MODULE 3

SEQUENTIAL CIRCUITS

1. **Name the flip flop used for data storage.**
   SR Flip Flop,D Flip Flop,JK flip flop,T flip flop
2. **List two types of sequential circuit based on timing of signals.**
   Asynchronous Ripple counters, Synchronous counters
3. **What are flip flops?Give examples**
   SR Flip Flop,D Flip Flop,JK flip flop,T flip flop
4. **Define latch.**

Latches	Flip Flops
Latches are building blocks of sequential circuits and these can be built from logic gates	Flip flops are also building blocks oof sequential circuits. But, these can be built from the latches.
Latch continuously checks its inputs and changes its output correspondingly.	Flip flop continuously checks its inputs and changes its output correspondingly only at times determined by clocking signal
The latch is sensitive to the duration of the pulse and can send or receive the data when the switch is on	Flipflop is sensitive to a signal change. They can transfer data only at the single instant and data cannot be changed until next signal change. Flip flops are used as a register.
It is based on the enable function input	It works on the basis of clock pulses
It is a level triggered, it means that the output of the present state and input of the next state depends on the level that is binary input 1 or 0.	It is an edge triggered, it means that the output and the next state input changes when there is a change in clock pulse whether it may a +ve or -ve clock pulse.

5. **Compare Sequential and combinational circuit.**

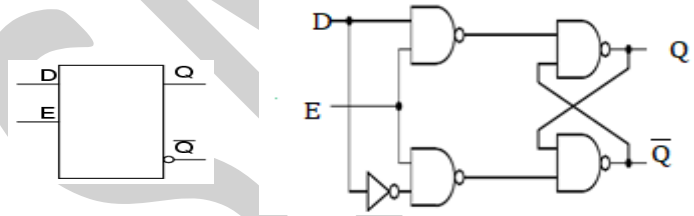
Difference between Combinational and Sequential circuit

S.No.	Combinational Circuit	Sequential Circuit
1.	It contains no memory elements.	It contains memory elements.
2.	The present value of it's outputs are determined solely by the present values of it's inputs.	The present value of it's outputs are determined by the present value of it's inputs and it's past state.
3.	It's behavior is described by the set of output functions.	It's behavior is described by the set of next-state(memory) functions and the set of output functions.

6. Differentiate synchronous and asynchronous sequential circuits.

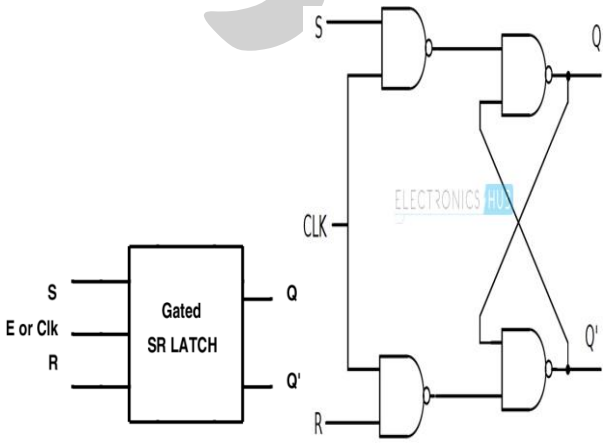
Asynchronous Ripple counters	Synchronous counters
The delay time of all flip flops are added. Therefore there is considerable propagation delay.	The clock pulses are applied to all flip flops simultaneously.Hence there is minimum propagation delay.
Frequency of operation is lesser than in a synchronous counter.	Frequency of operation can be much higher than that in ripple counter.
The maximum frequency depends on modulus.	The maximum frequency does not depends on modulus.
Circuit is simple.	Circuit is complex.
Minimum number of logic devices are needed.	The number of logic devices required is more than in ripple counter.
Less costly than synchronous counter	More costly than ripple counter

7. Construct a D latch using a gated SR latch.



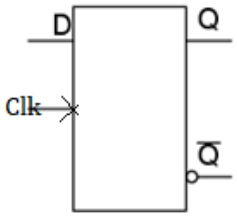
E	D	Q_n	Q_{n+1}	State
1	0	0	0	RESET
1	0	1	0	
1	1	0	1	SET
1	1	1	1	
0	X	0	0	No Change
0	X	1	1	

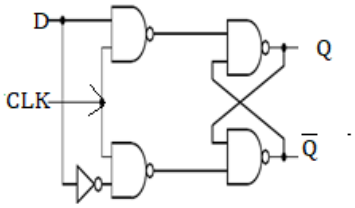
8. Design SR latch using NAND gate.



E	S	R	Q_n	Q_{n+1}	State
1	0	0	0	0	No Change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	x	Invalid or Indeterminate
1	1	1	1	x	

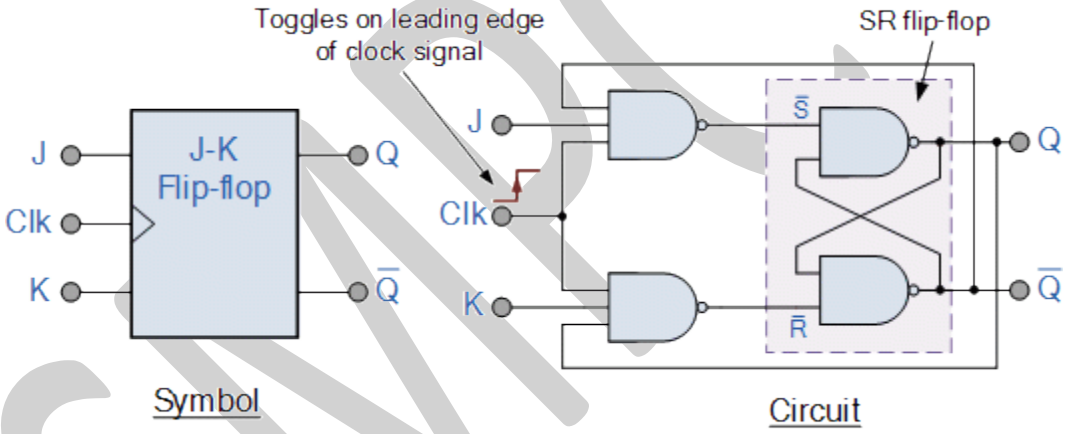
9. Write short note on D flip flop .Draw the logic symbol and truth table for a D flip flop. Or Explain the working of a D flip flop.





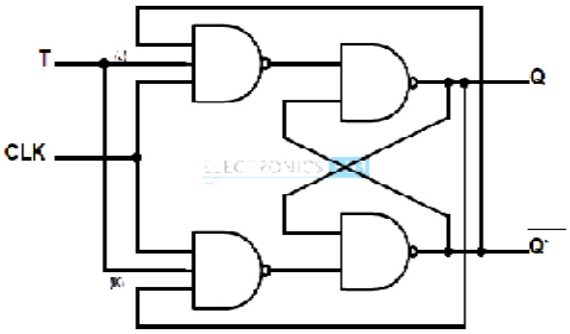
Clk	D	Q_n	Q_{n+1}	State
1	0	0	0	RESET
1	0	1	0	
1	1	0	1	SET
1	1	1	1	
0	X	0	0	No Change
0	X	1	1	

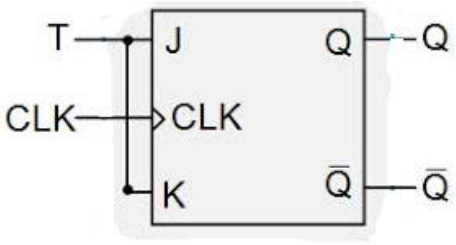
10. Demonstrate a JK flip flop with truth table.



Clk	J	K	Q_n	Q_{n+1}	State
1	0	0	0	0	No Change
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	1	Toggle
1	1	1	1	0	

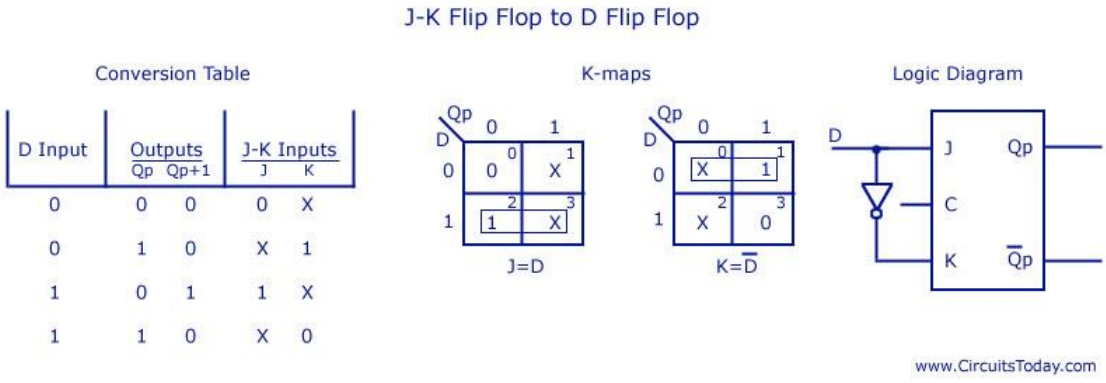
11. **Explain the working of a T flip flop with logic diagram and truth table.**





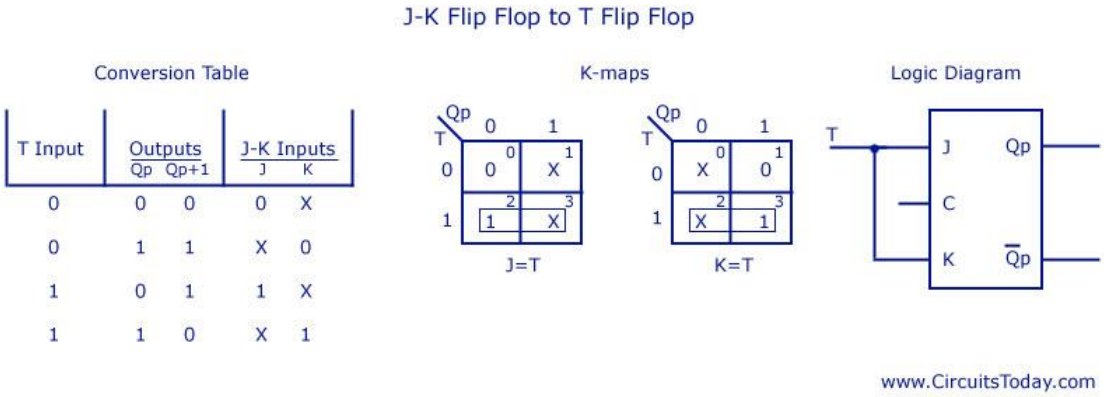
Clk	D	Q_n	Q_{n+1}	State
1	0	0	0	No Change
1	0	1	1	
1	1	0	1	Toggle
1	1	1	0	
0	X	0	0	No Change
0	X	1	1	

12. **Design JK flip flop using D flip flop and verify it using characteristics table and equation.**



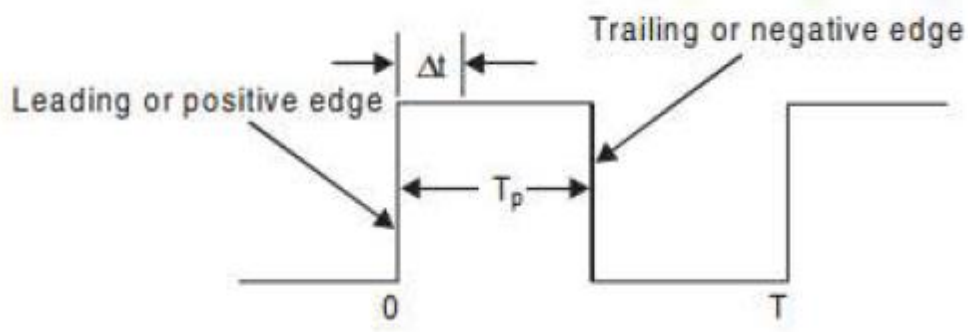
13. **Construct a T flip flop using a JK flip flop with truth table**

J and K are the actual inputs of the flip flop and T is taken as the external input for conversion. Four combinations are produced with T and Qp. J and K are expressed in terms of T and Qp.



