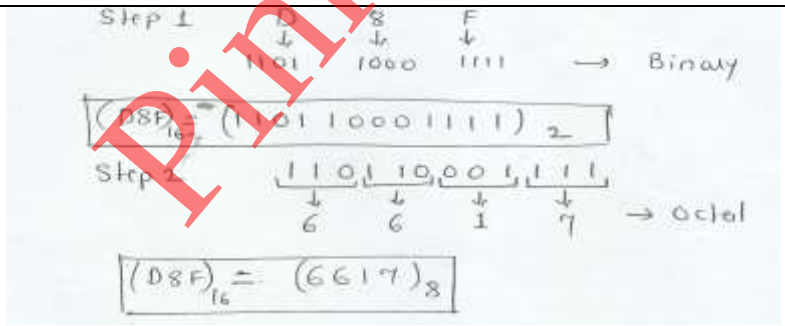
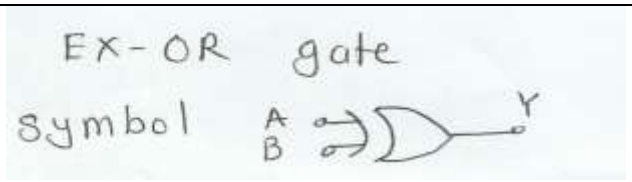
WINTER – 19EXAMINATIONS

Subject Name: Digital Techniques **Model Answer** Subject Code: 22320

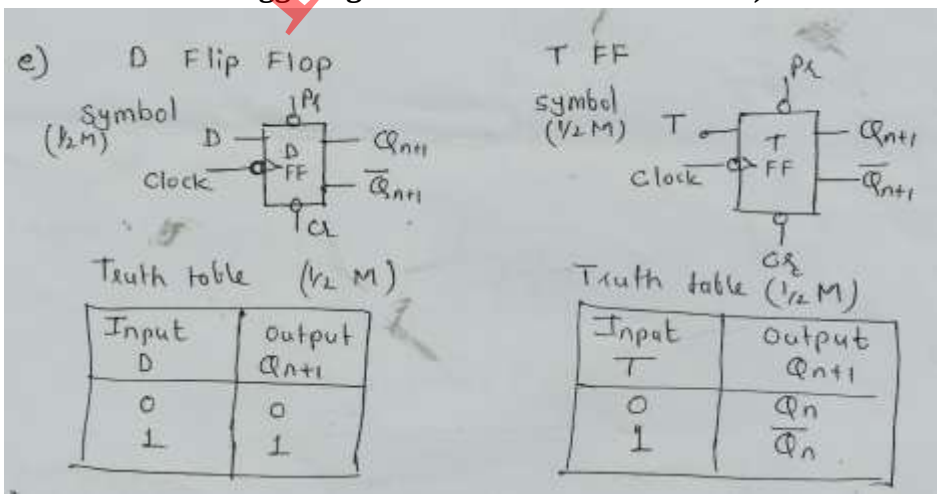
Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in themodel answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given moreImportance (Not applicable for subject English and Communication Skills.
4) While assessing figures, examiner may give credit for principal components indicated in thefigure. The figures drawn by candidate and model answer may vary. The examiner may give credit for anyequivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constantvalues may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

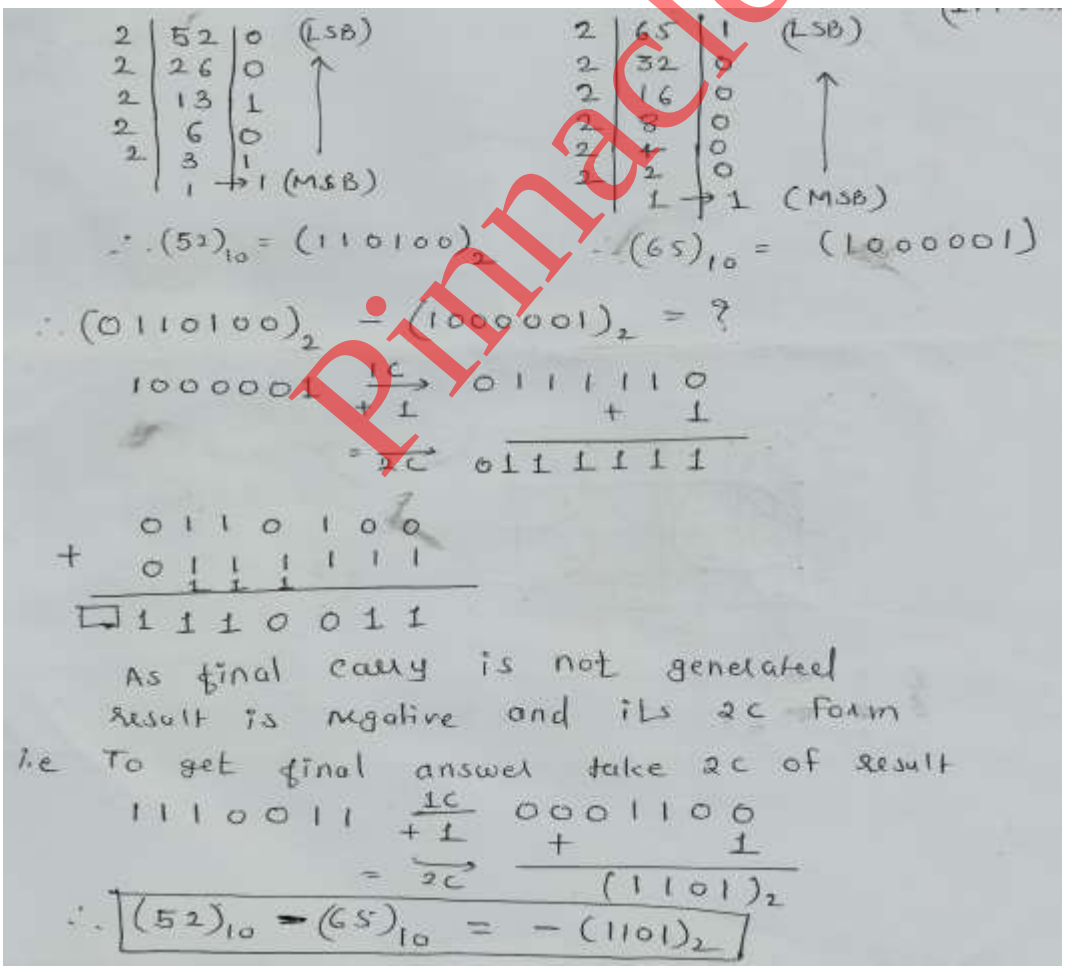
Q. No.	Sub Q. N.	Answer	Marking Scheme
Q.1		**Attempt any FIVE of the following:**	**10-Total Marks**
	a)	**Convert $(D8F)_{16}$ into binary and octal.**	**2M**
	Ans:		**1M** **1M**
	b)	**Draw symbol, Truth table and logic equation of Ex-OR gate.**	**2M**
	Ans:	 Logic Equation = $A\bar{B}+\bar{A}B$ OR $A \oplus B$ Truth Table:	**½ M** **½ M** **1M**

