KLS Vishwanathrao Deshpande Institute of Technology

(Accredited by NAAC with "A" Grade)

(Approved by AICTE, New Delhi, Affiliated to VTU, Belagavi)

(Recognized Under Section 2(f) by UGC, New Delhi)

Udyog Vidya Nagar, Haliyal - 581 329, Dist.: Uttara Kannada

www.klsvdit.edu.in | principal@klsvdit.edu.in | hodece@klsvdit.edu.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

University / Model Question Paper Scheme & Solution

Faculty Name	:	Mr. A. V. Kolaki
Course Name	:	Digital System Design using Verilog
Course Code	:	BEC302
Year of Question Paper	:	Dec 2024 / Jan 2025
Date of Submission	:	21/07/2025

Faculty Member

HoD

21.07.2025

Head of the Department
Dept. of Electronic & Communication Engg.
KLS V.D.I.T., HALIYAL (U.K.)

Dean (Acad.)

CBCS SCHEME

USN | | | | | | | | | | |

BEC302

Third Semester B.E./B.Tech. Degree Examination, Dec.2024/Jan.2025
Digital System Design using Verilog

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Design a combinational logic truth table so that an output is generated indicating when a majority of four inputs is tree.	4	L3	CO1
	b.	Find the prime implicants and the essential prime implicants of the following Boolean functions using Karnaugh maps. i) $f(a, b, c, d) = \Sigma(1, 5, 6, 7, 11, 12, 13, 15)$ ii) $f(a, b, c, d) = \sum(0, 1, 4, 5, 9, 11, 13, 15)$	8	L4	CO1
	c.	Simplify the given boolean function using Quine McCluskey minimization technique for the function $O = f(a, b, c, d) = \sum(0, 1, 2, 3, 6, 7, 8, 9, 14, 15)$	8	L3	CO1
		OR			
Q.2	a.	Place the following equations into the proper canonical form: i) $P = f(a, b, c) = ab' + ac' + bc$ ii) $G = f(w, x, y, z) = w'x + yz'$	4	L3	CO1
	b.	Find the minimal sum and minimal product for the following Boolean functions using Karnaugh maps i) $f(a,b,c,d) = \bar{a}\bar{b}d + bcd + a\bar{b}d + b\bar{c}\bar{d}$ ii) $f(a,b,c,d) = (a+\bar{b})(a+c+d)(\bar{a}+\bar{b}+\bar{d})(a+\bar{c}+d)$	8	L4	CO1
	c.	Simplify the given boolean function using quine. McCluskey minimization technique for the function. $s = f(a,b,c,d) = \Sigma(1,3,13,15) + \Sigma d(8,9,10,11)$	8	L3	CO1
		Module – 2			
Q.3	a.	Design and explain binary full adder with block diagram, Karnaugh map and logic circuit.	10	L3	CO2
	b.	Define decoder, write the symbol, truth table and logic circuit for 3:8 line decoder using minterm generator.	10	L2	CO2
		OR			
Q.4	a.	Define multiplexer, write the symbol, truthtable and logic circuit for 4:1 multiplexer using enable input.	10	L2	CO2
	b.	Realize the Boolean function $f(w, x, y, z) = \sum(0, 1, 5, 6, 7, 9, 12, 15)$ i) Using 8:1 MUX ii) Using 4:1 MUX	10	L2	CO2

		Module – 3			
Q.5	**a.**	Develop the characteristic equation for i) SR flip flop ii) JK flip flop iii) D flip flop iv) T flip flop.	**10**	**L3**	**CO3**
	b.	Explain serial in, parallel at unidirectional shift register and parallel in serious out unidirectional shift register.	**10**	**L2**	**CO3**
		OR			
Q.6	**a.**	Explain Mod-4 ring counter and Mod-8 twisted ring counter with logic diagram and counting sequence.	**10**	**L2**	**CO3**
	b.	Design a synchronous Mod-6 counter using clocked D-flip flop.	**10**	**L3**	**CO3**
		Module – 4			
Q.7	**a.**	Explain logical operators and relational operators used in verilog.	**8**	**L2**	**CO4**
	b.	Illustrate i) NETS ii) Register iii) Vector iv) integer data types with an example.	**8**	**L2**	**CO4**
	c.	Write a verilog code for full adder using data flow description style.	**4**	**L2**	**CO4**
		OR			
Q.8	**a.**	Illustrate the structure of behavioural description with an example using half adder.	**8**	**L2**	**CO4**
	b.	Illustrate the structure of verilog module with an example using half subtractor.	**8**	**L2**	**CO4**
	c.	Write a verilog code for binary to gray using behavioural description style.	**4**	**L2**	**CO4**
		Module – 5			
Q.9	**a.**	Write the syntax of IF and EISE-IF with an example.	**8**	**L2**	**CO4**
	b.	Write logic symbol, flowchart and program for D-latch using behavioural description style.	**8**	**L2**	**CO4**
	c.	Write a verilog code for 8:1 MUX using behavioural description style.	**4**	**L2**	**CO4**
		OR			
Q.10	**a.**	Explain the structure of structural model with built in gates using example of half adder. Also mention an primitive built in gates.	**8**	**L2**	**CO4**
	b.	Write a verilog code of a 3-bit ripple carry adder using structural description model.	**8**	**L2**	**CO4**
	c.	Write a verilog code of SR flip flop using behavioural description style.	**4**	**L2**	**CO4**

* * * * *

Q.No.1a. Design a combinational logic truth table so that an output is generated indicating when a majority of four inputs true

Ans.

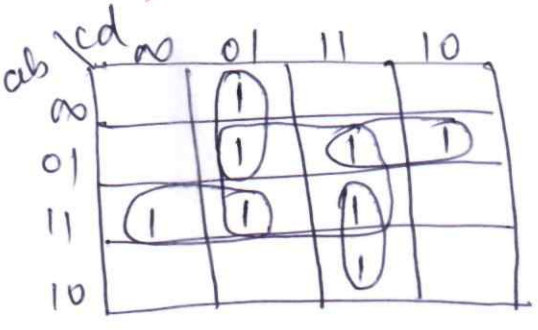
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

1b. Find the prime implicants & the essential prime implicants of the following Boolean funs using K-maps.

i) $f(a,b,c,d) = \Sigma(1,5,6,7,11,12,13,15)$

ii) $f(a,b,c,d) = \Sigma(0,1,4,5,9,11,13,15)$

Ans.