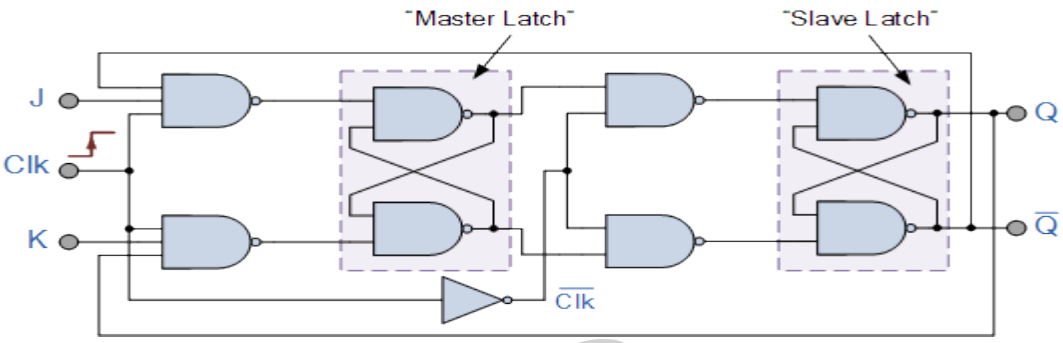
14. Using suitable example explain race condition .How can it be avoided.

Consider, for example, that the inputs are J = K = 1 and Q = 1, and a pulse as shown above is applied at the clock input. After a time interval Δt equal to the propagation delay through two NAND gates in series, the outputs will change to Q = 0. So now we have J = K = 1 and Q = 0. After another time interval of Δt the output will change back to Q = 1. Hence, we conclude that for the time duration of tp of the clock pulse, the output will oscillate between 0 and 1. Hence, at the end of the clock pulse, the value of the output is not certain. This situation is referred to as a race-around condition.



Generally, the propagation delay of TTL gates is of the order of nanoseconds. So if the clock pulse is of the order of microseconds, then the output will change thousands of times within the clock pulse. This race-around condition can be avoided if tp < Δt < T. Due to the small propagation delay of the ICs it may be difficult to satisfy the above condition. A more practical way to avoid the problem is to use the master-slave (M-S) configuration

15. Explain the working of JK master slave flip flop with logic diagram.



The input signals J and K are connected to the gated "master" SR flip flop which "locks" the input condition while the clock (Clk) input is "HIGH" at logic level "1".

As the clock input of the "slave" flip flop is the inverse (complement) of the "master" clock input, the "slave" SR flip flop does not toggle.

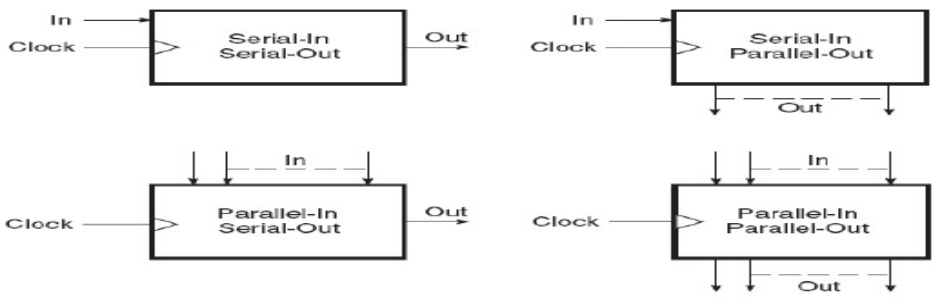
The outputs from the "master" flip flop are only "seen" by the gated "slave" flip flop when the clock input goes "LOW" to logic level "0".

When the clock is "LOW", the outputs from the "master" flip flop are latched and any additional changes to its inputs are ignored. The gated "slave" flip flop now responds to the state of its inputs passed over by the "master" section.

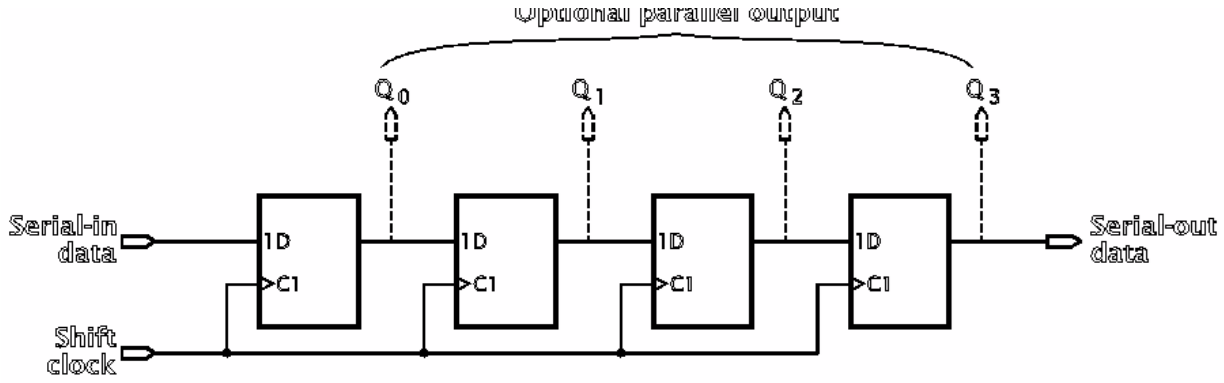
Then on the "Low-to-High" transition of the clock pulse the inputs of the "master" flip flop are fed through to the gated inputs of the "slave" flip flop and on the "High-to-Low" transition the same inputs are reflected on the output of the "slave" making this type of flip flop edge or pulse-triggered.

Then, the circuit accepts input data when the clock signal is "HIGH", and passes the data to the output on the falling-edge of the clock signal. In other words, the **Master-Slave JK Flip flop** is a "Synchronous" device as it only passes data with the timing of the clock signal.

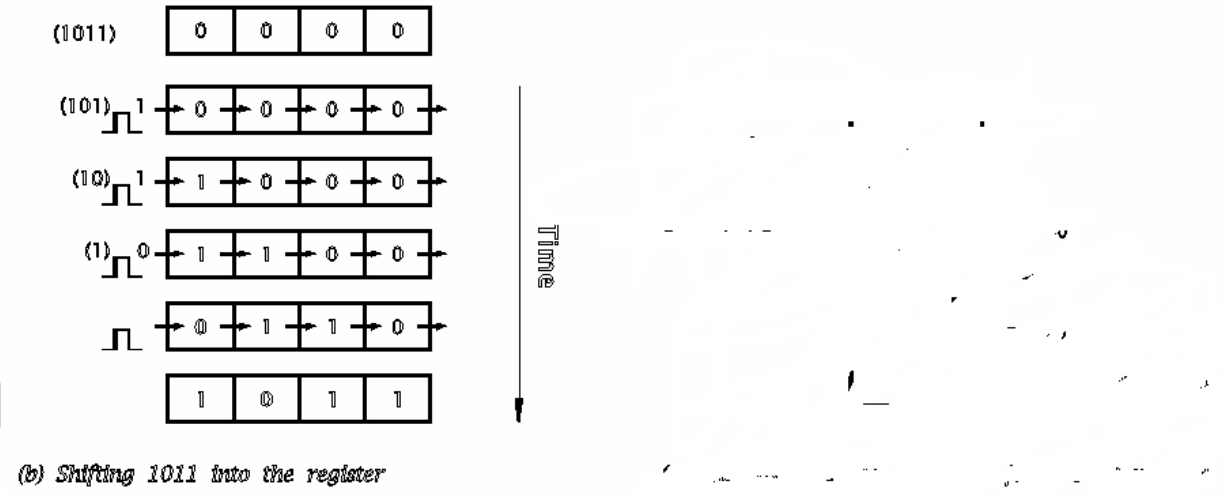
16. Explain different types of shift registers with data shifting diagrams.



SISO-In the serial mode, the registers have a single serial input and a single serial output. The information is transferred one bit at a time while registers are shifted in the same direction.