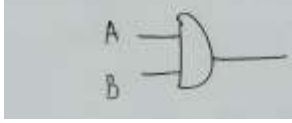
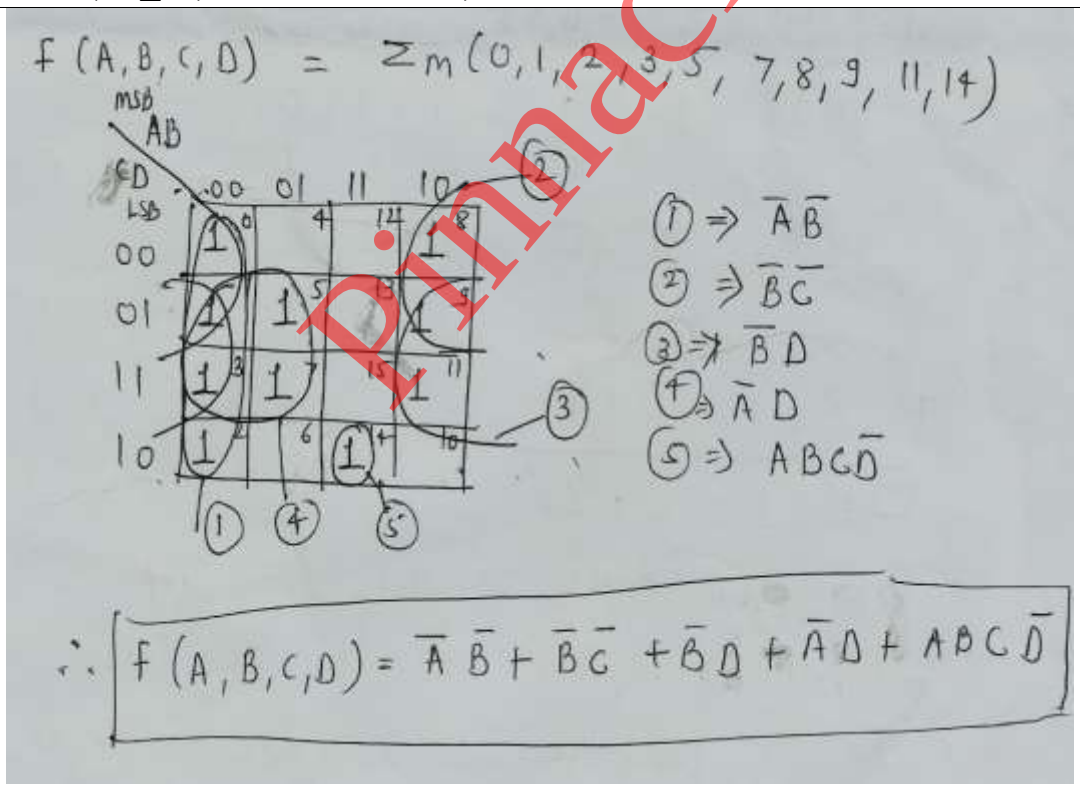
Inputs		Output
A	B	Y
0	0	0

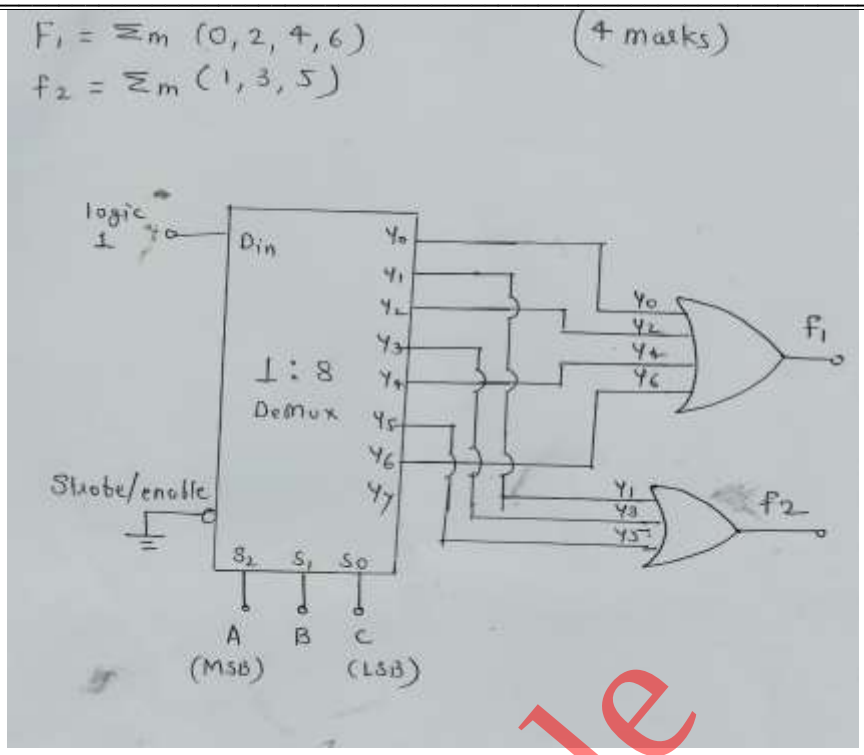
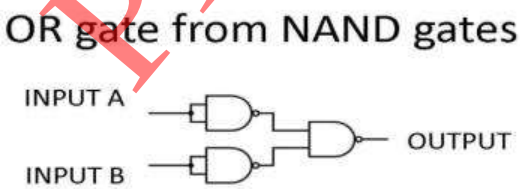
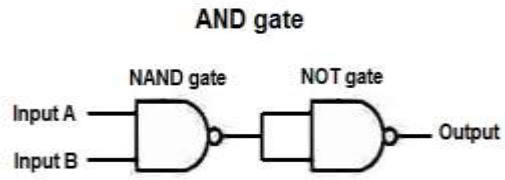
0	1	1		
1	0	1		
1	1	0		

	c)	**State the DeMorgan's Theorems.**	**2M**
	Ans:	De Morgan's 1st Theorem complement of sum is equal to product of their individual complements. OR $\overline{A+B} = \overline{A} \bullet \overline{B}$ De Morgan's 2nd theorem Complement of product is equal to sum of their individual complements. OR $\overline{A \bullet B} = \bar{A} + \bar{B}$	**1st-1M** **2nd -1M**
	d)	**Convert the following expression into standard SOP form.** **Y= AB+ A$\bar{C}$ + BC**	**2M**
	Ans:	Y = AB+ A$\bar{C}$ + BC Total variable ABC 1st Product term = AB (C is missing) 2nd Product term = A$\bar{C}$ (B is missing) 3rd Product term = BC (A is missing) Y= AB•1 + A$\bar{C}$ •1 + BC• 1 Y= AB(C+$\bar{C}$) A$\bar{C}$(B+$\bar{B}$) + BC(A+$\bar{A}$) Y= ABC + AB$\bar{C}$+ AB$\bar{C}$+ A$\bar{B}\bar{C}$ +ABC + $\bar{A}$BC ($\because A+\bar{A}=A$) Y= ABC + AB$\bar{C}$ + A$\bar{B}\bar{C}$ + $\bar{A}$BC Standard SOP Form	**2M**
	e)	**Draw symbol and write truth table of D and T Flip Flop.**	**2M**
	Ans:	**(Note: Symbol with other triggering method also can be consider)**  	**1M Symbol** **1M Truth table**
	f)	**Write down number of flip flops are required to count 16 clock pulses.**	**2M**
	Ans:	No of states= no.of clock pulses = 16	**2M**