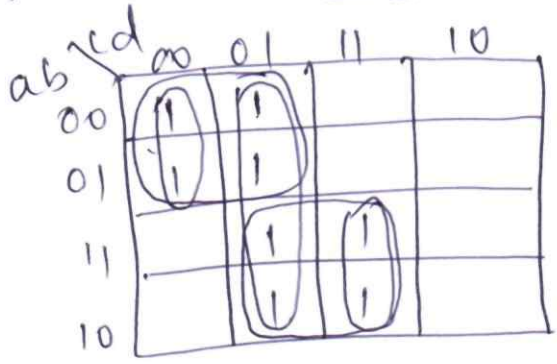
Prime Implicants

bd
$\bar{a}bc$
acd
$ab\bar{c}$
$\bar{a}\bar{c}d$

Essential Prime Implicants

$\bar{a}bc$
acd
$ab\bar{c}$
$\bar{a}\bar{c}d$

ii) $f(a,b,c,d) = \sum(0,1,4,5,9,11,13,15)$.



Prime Implicants	Essential Prime Implicants
ad	ad
$\bar{a}\bar{c}$	$\bar{a}\bar{c}$
$\bar{c}d$	
acd	
$\bar{a}\bar{c}\bar{d}$	

1.c. Simplify the given boolean fun using Quine McCluskey minimization technique for the fun

$$0 = f(a,b,c,d) = \sum(0,1,2,3,6,7,8,9,14,15)$$

Ans:

Group	Minterm	Variables a b c d
0	0	0 0 0 0 ✓
1	1	0 0 0 1 ✓
	2	0 0 1 0 ✓
	8	1 0 0 0 ✓
2	3	0 0 1 1 ✓
	6	0 1 1 0 ✓
	9	1 0 0 1 ✓
3	7	0 1 1 1 ✓
	14	1 1 1 0 ✓
4	15	1 1 1 1 ✓

Group	Minterm	Variables a b c d
0	(0,1)	0 0 0 – ✓
	(0,2)	0 0 – 0 ✓
	(0,8)	– 0 0 0 ✓
1	(1,3)	0 0 – 1 ✓
	(1,9)	– 0 0 1 ✓
	(2,3)	0 0 1 – ✓
	(2,6)	0 – 1 0 ✓
	(8,9)	1 0 0 – ✓
2	(3,7)	0 – 1 1 ✓
	(6,7)	0 1 1 – ✓
	(6,14)	– 1 1 0 ✓
3	(7,15)	– 1 1 1 ✓
	(14,15)	1 1 1 – ✓

Group	Minterm	Variables a b c d
0	(0,1,2,3)	0 0 – –
	(0,1,8,9)	– 0 0 –
1	(2,3,6,7)	0 – 1 –
2	(6,7,14,15)	– 1 1 –

Group	Minterm	Variables a b c d

PI | Minterm | 0 | 1 | 2 | 3 | 6 |

P2	minterm abcd	0	1	2	3	6	7	8	9	14	15
$\bar{a}\bar{b}$	(0,1,2,3)	x	x	(x)	(x)						
$\bar{b}\bar{c}$	(0,1,8,9)	x	x					(x)	(x)		
$\bar{a}c$	(2,3,6,7)			x	x	x	x				
bc	(6,7,14,15)					x	x			(x)	(x)

$\therefore f = \bar{a}\bar{b} + \bar{b}\bar{c} + bc$

2a) Place the following equations in to the proper canonical form.

i) $P = f(a,b,c) = a\bar{b} + a\bar{c} + bc$

Ans: $= a\bar{b}(c+\bar{c}) + a\bar{c}(b+\bar{b}) + bc(a+\bar{a})$

$= a\bar{b}c + ab\bar{c} + a\bar{b}\bar{c} + \bar{a}bc + abc$

ii) $q = f(w,x,y,z) = \bar{w}x + y\bar{z}$

$= \bar{w}x(y+\bar{y})(z+\bar{z}) + y\bar{z}(x+\bar{x})(w+\bar{w})$

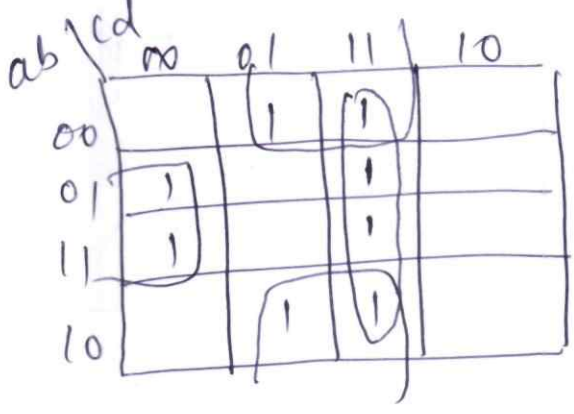
$= \bar{w}xyz + \bar{w}xy\bar{z} + \bar{w}x\bar{y}z + \bar{w}x\bar{y}\bar{z} + wxy\bar{z} + w\bar{x}y\bar{z} + \bar{w}\bar{x}y\bar{z}$.

2b) Find the minimal sum & minimal product for the following boolean fns using K-maps

i) $f(a,b,c,d) = \bar{a}\bar{b}d + bcd + a\bar{b}d + b\bar{c}\bar{d}$

Canonical form = $\bar{a}\bar{b}cd + \bar{a}\bar{b}\bar{c}d + abcd + \bar{a}bcd + a\bar{b}cd + a\bar{b}\bar{c}d + ab\bar{c}\bar{d} + \bar{a}b\bar{c}\bar{d}$

$\therefore f = \Sigma(1,3,4,7,9,11,12,15)$

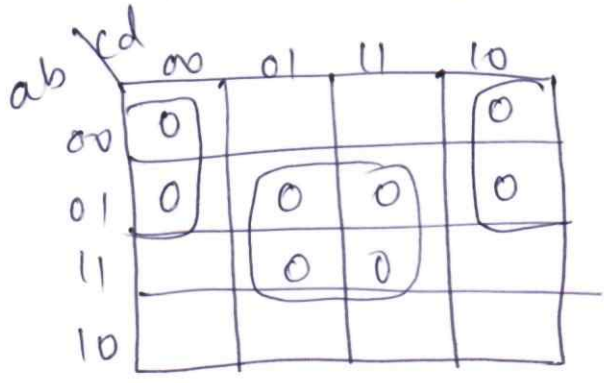


$f = cd + \bar{b}d + b\bar{c}\bar{d}$

ii) $f(a,b,c,d) = (a+\bar{b})(a+c+d)(\bar{a}+\bar{b}+\bar{d})(a+\bar{c}+d)$

Canonical form: $(a+\bar{b}+c+d)(a+\bar{b}+c+\bar{d})(a+b+c+d)(\bar{a}+\bar{b}+c+\bar{d})$
$(\bar{a}+\bar{b}+\bar{c}+\bar{d})(a+b+\bar{c}+d)(a+\bar{b}+\bar{c}+d)$
$(a+\bar{b}+\bar{c}+\bar{d})($

$f(a,b,c,d) = \Sigma(0,2,4,5,6,7,13,15)$



$f = (a+d)(\bar{b}+\bar{d})$.

2C) Simplify the given boolean function using Quine Mcclusky minimization technique for the function

$S = f(a,b,c,d) = \Sigma(1,3,13,15) + \Sigma d(8,9,10,11)$.