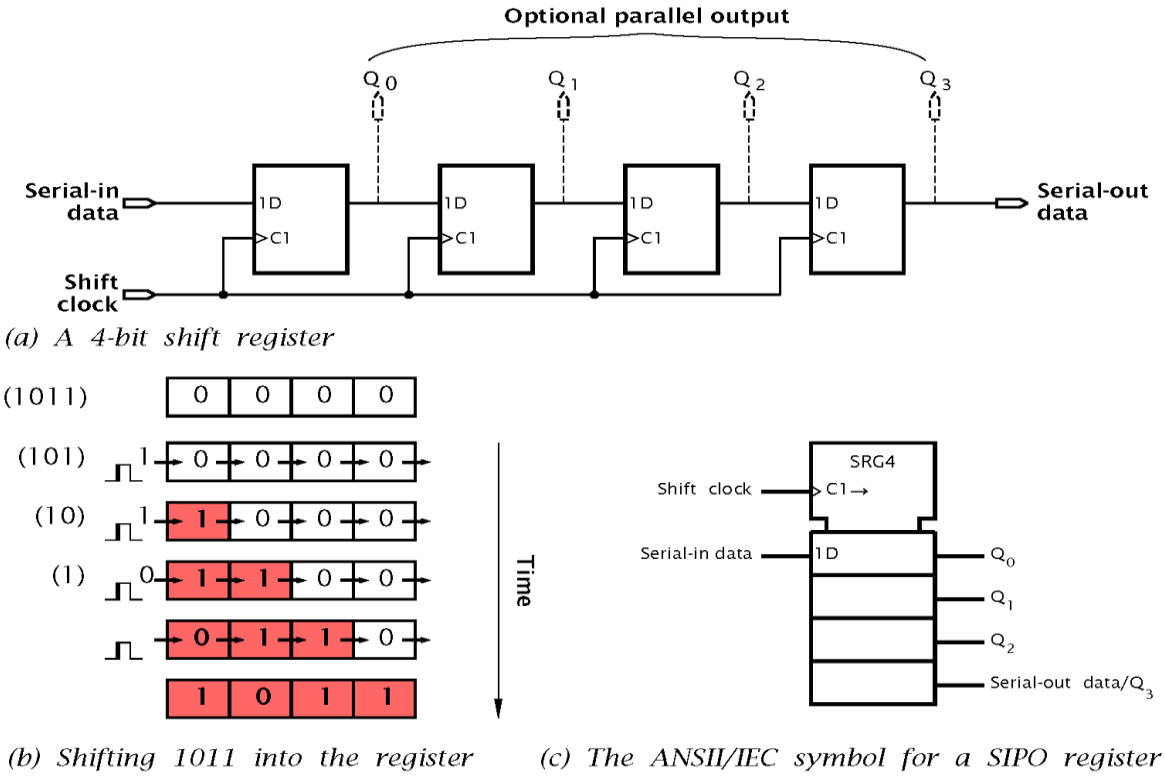


(a) A 4-bit shift register

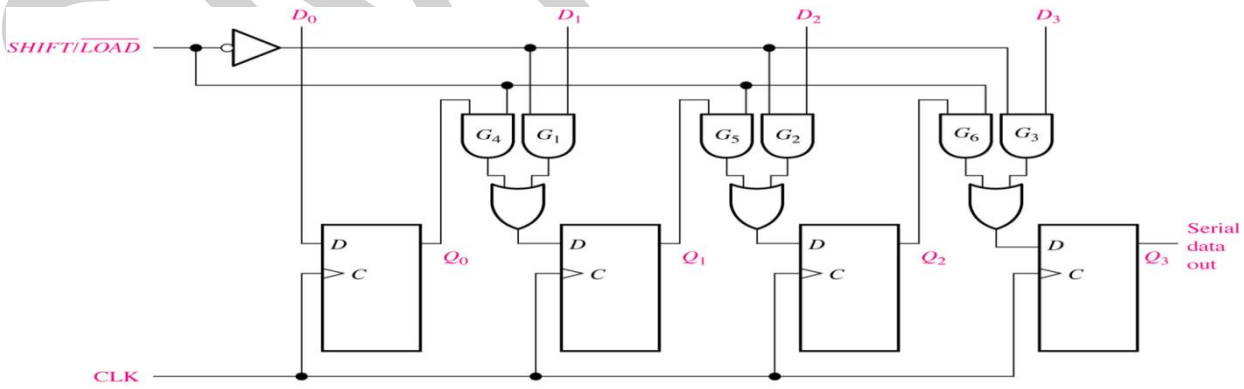


(b) Shifting 1011 into the register

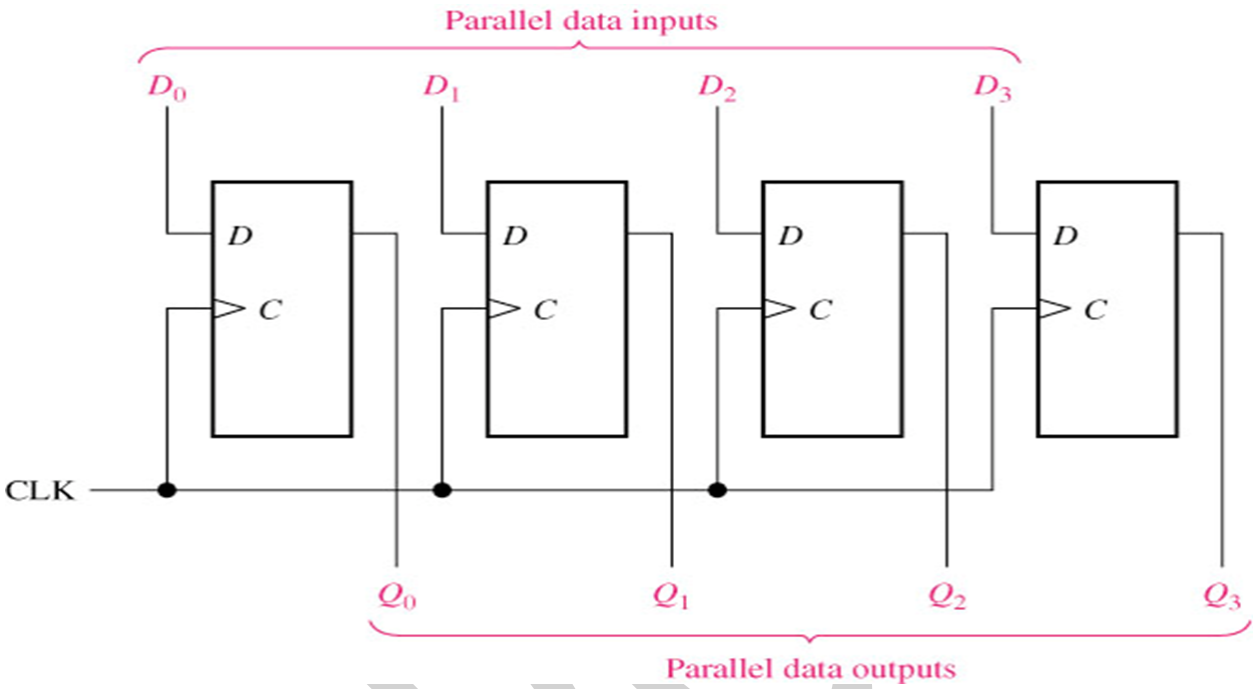
SIPO-data bits are entered serially in the same manner as discussed in the last section. The difference is the way in which the data bits are taken out parallel from the register.



PISO- data bits are entered parallel and outputs are taken out serially.

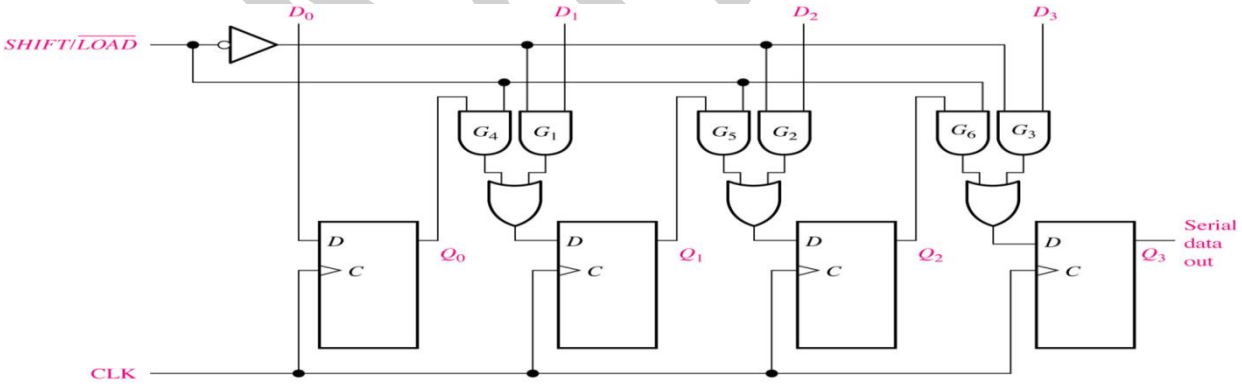


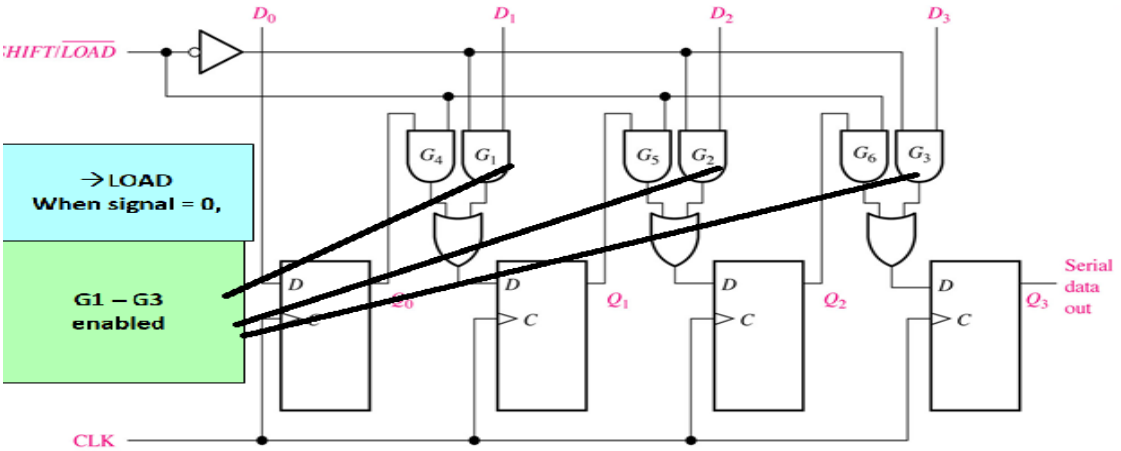
PIPO- all data bits appear on the parallel outputs immediately following the simultaneous entry of the data bits.

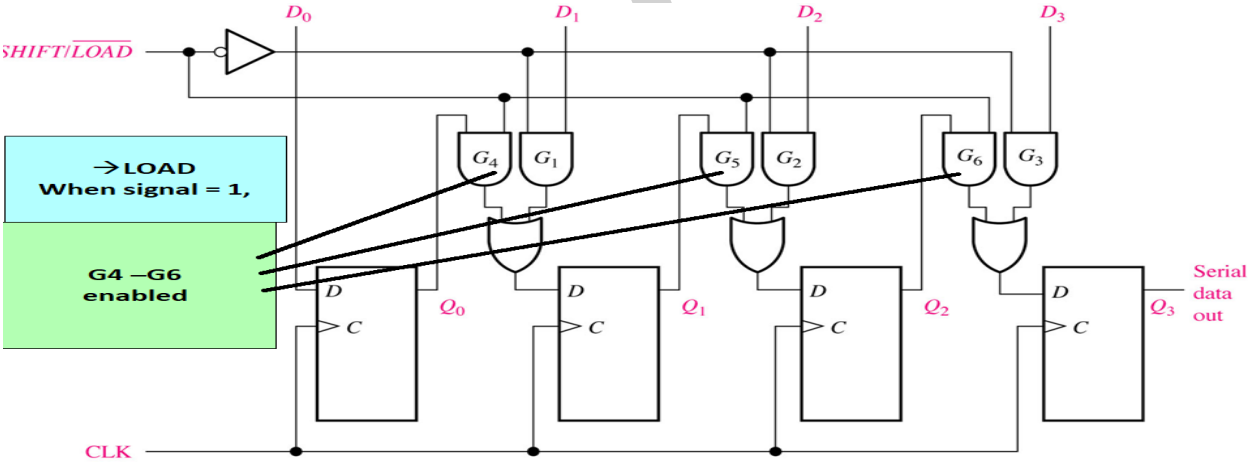


17. Explain the working of a parallel in serial out shift register.

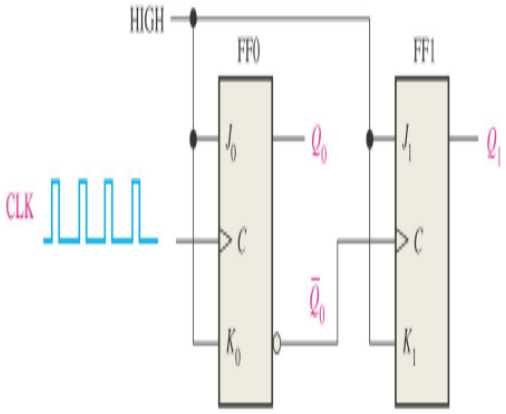
The parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.





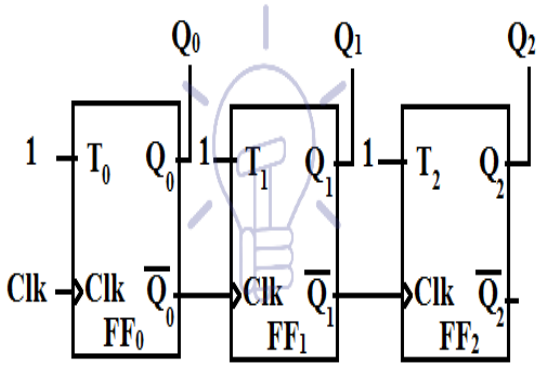


18. Draw a 2 bit or MOD 4 asynchronous counter.



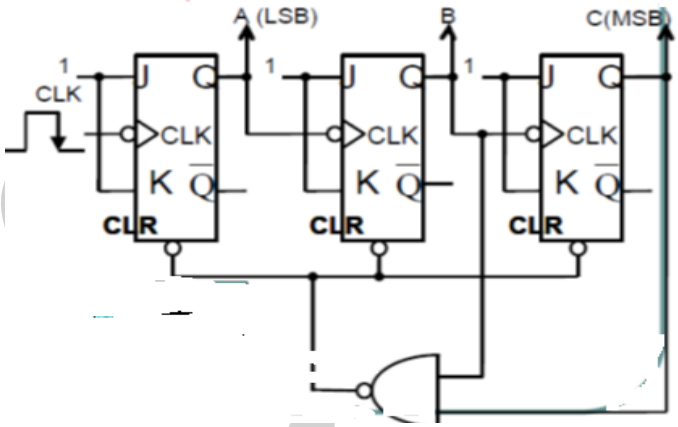
Clock Pulse	Q1	Q0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