		2^n =m n = no.of flip flops requried m= no.of states 2^n = 16 n = 4 4 flip flops are required to count 16 clock pulse.	
	g)	**List the types of DAC**	**2M**
	Ans:	1) Binary weighted DAC 2) R –2R ladder network DAC	**1M each**

Q.2		**Attempt any THREE of the following:**	**12-Total Marks**
	a)	**Perform the subtraction using 2'S Complement methods.** **$(52)_{10} - (65)_{10}$**	**4M**
	Ans:		**Conversion-1M each** **Complement-1M** **Final answer-1M**
	b)	**Simplify the following Boolean Expressionand Implement using logic gate.** **$AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD$**	**4M**

	Ans:	$AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD$ (2 mks) $= AB\bar{C}(\bar{D}+D) + ABC(\bar{D}+D)$ $(\because A+\bar{A}=1)$ $= AB\bar{C}\cdot 1 + ABC\cdot 1$ $= AB\bar{C} + ABC$ $\therefore (A\cdot 1 = A)$ $= AB(\bar{C}+C)$ $\therefore (A+\bar{A}=1)$ $= AB\cdot 1$ $\therefore (A\cdot 1 = A)$ $= AB$ Implementation (2 mks) 	**2M** **2M**
c)	**Minimize the four variable logic function using K map.** **$F(A,B,C,D) = \sum m(0,1,2,3,5,7,8,9,11,14)$**		**4M**
	Ans:		**Kmap with place value-1M** **Pair-1M** **Answer-2M**
d)	**Implement the following function using demultiplexer.** **$f_1 = \sum m(0,2,4,6)$** **$f_2 = \sum m(1,3,5)$**		**4M**

	Ans:		**4M**
Q.3		**Attempt any <u>THREE</u> of the following:**	**12-Total Marks**
	a)	**Realize the following logic expression using only NAND gates.** **(i) OR** **(ii) AND** **(iii)NOT**	**4M**
	Ans:	(i)OR  (ii)AND  (ii)NOT  (out put A bar)	**1½ M** **1½ M** **1M**

b) **Draw binary to gray converter and write its truth table.** **4M**

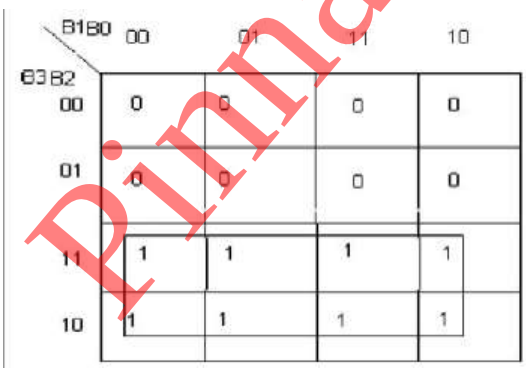
Ans: Truth Table for 4 bit Binary to Gray code converter **2M Truth table**

Binary Input				Gray Output			
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Note:

Kmap is optional

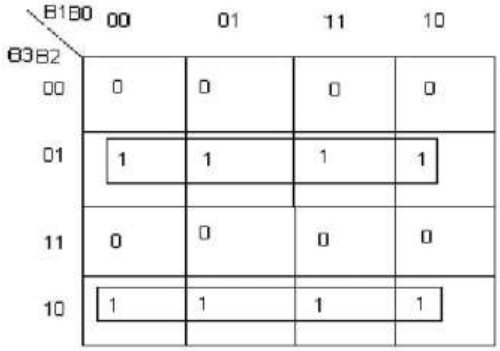
K-MAP FOR G3:



2M Logical diagram

G3=B3

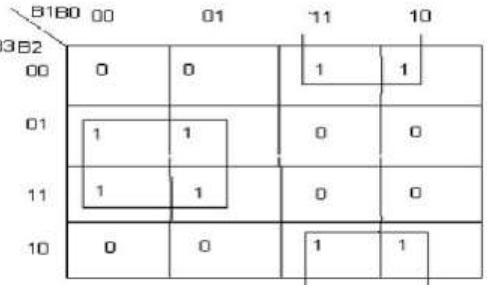
K-MAP FOR G2



G2 = $\overline{B3}$B2 + $\overline{B2}$B3

=B3 XOR B2

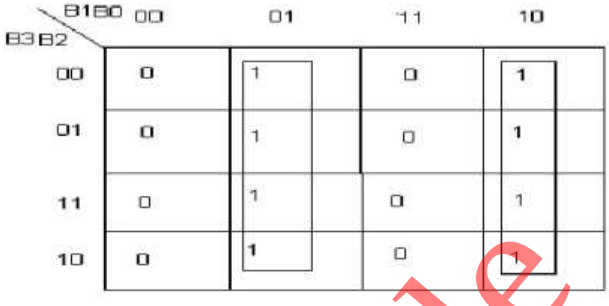
K-MAP FOR G1:



$G1=\overline{B2}\,B1 + B2\,\overline{B1}$

= B1 XOR B2

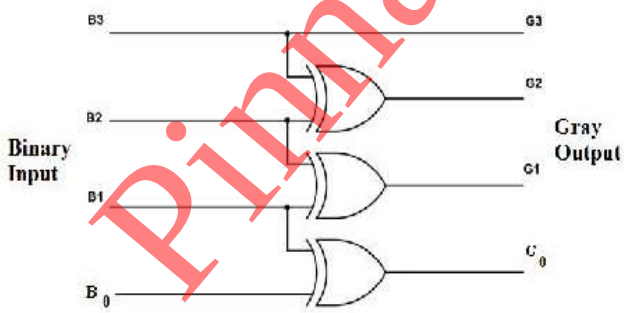
K-MAP FOR G0:



G0 = $\overline{B1}$B0+B1$\overline{B0}$

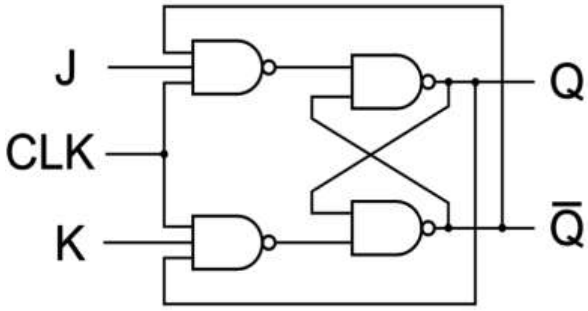
= B1 XOR BO

Diagram for 4 bit Binary to Gray code converter:



(Note: Realization of output equation can be done Basic or Universal)

c)	**Describe the working of JK flip flop with truth table and logic Diagram.**	**4M**
Ans:	logic Diagram:	**1M**



Truth Table

J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

1M

Working:

The JK flip flop is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level "1". Due to this additional clocked input, a JK flip-flop has four possible input combinations, "logic 1", "logic 0", "no change" and "toggle".