Ans

Group	Minterm	Variable a b c d
0	1	0 0 0 1 ✓
	8*	1 0 0 0 ✓
1	3	0 0 1 1 ✓
	9*	1 0 0 1 ✓
	10*	1 0 1 0 ✓
2	11*	1 0 1 1 ✓
	13	1 1 0 1 ✓
3	15	1 1 1 1 ✓

Group	Minterm	Variable a b c d
0	(1, 3)	0 0 - 1 ✓
	(1, 9*)	- 0 0 1 ✓
	(8*, 9*)	1 0 0 - ✓
	(8*, 10*)	1 0 - 0 ✓
1	(3, 11*)	- 0 1 1 ✓
	(9*, 11*)	1 0 - 1 ✓
	(9*, 13)	1 - 0 1 ✓
	(10*, 11*)	1 0 1 - ✓
2	(11*, 15)	1 - 1 1 ✓
	(13, 15)	1 1 - 1 ✓

Group	Minterm	Variable a b c d
	(1, 3, 9, 11)	- 0 - 1
	(8, 9, 10, 11)	1 0 - -
	(9, 11, 13, 15)	1 - - 1

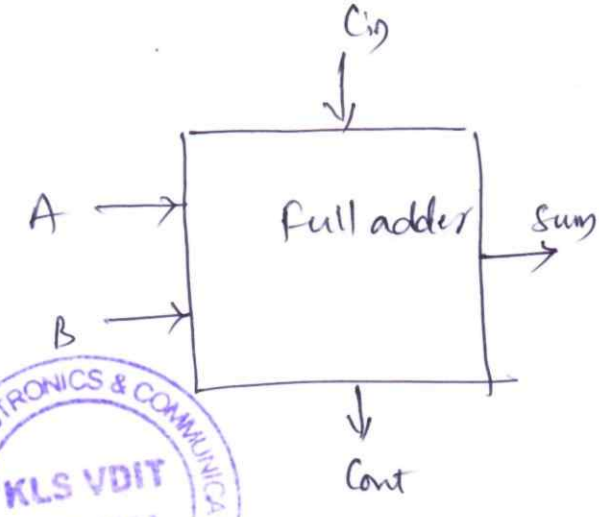
PI	minterms	1	3	8*	9*	10*	11*	13	15
$\bar{b}d$	(1,3,9,11)	⊗	⊗		×		×		
$a\bar{b}$	(8,9,10,11)			×	×	×	×		
ad	(9,11,13,15)				×		×	⊗	⊗

$$\therefore f(a,b,c,d) = \bar{b}d + ad$$

3a. Design & explain binary full adder with blocks diagram Karnaugh map & logic circuit.

Ans Full add blocks diagram

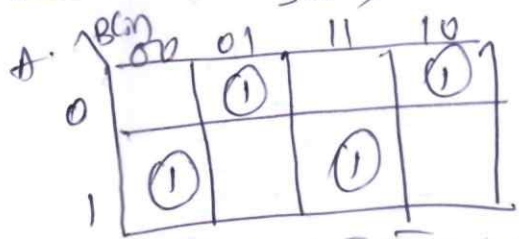
Inputs			outputs	
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



A full adder is a combinational ckt that forms the arithmetic operation of three bits. It consists of 3 i/p's & 2 o/p's. The 2 i/p variables denoted by A, B represent two significant bits. The third i/p Cin represents the carry from the previous lower significant bit.

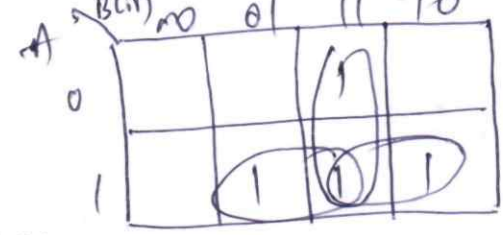
K-map for Sum & Carry

Sum



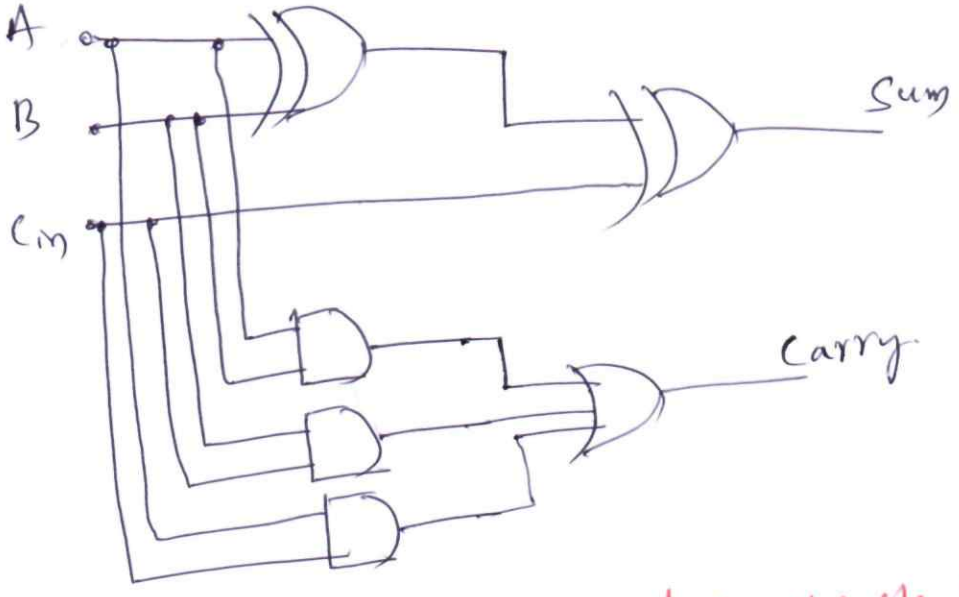
Sum = $\bar{A}BC_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

Carry



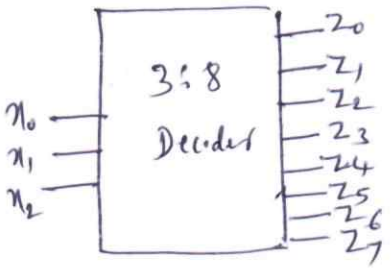
Carry = $AB + AC_{in} + BC_{in}$

Sum = $A \oplus B \oplus C_{in}$ Carry = $AB + BC_{in} + AC_{in}$

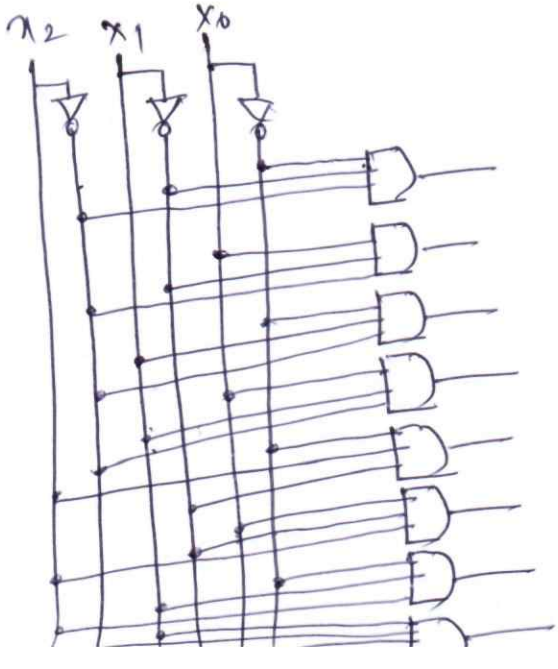


3b. Define decoder, write the symbol, truthtable, & logic ckt for 3:8 line decoder using minterm generation.

