Model solution for Digital Logic circuits

B. Tech. III sem., E.C.E

section - A

1. (i) - $(83)_{10}$

(ii) - b & d

(iii) - True

(iv) - one

(V) - False

(vi) - True

(vii) - (d)

(viii) - (b)

(ix) - False

(X) - (d)

Section – B.

Unit – I

Q(2) (a)- The check sum of the given message is

$$\begin{array}{cccccccc}
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
\hline
1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
\hline
\end{array} \rightarrow \text{check-sum}$$

(b)- In error detection and correction code with the help of parity bits we can detect the Location of bit in error and in case of digital, bits allowed are 0 and 1. so we can correct one error. some of important error detection and correction codes are ①- Block parity code
②- Hamming code

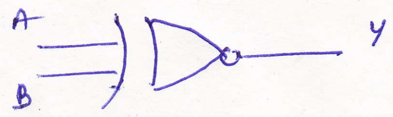
In block parity data block is arranged in a rectangular matrix form and two sets of parity bits are generated namely ①- Longitudinal Redundency-check (LRC) ②- Vertical redundency check (VRC)

VRC is the parity bit associated with the character-code and LRC is generated over the rows of bits for example a word 'COMPUTER' is encoded as

	C	O	M	P	U	T	E	R	LRC (Even parity)
	1	1	1	0	1	0	1	0	1
	1	1	0	0	0	0	0	1	1
	0	1	1	0	1	1	1	0	1
	0	1	1	0	0	0	0	0	0
	0	0	0	1	1	1	0	1	0
	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	0
VRC (Even parity)	1	1	0	0	0	1	1	1	1

Even a single error in any bit result in failure of LRC in one of the rows and VRC in one of the columns. The bit which is common to the row and column is the bit in error. Multiple errors in rows and column can be detected but can't be corrected as the bits which are in error can't be located.

Q(3) - Symbol $\rightarrow$



; $Y = \overline{\bar{A}B + A\bar{B}}$

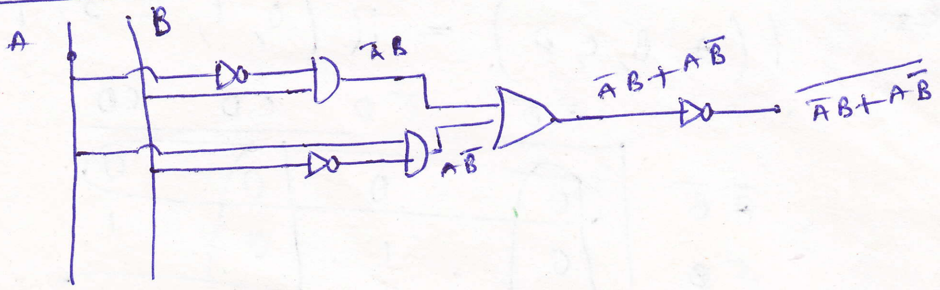
Truth table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

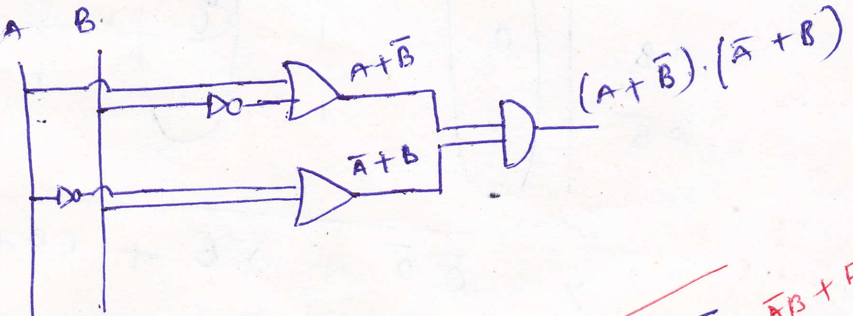
3

Ex- Nor gate.

