component XOR_circuit port (a, b : in bit; c : out bit), end component, component nand_gate (d, e : in bit; f : out bit); end component signal alpha : bit;

begin

XOR : Xor_circuit port map

(a⇒x, b⇒y, c⇒sum);

NAND : nand_gate port map

(d⇒x, e⇒y, f⇒alpha);

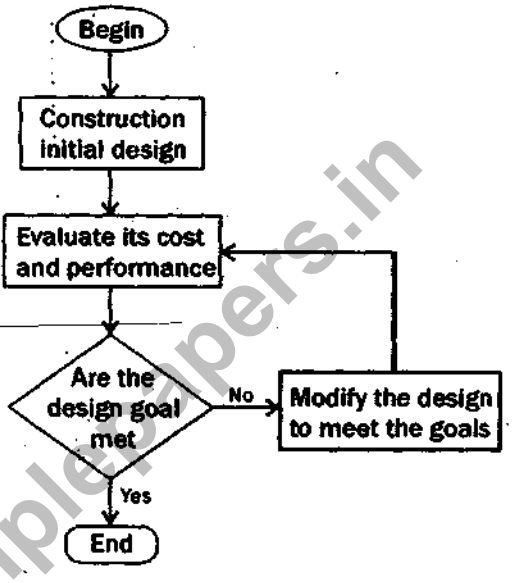
NAND2 : nand_gate port map

(d⇒alpha, e⇒alpha, f⇒carry)

end structure;

Q.4. With the help of flow chart, exaplain design process. How useful is CAD in overall designation.

Ans. An initial design is created, in adhoc fashion, by adopting an existing design of a similar system. The result is then evaluated to see if it meets the relevant design objectives. If noh, the design is revised and the results is reevaluated.



CAD: It is used to automate, and contribute in three important ways to design process:

***(i)* CAD editors or translators:** It convert design data into forms such as HDL descriptions or schematic diagrams.

***(ii)* Simulators:** It create computer models of a new design, which can mimie the designs behaviour and help designers to determine how well the design meets various performance and goals.

***(iii)* Synthesizers:** It automate the design process itself by deriving structures that implement all or part of some design step.

Q.5. Differentiate the computer design levels under *(i)* components, *(ii)* IC density *(iii)* Info. units *(iv)* Time units.

www.btechsamplepapers.in

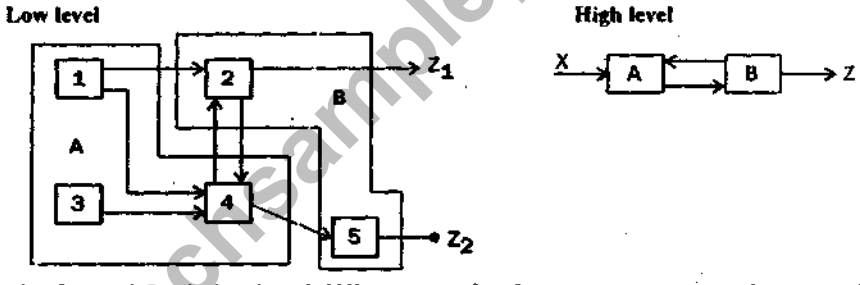
Ans.

Levels	Components	IC-density	Info. units	Time Units
(*i*) Gate (logic level)	Logic gates, f=f	SSI	Bits	10^{-12} to 10^{-1} s
(*ii*) Register (register transfer level)	Registers, Counters combinational and small sequential circuit	MSI	Words	10^{-9} to 10^{-6} s
(*iii*) Processor (architecture /system)	CPU's memories, I/O devices	VLSI	Blocks of words	10^{-3} to 10^{-3} s

Q.6. Explain the hierarchical system with the half of block diagram explain low high design levels.

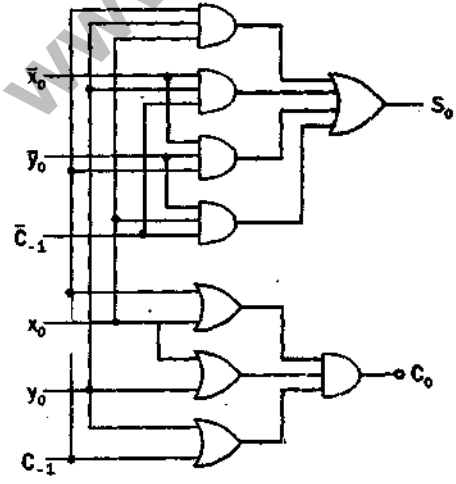
Ans. When there is one to one mapping 'h_i' between components in L_i and disjoint subsystems in level L_{i-1}, a system is called hierarchical system.

The system is said to high or low depending on the complexity of the components.

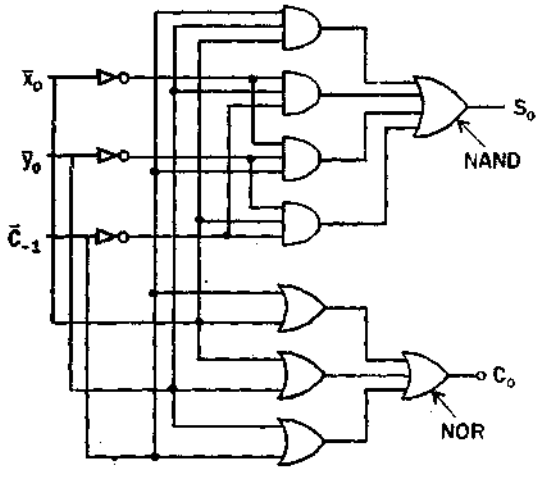


Q.7. What is Gate level design? What are design components in gate level? Give examples of two level circuit design and their disadvantages.

Ans. It is concerned with processing binary variables whose possible values are restricted to the bits. The design components are logic gates, flip-flops. The example of two level circuit design is



Full adder



Full adder

www.btechsamplepapers.in

Combinational function are used where mapping from the sert of 2^n input combinations of n-binary variables onto the output values and 1 is done. The function shown is a pair of 3 var.ables. So (x_0, y_0, C_{-1}) and $C_0(x_0, y_0, C_{-1})$ which are sum and carry of points of a full adder.

$$S_0 = x_0y_0C_{-1} + \bar{x}_0y_0\bar{C}_{-1} + \bar{x}_0\bar{y}_0C_{-1} + x_0\bar{y}_0\bar{C}_{-1}$$

$$C_0 = (x_0 + C_{-1})(x_0 + y_0)(y_0 + C_{-1})$$

The figure is called two level because there are only two gates, AND/OR along each path from their adder's external/primary inputs x_0, y_0, C_{-1} to its primary outputs S_0, C_0.