19. Draw a 3 bit or MOD 8 asynchronous counter using T flip flop.



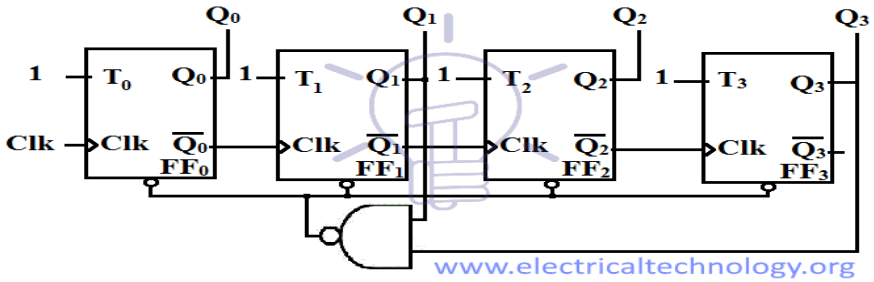
Counter State	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

20. Design a MOD 6 asynchronous circuit



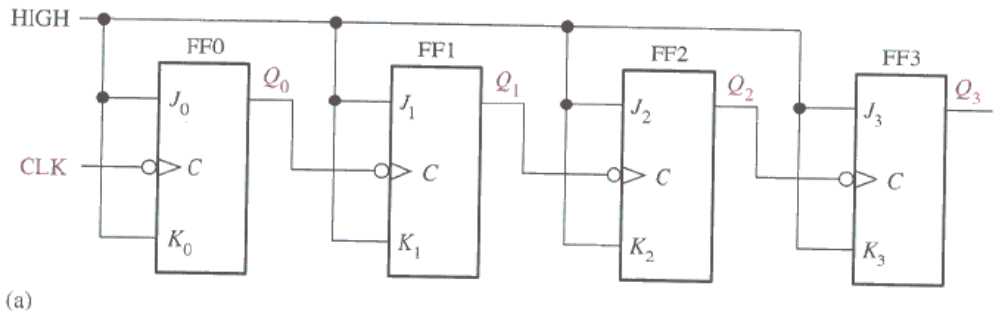
Clk	Q_0	Q_1	Q_2
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1

21. Design and implement a MOD 10 or Decade asynchronous counter using T flip flop and explain its working.



State	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 (reset)	1	0	1	0

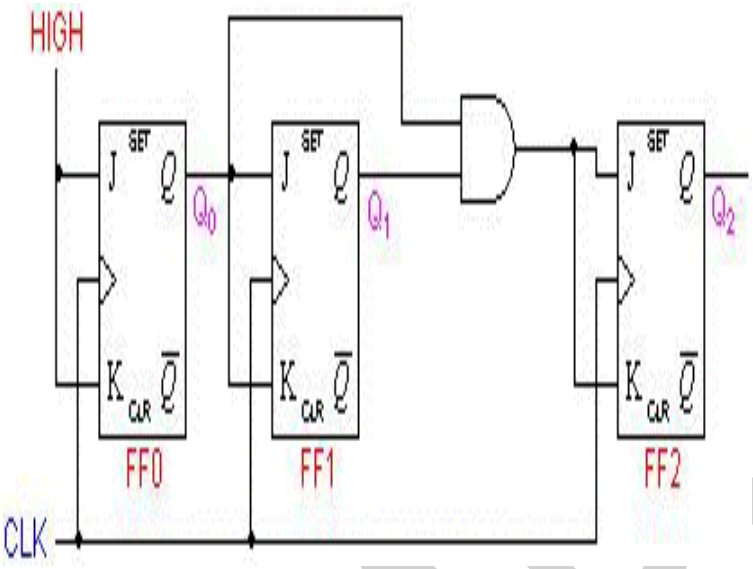
22. Design and implement a MOD 16 or 4 bit asynchronous counter using flip flop and explain its working.



(a)

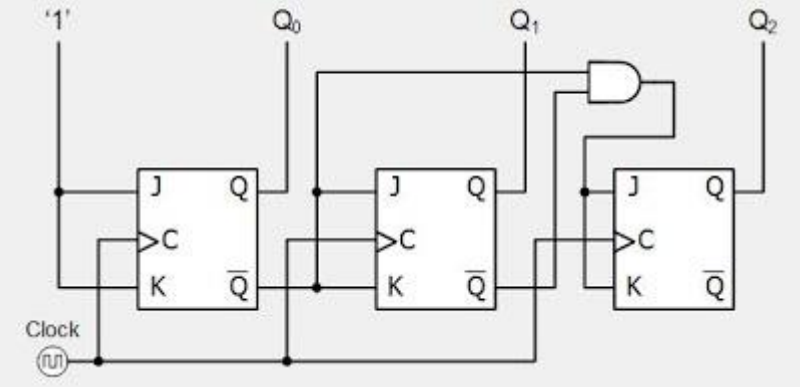
CLK	Q0	Q1	Q2	Q3
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	0
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

23. Design a synchronous 3 bit or MOD 8 up counter.



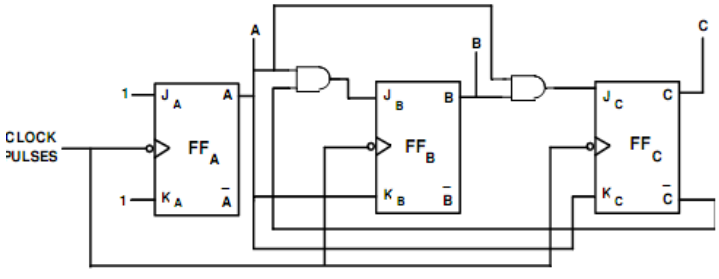
Clk	Q_0	Q_1	Q_2
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

24. Design a synchronous 3 bit down counter.



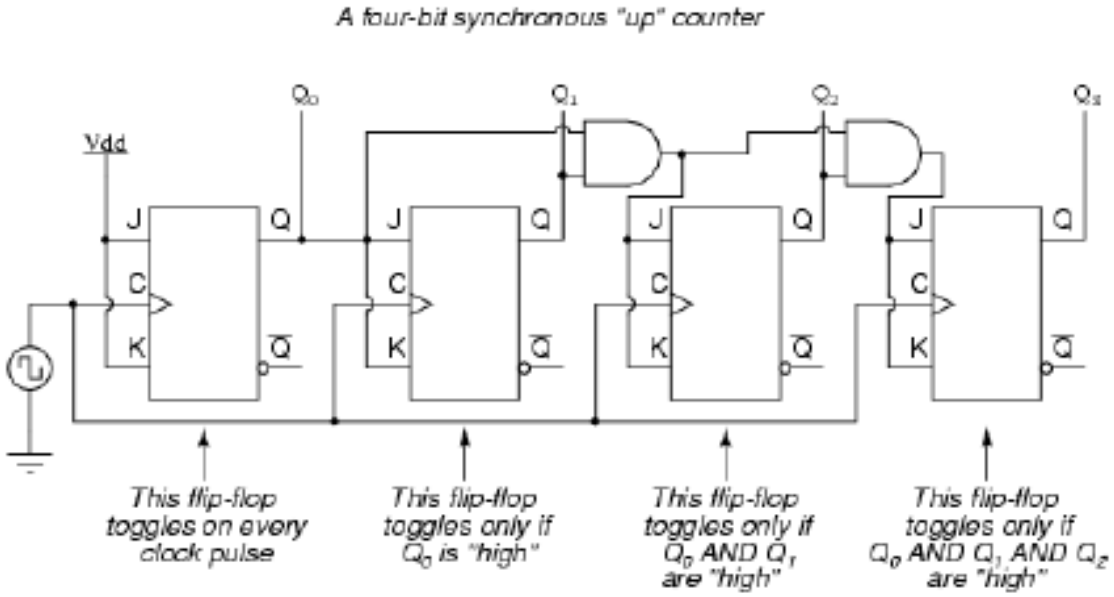
Clk	Q_0	Q_1	Q_2
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1

25. Design a synchronous MOD 6 counter using JK flip flop.



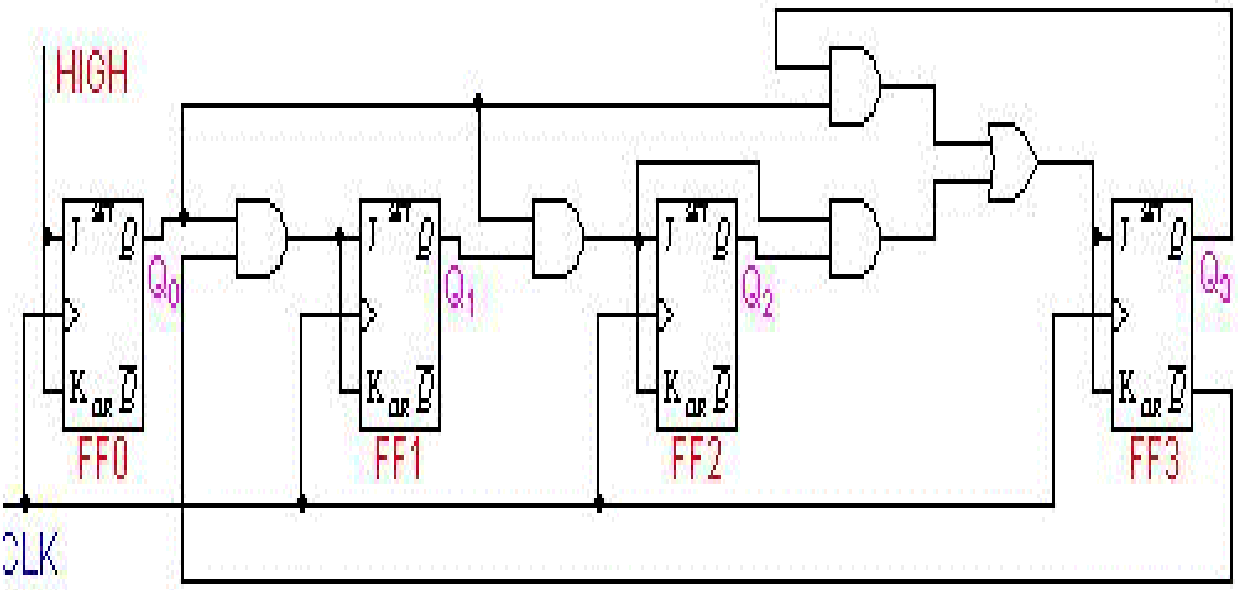
Clk	Q_0	Q_1	Q_2
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1

26. Design a synchronous MOD 16 or four 4 counter using JK flip flop



CLK	Q0	Q1	Q2	Q3
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	0
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

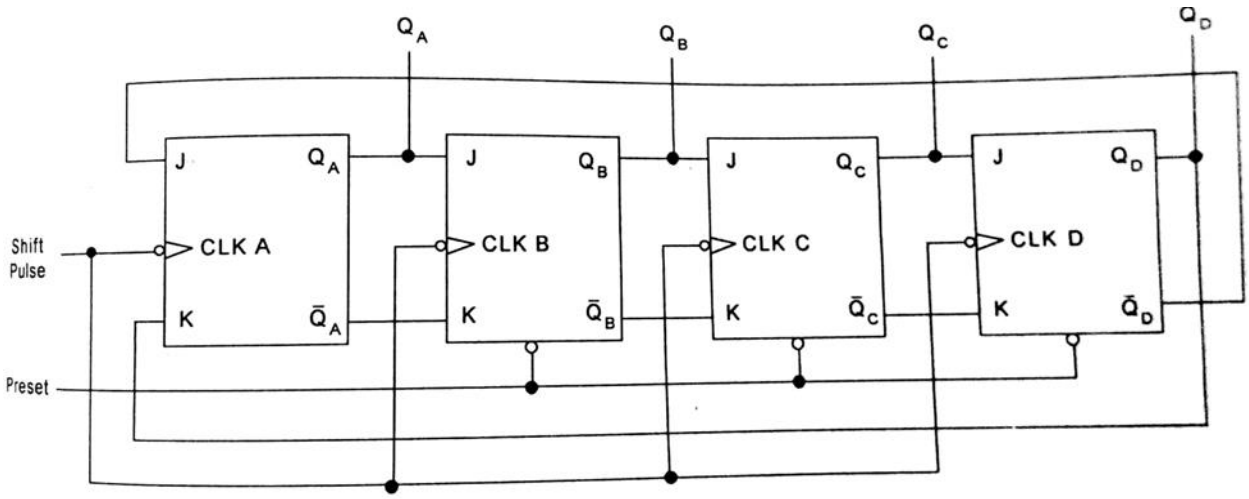
27. Design a synchronous MOD 10 or Decade counter using JK flip flop



State	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 (reset)	1	0	1	0

28. Explain the working of 3 bit Johnson counter using JK flip flop.

Johnson counters are a variation of standard ring counters, with the inverted output of the last stage fed back to the input of the first stage. They are also known as twisted ring counters. An n-stage Johnson counter yields a count sequence of length 2n, so it may be considered to be a mod-2n counter.