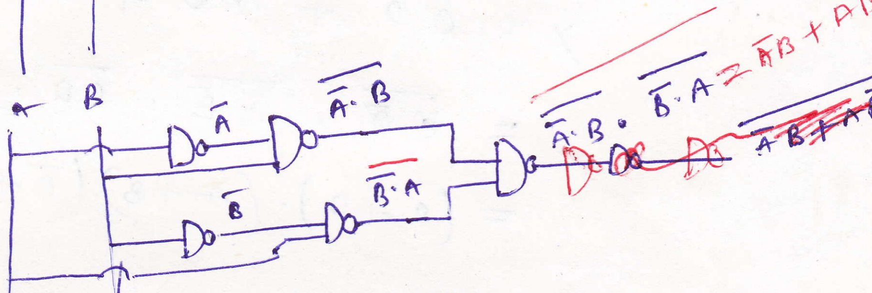
(i)- $y = \overline{\bar{A}B + A\bar{B}}$ =



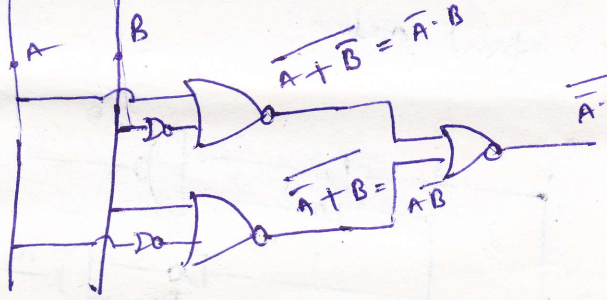
(ii)- $y = (A+\bar{B}).(\bar{A}+B)$ $\cong$



(iii)- $y = \overline{\bar{A}B + A\bar{B}}$ $\cong$


A
B
$\bar{A}$
$\overline{\bar{A}.B}$
$\bar{B}$
$\overline{\bar{B}.A}$
$\overline{\overline{\bar{A}.B} . \overline{\bar{B}.A}} = \bar{A}B + A\bar{B}$
$\overline{\bar{A}B + A\bar{B}}$

(iv)- $y = (A+\bar{B})(\bar{A}+B) \cong$

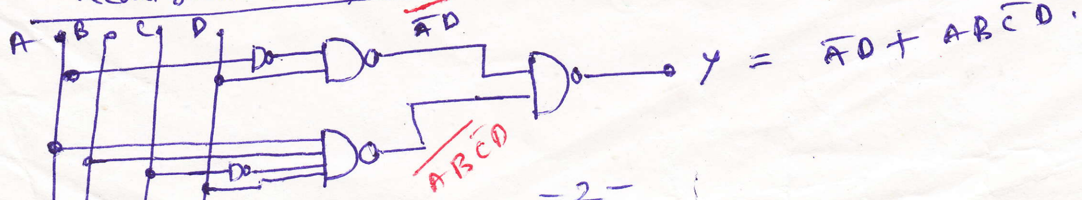

A
B
$\overline{A+\bar{B}} = \bar{A}.B$
$\overline{\bar{A}+B} = A\bar{B}$
$\overline{\bar{A}.B + A\bar{B}}$

In all above four circuit diagram their output is equivalent to output of ex-NOR.

Unit - II

Q4. $f = A'B'C'D + A'B'CD + A'BC'D + A'BCD + ABC'D$
$= \bar{A}\bar{B}D(\bar{C}+C) + \bar{A}BD(\bar{C}+C) + AB\bar{C}D = \bar{A}\bar{B}D + \bar{A}BD + AB\bar{C}D$
$= \bar{A}D(\bar{B}+B) + AB\bar{C}D = \bar{A}D + AB\bar{C}D = y$