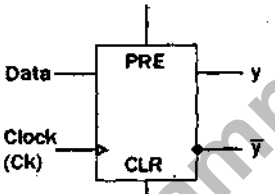
The number of logic levels is defined by the number of gates along the circuit's longest I_0 path.

Disadvantage of Two Level: Computationally complex, so they ae practical for de-signing smal circuits.

Q.8. What are flip-flops. Explain the behaviour of most common type of flip-flop giving state table and timing diagram.

Ans. Flip-flop are 1 bit storage rely on an external clock signal circuit to synchronize the times at which they respond to changes on their input data lines.

The most common type is edge triggered D_{ff}.

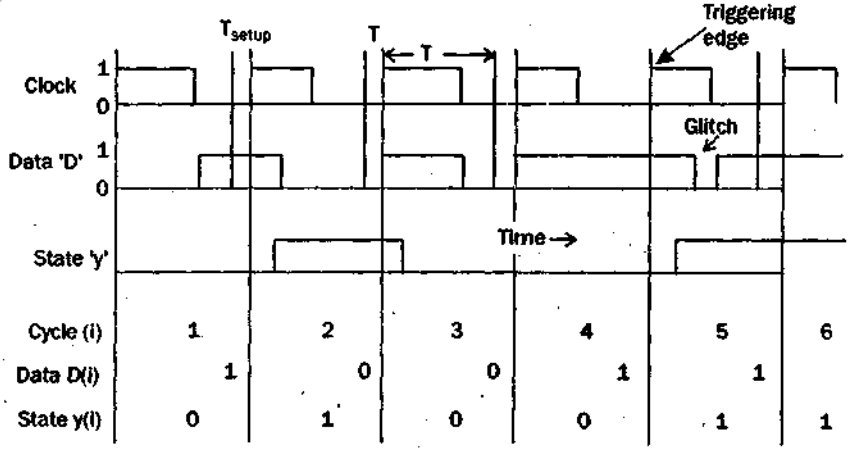


	I/P D(i) 0	1	
State 0	0	1	Next state
y(i) 1	0	1	y(i + 1)

The Δff reads in the data value on its D line when the 0 to 1 triggering edge of click signal circuit arrives; this D value becomes the new value of y. The input data line 'D' can be varied independetly and can go through several changes in any clock cycle 'I'. The data value $D(I)$ present just before the arrival of the triggering edge of circuit determines the next state $y(I + 1)$. To charge the flip-flop's state the D signal must be held steady for a min. periods known as set up time. $T_{set\ up}$ before $f - f$ is triggered.

In the beginning the flip-flop ($f - f$) is intialized (reset) which is independent of circuit. There are two control inputs PRESET (pre) to force $y \rightarrow 1$ and CLEAR (CLR) to force $y \rightarrow 0$.

Characteristic eg. of D_{ff} $y(I + 1) = D(I)$



www.freesamplepapers.in

For clock cycle, $C(i) = 1$, $D(1) = 1$, $y(1) = 0$

For clock cycle, $C = 2$, $y(1) = 1$, in reponse to $D(1) = 1$.

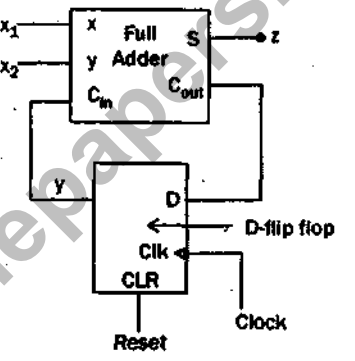
Since D stands for delay so 'y' charges in next circuit in response to previous data D.

Q.9. Explain sequential circuit with help of example.

Ans. It consists of combinational circuit and a set of flip flops. The combinational logic forms the computational or data processing part of circuit. The flip-flop stores info on the circuits's past behaviour; this defines the circuit's internal state Y. If primary inputs are 'X' and primary outputs are 'Z', then 'Z' is function fo both X and Y. Sequential circuit is provided with clock signal that determines the times at which the $f - f$ change state, the resulting circuit is clocked or synchronous. The behaviour can be specified by a state table that includes the possible values of the primary outputs and its internal states.

e.g. Serial Adder - $Z = X_1$ plus X_2.

	I/Rs	x_1	x_2	
	0 0	0 1	1 0	1 1
Present state				
S_0 (y=0)	S_0, 0	S_0, 0	S_0, 0	S_1, 0
S_1 (y=1)	S_0, 1	S_1, 0	S_1, 0	S_1, 1



The numbers are added serially, i.e. bit by bit and the result is also produced serially. In the clock cycle the serial adder has 2 input bits $x_1(i)$ and $x_2(i)$ and computes 1 bit $Z(i)$ of Z. It also computes a carry signal $C(i)$. Thus the output is

$$C(i)Z(i) = x_1(i) \text{ plus } x_2(i) \text{ plus } C(i-1)$$

There are only two internal states, its memory consists of single flip-flop stored in a state various 'y' we asign $y = 0$ for S_0 and $y = 1$ for S_1, this equates with stored carry signal $C(i - 1)$, we can use edge triggered D_{ff} to store 'y', character equation of f–f is $y(i + 1) = D(i)$. Thus 'C' can be implemented directly by a full adder circuit whose sum output is 'Z' and carry output is 'D'.

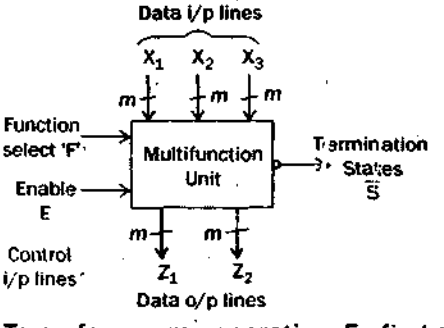
Q.10. What are the different register level components and their functions. Explain generic block representation of register level components.

Ans. Register level circuits are composed of word oriented devices e.g. registers, shift register, counters.