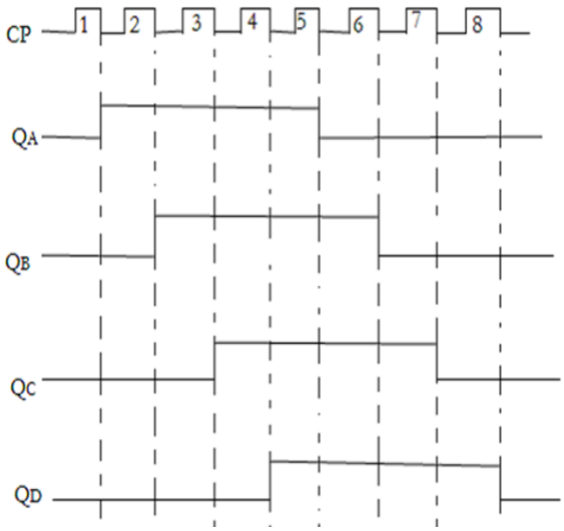
1.Since the J input of flip flop A is 1,the 1st shift pulse sets the A flip flop and other flip flop remain rest as the J inputs of these flip flops are 0 and K inputs are 1.

2.When the 2nd shift pulse is applied, since QB is still 1,flip flop A remains set and flip flop B is set,while flip flop C and D remains reset.

3.During the 3rd shift pulse flip flop C also sets,while flip flops A and B already set but flip flop D remains reset.

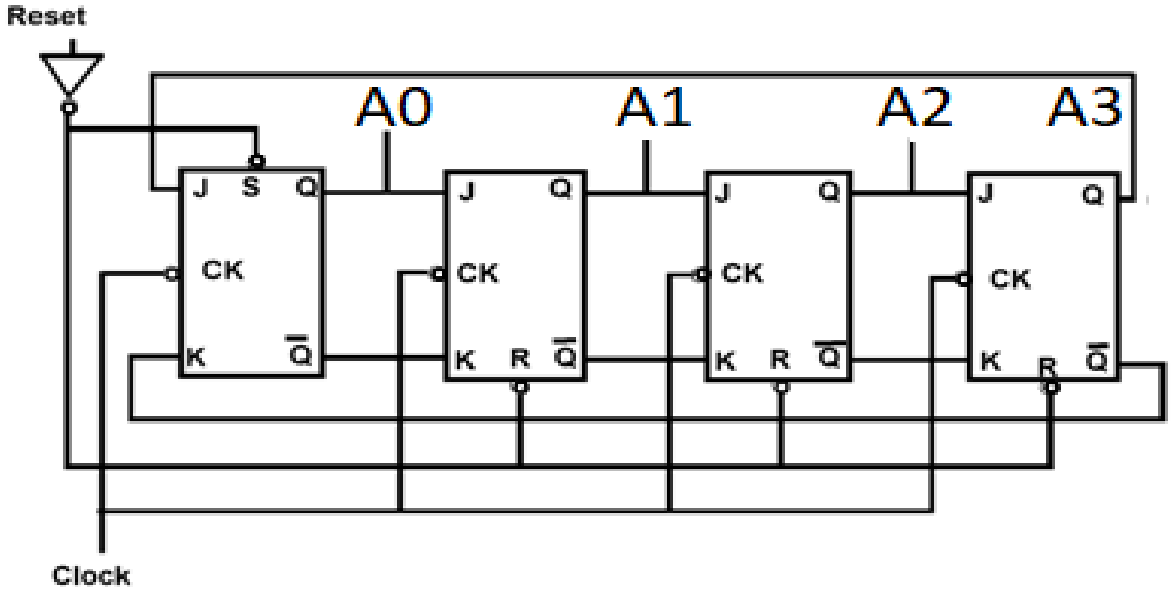
4.During the 4th pulse ,flip flop D also sets while flip flops, A,B and C are already set.

Clock pulse	**QD**	**QC**	**QB**	**QA**
1	0	0	0	0
2	0	0	0	1
3	0	0	1	1
4	0	1	1	1
5	1	1	1	1
6	1	1	1	0
7	1	1	0	0
8	1	0	0	0



29. Explain the working of a 4 bit ring counter. Or Design a 4 bit ring counter.Also represent it using timing diagram and state diagram.

A ring counter is basically a circulating shift register in which the output of the most significant stage is fed back to the input of the least significant stage. The following is a 4-bit ring counter constructed from JK flip-flops. The output of each stage is shifted into the next stage on the positive edge of a clock pulse. If the CLEAR signal is high, all the flipflops except the first one FF0 are reset to 0. FF0 is preset to 1 instead.

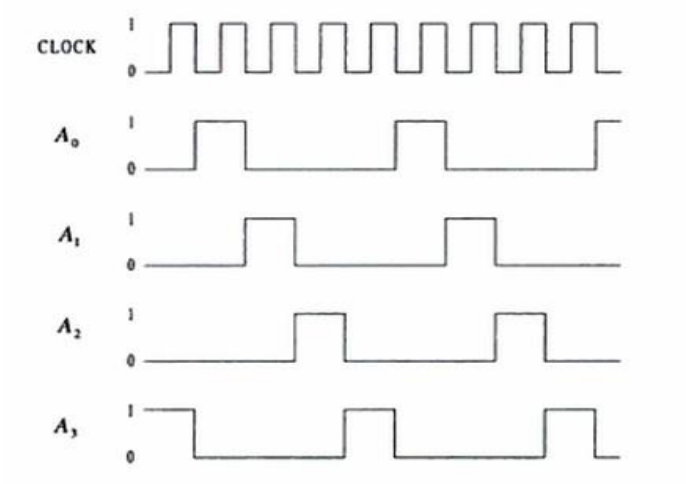


1.The Q4 and $\overline{Q4}$ outputs of the last flip flops are connected respectively,to the J and K inputs of flip flop A.

2.The preset input of flip flop A is connected to the reset inputs of flip flops for first,second,third.

If we place only one of the flip flops in the set state and the others in the reset state and then apply clock pulses,the logic 1 will advance by one flip flop around the ring for each clock pulse and the logic 1 will return to the original flip flop after exactly four clock pulses,as there are only 4 flip flops in the ring.

Timing Diagram

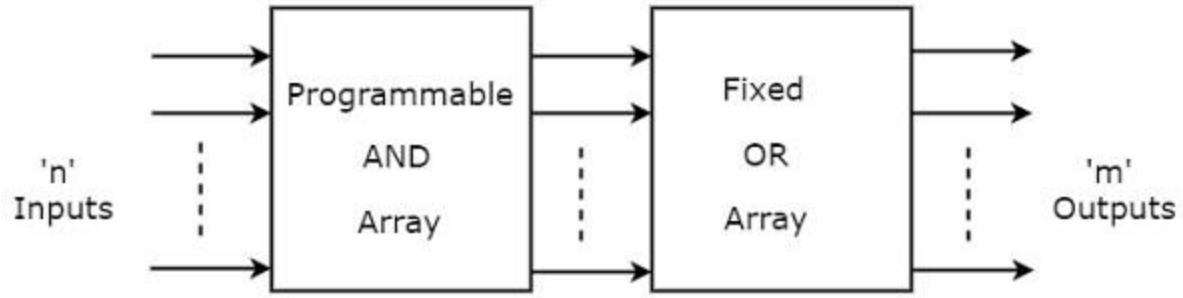


A3	A2	A1	A0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

MODULE 4

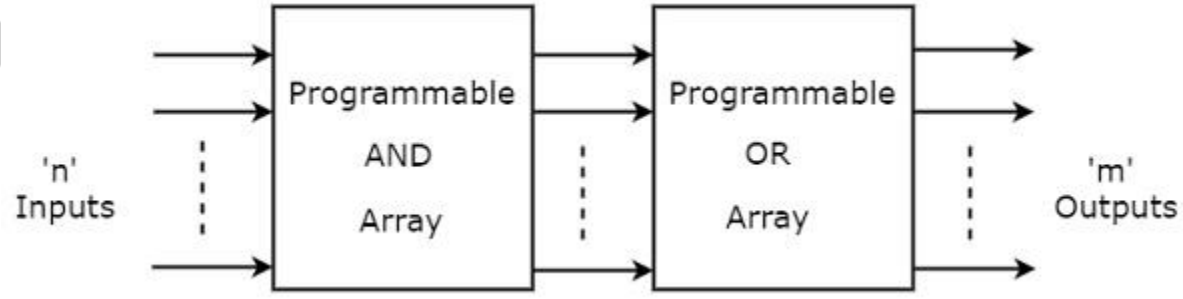
A/D,D/A,MEMORY AND PROGRAMMABLE LOGIC DEVICES

1. **Name an error detecting code.(OCT 16)**
   Hamming Code
2. **A group of 4 bits is called a byte and group of 8 bits is called nibble.(APR 17)**
3. **What is hamming code, also specify its applications.)(APR 19)**
   It is a code to detect error and correction .It is a single error correction
4. **What is PAL? .(OCT 18)**
   PAL is a programmable logic device that has Programmable AND array & fixed OR array.



5. **What is PLA?**

   PLA is a programmable logic device that has both Programmable AND array & Programmable OR array. Hence, it is the most flexible PLD.



6. **What is meant by resolution in DAC? (APR18)**
   Resolution of DAC is defined as the smallest change that can occur in the analog output as a result of a change in digital input.
7. **Describe the need of DAC and ADC in digital system. (APR 17)**
   DAC
   Microprocessors can only perform complex processing on digitized signals.
   When signals are in digital form they are less susceptible to the deleterious effects of additive noise.
   ADC Provides a link between the analog world of transducers and the digital world of signal processing and data handling.

ADC

To display the digital output of a digital system in analog form

To reconstruct the analog signal

To synthesize the speed ,video signals etc

8. List and explain various DAC specifications. (APR 17)(APR 19) (APR18)(OCT 17) (OCT 16)

Accuracy: Accuracy indicates how close the measured value is to the true value. The most common factors for specifying accuracy are full scale error or Gain error and linearity error.

Full Scale error is the maximum deviation of the output value from its expected(ideal) value expressed in percentage of full-scale.

Gain Error: Difference in slope of the ideal curve and the actual DAC output.

Linearity error is the maximum deviation in step size from the ideal step size.

More expensive DACs have full scale and linearity errors as low as .001% of full scale.

Offset Voltage: Ideally, the output of a DAC will be zero volts when binary inputs are are all 0s.

In practice, there will be a very small output voltage called offset voltage or offset error

Resolution of DAC is defined as the smallest change that can occur in the analog output as a result of a change in digital input.