Ans: Decoder is a combinational ckt with 'n' i/p & 2^n output $n:2^n$ where codewords are converted in to symbols



Inputs x_2 x_1 x_0	Outputs z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7
0 0 0	1	0	0	0	0	0	0	0
0 0 1	0	1	0	0	0	0	0	0
0 1 0	0	0	1	0	0	0	0	0
0 1 1	0	0	0	1	0	0	0	0
1 0 0	0	0	0	0	1	0	0	0
1 0 1	0	0	0	0	0	1	0	0
1 1 0	0	0	0	0	0	0	1	0
1 1 1	0	0	0	0	0	0	0	1



$z_0 = \bar{x}_2 \bar{x}_1 \bar{x}_0$

$z_1 = \bar{x}_2 \bar{x}_1 x_0$

$z_2 = \bar{x}_2 x_1 \bar{x}_0$

$z_3 = \bar{x}_2 x_1 x_0$

$z_4 = x_2 \bar{x}_1 \bar{x}_0$

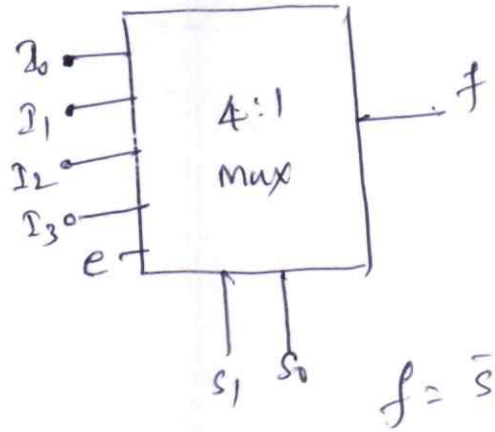
$z_5 = x_2 \bar{x}_1 x_0$

$z_6 = x_2 x_1 \bar{x}_0$

$z_7 = x_2 x_1 x_0$

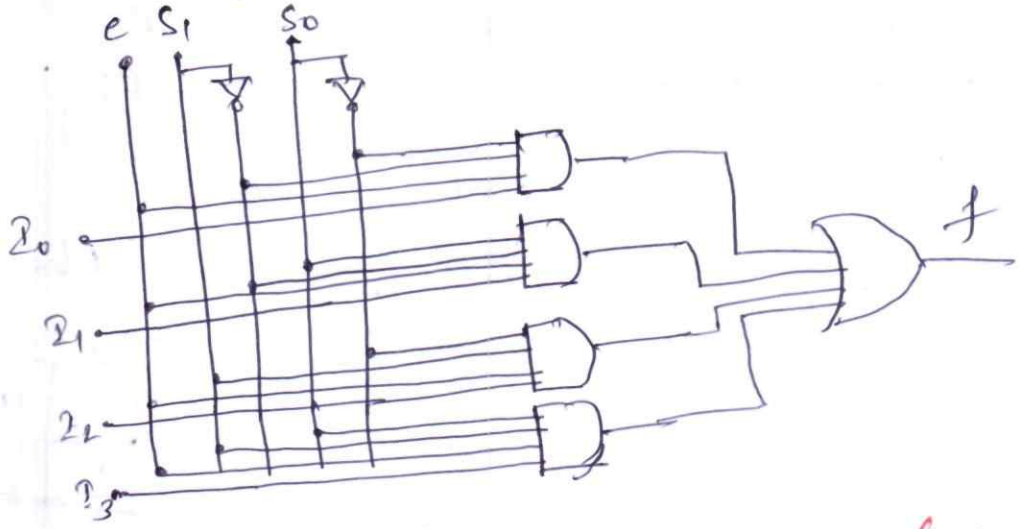
4a) Define multiplexer write the symbol, truth table & logic ckt for 4:1 multiplexer using enable i/p.

Ans: Multiplexer is a combinational logic ckt with 2^n i/ps & 'n' select lines, one output



e	S_1	S_0	f
0	x	x	0
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3

$f = \bar{S}_1 \bar{S}_0 I_0 e + \bar{S}_1 S_0 I_1 e + S_1 \bar{S}_0 I_2 e + S_1 S_0 I_3 e$

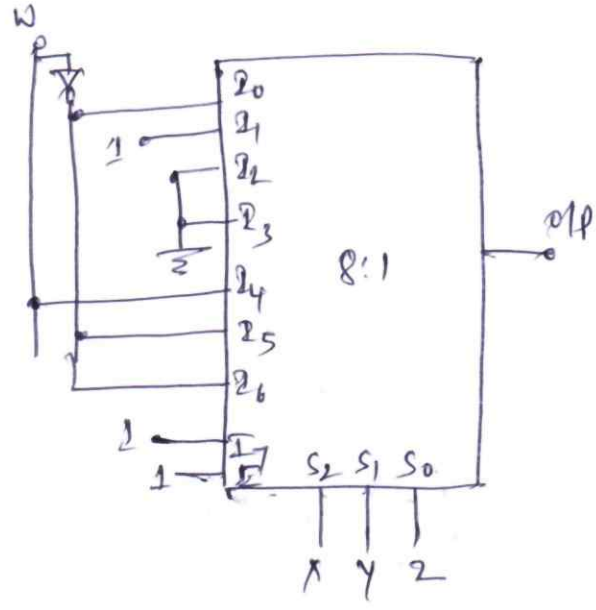


4b. Realize the boolean function $f(w, x, y, z) = \sum m(0,1,5,6,7,9,12,15)$

i) using 8:1 mux ii) using 4:1 mux

Ans: i)

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\bar{w}$	0	1	2	3	4	5	6	7
w	8	9	10	11	12	13	14	15
	$\bar{w}$	1	0	0	w	$\bar{w}$	$\bar{w}$	1



ii)