Type	*Component*	*Function*
Combinational	Word gates	Logical operations
	Word gates on use.	Data routing
	Decodes and encodes adders	Code checking and conversion
	Adders	Addition, subtraction
	ALU	Numerical and logical operation
	PLD	Gen. comb. function
Sequential	Parallel registers	Info storage
	Shifts registers	
	Counters	Control/timing sig. gen.
	PLD	Gen. sequential function.

www.btechsamplepapers.in

Generic rep. of Reg. level comp.: The input control lines associated with a multifunction block fall into enable lines, which specify the time or condition for a selected operation to be performed and select lines which specify one of several possible operations that unit is to perform.



To perform some operation F_b, first set the select line 'F' to a bit pattern denoting F_1 and then activate the edge triggered enable line 'E'.

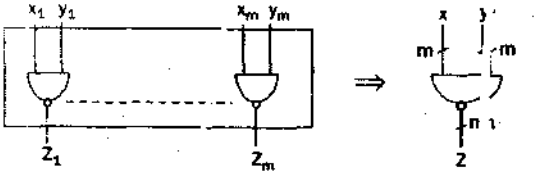
The components I_0 lines are separated into data and control lines. M-bit identifies the bus's role, the type of data transmitted over a data bus.

Q.11. Explain reference to register level:

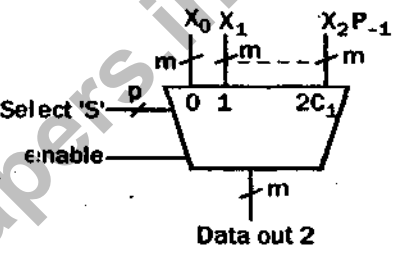
(i) Word gates (iii) Mux (iii) Decoders (iv) Encoders.

Ans. Word gates: Let $X = (x_1, x_2, x_m)$, $Y = (y_1, y_2, \ldots y_m)$ be two m-bit binary words. The gate operations are performed on X and Y bitwise to obtain m-bit word $Z = (Z_1, Z_2 \ldots Z_m)$. This is called wordgate operatioin e.g. If Nand operation, $Z = \overline{XY}$.

$\therefore Z(Z_1, Z_2, \ldots Z_m) = (\overline{x_1 y_1}, \overline{x_2 y_2} \ldots \overline{x_m y_m})$



MUX: It is a device intended to route data from one of several sources to a common destination, the source is specified by applying appropriate control signals m-bit datalines, $X_i = (x_{i,0}, x_{i,1}, \ldots x_{i,m-1})$. When $a_i = 1$, then select bus 'S' select the line with subscripts 'i'. The binary variable a_i denotes the selectioln of input of data bus $X_i - a_i$. The data word on X_i is transferred to 'Z'.



$Z_j = (x_{0,j} a_0 + x_{1,j} a_1 + \ldots + x_2 b_{-1,j} a_2 b_{-1})e$

or for single word based equation

$Z = (X_0 a_0 + X_1 a_1 + \ldots + X_2 b_{-1} a_2 b_{-1})e$

Decoders: A 1 out 2^n or ½n decoder is a combinational circuit with 'n' input lines X and 2^n output lines 'Z' such that each of the 2^n possible input combinations A_i applied to X activates a corresponding output line Z_i.

It is used in address decoding, routing data from common source to one of several destinations. It is similar to demense, where control input enable is viewed as 1 bit data source to be routed to one of 2^n destinations.

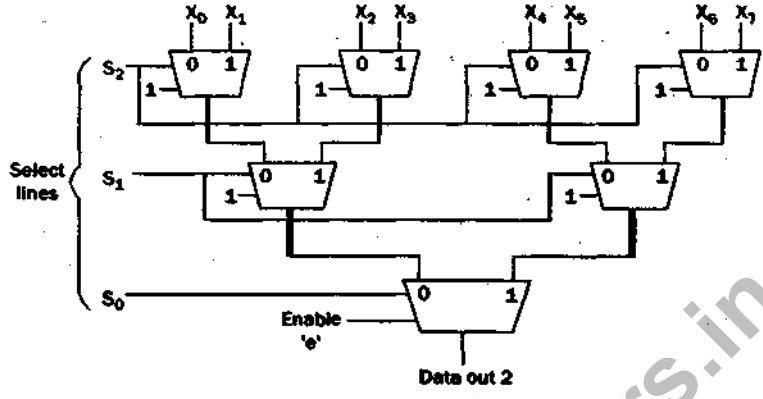
Encoders: It generates the address or index of an active input line. It has 2^K input data lines and 'K' output data lines. Priority encoder, fix the priority to each of input such that x_i has highest priority than x_j if $i > j$.

Q.12. (i) construct eight input mux from two input mux.

(ii) Contruct MUX based full adder.

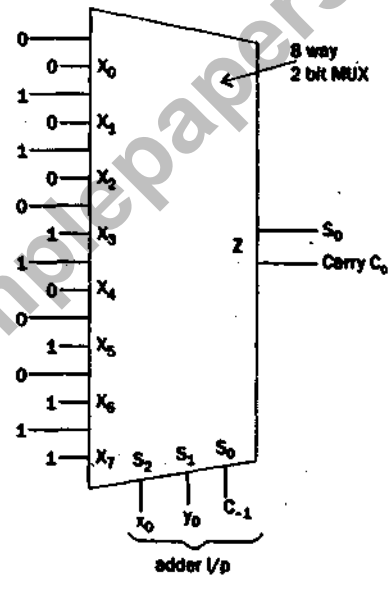
Ans. (i) Construct eight input MUX from two input MUX.

www.btechsamplepapers.in



(ii)

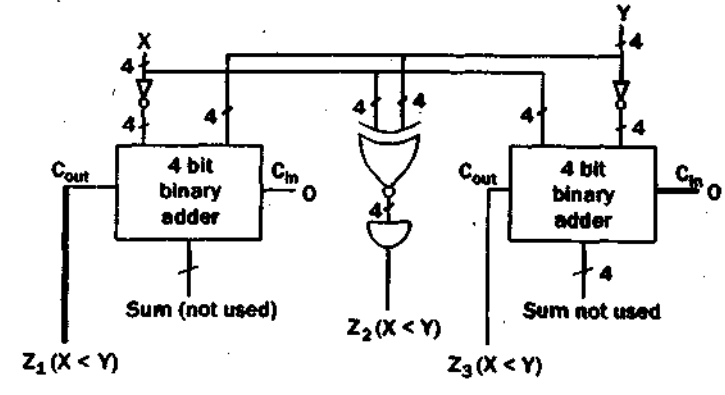
I/P			O/P	
x_o	y_o	C_{-1}	S_o	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	1	1	0
1	1	0	1	0
1	1	0	0	1
1	1	1	0	1
			1	1



Q.13. Name useful arithmetic component at register level. Design 4 bit magnitude comparater at register level.

