

Chapter-5

Combination logic

Logic CKT for digital system

are combinational or sequential.

A combinational CKT consists of logic gates whose output at any time are determined directly from the present combination of input and does not depend upon past condition. It does not have memory.

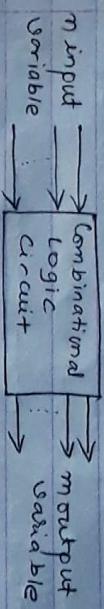


Fig: Block Diagram of Combinational CKT.

As shown in above figure a combinational CKT consist of ^m input variables

logic gates and ^m output variables

Input variables are of

binary signal which has two possible values logical 1 and 0. For n input variables, there are 2^n possible combination of binary input values and different output combination for each input.

* Design procedure

Combinational CKTs are

designed by following steps

- problem is stated.

- the number of input variables and output variables

required is determined.

The input output variables are assigned letter symbols.

Truth table is determined using given the relationship between input and output.

Simplified Boolean function of each output is obtained.

6. The logic diagram is drawn for the output Boolean function.

As been stated, the simplified Boolean function are simplified

a) Algebraic manipulation

b) Map method

c) Tabulation procedure

keeping in consideration of restriction, limitation and criteria stated in the design. However, for practical design we answer

- minimum number of gates
- minimum number of inputs
- minimum propagation time of signal through the circuit
- minimum number of inter connection
- limitation of driving capabilities of each gate

* Adder

Digital computers performs processing of task, one of the important task is arithmetic operation. Most basic arithmetic operation is the addition of binary digit which is done by combinational circuit called Adder. There are two types of Adder.

1. Half Adder :→

A combinational circuit that performs the addition of two binary bits. Half Adder consists of two binary input bits and produces two output bits, a sum bit and a carry bit.

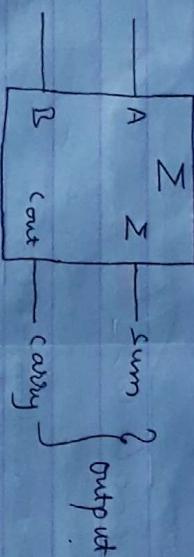


fig : logic symbol of Half Adder

A	B	carry (cout)	sum (Σ)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Truth table

- * recall basic rules for binary addition. For half adder, the carry is 0 unless both the input are 1.

By observation of the truth table the sum can be expressed as the Exclusive-OR and carry by the AND gate which is shown below.



fig :→ HALF ADDER

2. Full Adder :→

A combinational circuit that performs the addition of three binary bits i.e. two bits and a previous carry producing two bits of sum and carry in called Full Adder.

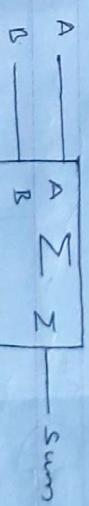


fig : logic symbol of Full Adder

A	B	cout	cout	cout	cout	cout	cout
0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1
1	0	0	1	0	1	0	1
1	1	1	0	1	0	1	1

Truth table

* For sum

$$S = A'B'C_{in} + A'B'C_{in} + A'BC_{in} + ABC_{in}$$

$$S = A'(B'C_{in} + C_{in}) + A(C'B'C_{in} + B'C_{in})$$

$$S = A'(B \oplus C_{in}) + A(C \oplus C_{in})$$

$$S = A'(A \oplus C_{in}) + A(\overline{A \oplus C_{in}})$$

$$S = A \oplus A \oplus C_{in}$$

For carry

$$C_{out} = A'B'C_{in} + A'BC_{in} + ABC_{in}$$

$$C_{out} = A'C_{in} + AB + BC_{in}$$

$$C_{out} = A'C_{in} + A'B'C_{in} + ABC_{in}$$

$$\begin{array}{c|cc|cc|cc} & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & A' & & A' & & A' & \\ 1 & A & & A & & A & \\ \hline & L & L & L & L & L & L \end{array}$$

$$A \otimes C_{in}$$

$$S = A \otimes C_{in}$$

$$A \otimes B$$

$$A \otimes C_{in}$$

$$A \otimes B$$

$$C_{out}$$

$$A \otimes C_{in}$$

fig : logic ckt for full adder.