Realization by NAND gate



Q5. $f(A,B,C,D) = \Pi(0,1,2,3,4,7,8,11,12,14,15)$

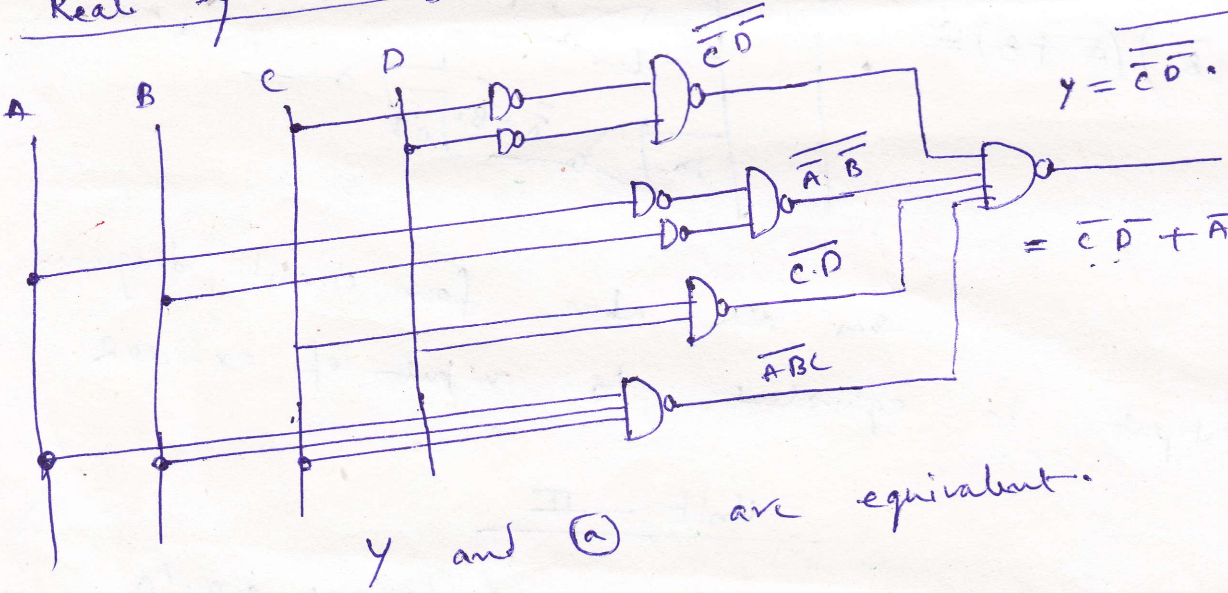
	$\overline{C}\,\overline{D}$	$\overline{C}D$	CD	$C\overline{D}$
$\overline{A}\,\overline{B}$	0	0	0	0
$\overline{A}B$	0	1	0	1
AB	0	1	0	0
$A\overline{B}$	0	1	0	1

$$y = \overline{\overline{C}\,\overline{D} + \overline{A}\,\overline{B} + CD + ABC} \quad \text{— ⓐ}$$

$$= \overline{\overline{C}\cdot\overline{D}} \cdot \overline{\overline{A}\,\overline{B}} \cdot \overline{CD} \cdot \overline{A\cdot B C}$$

$$= (C+D)\cdot(A+B)(\overline{C}+\overline{D})(\overline{A}+\overline{B}+\overline{C})$$

Realⁿ by Nand gates:



A
B
C
D
$\overline{\overline{C}\,\overline{D}}$
$\overline{\overline{A}\cdot\overline{B}}$
$\overline{C\cdot D}$
$\overline{ABC}$

$$y = \overline{\overline{\overline{C}\,\overline{D}} \cdot \overline{\overline{A}\cdot\overline{B}} \cdot \overline{C\cdot D}\;\overline{ABC}}$$

$$= \overline{C}\,\overline{D} + \overline{A}\cdot\overline{B} + CD + ABC$$

y and ⓐ are equivalent.