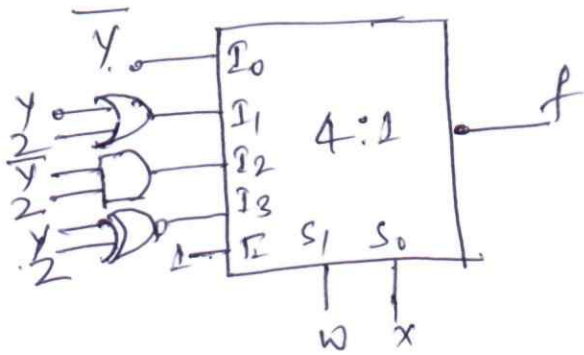
	$\bar{Y}\bar{Z}$	$\bar{Y}Z$	$Y\bar{Z}$	YZ
I_0	(0)	(1)	2	3
I_1	4	(5)	(6)	(7)
I_2	8	(9)	10	11
I_3	(12)	13	14	(15)

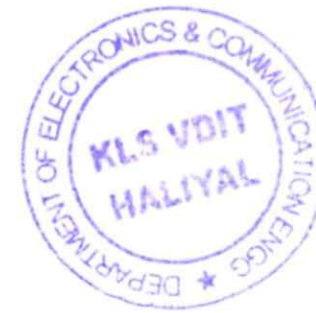
$\bar{Y}\bar{Z} + \bar{Y}Z = \bar{Y}$

$\bar{Y}Z + Y\bar{Z} + YZ = Y + \bar{Y}Z = Y + Z$

$\bar{Y}Z +$

$\bar{Y}\bar{Z} + YZ.$





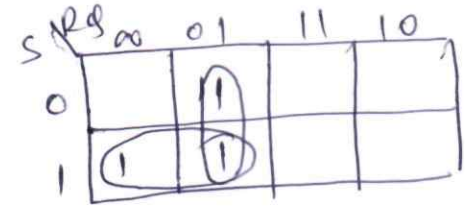
5a) Develop the characteristic equation for.
i) SR FF ii) JK FF iii) D. FF iv) T. FF

Ans: Flip-Flop function: FF Next State Table

S	R	Q^+
0	0	Q
0	1	0
1	0	1
1	1	x

S	R	Q	Q^+	
0	0	0	0	
0	0	1	1	
0	1	0	0	
0	1	1	0	
1	0	0	1	
1	0	1	1	
1	1	0	x	} DNS Not allowed
1	1	1	x	

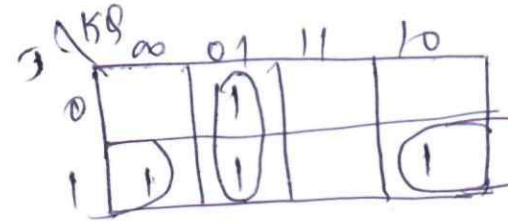
Characteristic Eqn



$Q^+ = S + \bar{R}Q$

J	K	Q^+
0	0	Q
0	1	0
1	0	1
1	1	$\bar{Q}$

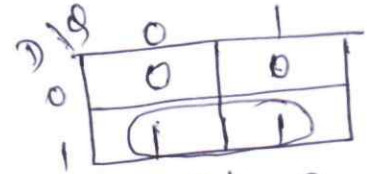
J	K	Q	Q^+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0



$Q^+ = J\bar{Q} + \bar{K}Q$

D	Q^+
0	0
1	1

D	Q	Q^+
0	0	0
0	1	0
1	0	1
1	1	1



$Q^+ = D.$

T	Q_{t+}
0	Q
1	$\bar{Q}$

T	Q	Q_{t+}
0	0	0
0	1	1
1	0	1
1	1	0

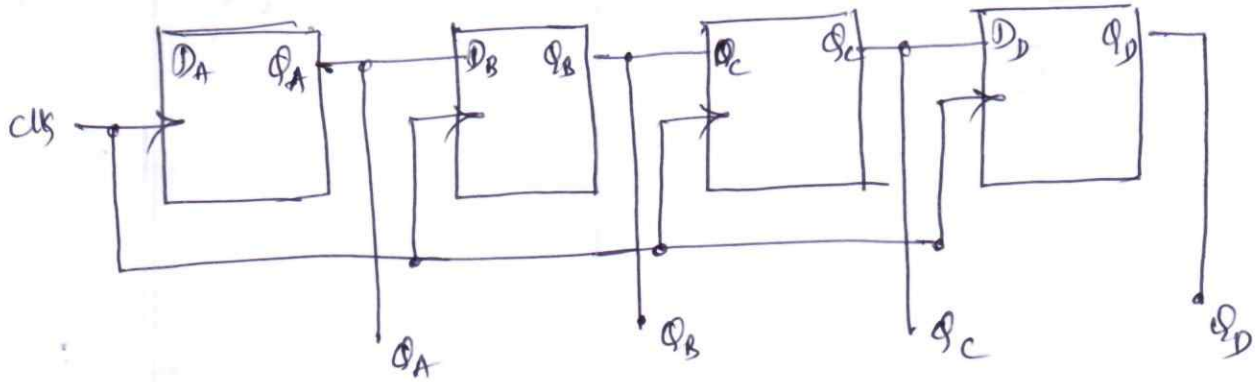
T \ Q	0	1
0		1
1	1	

$Q_{t+} = \bar{T}Q + T\bar{Q} = T \oplus Q$

6 (a) Explain mod-4 ring counter & mod-8 twisted ring counter with logic diagram & counting sequence

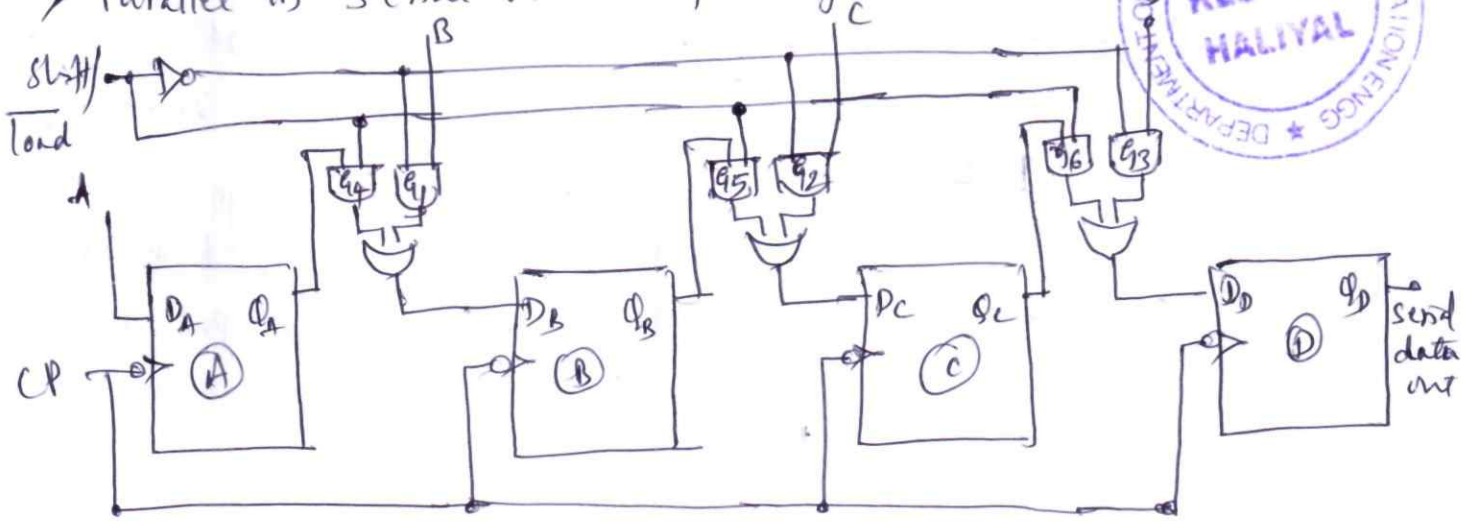
5 b) Explain serial in, Parallel & mod-8 twisted ring counter with logic diagram & at unidirectional shift register & Parallel in serial out unidirectional shift Register.