* prove that full adder is the combination of

two half adder with an extra OR gate.

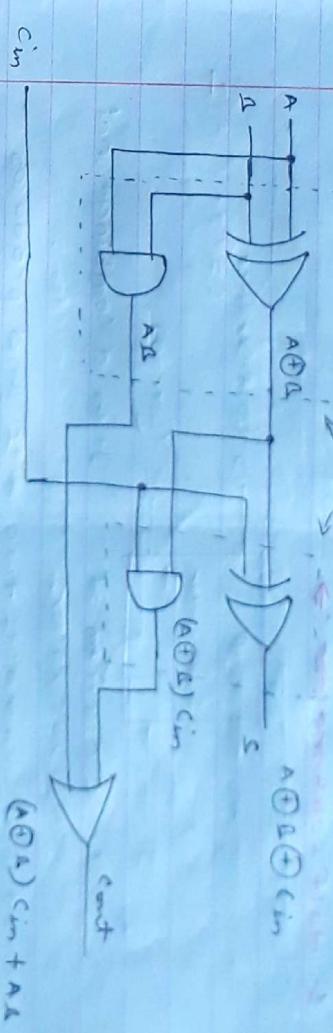
$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = A'C_{in} + A'B'C_{in} + ABC_{in}$$

$$C_{out} = (A \oplus B) C_{in} + A \oplus B \oplus C_{in}$$

$$C_{out} = (A \oplus B) C_{in} + A \oplus B \oplus C_{in}$$

Half adder



The truth-table of full adder is

$$(A \oplus B)C_{in} + A \oplus B$$

$$A \otimes C_{in}$$

$$A \otimes B$$

$$A \otimes C_{in}$$

$$A \otimes B$$

$$C_{out}$$

$$A \otimes C_{in}$$

$$A \otimes B$$

$$A \otimes C_{in}$$

$$A \otimes B$$

$$C_{out}$$

$$A \otimes C_{in}$$

$$A \otimes B$$

$$A \otimes C_{in}$$

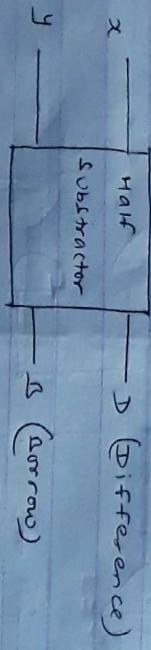
$$A \otimes B$$

$$A \otimes C_{in}$$

* Subtractor

combinational circuit performs subtraction of binary numbers by taking the complement of the subtrahend and adding it to the minuend. There are two types of subtractor as Adder.

1. Half Subtractor :



Block diagram of $g+$ is a combinational half subtractor

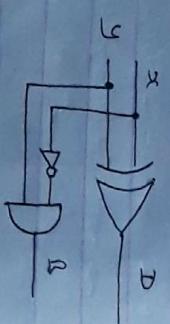
that subtracts two bits and produces their difference. $g+$ has two inputs designated x as minuend and y as subtrahend. D is the difference and B is the borrow output.

x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Therefore

$$D = x'y + xy'$$

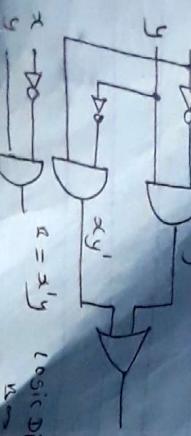
$$\text{and } B = x'y$$



$$D = x \oplus y$$

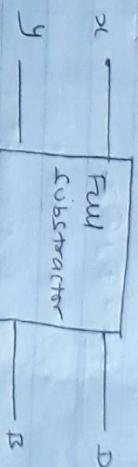
$$B = x'y$$

logic circuit for half subtractor



logic diagram using
inverter

2. Full Subtractor :



$g+$ performs the subtraction

between two bits taking account that 1 may be borrowed from a lower significant stage. The combinational circuit consists of three inputs and two outputs. The three inputs x, y, z denote the minuend, subtrahend and previous borrow respectively.