- The resolution is always equal to the weight of the LSB and is also known as the step size,since it is the amount of V_O that will change when the digital input data goes from one step to the next.

Settling time: The time required for the output of the DAC to settle within (1/2)LSB of the final value for a given digital input

9. Explain a weighted resistor type DAC. (OCT 16)

- If R is any arbitrary resistance selected to suit the impedance level of the circuit
- Resistance from LSB are R,R/2,R/2^2,R/2^3,R/2^4,R/2^5,R/2^6
- The current I to the non Inverting terminal is
- $I=VR(\frac{s6}{R/2^6}+\frac{s5}{R/2^5}+\frac{s4}{R/2^4}+\frac{s3}{R/2^3}+\frac{s2}{R/2^2}+\frac{s1}{R/2^1}+\frac{s0}{R/2^0})$
- $=\frac{VR}{R}(2^6s6+2^5s5+2^4s4+2^3s3+2^2s2+2^1s1+2^0s0)$
- The output voltage V0
- $V0=-IRf=\frac{-V_R R_f}{R}(2^6s6+2^5s5+2^4s4+2^3s3+2^2s2+2^1s1+2^0s0)$

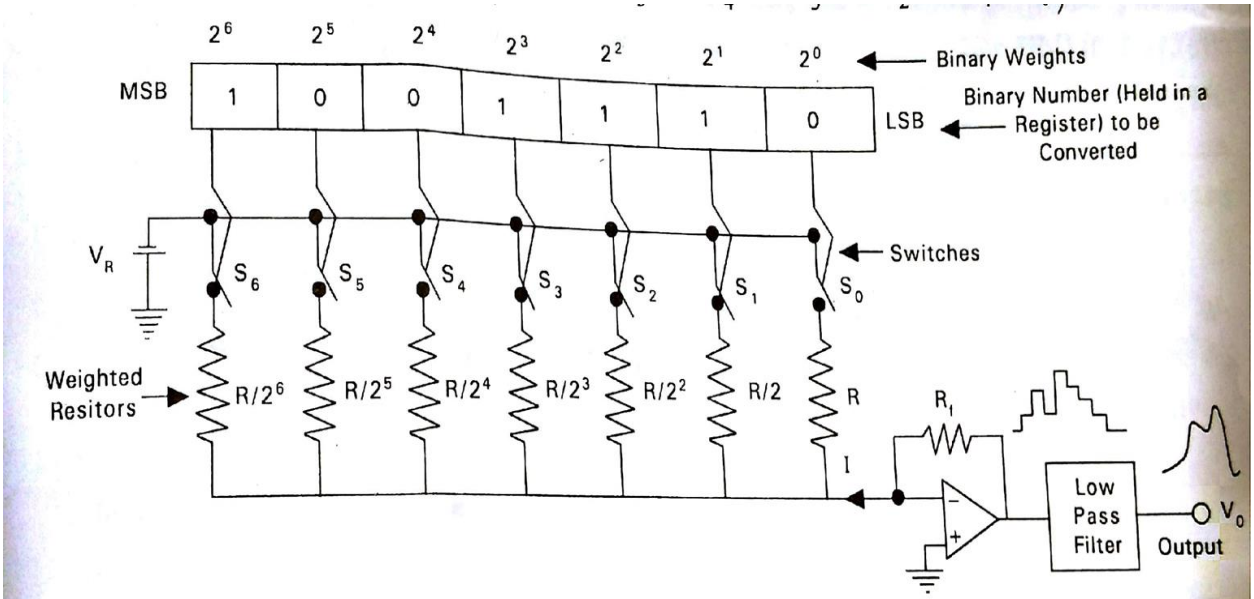


Fig. 4.9 Weighted Resistor D/A Converter

10. Explain a 4 bit DAC with neat block diagram. (APR 17)

or

11. Explain a R-2R ladder DAC.(OCT 18) (APR18)

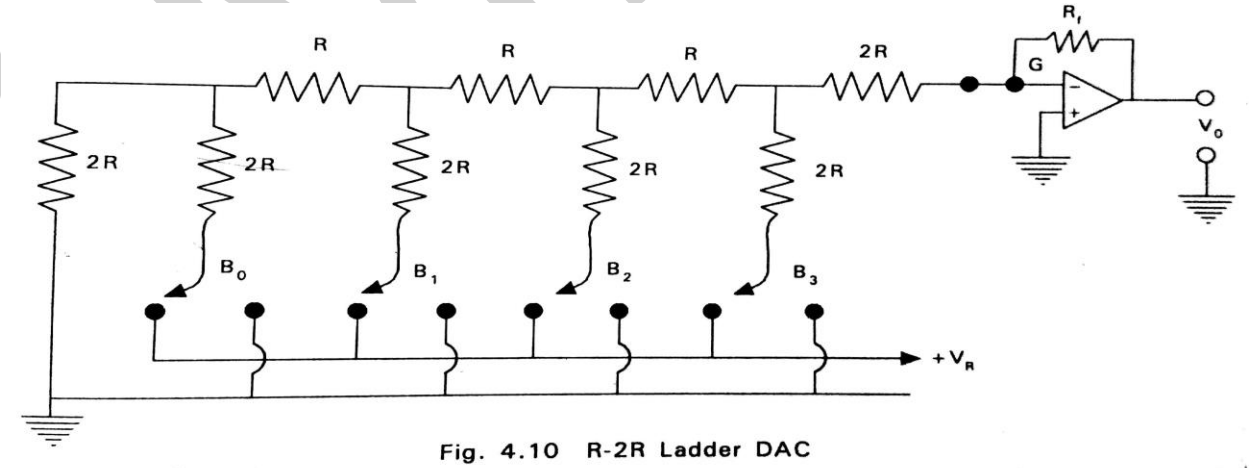


Fig. 4.10 R-2R Ladder DAC

- The operation of the circuit assume that the terminal B0 is connected to VR and all other terminals namely B1,B2,B3 are connected to ground.
- Applying Thevenin's successively to the nodes a0,a1,a2,a3 with respect to ground .

- The current i obtained
- $i=\frac{V_R}{3R}\left(\frac{B0}{16}+\frac{B1}{8}+\frac{B2}{4}+\frac{B3}{2}\right)$
- The output voltage of the OP AMP
- $V_0=-iRf=\frac{R_f}{3R}V_R\left(\frac{B0}{16}+\frac{B1}{8}+\frac{B2}{4}+\frac{B3}{2}\right)$
- $V_0=-\frac{R_f V_R}{48R}(2^3B3+2^2B2+2^1B1+2^0B0)$

12. Explain the working of counter ramp type ADC with diagram. (APR18))(OCT 17)

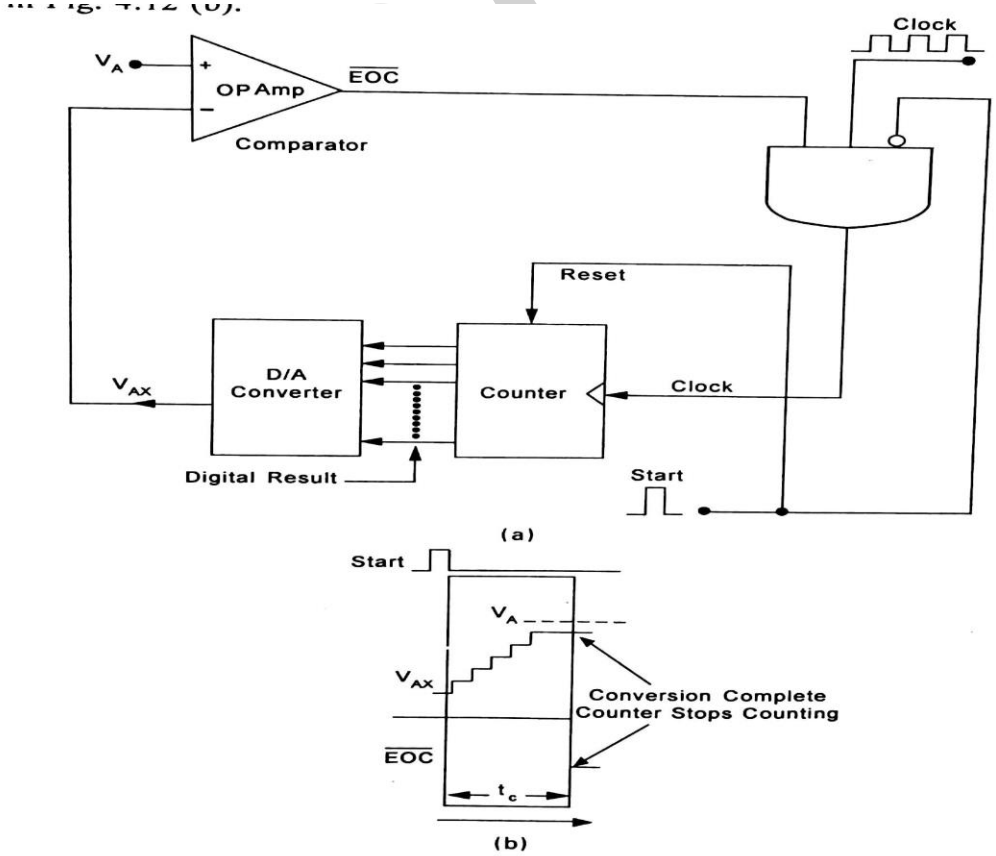


Fig. 4.12 Counter Ramp Type A/D Converter

Operation

Let VA is the analog input voltage to be converted,

1.A START pulse is applied to reset the counter to zero.The HIGH at start also inhibits clock pulses from passing through the AND gate into the counter.

2.With all 0s at its input,the DAC's output will be V_{AX}=0V.Since V_A>V_{AX},the comparator output $\overline{EOC}$ will be HIGH.

3.When START returns to LOW the AND gate is enabled and clock pulses get through to the counter.

4.This continues until V_{AX} reaches a step that exceeds V_A by an account equal to or greater than V_T.At this point $\overline{EOC}$will go to LOW and inhibit the flow of pulses into the counter and the counter will stop counting.

5.The conversion process is now complete as signaled by the HIGH to LOW transition at $\overline{EOC}$ and the contents of the counter are the digital representation of V_A.

6.The counter will hold the digital value until the next START pulse initiates a new conversion.

13. Explain the working of Successive approximation tpeADC with diagram.

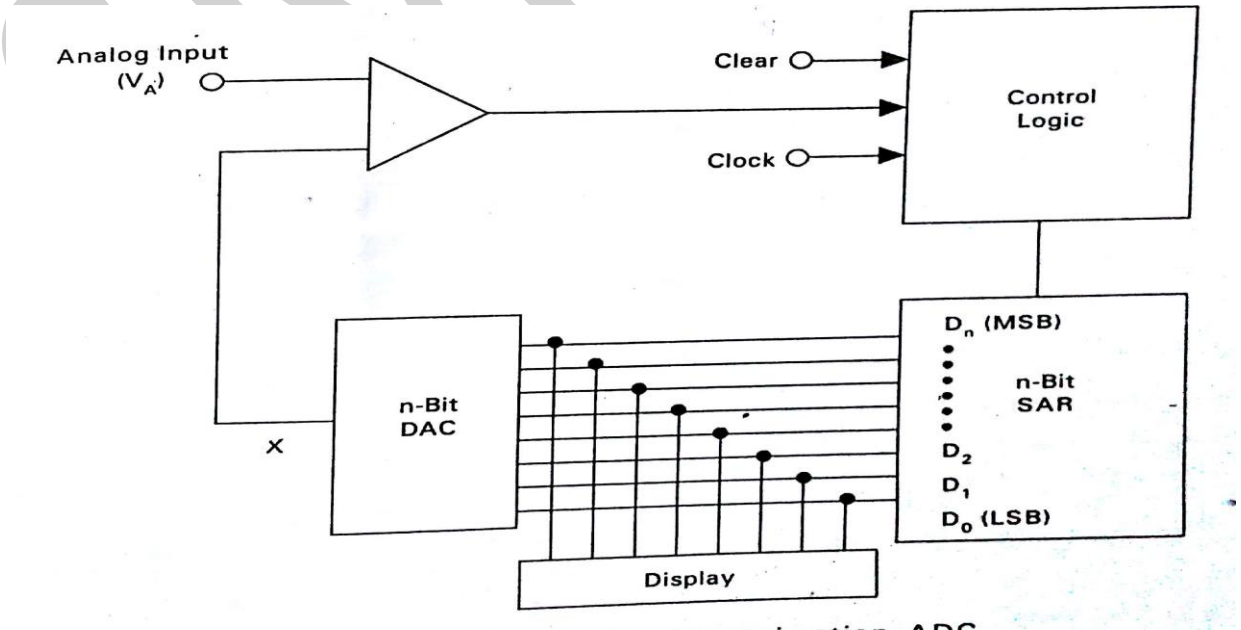


Fig. 4.14 Successive Approximation ADC

- The ring counter is initially reset to 0.Normally ring counter is apart of successive approximation register(SAR).
- The MSB is initially set to 1and the digital equivalent is compared with the unknown analog input voltage.
- If the analog input voltage is higher than the digital equivalent,the MSB is retained as 1 and the second MSB is set to 1.Otherwise the MSB is reset to 0 and the second MSB is set to 1.
- Comparison is made as given in step 2 to decide whether to retain or reset the second MSB and then the third MSB is set to 1.
- The above process is repeated down to LSB and by this time the converted digital value is available in the SAR.