Unit - III

Q6- (a)-There are several techniques for reducing the carry propagation time in a parallel adder. The most widely used technique employs the principle of carry look ahead

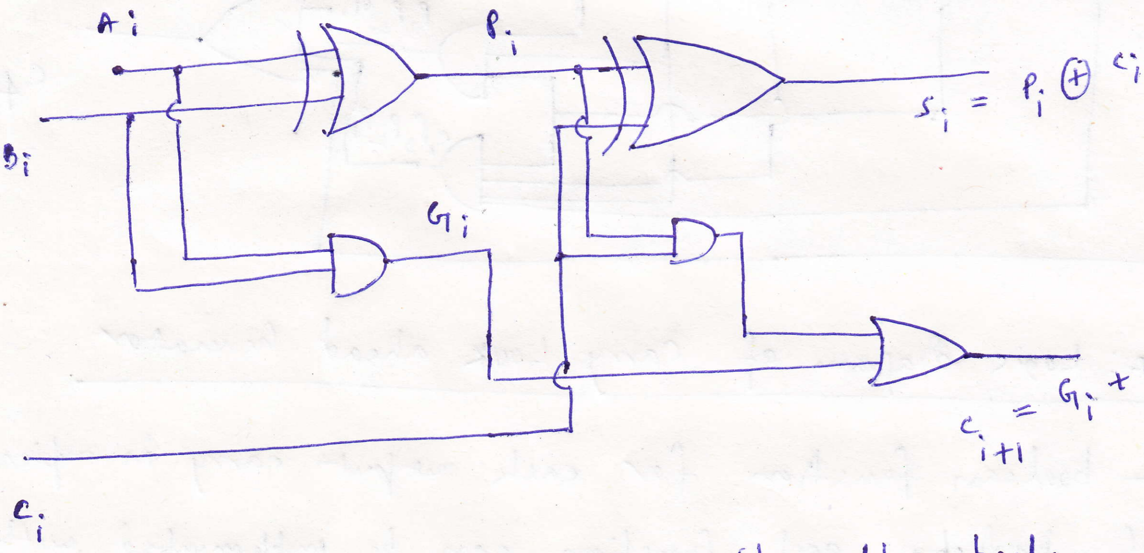
consider the circuit of the full adder shown below

$P_i = A_i \oplus B_i$; $G_i = A_i B_i$;

The output sum and carry can be expressed as

$S_i = P_i \oplus C_i$; $C_{i+1} = G_i + P_i C_i$

where G_i = carry generated.



Now we can write the boolean functions for the carry outputs of each stage.

$C_2 = G_1 + P_1 C_1$; $C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 C_1$;