therefore

$$D = x'y'z + x'y'z' + x'y'z + xyz$$

x	y	z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

x	y	z	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = x'(y \oplus z) + x(y \oplus z)$$

$$B = x'y'z + x'y'z' + x'y'z + xyz$$

At $x=1$ mapping

$$B = x'z + xy + yz$$

1 x

x	y	z	B
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

x
y
z

$$D = x \oplus y \oplus z$$



x'z

x'yz

$$D = x'z + x'y'z + y'z$$

$$D = x'(y'z + yz) + x(yz + y'z)$$

* fig : logic ckt for Full subtractor.

* prove that full subtractor can be derived from full adder with an inverter

The combinational ckt of full

subtractor consist of three inputs and two outputs. If the x, y, z is the minuend, subtrahend and previous borrows respectively. Output designated by D refers to the difference and A refers to the borrow. The truth table for full subtractor is

x
y
z

x	y	z	D	A
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = x'y'z + x'y'z + x'y'z + xyz$$

$$D = x'(y'z + yz) + x(yz + y'z)$$

Similarly,

$$A = x'(y'z + yz) + yz(x' + x)$$

$$A = x'(y \oplus z) + yz$$

y
z

$$y \oplus z$$

x

x'

y

y'

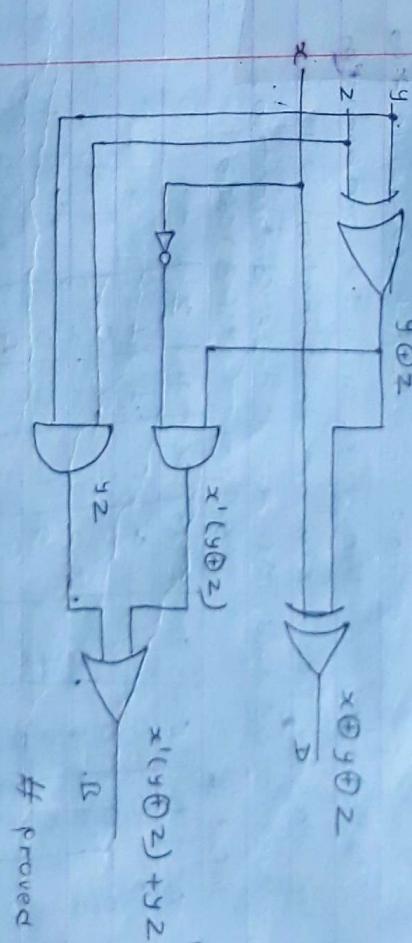
z

z'

D

$x'(y \oplus z)$

$x'(y \oplus z) + yz$



proved

* Code conversion

Digital system consists of different system within it. If these systems uses different code conversion circuit is required to make it compatible. This circuit is known as code converter and performs conversion of code.

* Example

Design a circuit that converts BCD to

Excess-3 code implement using gates.

BCD codes uses four bits to represent decimal digit. 0 has four input variable A, B, C, D and four output variable w, x, y, z.

As we know four input variable has $2^4 = 16$ combination, only 10 combination is valid. Rest six bit combination will never occur and are don't care combination.

Input (BCD) Output (Excess-3) Code

A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	1	0	1	0
0	0	1	1	0	1	0	1
0	1	0	0	1	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0
1	0	1	0	0	1	1	0
1	0	1	1	1	0	0	1
1	1	0	0	0	0	0	0
1	1	0	1	1	1	0	0
1	1	1	0	1	0	1	1
1	1	1	1	0	1	1	1