14. Categorize and explain different types of ROMS.)(APR 19)

The required paths in a ROM may be programmed in four different ways.

1. Mask programming: fabrication process
2. Read-only memory or PROM: blown fuse /fuse intact
3. Erasable PROM or EPROM: placed under a special ultraviolet light for a given period of time will erase the pattern in ROM.
4. Electrically-erasable PROM(EEPROM): erased with an electrical signal instead of ultraviolet light.

15. Explain how memory decoding is performed. .(OCT 18)

or

Explain construction of memory cell with logic diagram. .(OCT 18))(OCT 17)

The equivalent logic of a binary cell that stores one bit of information is shown below.

Read/Write = 0, select = 1, input data to S-R latch
Read/Write = 1, select = 1, output data from S-R latch

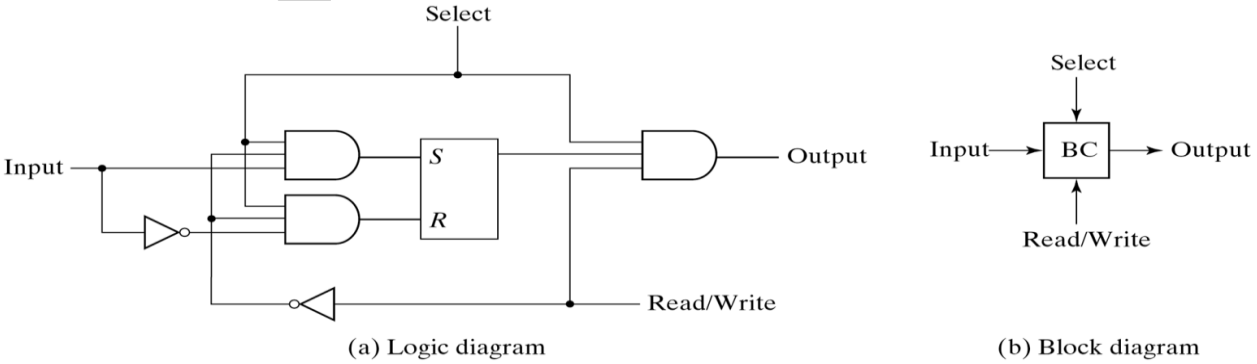


Fig. 7-5 Memory Cell

16. Draw an internal logic diagram of 32*8 ROM(OCT 16)

Each output of the decoder represents a memory address.
Each OR gate must be considered as having 32 inputs.
A 2^k * n ROM will have an internal k X 2^k decoder and n OR gates.

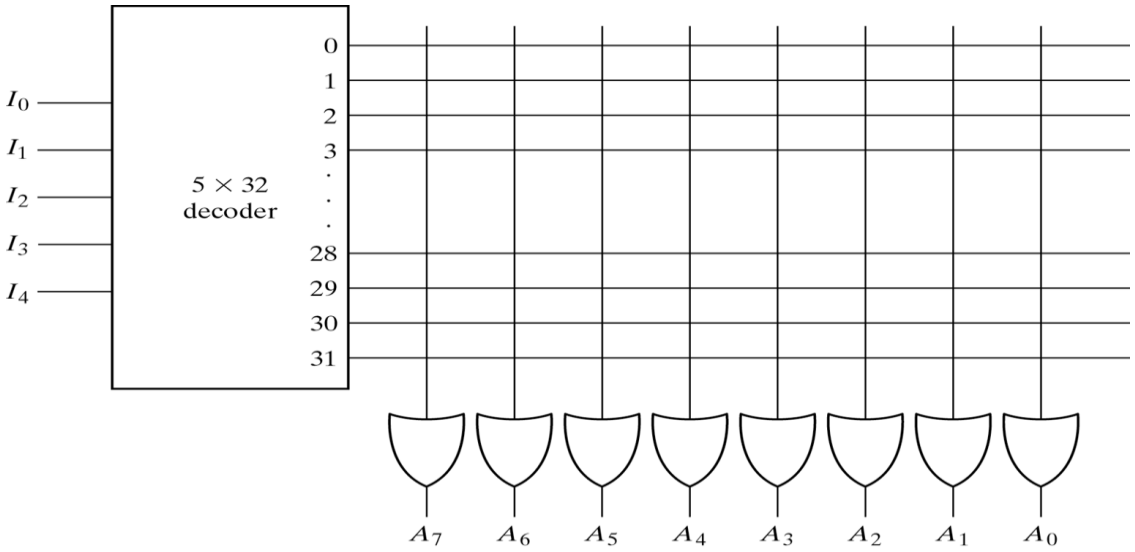


Fig. 7-10 Internal Logic of a 32 × 8 ROM

A programmable connection between to lines is logically equivalent to a switch that can be altered to either be close or open.
Intersection between two lines is sometimes called a cross-point

Table 7-3
ROM Truth Table (Partial)

Inputs					Outputs							
I4	I3	I2	I1	I0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮					⋮					
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

Address 3 = 10110010 is permanent storage using fuse link

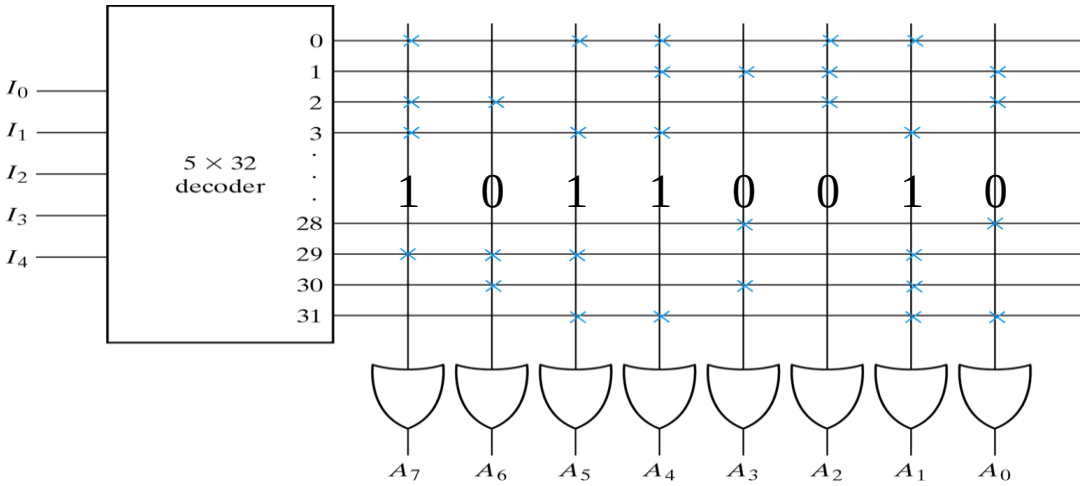


Fig. 7-11 Programming the ROM According to Table 7-3

17. Draw and explain two dimensional decoding structure for a 1K memory. (APR 17)

A decoder with k inputs and 2^k outputs requires 2^k AND gates with k inputs per gate. Two decoding in a two-dimensional selection scheme can reduce the number of inputs per gate.

1K-word memory, instead of using a single 10X1024 decoder, we use two 5X32 decoders.