Ans. Adders, subtractors, magnitude comparator are the useful components.

Magnitude comparator:

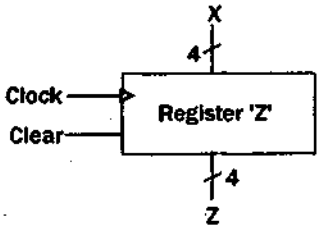


www.btechsamplepapers.in

Q.14. Explain main sequential components in reference to register level:

(a) Registers (π) counters (m) Buses.

Ans. Registers: A m-bit register is an ordered set of m flip-flops designed to store an m-bit word ($Z_0, Z_1, \ldots Z_{m-1}$). Each bit of the word is stored in a separate flip-flop, but have common control lines. Registers 'Z' reads in the data word 'X' each time it is clocked. To maintain the contents a state of 'Z' at a constant value, it is necessary to apply that value, continuously to Z's input bus. For load in a new value of 'X' into 'Z' in a particular clock cycle and subsequentally change 'X' without changing 'Z', control line is used, which cause the register to read the current value of 'X' when it is clocked and load has been set to 1.



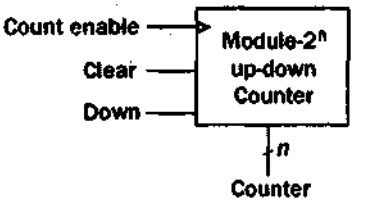
Registers are used so that external data can be transferred to or from all its flip-flops.

Counters: It is a sequential circuit designed to cycle through a predetermined sequence of 'K' distinct states $S_0, S_1, \ldots S_{K-1}$ in response to signals on an input line.

State transition

$S_{i+1} = S_i$ plus

Each 1 input increments the state by one, the circuit can be viewed as counting the input 1s.



Counter can store the state of control unit, can generate timing signals and introduce delays.

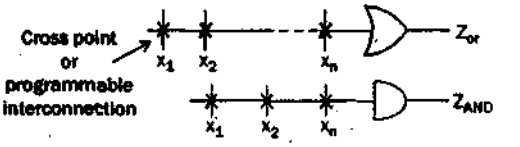
Buses: It is a set of lines designed to transfer all bits of a word from a specified source to a specified destination on the same or a different IC. A bus can be undirectional/bidirectional. Buses are often shared to reduce costs which also reduces the number of connecting lines.

Q.15. Compare PLD's and PLA.

Ans. PLD's: It is applied to IC's containing many gates whose interconnectioins can be programmed to implement any desired function (combinational/sequential):

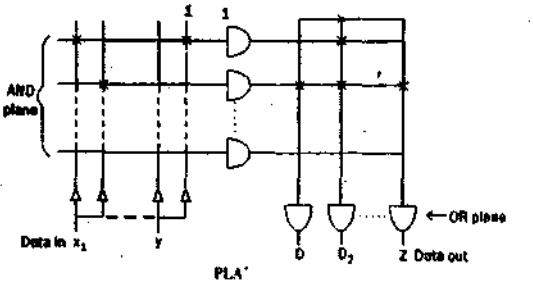
- These are easy to design and inexpensive.
- They constitute a key tech. for building ASIC's.
- To program PLD's mask programming and field programming is used.
- The convections are transitor switches which are programmed by switching them on/off.

Both AND and OR plane are programmable.



PLA: It is intended to realize a set of combinational logic function minimal SOP form. It consists of an array of AND gates, relying on pait terms and if a set of OR gates to form logical sum of poit terms. The OR plane has fixed connections while AND connections are programmable.

It is easy to use, high speed as output fan-out is that



PLA

www.btechsamplepapers.in

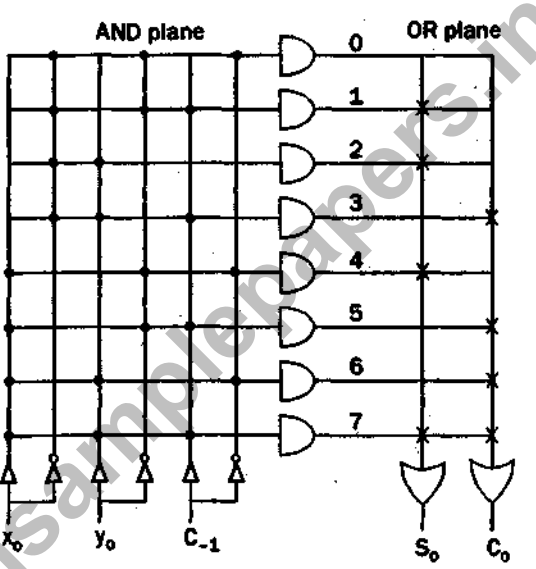
Q.16. Explain ROM using ROM implement full adder.

Ans. ROM generates all 2^n possible *n*-variable polt terms in its AND plane. This enables each output column of the OR plane to realize any desired function of '*n*' or fewer variables in sum of minterms.

It is a memory device only, it OR plane stores the 2^n data words that have been programmed into it. A stored word is read put each time the ROM receives a new input combination or address. The process of reading the stored info from a ROM is called table look up.