The maps are drawn to obtain simplified logic function for each output

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

$$x = \bar{A} \bar{B} \bar{C} D + \bar{A} \bar{B} C D + A \bar{B} \bar{C} D' + A \bar{B} C D'$$

$$y = C \bar{D} + \bar{B} D + \bar{C} \bar{D}'$$

$$z = C \bar{C} + \bar{B} \bar{D} + \bar{C} \bar{C} \bar{D}'$$

$$\begin{array}{c|ccccc}
& \bar{A} & \bar{B} & \bar{C} & \bar{D} & w \\
\bar{A} & 0 & 0 & 0 & 0 & 1 \\
\bar{B} & 0 & 0 & 0 & 1 & 1 \\
C & 0 & 0 & 1 & 0 & 1 \\
D & 0 & 0 & 1 & 1 & 0 \\
\hline
w & 1 & 1 & 1 & 1 &
\end{array}$$

$$w = A + B + C + D$$

$$x = A + B + C + D$$

$$\begin{array}{c|ccccc}
& \bar{A} & \bar{B} & \bar{C} & \bar{D} & x \\
\bar{A} & 0 & 0 & 0 & 0 & 1 \\
\bar{B} & 0 & 0 & 0 & 1 & 1 \\
C & 0 & 0 & 1 & 0 & 1 \\
D & 0 & 0 & 1 & 1 & 0 \\
\hline
x & 1 & 1 & 1 & 1 &
\end{array}$$

$$x = C \bar{D} + \bar{B} D + \bar{C} \bar{D}'$$

$$\begin{array}{c|ccccc}
& \bar{A} & \bar{B} & \bar{C} & \bar{D} & y \\
\bar{A} & 0 & 0 & 0 & 0 & 1 \\
\bar{B} & 0 & 0 & 0 & 1 & 1 \\
C & 0 & 0 & 1 & 0 & 1 \\
D & 0 & 0 & 1 & 1 & 0 \\
\hline
y & 1 & 1 & 1 & 1 &
\end{array}$$

$$y = C D + C' D'$$

$$Z = A' B' C' D' + A' B' C D' + A' B' C D + A' B' C D' + A' B' C D$$

$$\begin{array}{c|ccccc}
& \bar{A} & \bar{B} & \bar{C} & \bar{D} & Z \\
\bar{A} & 0 & 0 & 0 & 0 & 1 \\
\bar{B} & 0 & 0 & 0 & 1 & 1 \\
C & 0 & 0 & 1 & 0 & 1 \\
D & 0 & 0 & 1 & 1 & 0 \\
\hline
Z & 1 & 1 & 1 & 1 &
\end{array}$$

$$Z = C D + C' D'$$

$$\begin{array}{c|ccccc}
& \bar{A} & \bar{B} & \bar{C} & \bar{D} & Z \\
\bar{A} & 0 & 0 & 0 & 0 & 1 \\
\bar{B} & 0 & 0 & 0 & 1 & 1 \\
C & 0 & 0 & 1 & 0 & 1 \\
D & 0 & 0 & 1 & 1 & 0 \\
\hline
Z & 1 & 1 & 1 & 1 &
\end{array}$$