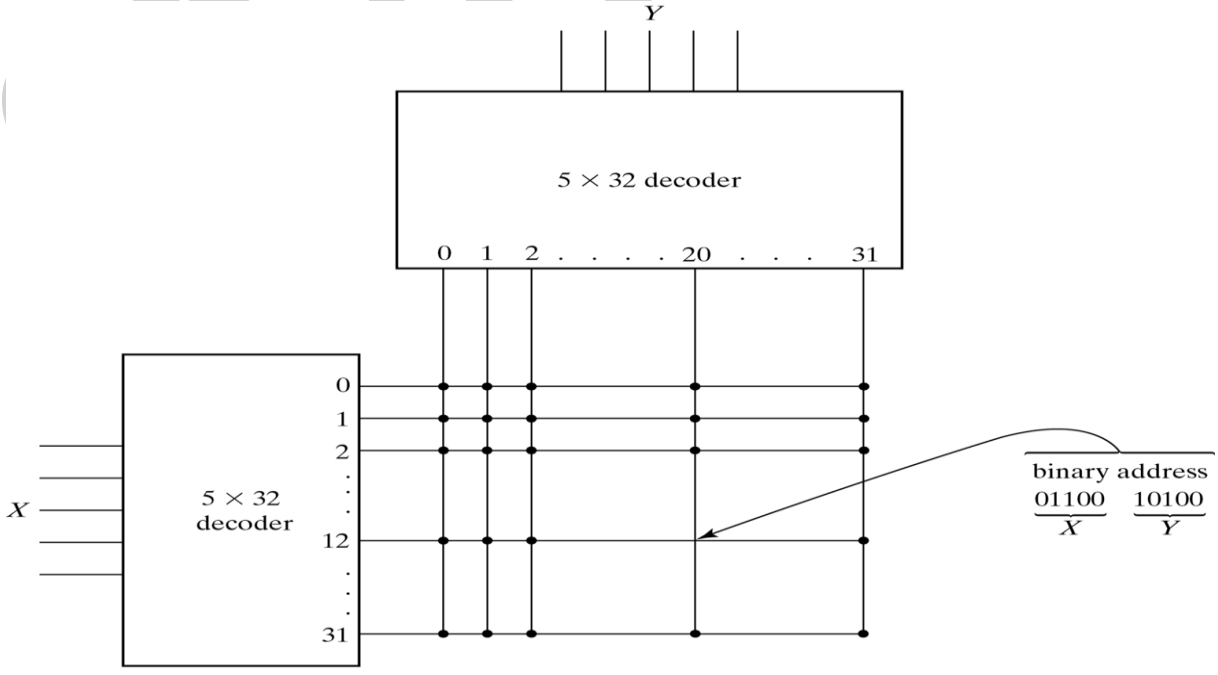


Fig. 7-7 Two-Dimensional Decoding Structure for a 1K-Word Memory

18. Explain the technique of error detection and correction using hamming code with example. (APR 17))(APR 19)

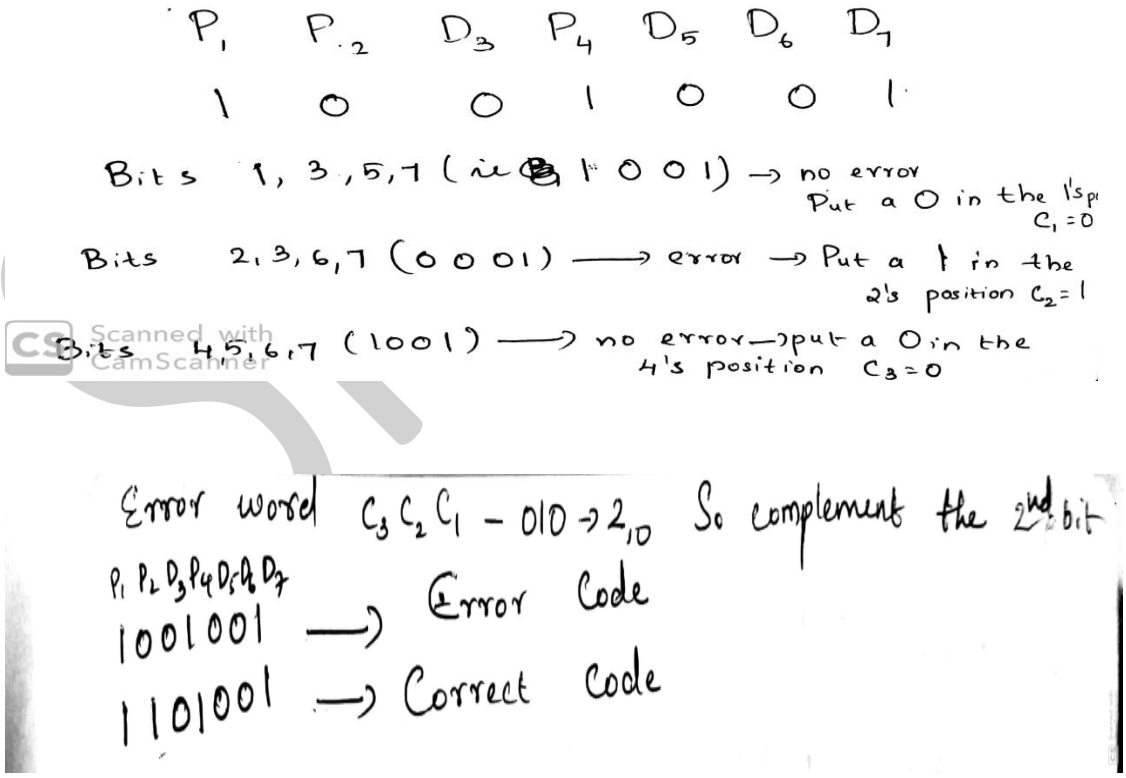
or

19. Write notes on error correction codes. (APR18)

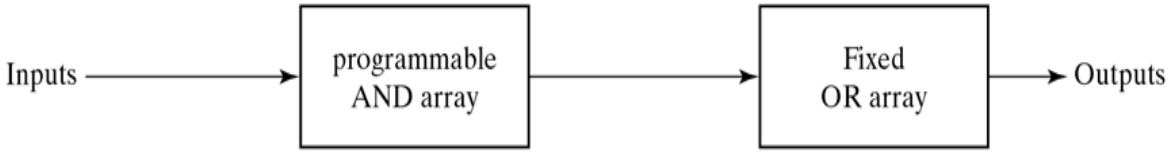
When data is transmitted through a channel,noise generated by various sources existing in the channel corrupting it.However,with digital data,if some bits are lost by noise corruption,there is a probability that the etire data itself may be lost .Data can be recovered from noise with the help of error detecting and correcting codes. Hamming code is used to find the error and then correcting it.

- It is protecting the occasional errors in storing and retrieving the binary information.
- Parity can be checked the error, but it can't be corrected.
- An error-correcting code generates multiple parity check bits that are stored with the data word in memory.

20. Decode the message "1001001 " coded in hamming code assuming that at most a single error occurred in the code.



21. Explain the working of Programmable logic array with example. (OCT 18)



(b) Programmable array logic (PAL)

- The product terms are then connected to OR gates to provide the sum of products for the required Boolean functions.
- The output is inverted when the XOR input is connected to 1 (since x⊕1 = x'). The output doesn't change and connect to 0 (since x⊕0 = x).
  Example:
  F1 = AB'+AC+A'BC'
  F2 = (AC+BC)'

Table 7-5
PLA Programming Table

		Inputs			Outputs (T)	(C)
	Product Term	A	B	C	F_1	F_2
AB'	1	1	0	–	1	–
AC	2	1	–	1	1	1
BC	3	–	1	1	–	1
A'BC'	4	0	1	0	1	–

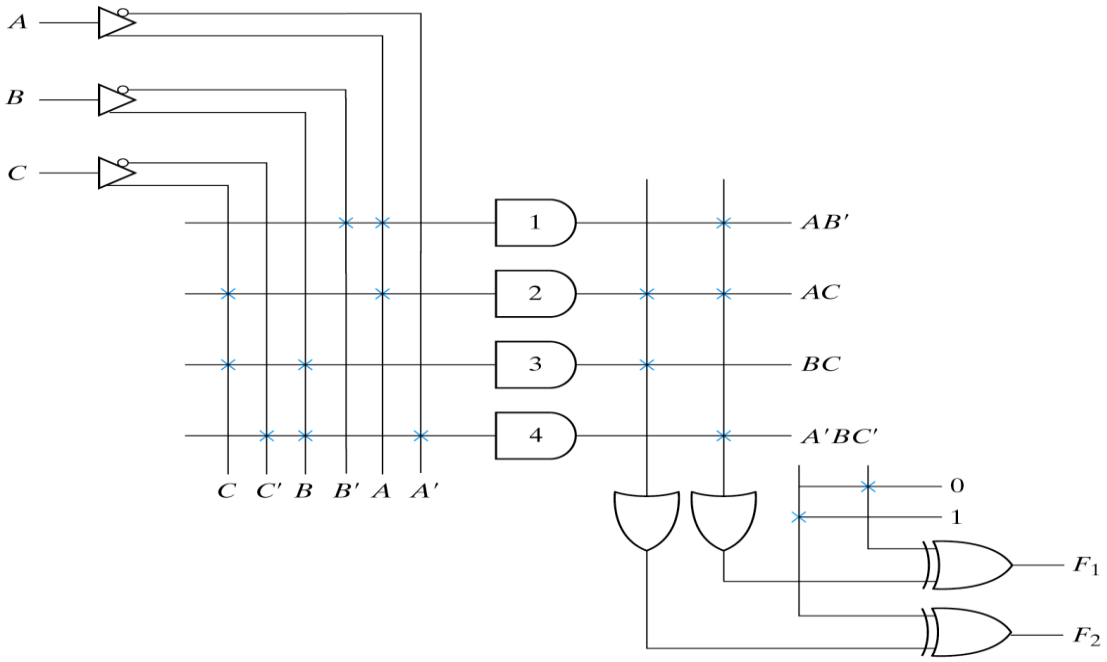
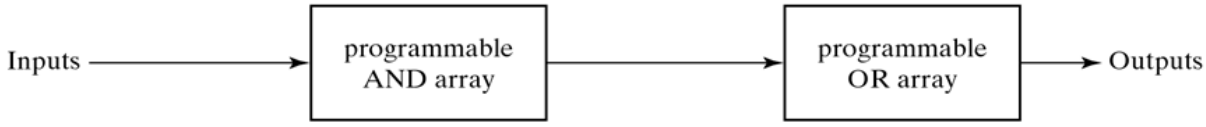


Fig. 7-14 PLA with 3 Inputs, 4 Product Terms, and 2 Outputs

22. Explain the working of Programmable array logic with example.



(c) Programmable logic array (PLA)

- When designing with a PAL, the Boolean functions must be simplified to fit into each section.
- Unlike the PLA, a product term cannot be shared among two or more OR gates. Therefore, each function can be simplified by itself without regard to common product terms.

2. $F_1 = \Sigma m(6, 8, 9, 12, 13, 14, 15)$

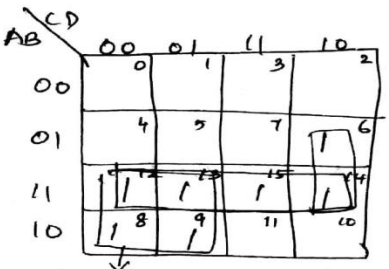
$F_2 = \Sigma m(1, 4, 5, 6, 7, 10, 11, 12, 13)$

$F_3 = \Sigma m(4, 5, 6, 7, 10, 11)$

$F_4 = \Sigma m(4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15)$

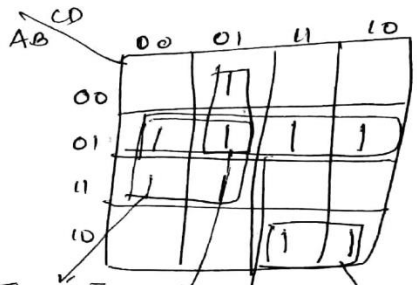
with PAL

F_1



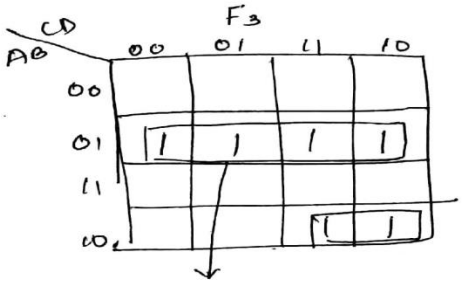
$F_1 = A\bar{C} + AB + BC\bar{D}$

F_2



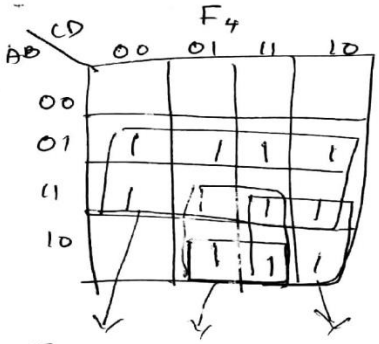
$F_2 = B\bar{C} + \bar{A}\bar{C}D + \bar{A}B + A\bar{B}C$

F_3



$F_3 = \bar{A}B + A\bar{B}C$

F_4



$F_4 = B + A\bar{D} + AC$

$F_1 = A\bar{C} + AB + BC\bar{D}$

$F_2 = B\bar{C} + \bar{A}\bar{C}D + \bar{A}B + A\bar{B}C = B\bar{C} + \bar{A}\bar{C}D + F_3$

$F_3 = \bar{A}B + A\bar{B}C$

$F_4 = B + A\bar{D} + AC$

Programming Table

Product Term	AND inputs A	B	C	D	F_3	Outputs
1	1	1	—	—	—	$F_1 = AB + A\bar{C} + BC\bar{D}$
2	1	—	0	—	—	
3	—	1	1	0	—	
4	—	—	—	—	1	$F_2 = F_3 + B\bar{C} + \bar{A}\bar{C}D$
5	0	—	0	1	—	
6	—	1	0	—	—	
7	0	1	—	—	—	$F_3 = \bar{A}B + A\bar{B}C$
8	1	0	1	—	—	
9	—	—	—	—	—	
10	—	1	—	—	—	$F_4 = B + AD + AC$
11	1	—	—	1	—	
12	1	—	1	—	—	

ANSWER FOR ALL THESE QUESTION ARE IN YOUR NOTE

23. **Draw a logic diagram to implement the Boolean functions**
    F1=
    F2= **with aPLA programming table. (OCT 16)**

24. **Draw a logic diagram to implement the Boolean function**
    F1=
    F2= **in PLA(APR18)**
25. **Realize the following functions using a PAL with 4 inputs and 3 wide AND OR structure along with the PAL programming table.)(APR 19)**
    F1=
    F2=
    F3=
    F4
26. **Develop a programming table for PAL for Boolean functions,**
    W=
    X=
    Y=
    Z=