$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_1$

This function can be implemented by

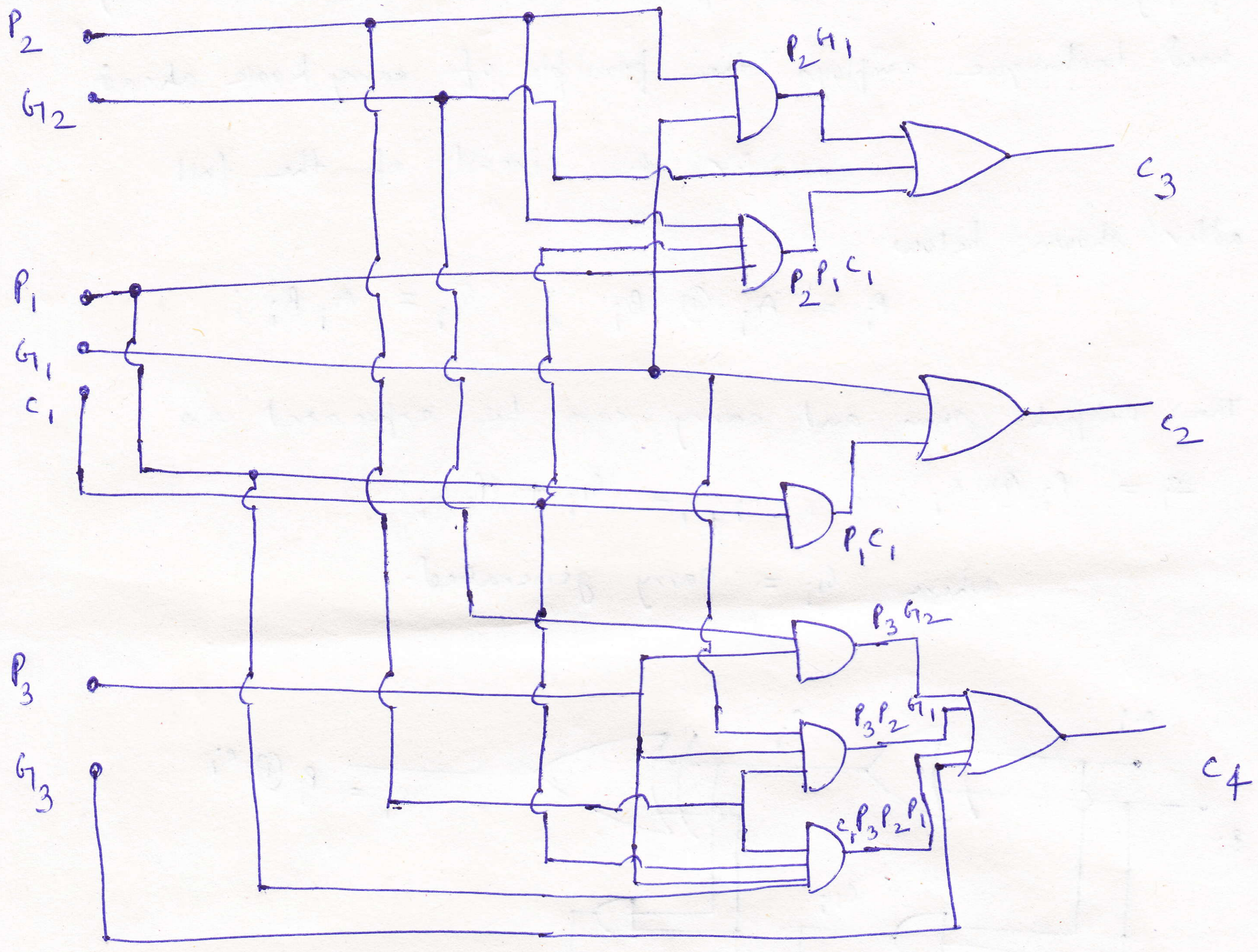


Figure: Logic diagram of Carry Look ahead Generator

Since the boolean function for each output carry is expressed in sum of products, each function can be implemented with one level of AND gates followed by an OR gate. The three boolean function for C_1, C_2 and C_3 are implemented in the carry look ahead generator shown in above figure. where C_3 does not have to wait for C_2 and C_1 to propagate i.e C_3 is propagated at the same time as C_1 and C_2.

Q 6

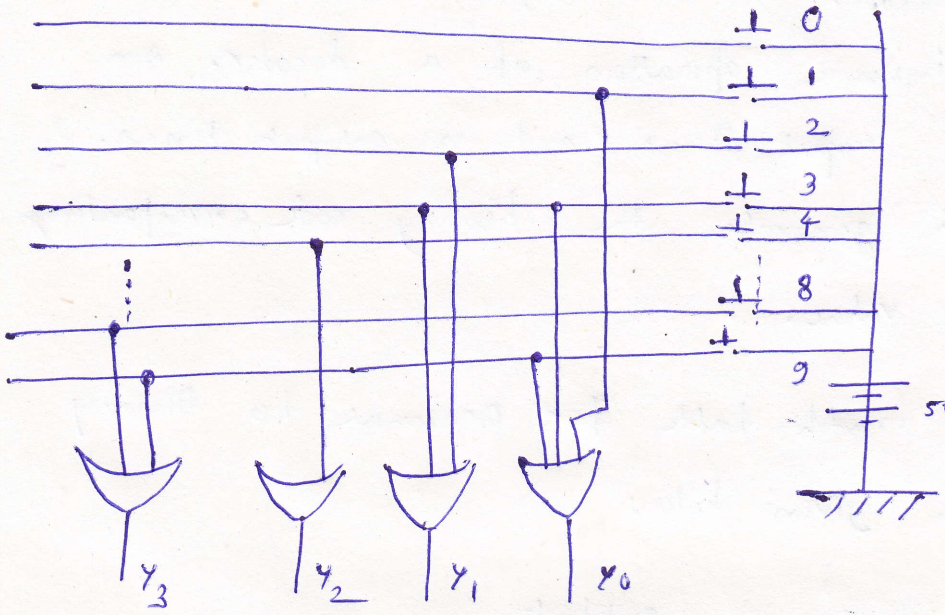
(b)- Encoder :- An encoder is a digital circuit that performs the inverse operation of a decoder. An encoder has 2^n input lines and n output lines. The output lines generate the binary code corresponding to the input value.