\oplus

The maps are drawn to obtain simplified Boolean function for each output

$$w_0 = A' D' C' D' + A' A C D$$

$$d = A' A C D' + A' A C' D'$$

$$+ A' C' D' + A' A' C' D$$

$$+ A' A' C D + A' A' C' D$$

A'	C'	D'	w_0
0	0	0	X
0	1	0	X
1	0	0	X
1	1	0	X

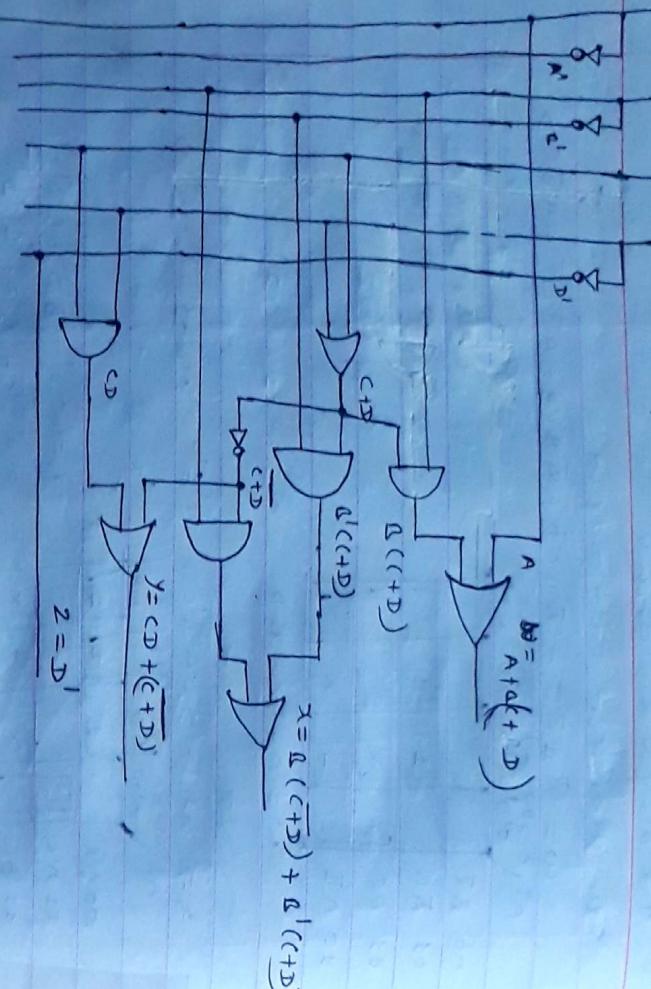
$$w_0 = A' A + C' D$$

A'	C'	D'	w_1
0	0	0	X
0	1	0	X
1	0	0	X
1	1	0	X

$$w_1 = A' C + A' D$$

A'	C'	D'	x
0	0	0	X
0	1	0	X
1	0	0	X
1	1	0	X

$$x = A' A' C D' + A' A' C' D + A' B' C D' + A' B' C' D$$



2. Design a Combinational circuit that converts 8, 4, -2, -1 code to LCD code (to binary)

gmp(8, 4, -2, -1) LCD code

A	B	C	D	8, 4, -2, -1	LCD code
0	0	0	0	10	1110
0	0	1	0	11	1101
0	1	1	1	12	1100
1	0	1	1	12	1100
2	0	1	0	10	0010
3	0	1	1	10	0011
4	0	1	0	00	0000
4	0	2	1	01	0001
4	0	1	0	10	0010
4	0	0	2	01	0011
4	0	0	0	00	0000
5	1	1	1	11	0011

$$y = A' B C D' + A' B C' D + A B' C D' + A B' C' D$$

A'	C'	D'	y
0	0	0	X
0	1	0	X
1	0	0	X
1	1	0	X

$$y = C' D + C D$$

A'	C'	D'	x
0	0	0	X
0	1	0	X
1	0	0	X
1	1	0	X

$$x = C' D + C D$$

$$Z = A'DC + A'DC'D + A'D'CD + A'D'C'D + A'DCD$$

$A'DC$

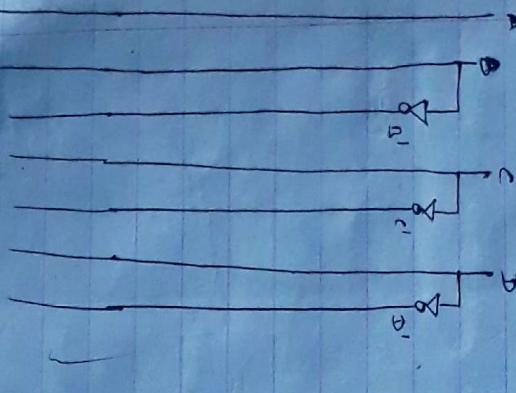
$C'D$

CD

CD'