Ans. > Serial in Parallel out shift Register:-



The data bits are entered in to the register in the same manner as mentioned above fig. Serially & o/p is taken in Parallel. Once the data bits are entered in the shift Register & are stored. Each bit appears on its respective o/p line & all bits are available simultaneously.

> Parallel in Serial out shift Register:-

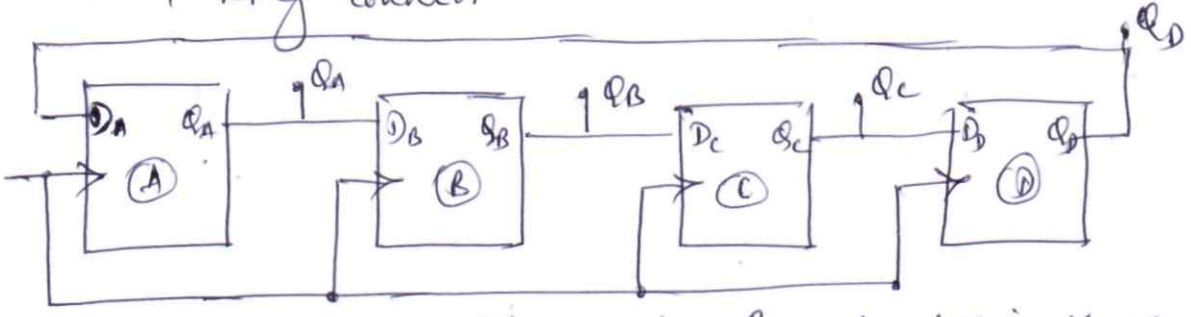


The 4-bit Parallel in serial out register as shown above fig The 4 i/p lines entering data parallel in to the register. Shift/$\overline{Load}$ is the control i/p which allows shift or loading data. When Shift/$\overline{Load}$ is low, gates G_1, G_2, G_3 are enabled allowing each i/p data bit to be applied to D i/p. of respective FF. When clock is applied the FF's with D=1 will set & those with D=0 will Reset & all 4 bits are stored simultaneously.

When Shift/$\overline{Load}$ is high gates G_1, G_2, G_3 are disabled & gates G_4, G_5, G_6 are enabled. This allows data bits to shift left from one stage to next. The OR gates at D i/p's of FF allows either the parallel data entry operation or shift operation depending on which AND gates are enabled by the level on the Shift/$\overline{Load}$ i/p.

6 a) Explain Mod-4 ring counter & Mod-8 twisted ring counter with logic diagram & Counting sequence

Ans: Mod-4 Ring Counter.

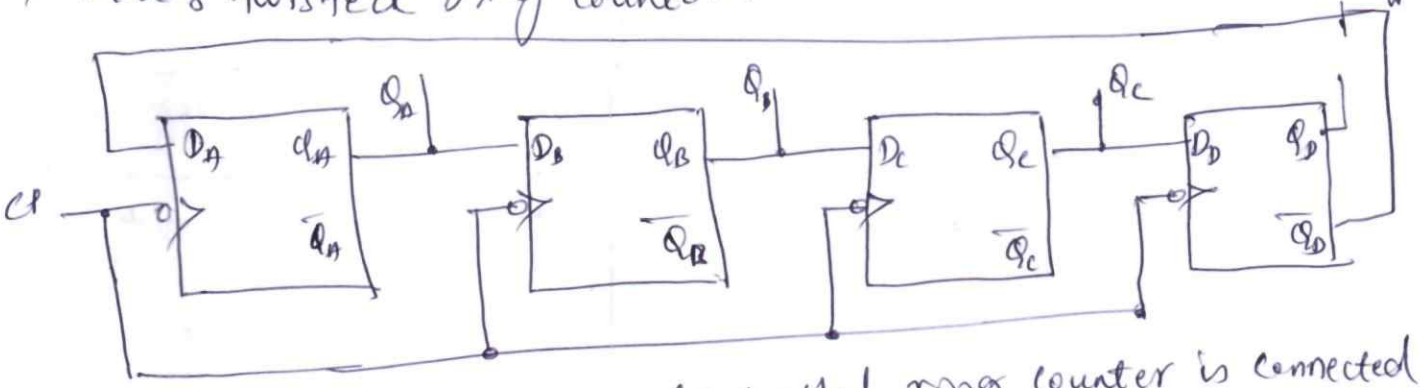


Counting Sequence

Q_A	Q_B	Q_C	Q_D
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

The Mod-4 Ring Counter is shown above. The Q o/p of each stage is connected to the D i/p of next stage & the o/p of last stage is fed back to the i/p of 1st stage. The 1st stage is preset to 1 & remaining o/p's are zero i.e $Q_A = 1$ & $Q_B = 0, Q_C = 0, Q_D = 0$. The first clk produces $Q_B = 1$ & remaining o/p's are zero. As the clk pulses applied at the i/p clk the above sequence states is produced.

7 Mod-8 twisted ring counter:-



Q_A	Q_B	Q_C	Q_D
0	0	0	0
1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	1
0	0	0	0

The Mod-8 twisted ring counter is connected as shown above. All FF's are cleared so all o/p's Q_A, Q_B, Q_C, Q_D are zero. The o/p of last stage Q_D is zero. Therefore complement o/p of last stage $\bar{Q}_D$ is one. This is fed back to the 'D' i/p of 1st stage, so D_A is one. The 1st falling clock edge produces $Q_A=1, Q_B=0, Q_C=0, Q_D=0$ since D_B, D_C, D_D are zero. The next pulse produces $Q_A=1, Q_B=1, Q_C=0, Q_D=0$. The sequence of states is repeated as mentioned above.

6 b.) Design synchronous Mod-6 counter using clocked D-FF.