The truth table for Decimal to Binary Encoder is as given below

Input	A	B	C	D
		output		
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

As shown in diagram when push button '0' is pressed none of the OR gates is energised all outputs are low so that the output word $Y_3 Y_2 Y_1 Y_0$ = 0 0 0 0. of push button '4' is pressed output word is $Y_3 Y_2 Y_1 Y_0$ = 0 1 0 0.

Figure: Logical circuit for Decimal to BCD Encoder



7. Multiplexers: A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

In shown diagram of 8:1 multiplexer, 8 users i,e $I_0 - I_7$ are connected. when selection line A BC = 0 0 0 then user I_0 can transmit its information. when A BC = 111 then I_7 can transmit because the concerned AND gate is enabled.

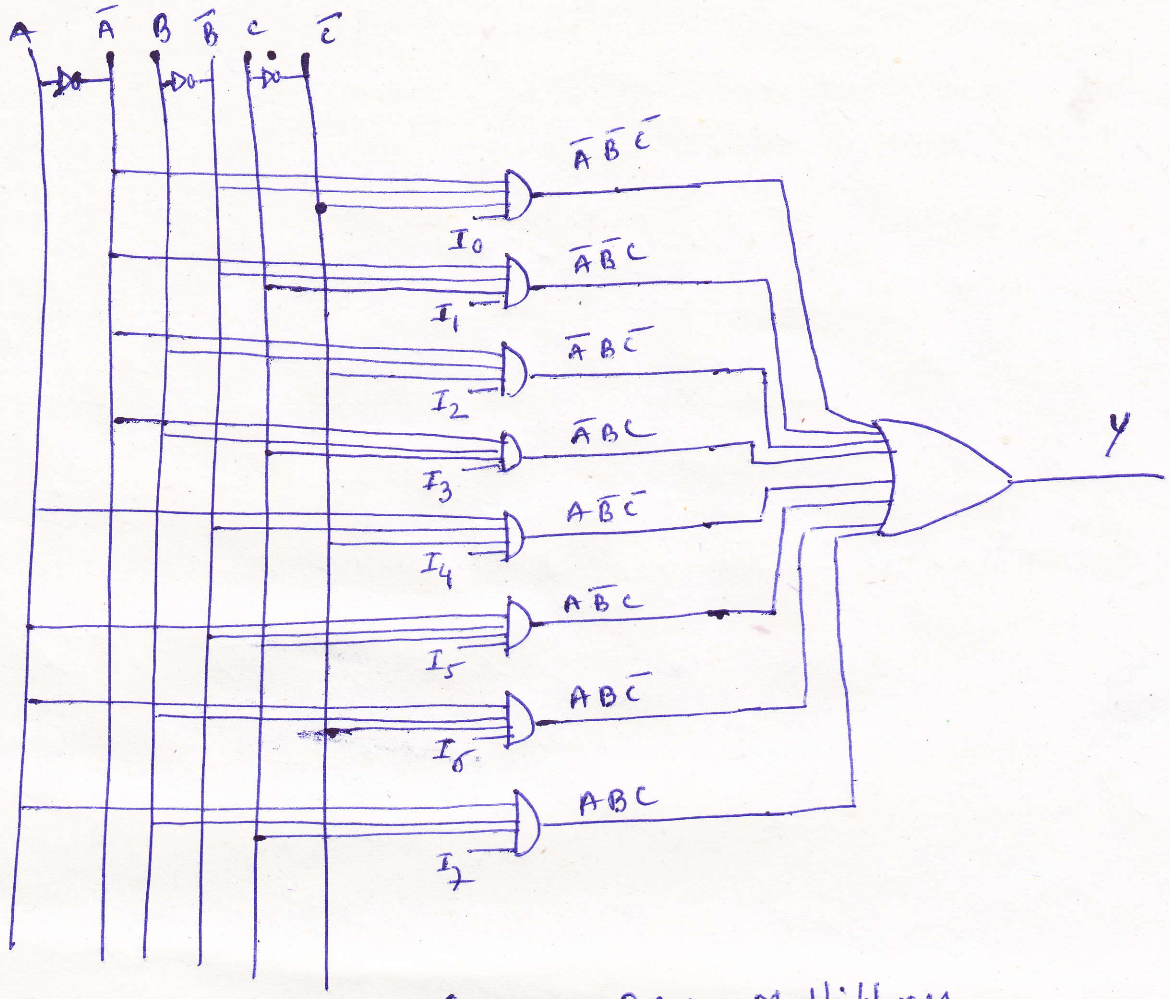


figure: 8:1 Multiplexer

Unit - IV

Q8- In case of RS flip-flop with NOR gate, when R and S inputs are 1 then outputs Q and $\bar{Q}$ both are same which is against the assumption and gets an undefined state this condition is termed as 'Race condition'.

To eliminate the possibility of a Race condition we use D flip-flop.