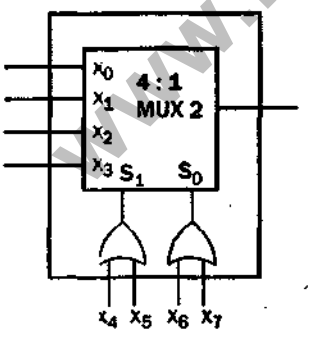
Adder:

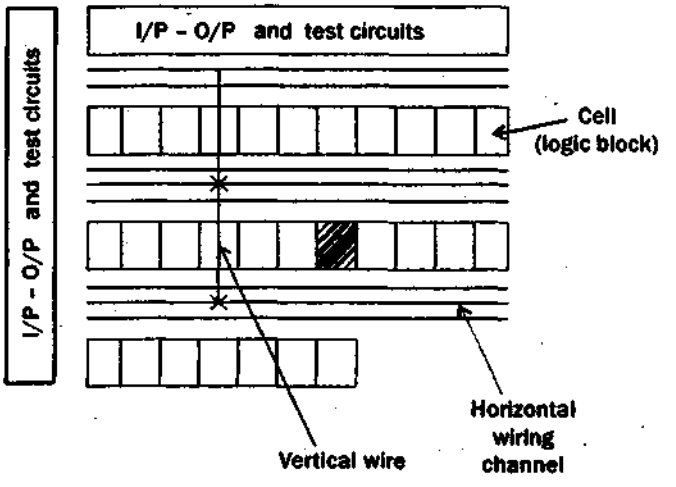
Adder			O/P	
x_0	y_0	C_{-1}	S_0	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Q.17. Explain basic cell and chip architecture of FPGA. What are steps of mapping a new design on FPGA chips.

Ans.





FPGA: It is 2-D array of general purpose logic circuits called cells or logic blocks, whose function are programmable, the cells are linked to one another by programmable

www.btechsamplepapers.in

buses. The cell types are small multi function circuits capable of realizing all Boolean function of a few variables, a cell may also contain one or two flip-flops.

FPGA can store the program that determines the circuit to be implemented in RAM or PROM on the FPGA chip. The pattern of the data in this configuration memory (CM) determines the cells functiona and their interconnection wiring. Each bit of CM control a transistor switch in the target circuit that can select some cell function or make some connections.

Steps to translate new design on FPGA chips:

Step 1. First translate or compile the design specifications on a schematic diagram or an HDL description.

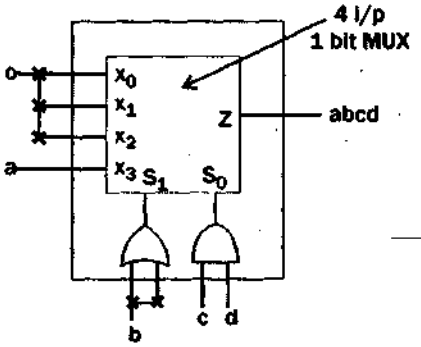
Step 2. Specialized place and route CAD software to assign the logic elements to cells, to determine the switch settings needed to set each cell's function and to establish the internal connections.

Step 3. The design is physically transferred to one or more copies of FPG. A chip is an appropriate programming unit.

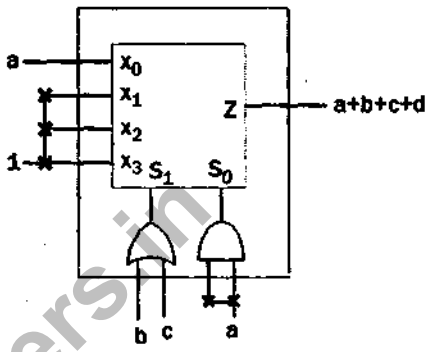
Q.18. How the FPGA is programmed to implement the following:

(*i*) 4 input AND gate
(*ii*) 4 input OR gate
(*iii*) Inverter
(*iv*) Δ-flip flop.

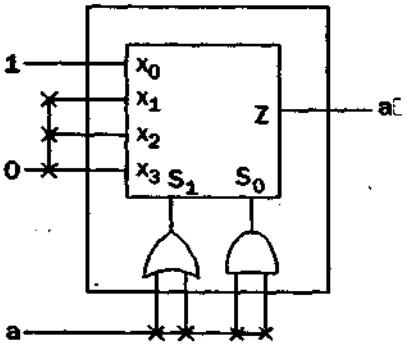
Ans. (*i*) 4 input AND gate:



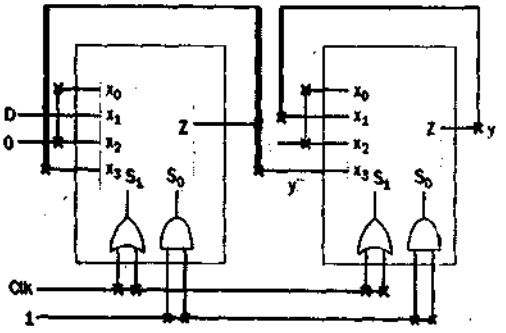
(*ii*) 4 input OR gate:



(*iii*) Inverter:



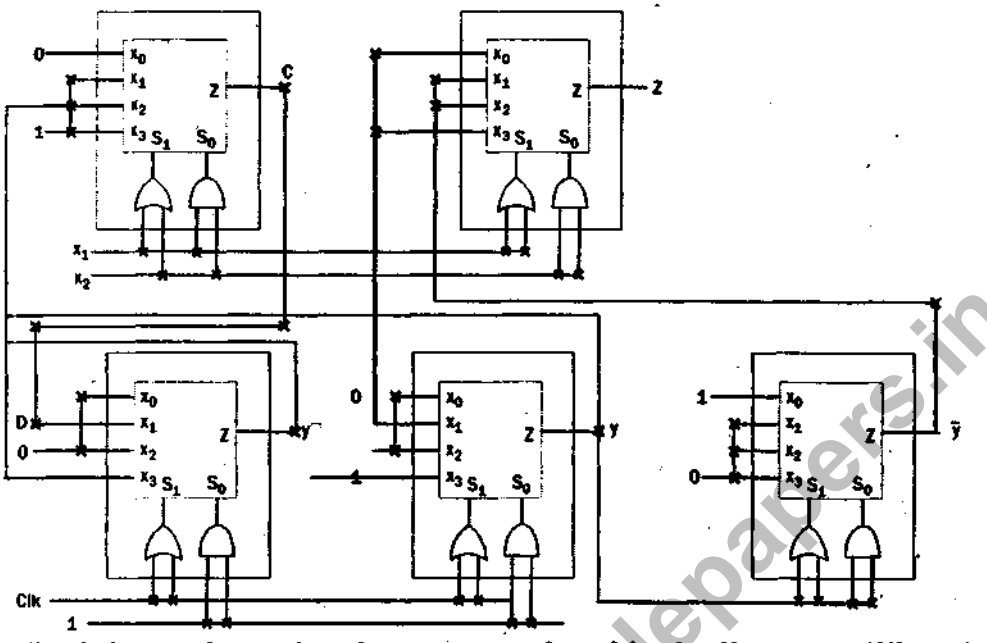
(*iv*) Δ-flip flop:



Q.19. Implement serial adder from FPGA cell.

www.btechsamplepapers.in

ns.