Ans: Synchronous Mod-6 counter using D-FF:-

Appln state

Q	Q+	D
0	0	0
0	1	1
1	0	0
1	1	1

Excitation state

Present state $Q_2 Q_1 Q_0$	Next state $Q_2 Q_1 Q_0$	Flip-Flop i/p's D_2 D_1 D_0
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 0
0 1 0	0 1 1	0 1 1
0 1 1	1 0 0	1 0 0
1 0 0	1 0 1	1 0 1
1 0 1	0 0 0	0 0 0
1 1 0	x x x	x x x
1 1 1	x x x	x x x

D_2

Q_2 \ Q_1Q_0	00	01	11	10
0	0	0	1	0
1	1	0	x	x

$D_2 = Q_1 Q_0 + Q_2 \bar{Q}_0$

D_1

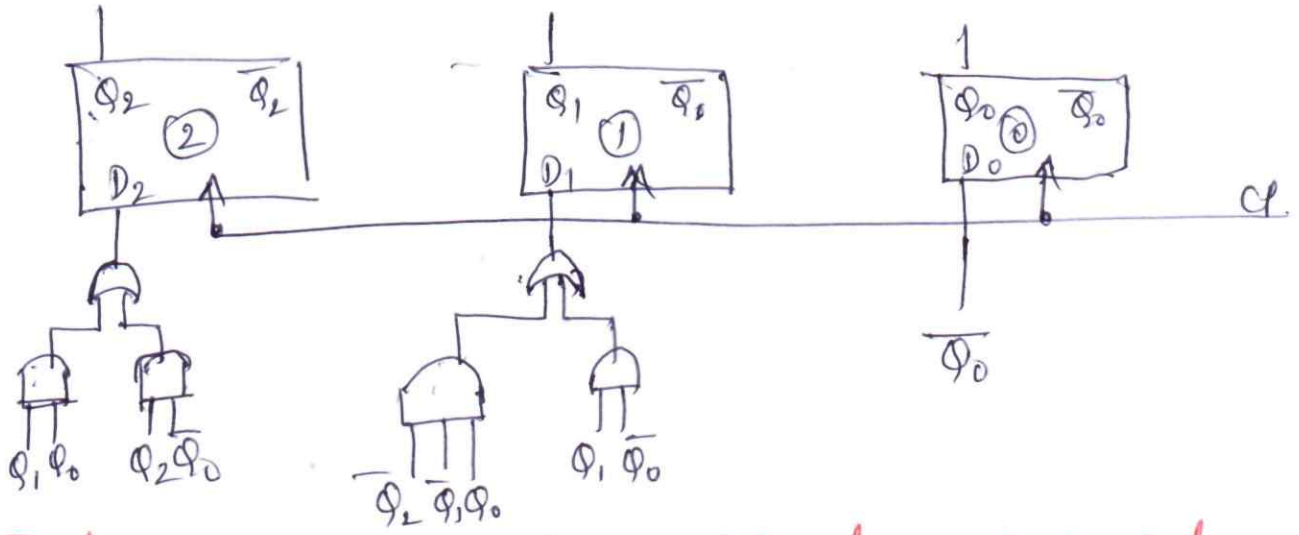
Q_2 \ Q_1Q_0	00	01	11	10
0	0	1	0	1
1	0	0	x	x

$D_1 = \bar{Q}_2 \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0$

D_0

Q_2 \ Q_1Q_0	00	01	11	10
0	1	0	0	1
1	1	0	x	x

$D_0 = \bar{Q}_0$



7 a). Explain Logical operators & relational operators used in Verilog

Ans. Logical operators:

> Verilog Bitwise logical operator:

Operator	Equivalent logic	Operand type	Result type
&		Bit	Bit
|		Bit	Bit
~&		Bit	Bit
~|		Bit	Bit
^		Bit	Bit
~^		Bit	Bit
~		Bit	Bit

> Verilog Boolean logical operators:

Operator	operators	No. of operands
&&	AND	TWO
||	OR	TWO

> Verilog Reduction Logical operators

Operator	Operation	No. of operands
&	Reduction AND	one
|	" OR	one
~&	" Nand	one
~|	" NOR	one
^	" X-OR	one
~^	" X-NOR	one
!	" Negation	one

> Relational operator

Operator	Description	Result Type
==	Equality	0, 1, x
!=	Inequality	0, 1, x
===	equality; inclusive	0, 1
!==	Inequality inclusive	0, 1
<	less than	0, 1, x
<=	less than or equal	0, 1, x
>	greater than	0, 1, x
>=	Greater than or equal	0, 1, x

7 b) Illustrate

i) Nets ii) Register iii) Vector iv) Integer data types with an example

i) Nets: Nets are declared by the predefined word wire

Value	Definition
0 —	Logic 0 false
1 —	Logic 1 true
x —	unknown
z —	High impedance

example: wire sum;
wire s1 = 1'b0;

ii) Register: Registers represent data storage elements Register is declared by predefined word reg

Value	Definition
0 —	Logic 0
1 —	Logic 1
x —	unknown
z —	High impedance

example: reg sum_total.

iii) Vector: Vectors are multiple bits, register or net can be declared as a vector.

wire [3:0] a = 4'b1010;
reg [7:0] total = 8'd12;

iv) Integers: Integers are declared by the predefined word integer

example: integer no_bits;

7 c) Write a verilog code for full adder using data flow description style:

Ans:
```
Module fulladder (a, b, cin, s, cout);
    input a, b, cin;
    output s, cout;
    assign s = a^b^cin;
    assign cout = ((a&b)|(a&cin)|(b&cin))
end module
```

8 a) Illustrate the structure of behavioural description with an example using half adder.

Ans. Behavioural Description: A behavioural description models the system as to how the o/p's behave with the i/p's
The module includes predefined word as always or initial

example:
```
Module half_add (I1, I2, O1, O2);
    input I1, I2;
    output O1, O2;
    reg O1, O2;
    always @ (I1, I2)
    begin
        #10 O2 = I1 & I2;
        #10 O2 = I1 & I2;
    end
end module
```

8b) Illustrate the structure of verilog module with an example using half subtractor.

Ans: Structure of the verilog module: The verilog module has a declaration & a body inputs & outputs of the module

Example:

```
Module half_sub (I1, I2, O1, O2)
  input I1, I2;
  output O1, O2;
  assign O1 = I1 ^ I2;
  assign O2 = ((~I1) & I2);
endmodule
```

8c) Write a verilog code for binary to gray using behavioural description style.