Both the S and the R inputs of the previous SR bistable have now been replaced by two inputs called the J and K inputs, respectively after its inventor Jack Kilby. Then this equates to: J = S and K = R.

The two 2-input AND gates of the gated SR bistable have now been replaced by two 3-input NAND gates with the third input of each gate connected to the outputs at Q and Q. This cross coupling of the SR flip-flop allows the previously invalid condition of S = "1" and R = "1" state to be used to produce a "toggle action" as the two inputs are now interlocked.

If the circuit is now "SET" the J input is inhibited by the "0" status

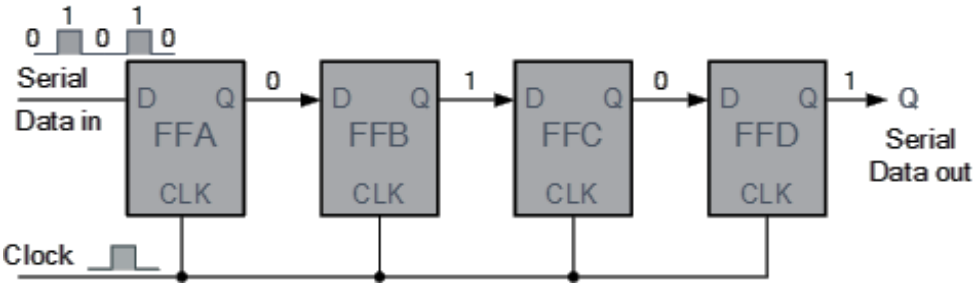
Of Q through the lower NAND gate. If the circuit is "RESET" the K input is inhibited by the "0" status of Q through the upper NAND gate. As Q and Q are always different we can use them to control the input. When both

inputs J and K are equal to logic "1", the JK flip flop toggles

2M

d) **Describe the working of 4 bit SISO (serial in serial out) shift register with diagram and waveform if input is 01101.** **4M**

Ans: Diagram:(use SR or JK or D type flip flop) **1M**

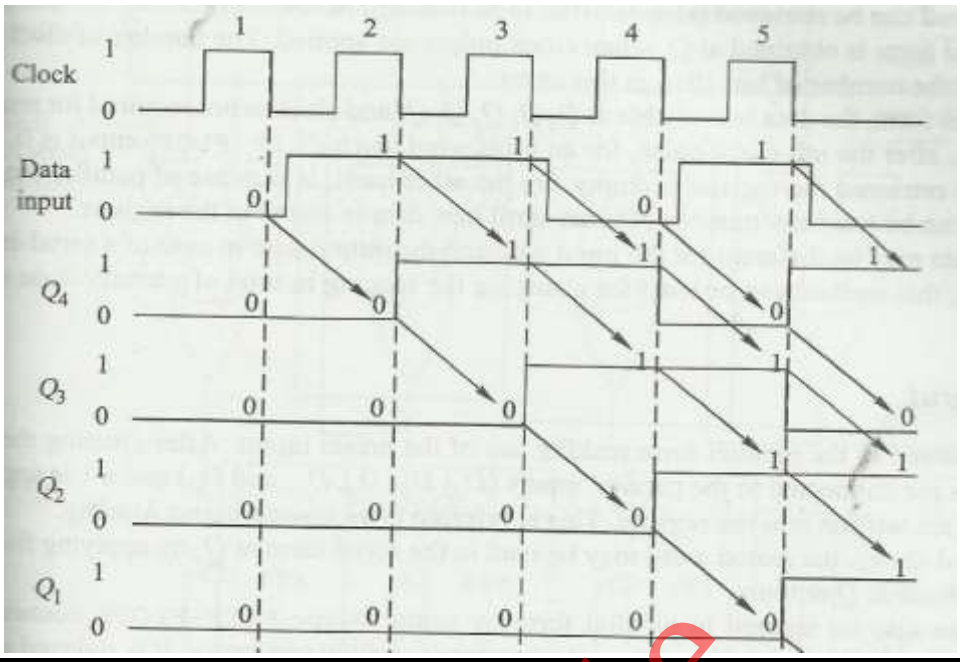
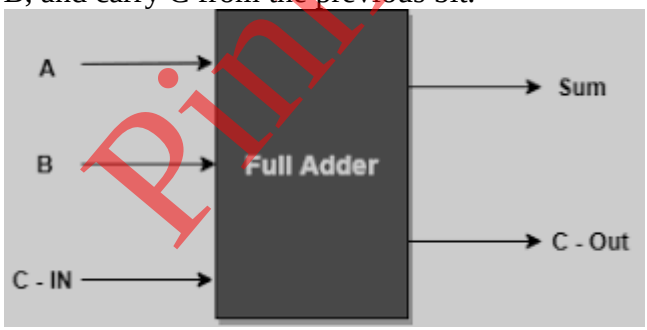


Working:

The DATA leaves the shift register one bit at a time in a serial pattern, hence the name **Serial-in to Serial-Out Shift Register** or **SISO**.

The SISO shift register is one of the simplest of the four configurations as it has only three connections, the serial input (SI) which determines what enters the left hand flip-flop, the serial output (SO) which is taken from the output of the right hand flip-flop and the sequencing clock signal (Clk). The logic circuit diagram below shows a generalized serial-in serial-out shift register, Output of FFA is Q_4,FFB Q_3,FFC Q_2 and FFD is Q_1

1½ M

		Waveform:(Input is 01101) 	**1½ M**
Q.4		**Attempt any THREE of the following :**	**12-Total Marks**
	a)	**Design a full Adder using Truth Table and K-map.**	**4M**
	Ans:	A full adder is a combinational logic circuit that performs addition between three bits, the two input bits A and B, and carry C from the previous bit.  Truth Table:	**Truth table 1½ M**

Input			Output	
A	B	Cin	Sum	Carry
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

K-map simplification for carry and sum

For Carry (C_{out})

A \ BC_{in}	00	01	11	10
0	0	0	1	0
1	0	1	1	1

For Sum

A \ BC_{in}	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$C_{out} = AB + A C_{in} + B C_{in}$

$Sum = \bar{A}\,\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\,\bar{C}_{in} + ABC_{in}$

1M

Logical diagram:

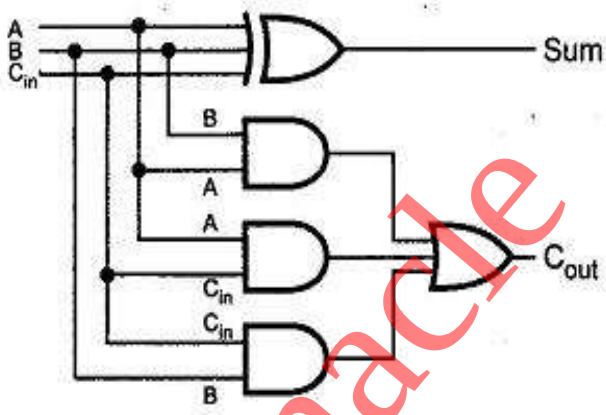


Fig. 3.17 Implementation of full-adder

1½ M

b) **Describe the working of ring counter using D flip flop with diagram and waveforms.** **4M**

Ans:

Diagram:

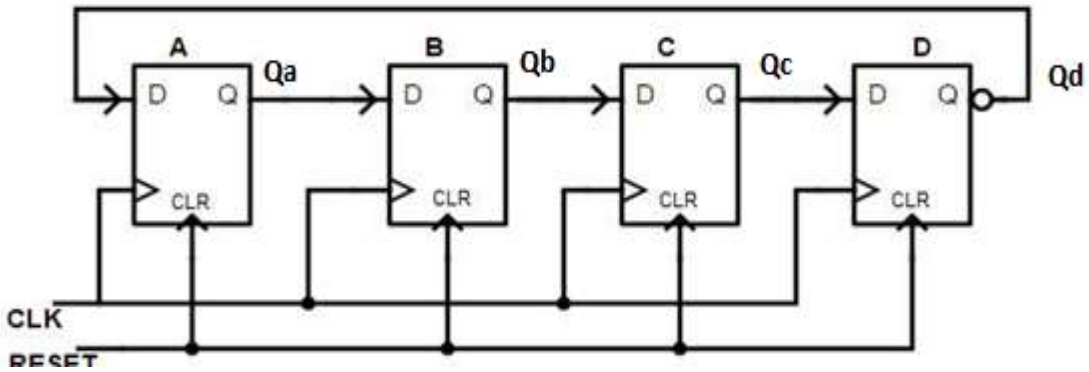
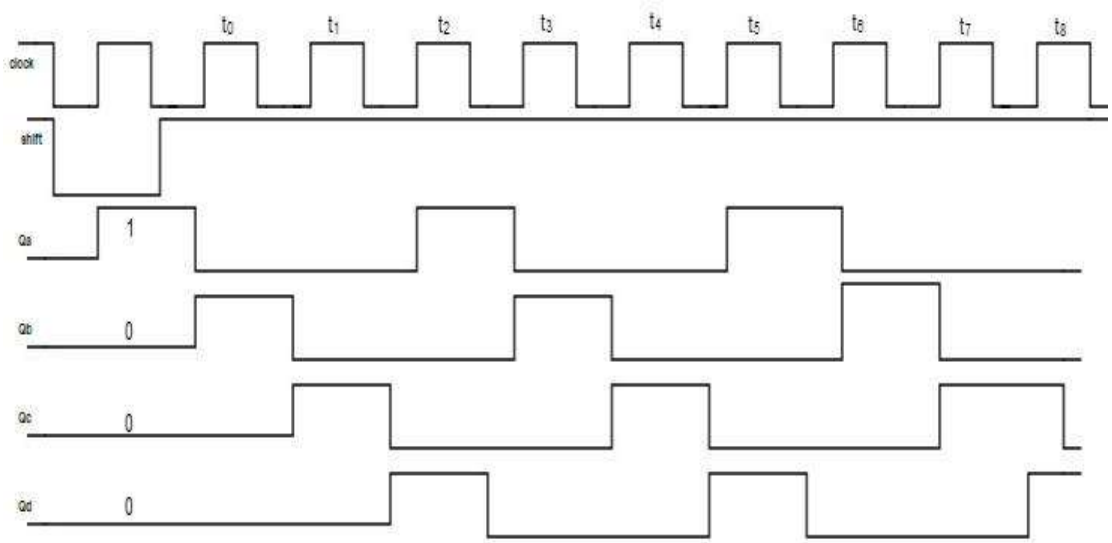


Diagram:1 ½ M

Waveforms:



Working:

The ring counter is a cascaded connection of flip flops, in which the output of last flip flop is connected to input of first flip flop. In ring counter if the output of any stage is 1, then its reminder is 0. The Ring counters transfers the same output throughout the circuit.

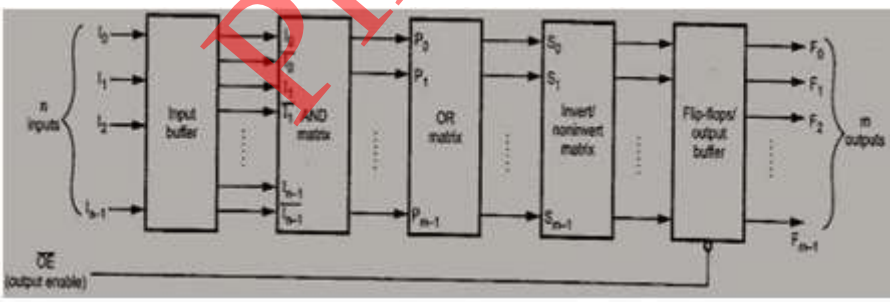
That means if the output of the first flip flop is 1, then this is transferred to its next stage i.e. 2nd flip flop. By transferring the output to its next stage, the output of first flip flop becomes 0. And this process continues for all the stages of a ring counter. If we use n flip flops in the ring counter, the '1' is circulated for every n clock cycles.

Waveform :1½ M

Explaination:1 M

c) **Draw block diagram of programmable logic Array.** **4M**

Ans: Diagram: **4M**



d) **Compare the following:**
(i) Volatile with Non Volatile.
(ii) EPROM with EEPROM. **4M**

Ans: (i)Volatile with Non Volatile. **2M (Any two point 1 M each)**

Parameter	Volatile memory	Non-Volatile memory
definition	Memory required electrical power to keep information stored is called volatile memory	Memory that will keep storing its information without the need of electrical power is called nonvolatile memory.
classification	All RAMs	ROMs, EPROM, magnetic memories

Effect of power	Stored information is retained only as long as power is on.	No effect of power on stored information
applications	For temporary storage	For permanent storage of information

ii)EPROM with EEPROM.