01A'1	X	X	X
1A'B'	X	1	X
10A'1	L	1	

$$Z = D$$



2. Design a combinational circuit that converts a decimal digit from the 2,4,2,1 code to the 8,4,2,1 code.

$$\begin{array}{l} wxyz \\ \hline 0 & 0000 \\ 1 & 0001 \\ 2 & 0010 \\ 3 & 0011 \\ 4 & 0100 \\ 5 & 1011 \\ 6 & 1100 \\ 7 & 1101 \\ 8 & 1110 \\ 9 & 1111 \end{array}$$

$$d = AB'C'D' + A'D'C'D + ABC'D' + A'DC'D$$

2

3

4

5

6

7

2	1000	xxx	2
3	1001	xxx	3
4	1010	xx	4
5	0101	xx	5
6	0110	xx	6
7	0111	xx	7

$$\begin{aligned} d &= AB'C'D' + A'D'C'D + ABC'D' + A'DC'D \\ &+ A'BC'D' + A'BCD \end{aligned}$$

$$w = A'DC + ABC'D' + A'BC'D + ABCD' + A'BCD$$

A'D	X	X	X
A'D	X	X	X
A'D	1	1	1

$$w = A$$

$$x = A'B'C'D + A'B'CD' + A'BC'D + A'BC'D' + ABCD$$

	$A'B'$	$C'D'$	$C'D$	CD	CD'
00 $A'B'$	1	1	1		
01 $A'B$	1	X	X	X	
11 AB			1		
10 AB'	X	X		X	

	$A'B'$	$C'D'$	$C'D$	CD	CD'
00 $A'B'$	1	1	1		
01 $A'B$	1	X	X	X	
11 AB		1			
10 AB'	X	X		X	

$$x = A'B + A'D + A'C + ABCD$$

$$y = A'B'C'D + A'B'CD' + AB'C'D + ABC'D' + ABCD$$

	$A'B'$	$C'D'$	$C'D$	CD	CD'
00 $A'B'$	1			1	
01 $A'B$		X	X	X	
11 AB	1		1		
10 AB'	X	X	1	X	

	$A'B'$	$C'D'$	$C'D$	CD	CD'
00 $A'B'$	1			1	
01 $A'B$		X		X	X
11 AB	1			1	
10 AB'	X		X	1	X

$$z = A'B'C'D + A'B'CD + AB'C'D + ABC'D + ABCD$$

	$A'B'$	$C'D'$	$C'D$	CD	CD'
00 $A'B'$		1	1		
01 $A'B$		X	X	X	
11 AB	1	1	1		
10 AB'	X	X	1	X	

$$z = D$$

3. Design a combinational Ckt that converts a decimal digit from Gray code to 4 bit binary number. Implement

Gray code

Ckt with exclusive-or gate.

Implementation

$$w = A'ac'D' + A'ac'D + ABCD \\ + A'c'D' + A'c'D + ABC'D'$$

$$Z = A'ac'D' + A'c'D + ABCD \\ + A'c'D' + A'c'D + ABC'D'$$

A	c'D'	c'D	CD	cD'
A'	1	4	4	4
A	4	4	2	2
A'	1	1	1	1

Gray code	4 bit Binary
0000	0000
0001	0101
0010	0010
0011	0111
0100	0100
0101	0110
0110	0111
0111	0101
1000	1000
1001	1101
1010	1100
1011	1011
1100	1010
1101	1001
1110	1111
1111	1110

$$w = A$$

$$\begin{array}{|c|c|c|c|} \hline & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$x = A'ac'D' + A'ac'D + ABCD$$