20. Explain register level system using block diagram. What is ultifunction system? Give example show by its block diagram.

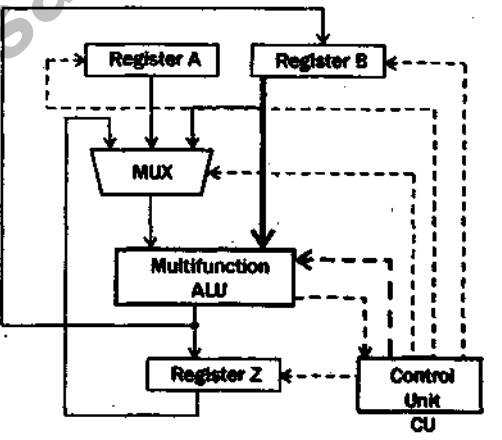
Ans. A register level system consists of a set of registers linked by combinational data trans- and data processing cicuits. Each operation is typically implemented by one or more elemen- y register transfer steps of form.

Cond: $Z := f(X_1, X_2, \ldots X_k)$

$X_1, X_2, \ldots X_k$ & $Z \rightarrow$ data words.

Cond. $\rightarrow$ control condition.

The statemnt above is when cond. holds, mpute the function f on $X_1, X_2, \ldots X_k$ and ign resulting value to 'Z'.



Multifunction System: It performs eral different operations. Such system is titioned into data processing part called apath and a controlling part, control unit, ponsible for selecting and controlling the ions of the datapath. e.g. $Z := A - B$.

To execute the controller CU must send select signals to ALU to select its subtract func- , it must send select signals to the mux that connects reg. A to ALU's left path and it must d a "load data" control signal to output register - Z.

Eg. of multifunction system is computer's CPU. Its control unit is responsible for the rpretation of instructions called program control or I-unit.

21. With the help of HDL describe the behaviour of 8 bit binary ltiplier at register level.

Ans. Multiplier 8 (in : INBUS; out : OUT BUS); register A[0 : 7], M[0 : 7], Q[0 : 7], COUNT[0 : 2]; but INBUS [0 : 7], OUTBUS[0 : 7];

www.btechsamplepapers.in

```
BEGIN:      A:=0, COUNT:=0, M:=INBUS; Q:=INBUS;
ADD:        A[0:7] : = A[1:7] + M[1:7] × Q[7];
SHIFT:      A[0] := 0, K[1:7] Q := A.Q[0:6];
TEST:       COUNT := COUNT + 1;
            if COUNT ≠ 7 then go to ADD,
FINISH:     A[o] := M[o] XOr Q[7], Q[7] := 0;
OUTPUT:     OUTBUT := Q;
            OUTBUS := A;
end multiplier 8;
```

Q.22. Explain the general approach to the design problem for register level systems.

Ans. (a) Define the desired behaviour by a set of sequences of register transfer operations such that each operation can be implemented directly using the available design components. This constitutes an algorithm AL to be executed. This involves a creative design process analogous to writing a computer program.

(*b*) Analyze AL to determine the types of components and the number of each type required for the datapath DP: the main components are registers, buses and combinational circuits.

(*c*) Contruct a block diagram for DP using the components identified in step 2. Make the connections between the components so that all data paths implied by AL are present and the given performance cost constraints are met: it requires defining an interconnection structure that links the components needed by the various parts of AL>

(*d*) Analyze AL and DP to identify the control signals needed. Introduce into DP the logic or control points necessary to apply these signals.

(*e*) Design a control unit CU or DP that meets all the requirements of AL>

The steps "d & e" i.e. specification and design of the control unit are relatively independent process.

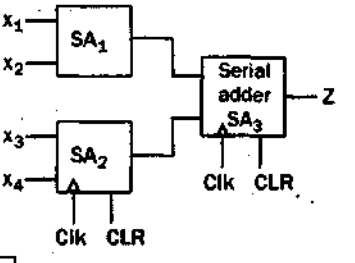
(*f*) Verify, typically by computer simulation, that the final design operates correctly and meets all performance cost goals: Design verificatioin plays a crucial role in the development proces. Simulation via CAD tools is used to identify and correct functional errors before the new design is committed to hardware.

Q.23. Design a pipelined 4 bit stream serial adder at register level.

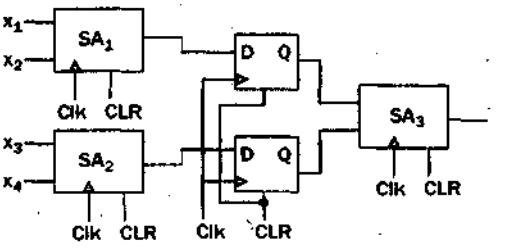
Ans. (*i*) Basic Design, (*ii*) Buffered:

Two stage pipeline design:

Ans. (*i*) Basic Design:

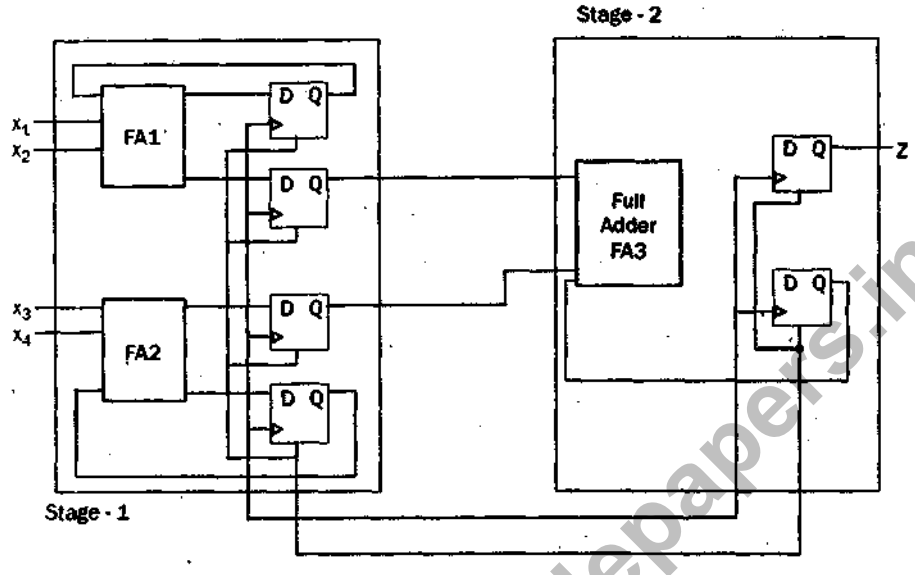


(*ii*) Buffered:



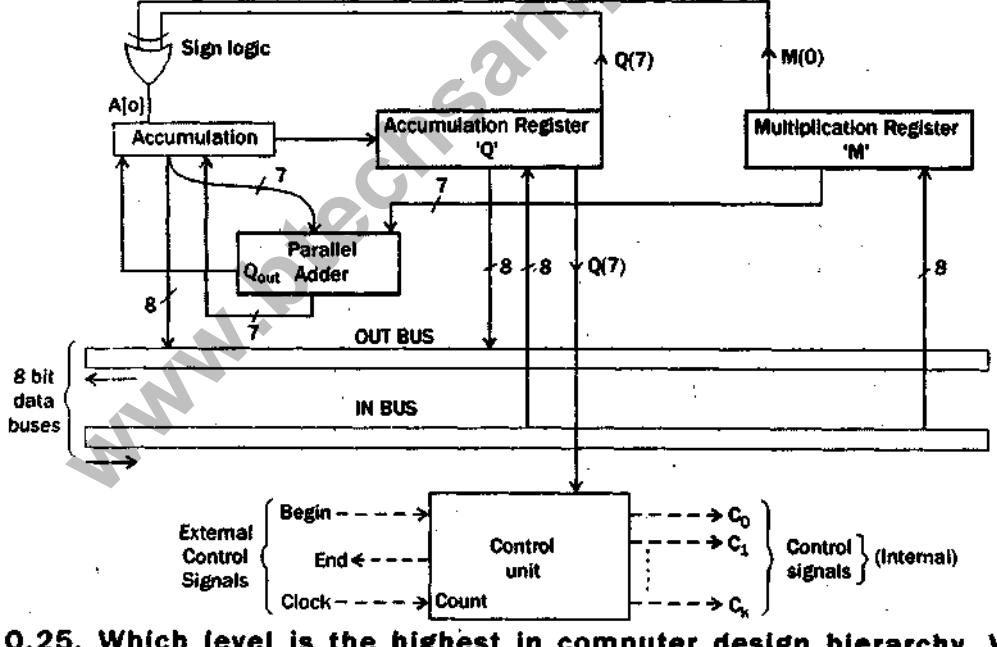
www.btechsamplepapers.in

Two stage pipeline design:



Q.24. Draw the block diagram of 8 bit binary multiplier.

Ans.

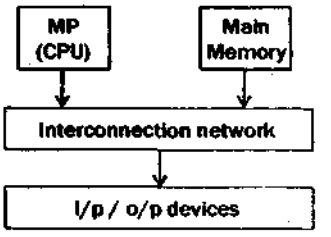


Q.25. Which level is the highest in computer design hierarchy. What are the main components, explain using internal organization block diagram of a CPU and cache memory.

Ans. The processor or system level is the toughest level in computer design hierarchy. It is concerned with the storage and processing of blocks of information such as programs and data files.

www.btechsamplepapers.in

Major Components:



CPU: It is general purpose instruction set processor that has overall responsibility for program interpretation and execution in computer system. These are microprocessors (μP).

Memories: It store programs and data required by the processors. The memories are:

(*i*) **Main memory:** Connected directly to and controlled by, the CPU. It is RAM.

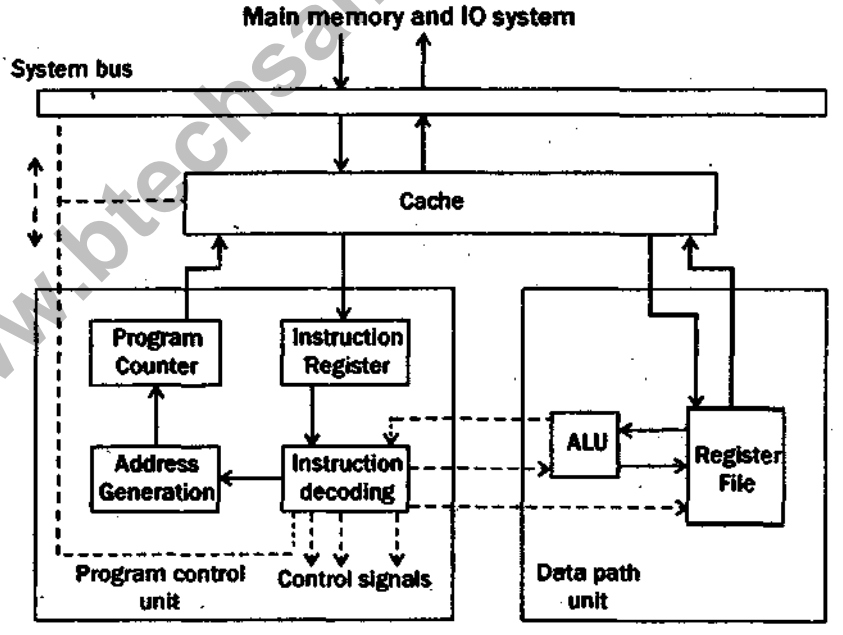
(*ii*) **Secondary memory:** High storage capacity slower than main memory connected indirectly to CPU.

(*iii*) **Cache memory:** It is positioned between CPU and memory. It reduces the average time taken by the CPU to access the memory system.

I/O devices: These devices are the means by which computer communicates with the outside world. It act as data transducers. i.e. to convert info from one physical representation to another. These do not alter the info content or its meaning. Their speed of operation is slow.

Interconnection network: It's function is to establish dynamic communication paths among the components via the buses under its control. It resolves the contention issue when several system component request use of the bus. The contention issue is resolved on priority basis.

Internal organisation of CPU and Cache:



CPU contains the logic needed to execute its particular instruction set and is divided into datapath and control units. The control part generates the addresses of instructions and data stored in external memory. Each memory request gen. by CPU is first directed to cache. The program control unit fetches instructions from cache or M and decodes them to derive the

www.btechsamplepapers.in

control signals needed for their execution. The CPU's datapath bias the arithmetic logic circuits that execute must inst. The CPU manages a system bus to communicate among CPU cache subsystem, main memory and I/O devices.