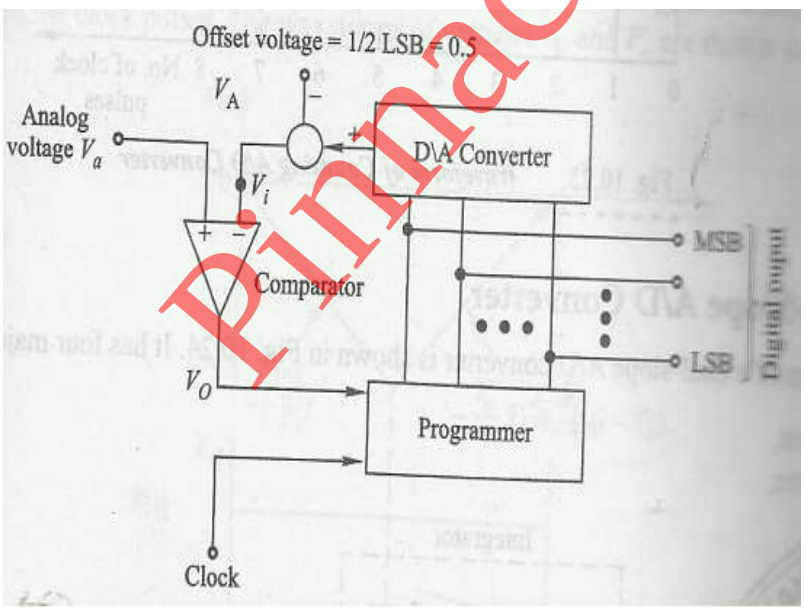
Parameter	EPROM	EEPROM.
Stands for	Erasable Programable Read-Only Memory.	Electrically Erasable Programmable Read-Only Memory.
Basic	Ultraviolet Light is used to erase the content of EPROM.	EEPROM contents are erased using electrical signal.
Appearance	EPROM has a transparent quartz crystal window at the top.	EEPROM are totally encased in an opaque plastic case.
Technology	EPROM is modern version of PROM.	EEPROM is the modern version of EPROM.

1M(Any two point each)

e) Describe the working principal of successive approximation ADC. **4M**

Ans: **Note:** Other relevant diagram and explanation also can be considered.

Diagram:



2M

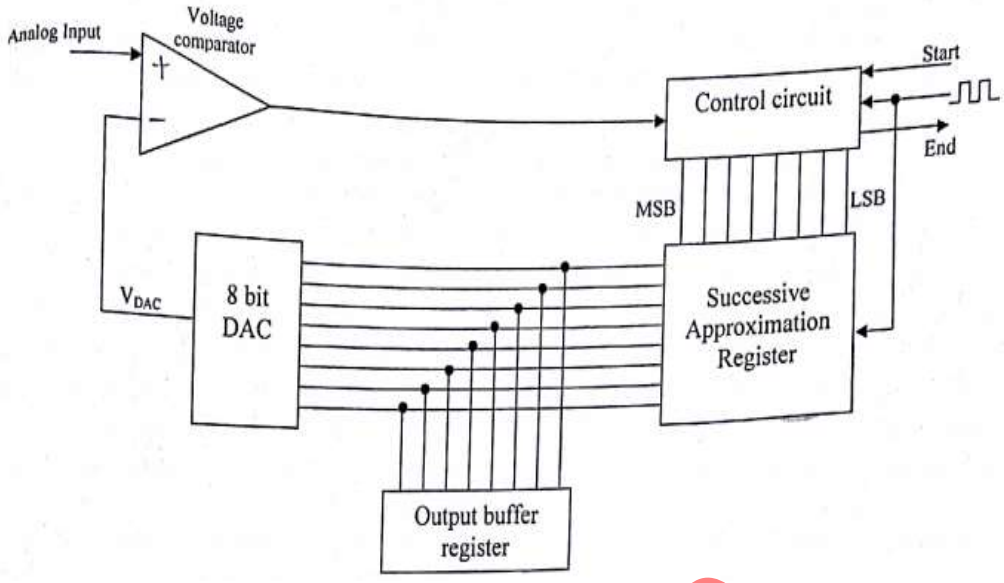
Working:

The successive approximation A/D converter is as shown in fig. An analog voltage (Va) is constantly compared with voltage Vi, using a comparator. The output produced by comparator (Vo) is applied to an electronic Programmer. If Va=Vi, then Vo=0 & then no conversion is required. The programmer displays the value of Vi in the form of digital O/P. But if Va Vi, then the O/P is changed by the programmer. If Va> Vi, then value of Vi is increased by 50% of earlier value. But if Va< Vi, then value of Vi is decreased by 50% of earlier value.

This new value is converted into analog form, by D/A converter so as to compare it with Va again. This procedure is repeated till we get Va=Vi. As the value of Vi is changed successively, this method is called as successive-approximation A/D converter.