$$+ A'c'D' + A'c'D + ABC'D'$$

$$x = A'ac + AB' = A \oplus B$$

$$y = A'ac'D + A'ac'D + ABCD + A'c'D'$$

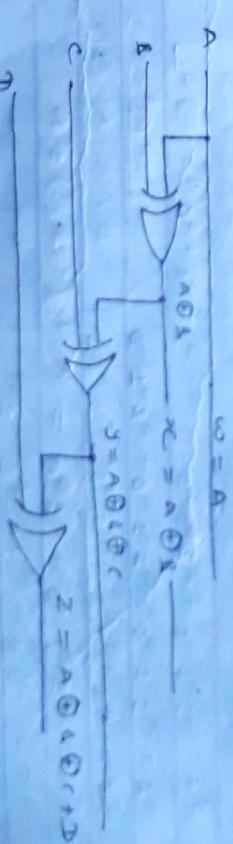
$$+ A'c'D + A'c'D + ABCD + A'c'D'$$

$$Z = A \oplus B \oplus C$$

$$x = A \oplus B$$

$$y = A \oplus B \oplus C$$

$$Z = A \oplus B \oplus C$$



(00) A'c'D'	1	1	1	1
(01) A'c'D'	1	1	0	0
(10) A'c'D'	0	1	1	0
(11) A'c'D'	0	1	0	1

$$y = A'ac' + A'ac + ABCc + A'cc'$$

$$+ A'c'c + A'c'c + ABCc + A'cc'$$

4. Design a combinational CKT that accepts a three bit number and generates an output binary number equal to the square of the input numbers.

Let three input bit number be represented by variable A_0, A_1, A_2 and output be represented by variable $B_0, B_1, B_2, B_3, B_4, B_5, B_6$.

$$\begin{array}{c|c} \text{Input number} & \text{Output number} \\ \hline A_2 A_1 A_0 & B_6 B_5 B_4 B_3 B_2 B_1 \\ A_6 A_5 A_4 A_3 A_2 A_1 & \end{array}$$

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

$$\begin{aligned} B_2 &= 0 \\ * B_1 &= A_2' A_1' A_0 + A_2' A_1 A_0 + A_2 A_1' A_0 + A_2 A_1 A_0 \\ Q_4 &= A_2' A_0 (A_1' + A_1) + A_2 A_0 (A_1' + A_1) \\ B_2 &= A_0 \end{aligned}$$

$$\begin{array}{c} 32 \\ - 8924 \\ \hline 1001 \end{array}$$

0	000	wxyz
1	01	0001
2	10	0100
3	11	1001

0	000	wxyz
1	01	0001
2	10	0100
3	11	1001

$$B_6 = A_2 A_1 A_0' + A_2 A_2 A_0 = A_2 A_2 (A_0 + A_0') = A_2 A_2$$

$$B_5 = A_2 A_1' A_0' + A_2 A_1' A_0 + A_2 A_2 A_0$$

$$B_4 = A_2 A_1' A_0' + A_2 A_0 (A_2' + A_2)$$

$$B_3 = A_2 A_1' A_0' + A_2 A_2 A_0$$

$$B_2 = A_2 (A_0 + A_1')$$

$$B_1 = A_2 A_1 + A_2 A_0$$

$$B_0 = A_2' A_1 A_0 + A_2' A_1 A_0' = A_0 (A_1 \oplus A_2)$$

$$B_3 = A_2' A_1 A_0' + A_2 A_2 A_0' = A_2 A_0' (A_2' + A_2) = A_2 A_0'$$

1		
	1	
		1

5. Design a combination logic whose output is the 2's complement of the input number.

$$F_3 = A'B'C'D + A'BC'D' + A'BCD + A'BC'D' + A'BCD + A'BCD + A'BCD'$$

$$\begin{array}{c|ccccc} A & B & C & D & F_4 & F_3 & F_2 & F_1 \\ \hline 0 & 0 & 0 & 0 & 0000 & 0000 & 0000 & 0000 \\ 1 & 0 & 0 & 0 & 0011 & 1111 & 1110 & 1110 \\ 2 & 0 & 0 & 1 & 0010 & 1110 & 1110 & 1110 \\ 3 & 0 & 1 & 1 & 0011 & 1110 & 1101 & 1101 \\ 4 & 0 & 1 & 0 & 0100 & 1100 & 1100 & 1100 \\ 5 & 0 & 1 & 0 & 0101 & 1011 & 1011 & 1011 \\ 6 & 0 & 1 & 1 & 0110 & 1010 & 1010 & 1010 \\ 7 & 0 & 1 & 1 & 0111 & 1001 & 1001 & 1001 \\ 8 & 1 & 0 & 0 & 1000 & 1000 & 1000 & 1000 \\ 9 & 1 & 0 & 1 & 0111 & 1001 & 1001 & 1001 \\ 10 & 1 & 0 & 1 & 0110 & 1000 & 1000 & 1000 \\ 11 & 1 & 0 & 1 & 0101 & 1001 & 1001 & 1001 \\ 12 & 1 & 1 & 0 & 1100 & 0100 & 0100 & 0100 \\ 13 & 1 & 1 & 0 & 1101 & 0021 & 0021 & 0021 \\ 14 & 1 & 1 & 0 & 1110 & 0010 & 0010 & 0010 \\ 15 & 1 & 1 & 1 & 1111 & 0001 & 0001 & 0001 \end{array}$$