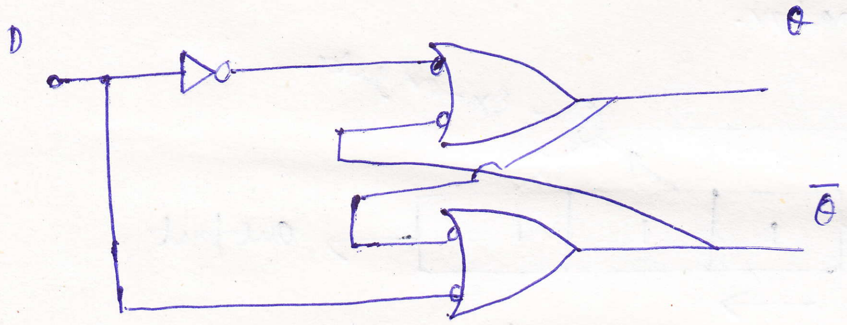


Figure :- D flip-flop.

The above diagram shows one way to build a D-flip flop. The inverter in above circuitry guarantees that (S and R) inputs will always be in opposite states. Therefore, it is impossible to set up a race condition in D-flip-flop.

Q9 - A shift register moves the stored bits left or right. There are two types of shift register ①- shift left register ②- shift right register

In case of shift right register when suitable clock arrives, the stored bits move one position to the right. In shift left register, the stored bits move one position to the left on suitable clock.

One application of shift register is pn sequence generator.

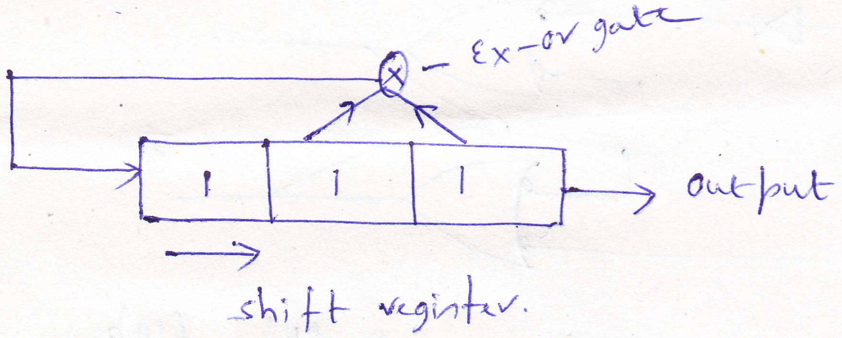


Figure : PN sequence generator

Using a shift register with feedback to generate the PN sequence 1110010 is shown in above figure. The Modulo-2 sum of the right-most two bits is shifted into the register at every clock pulse. The initial content of shift register is 111. The content of flip-flop after every clock is 111 (initial), 011, 001, 100, 010, 101, 110, 111 and generated sequence is 1110010.