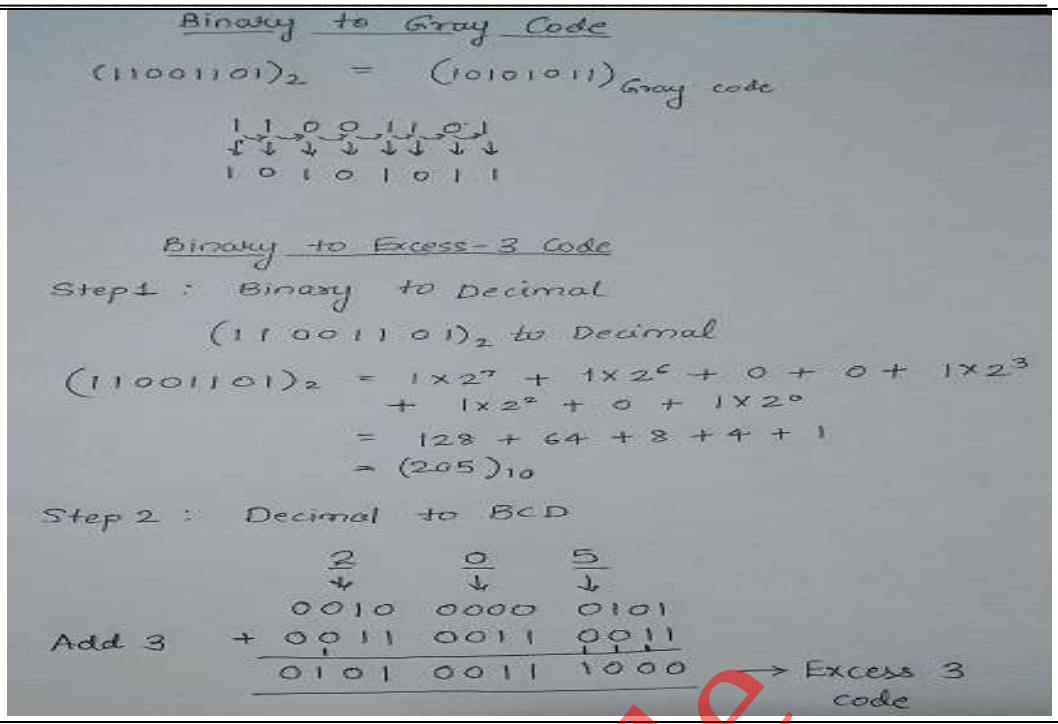
2M

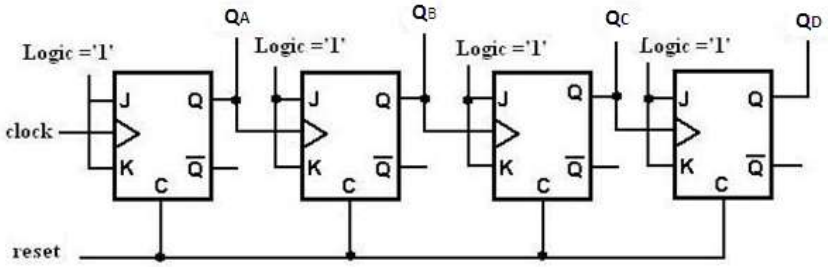
OR



When the starts signal goes low the successive approximation register SAR is cleared and output voltage of DAC will be 0v. When start goes high the conversion starts.
After starts, during first clock pulse the control circuit set MSB bit so SAR output wiil be 1000 0000. This is connected as input to DAC so output of DAC is compared with Vin input voltage. If V_{DAC} is more than Vin the comparator output –Vsat, if V_{DAC} is less than Vin, the comparator output is +Vsat.
If output of DAC i.e. V_{DAC} is +Vsat (i.e. unknown analog input voltage Vin> V_{DAC}) then MSB bit is kept set, otherwise it is reset.
Consider MSB is set so SAR will contain 1000 0000.
The next clock pulse will set next bit i.e. D6 bit is kept as it is, but if it –Vsat the D6 bit reset. The process of checking and taking decision to keep bit set or to reset is continued upto D0. Then the DAC input will be digital data equal to analog input.
When the conversion is finished the control circuits sends out an end of conversion signal and data is locked in buffer register.

Q.5		**Attempt any <u>TWO</u> of the following :**	**12- M**
	(a)	**(i)Convert the following binary number $(11001101)_2$ into Gray Code and Excess-3 Code.**	**2M**

	Ans:		**1M each conversion**
		(ii)Perform the BCD Addition. **$(17)_{10} + (57)_{10}$**	**2M**
	Ans:	$(17)_{10}$ 0001 0111 $(57)_{10}$ + 0101 0111 -------(1/2 M) 0110 1110 Valid Invalid BCD BCD ----------(1/2 M) ADD 0110 TO Invalid BCD 1 11 0110 1110 + 0000 0110 01110100 ----------(1/2 M) 7 4 $= (74)_{10}$ ----------(1/2 M)	**½ Each step**
		(iii)Perform the binary addition. **$(10110 \bullet 110)_2 + (1001 \bullet 10)_2$**	**2M**
	Ans:	$10110.110)_2 - (1001.10)_2 = (100000.010)_2$ 11111 10110.110 + 1001.10 100000.010	**2M**
	(b)	**Design a 4bit ripple counter using JK flip flop, with truth table and waveforms.**	**6M**
	Ans:	Circuit Diagram:	**2M**

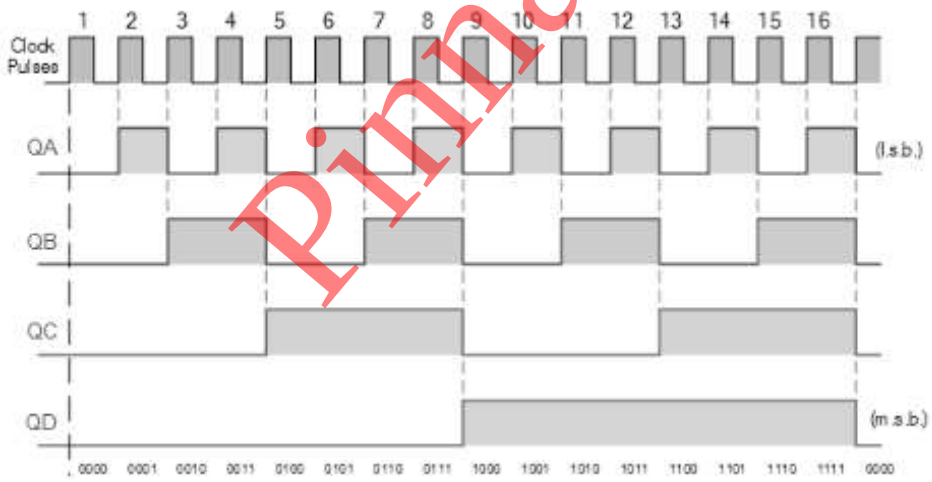


Truth Table:

State	Q_D	Q_C	Q_B	Q_A
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	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0

2M

Timing Diagram / Waveforms:



2M

(c) **Calculate the analog output for 4 bit weighted register type DAC for inputs**
(i) 1011
(ii) 1001
Assume (V_{fs}) full scale range of voltage is 5V — **6M**

Ans: — **3M each**

Given:

VR = Vfs = 5V

Formula Used:

$Vo = - VR (B1.2^{-1} + B2.2^{-2} + B3.2^{-3} + B4.2^{-4})$

1. 1011

$Vo = - VR (B1.2^{-1} + B2.2^{-2} + B3.2^{-3} + B4.2^{-4})$

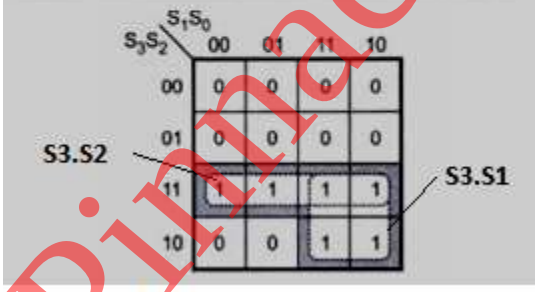
		= - 5 (1*1/2 + 0 + 1*1/2^3 + 1 *1/2^4) = - 5 (1*1/2 + 1*1/8 + 1 *1/16) = - 5 (0.5 + 0.125 + 0.0625) = 3.4375V Vo = 3.4375 V 2. 1001 Vo = - VR (B1.2^{-1} + B2.2^{-2} + B3.2^{-3} + B4.2^{-4}) = - 10 (1*1/2 + 0 + 0 + 1 *1/2^4) = - 10 (1*1/2 + 0 + 0 + 1 *1/16) = - 10 (0.5 + 0.0625) = 2.8125V Vo = 2.8125 V	
Q.6		**Attempt any TWO of the following:**	**12-Total Marks**
	(a)	**Compare TTL, CMOS and ECL logic family on the following points.** **(i) Basic Gates** **(ii) Propogation dealy** **(iii)Fan out** **(iv)Power Dissipation** **(v) Noise immunity** **(vi)Speed power product**	**6M**
	Ans:		**1M Each parameter**

Parameter	TTL	CMOS	ECL
Basic gates	NAND	NOR/NAND	OR/NOR
Propagation delay	10	70-105	2
Fan out	10	50	25
Power Dissipation	10mW	1.01mW	40-55mW
Noise Immunity	0.2V	5V	0.25V
Speed Power Product	100	0.7	100

	(b)	**Design a BCD adder using IC 7483.**	**6M**
	Ans:	**(Note: Labeled combinational circuit can be drawn using universal gate also)** 1) To implement BCD adder we require: • 4-bit binary adder for initial addition • Logic circuit to detect sum greater than 9 • One more 4-bit adder to add 0110201102 in the sum if sum is greater than 9 or carry is 1 2) The logic circuit to detect sum greater than 9 can be determined by simplifying the	

Boolean expression of given truth Table.