Ans: Binary to gray code in behavioural description style 4 bit

```
Module btg (b, g);
  input [3:0] b;
  output [3:0] g;
  reg [3:0] g;
  always @ (b)
    begin
      g[3] = b[3];
      g[2] = b[3] ^ b[2];
      g[1] = b[2] ^ b[1];
      g[0] = b[1] ^ b[0];
endmodule
```

9a) write the syntax of If & Else If with an example

Ans. If statement.

```
If (boolean expression)
  begin
      statement 1;
      statement 2;
      statement 3;
  end
else begin
      statement a;
      statement b;
      statement c;
end
```

ex:

```
if (clk == 1)
  temp = S2;
else
  temp = S1;
```

If & Else-If statement :

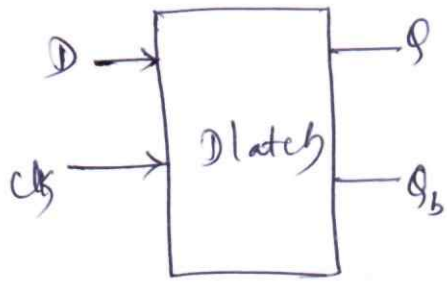
```
If (Boolean expression1)
begin
    statement 1 ;
    statement 2 ;
end
else if (Boolean expression2)
  begin
    statement i;
    statement ii;
  end
 else
  begin statement a;
        statement b ;
  end
```

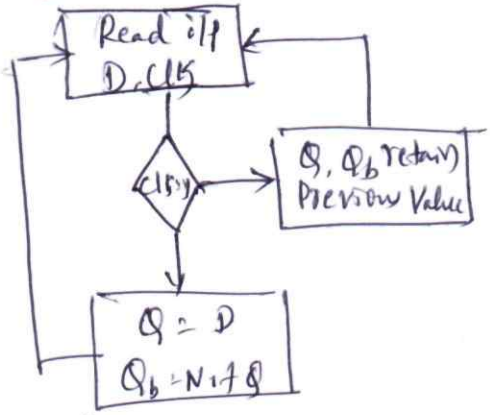
ex:

```
if (signal1 == 1)
  temp = s0;
else if (signal2 == 1)
  temp = s2 ;
else
temp = s3 ;
```

9b) Write Logic symbol, flowchart & Program for D-latch using behavioural description style

Ans:






```
Module D_latch ( D, E, Q, Qb)
  input D, E;
  output Q, Qb;
    reg Q, Qb;
  always @ (D, E)
  begin if (E==1)
          begin Q = D ;
                Qb = ~Q;
          end
```

9c) Write verilog code for 8:1 mux using behavioural description style

Ans: 8:1 mux using behavioural description style

```
Module mux2 (i, s, Y);
    input [7:0] i;
    input [2:0] s;
    output Y;
    reg Y;
    always @ (i, s)
    begin
      case (s)
      3'b000 : Y = i[0];
      3'b001 : Y = i[1];
      3'b010 : Y = i[2];
      3'b011 : Y = i[3];
      3'b100 : Y = i[4];
      3'b101 : Y = i[5];
      3'b110 : Y = i[6];
      3'b111 : Y = i[7];
      endcase
    end
    end module
```

10a) Explain the structure of structural model with built in gates using example of half adder. Also mention an primitive built in gates.

Ans: Structure of structural Model.

```
Module system (a, b, Sum, Cout);
    input a, b;
    output Sum, Cout;
    xor x1 (Sum, a, b);
    and a1 (Cout, a, b);
```

> Module Name is system with 2 i/ps a, b & 2 o/p's sum, Cout

> Verilog has a large number of built in gates

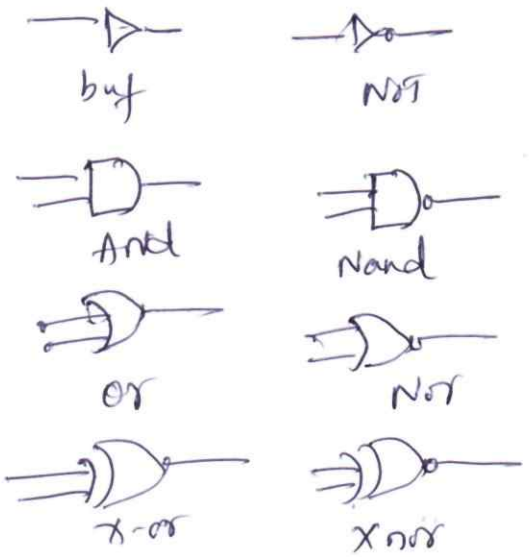
i.e xor x2 (sum, a, b);

Describes 2 i/p xor & optional identifier with i/p a, b & o/p sum

and a2 (Cout, a, b);

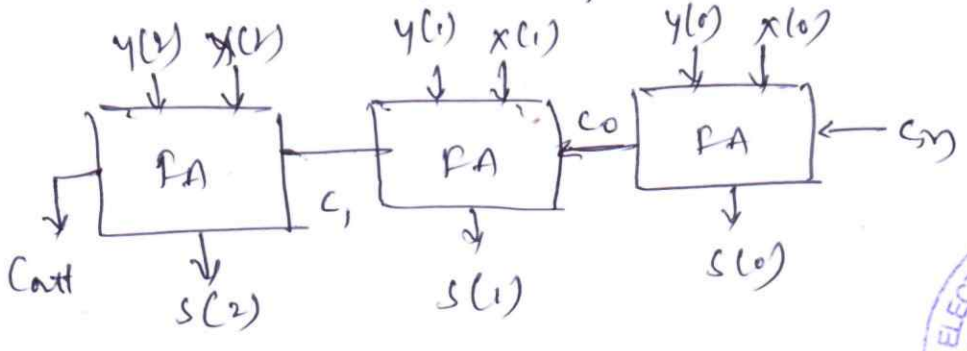
Describes two i/p's and with i/p's a, b & o/p Cout

Built in Primitive gates:



10b) write a Verilog code of a 3-bit ripple carry adder using structural description model

Ans: Structural description of a 3-bit Ripple Carry Adder:



```
Module three_bit_add (x, y, cin, sum, cout);
    input [2:0] x, y;
    input cin;
    output [2:0] sum;
    output cout;
    wire [1:0] carry;
      full_add m0 (x[0], y[0], cin, sum[0], carry[0]);
      full_add m1 (x[1], y[1], carry[0], sum[1], carry[1]);
      full_add m2 (x[2], y[2], carry[1], sum[2], cout);
endmodule
```

10c) Write a Verilog code of SR FF using behavioural description style

Ans: Verilog code for SR FF using behavioural description style

```
Module srff (s, r, clk, q, qb);
    input [1:0] s, r;
    input clk;
    output q, qb;
    reg q, qb
    initial begin
        q = 1'b0;
        qb = ~q;
    always @ (posedge clk)
    begin
    if (reset == 1)
    begin
        q = 1'b0;
        qb = ~q;
    end
    else
```

```
case (s.r)
    2'b00 : qn = qn;
    2'b01 : qn = 0;
    2'b10 : qn = 1;
    2'b11 : qn = 1'bz;
endcase
    qnb = ~qn;
    end
endmodule
```

Head of the Department
Dept. of Electronic & Communication Engg.
KLS V.D.I.T.. HALIYAL (U.K.)