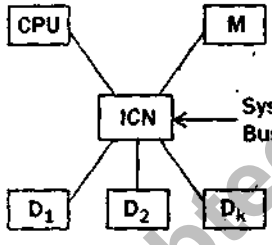
Q.26. What are the performances specifications of computer with the help of block diagram explain different computer structures.

Ans. The performance specificatioins take following forms:

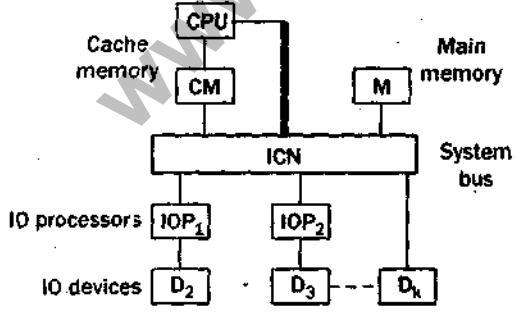
1. The computer should be capable of executing a instructions of type '*b*'/sec.
2. The computer should be able to support 'c' memory or IO devices of type '*d*'.
3. The computer should be compatabile with computers of type '*e*'.
4. The total cost of the system should not exceed '*f*'.

Prototyper Structures:

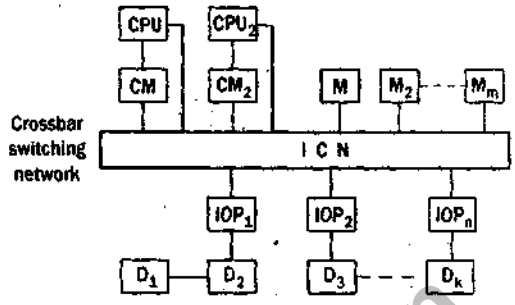
(*i*) Basic computer structure:



(*ii*) Computer with cache and IO processor:



(*iii*) Computer with multiple CPU and main memory banks:



Q.27. Explain the following:

(*i*) MIPS (*ii*) CPI (*iii*) Average time 't_E'

Ans. (*i*) MIPS: (Millions of instructions/sec): It is the average instruction executed speed.

(*ii*) CPI: Average number of CPU clock cycles required/instruction.

(*iii*) Average time '*T*': It is time in μs required to execute '*N*' instructions.

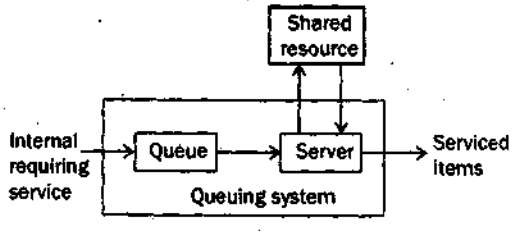
$$T = \frac{N \times CPI}{f} \qquad f \rightarrow \text{MHz}$$

Average time 't_E': It is average time to execute an instruction

$$t_E = \frac{T}{N} = \frac{CPI}{f} \ \mu s$$

Q.28. Explain M/M/1 model with the help of diagram. Derive the Little's equation.

Ans. M/M/1: It represents a server such as CPU or a computer with a set of tasks to be executed. The tasks are activated at random times and all queued in memory until they can be processed or serviced by the CPU on a first come first serve basis.



www.btechsamplepapers.in

The important parameters are the rate at which tasks requiring device arrive and tasks are serviced.

Little's equation: The average task 'X' passing through the system under steady state conditions encounter the same number of waiting tasks l_Q, when it enters the system.

The number left behind = λt_Q.

(It is number of tasks that enter the system at rate 'λ' during period t_Q)

$\lambda \rightarrow$ mean arrival rate

$t_Q \rightarrow$ mean waiting time.

$$\therefore \quad \lambda_Q = \lambda t_Q$$

$$\therefore \quad t_Q = \frac{l_Q}{\lambda} \quad \text{Little's equation.}$$

Q.29. With reference to M/M/1 explain the following:

(*i*) Mean utilization

(*ii*) Mean queue length

(*iv*) Mean waiting time.

Ans. (*i*) Mean utilization: The fraction of time server is busy, on average.

$$\text{Mean utilization } P = \frac{\text{Arrival rate}}{\text{Service rate}}$$

Inter arrival time distribution: ($p_i(t)$) It is the problem that at least one task arrives during a period of length 't'.

$$P_i(t) = 1 - e^{-\lambda t}$$

Mean Queue length: The average number of tasks queued in the system, including tasks waiting for service and those actually being served.

Mean Queue length $\rightarrow l_Q = P(1 - \rho)$

Mean Waiting Time: The average time that arriving tasks spend in the system, both waiting for service and being served (t_Q).

$$t_Q = l_Q/\lambda$$

$$t_Q = \frac{\rho}{(1-\rho)\lambda} = \frac{1}{(\mu - \lambda)} \qquad \left[\rho = \frac{\lambda}{\mu}\right]$$

Q.30. Consider a server computer that is processing jobs in a way that can be approx. by the M/H/1 model. Arriving jobs are queued in main memory until they are fully executed in one step by CPU. New jobs arrived an average rate of 10/min and computer is idle on average for 25% of time. Calculate.

(*i*) What is the average number of jobs '*N*' in main memory that are waiting to begin execution.

(*ii*) What is average time '*T*' that each job spend in the computer.

Ans. Accessing steady state conductions *T* is t_Q and *N* is l_w.

System is busy for 75% of time, $\rho = \frac{\lambda}{\mu} = 0.75$

Given $\lambda = 10$ jobs/min

$$\therefore \quad \mu = \frac{\lambda}{\rho} = \frac{10}{.75} = \frac{40}{3} \text{ jobs/min}$$

Now

(*i*) $$T = t_Q = \frac{1}{(\mu - \lambda)} = \frac{1}{\left(\frac{40}{3} - 10\right)}$$

$= 0.3$ min.

(*ii*) $N = l_Q = \lambda t_Q = 10 \times 0.3 = 3$

$$t_w = \frac{lw}{\lambda} \Rightarrow lw = \lambda t_w$$

$$t_w = t_Q - \frac{1}{\mu}$$

$$\therefore \quad l_w = 10\left[\frac{1}{(\mu - \lambda)} - \frac{1}{\mu}\right] = 10 \times [0.3 - 0.075]$$

$= 10 \times 0.225 = 2.25$ jobs.

www.btechsamplepapers.in