Inputs				Output
S_3	S_2	S_1	S_0	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



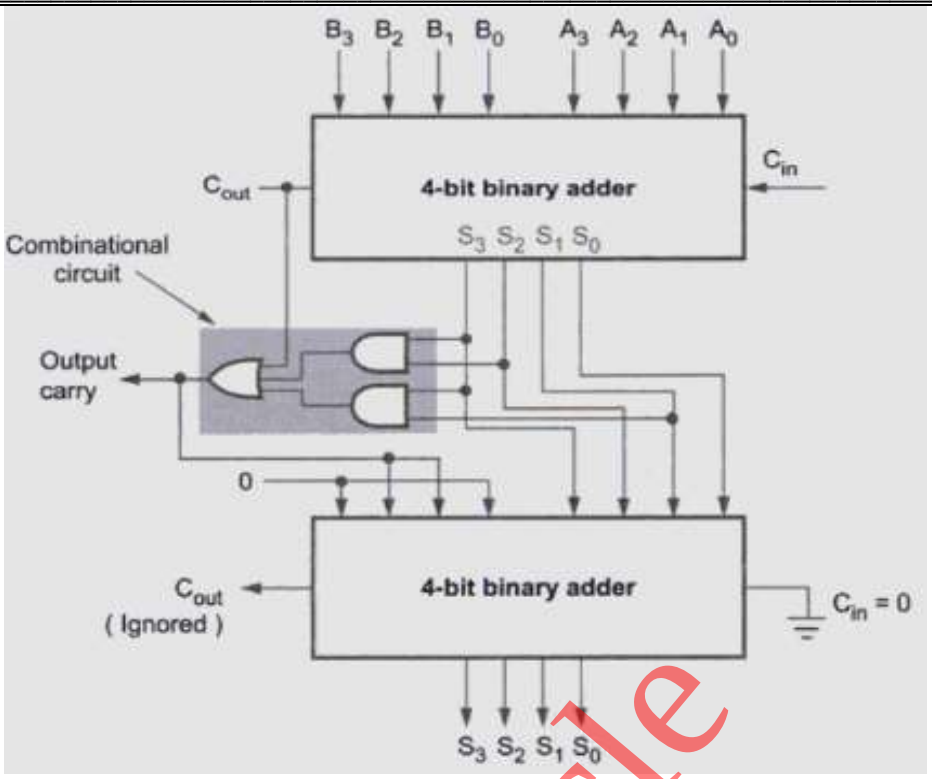
= S3.S2+S3.S1

3) Y=1 indicates sum is greater than 9. We can put one more term, C_out in the above expression to check whether carry is one.
4) If any one condition is satisfied we add 6(0110) in the sum.
5) With this design information we can draw the block diagram of BCD adder, as shown in figure below.

Truth Table: 2M

K-Map: 1M

Circuit Diagram: 3M



(c) **Design a 3 bit synchronous counter using JK Flip Flop.** **6M**

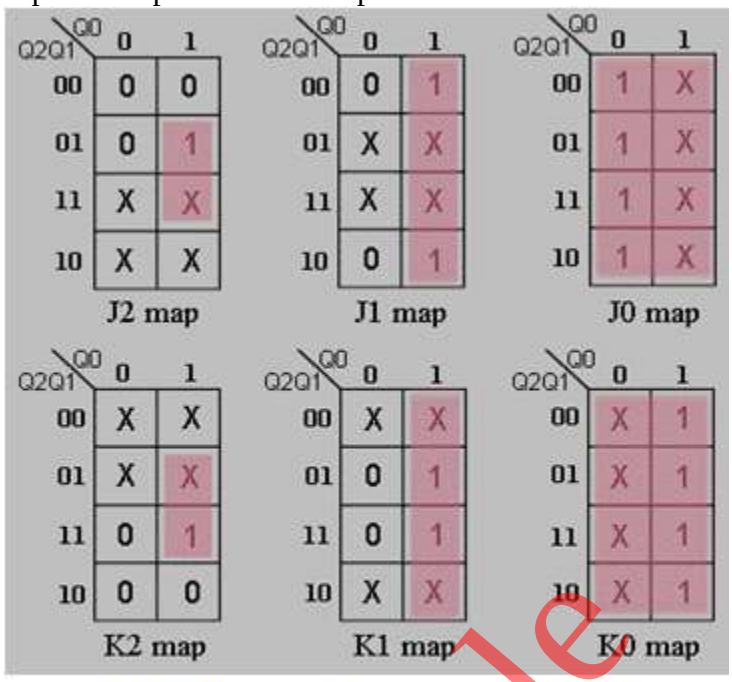
Ans: **2M**

1) Step1:
Construct JK state table with corresponding excitation table:

Output State Transitions				
Present State Q2 Q1 Q0	Next state Q2 Q1 Q0	Flip-flop inputs J2 K2	J1 K1	J0 K0
0 0 0	0 0 1	0 X	0 X	1 X
0 0 1	0 1 0	0 X	1 X	X 1
0 1 0	0 1 1	0 X	X 0	1 X
0 1 1	1 0 0	1 X	X 1	X 1
1 0 0	1 0 1	X 0	0 X	1 X
1 0 1	1 1 0	X 0	1 X	X 1
1 1 0	1 1 1	X 0	X 0	1 X
1 1 1	0 0 0	X 1	X 1	X 1

State Table and Corresponding Excitation Table (d=don't care)

Step 2:
Build Karnaugh Map or Kmap for each JK inputs:

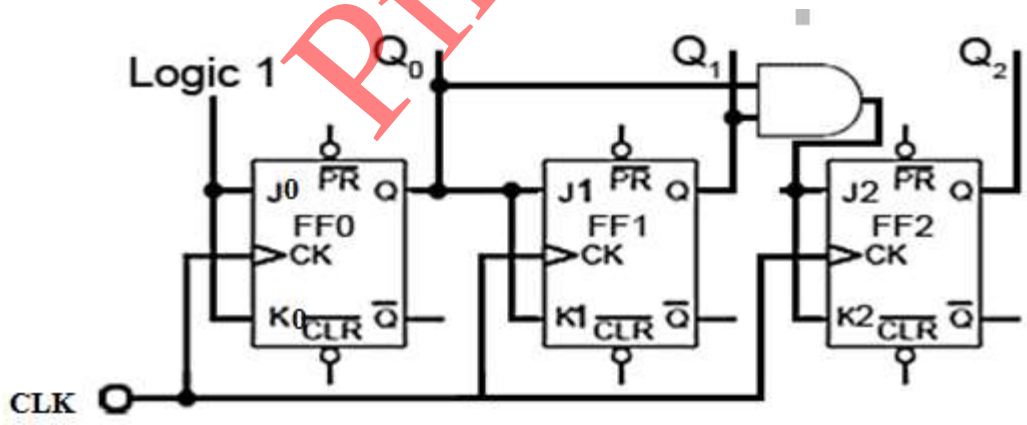


2M

J2=Q1.Q0 **J1=Q0** **J0=1**

K2=Q1.Q0 **K1=Q0** **K=1**

Step3:
Draw the complete design as below:



2M

Note: It can also be designed using any other Flip Flop.