Unit - V

Q10:- Implementation of AND gate using diodes.

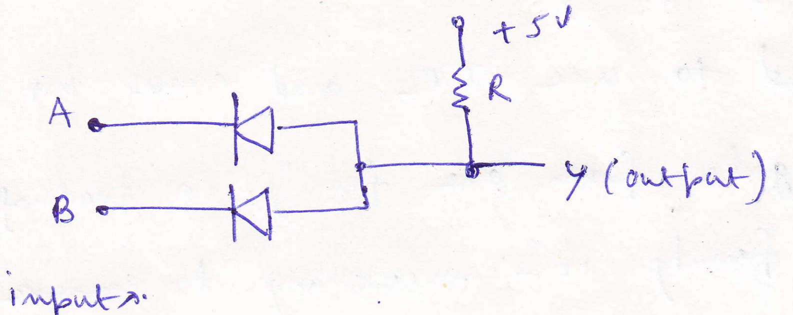


fig. - a

Truth - table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

In above diagram(a) when both inputs are 0 or one of inputs are 0 then both diodes or one of diode is forward biased and output is 0. when both inputs are 1 then diodes are reverse biased and output is 1.

Implementation of OR gate using diodes.

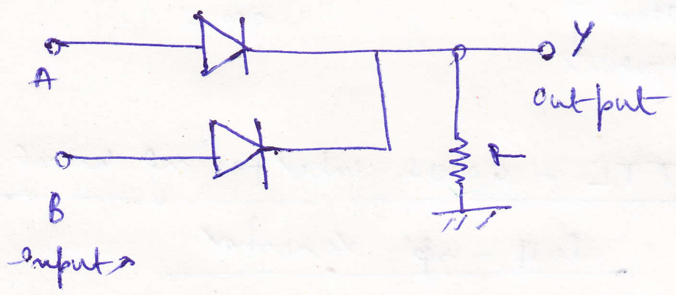


fig - b

Truth - table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

In above diagram (b) when both inputs are 0 both diodes are off and output is zero. when both inputs are 1 or one of input is 1 then both or one of diode are ON and output is 0 as shown in truth - table.

Q11 - Interfacing - Many times it is necessary to use more than one logic family in a digital circuit e.g. it may be required to use TTL and CMOS in same circuitry. When gate from one family is feeding the gate of another family it's necessary to ensure the matching of voltage and current levels. This is called interfacing.

TTL to CMOS interfacing.

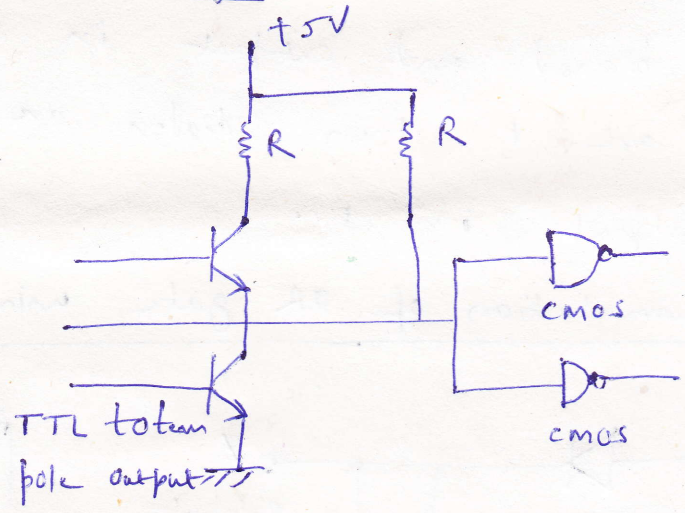


fig - TTL - CMOS interfacing using Pull-up resistor

The input current of CMOS is so small that can be easily satisfied. but V_{OH} of TTL is less than V_{IH} of CMOS. V_{IH} of CMOS can't be lowered. Therefore the only alternative is to pull up (increase) V_{OH} of TTL gate by adding a pull up resistance to nearly +5V.