$$\begin{aligned} F_1 &= \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D \\ &\quad + \bar{A}BCD + A\bar{B}\bar{C}D + A\bar{B}CD + ABCD \\ &\quad + ABD \\ F_2 &= \bar{A}\bar{B}(\bar{C}+C)D + \bar{A}B(\bar{C}+C)D \\ &\quad + A\bar{B}D(\bar{C}+C) \\ &\quad + ABD(C\bar{C}+C) \\ F_3 &= \bar{A}\bar{B}D + \bar{A}BD + A\bar{B}D + ABD \end{aligned}$$

A	B	C	D	F ₄
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
1	0	0	0	1
1	0	0	1	1
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

$$F_2 = B'C'D + B'D + A'C$$

$$\begin{aligned} F_4 &= A'D'C'D + A'D'CD + A'DC'D + A'DC'D \\ &\quad + A'DC'D + A'DC'D + A'DC'D \end{aligned}$$

$$\begin{array}{c|ccccc} A & B & C & D & F_4 & F_3 & F_2 & F_1 \\ \hline 0 & 0 & 0 & 0 & 0000 & 0000 & 0000 & 0000 \\ 1 & 0 & 0 & 0 & 1111 & 1111 & 1111 & 1111 \\ 2 & 0 & 0 & 1 & 1110 & 1110 & 1110 & 1110 \\ 3 & 0 & 1 & 1 & 1101 & 1101 & 1101 & 1101 \\ 4 & 0 & 1 & 0 & 1010 & 1010 & 1010 & 1010 \\ 5 & 0 & 1 & 0 & 1011 & 1001 & 1001 & 1001 \\ 6 & 0 & 1 & 1 & 1010 & 1001 & 1001 & 1001 \\ 7 & 1 & 0 & 0 & 0100 & 0100 & 0100 & 0100 \\ 8 & 1 & 0 & 1 & 0021 & 0021 & 0021 & 0021 \\ 9 & 1 & 0 & 1 & 0022 & 0011 & 0011 & 0011 \\ 10 & 1 & 1 & 0 & 0110 & 0010 & 0010 & 0010 \\ 11 & 1 & 1 & 0 & 0111 & 0001 & 0001 & 0001 \\ 12 & 1 & 1 & 1 & 1111 & 0000 & 0000 & 0000 \\ 13 & 1 & 1 & 1 & 1111 & 0000 & 0000 & 0000 \\ 14 & 1 & 1 & 1 & 1111 & 0000 & 0000 & 0000 \\ 15 & 1 & 1 & 1 & 1111 & 0000 & 0000 & 0000 \end{array}$$

A	B	C	D	F ₄
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
1	0	0	0	1
1	0	0	1	1
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

$$F_4 = A'D'C'D + A'D'CD + A'DC'D + A'DC'D$$

$$F_4 = A'D'C'D + A'(B+C+D)$$

$$F_4 = A'(B+C+D) + A'(B+C+D)$$

$$F_4 = A \oplus (B+C+D)$$

$$F_1 = D$$

$$\begin{aligned} F_2 &= A'A'C'D + A'B'C'D + A'BC'D + A'BCD + A'BCD + A'BCD + A'BCD \\ F_2 &= \bar{C}D + C\bar{D} \end{aligned}$$

6. Design a combinational circuit with four input lines that represent a decimal digit in ACD and four output lines that generate a's complement of the input digit.

A'ACD	wxyz	A'ACD	wxyz
0 0000	1001	10 4020	XXXX
1 0001	1000	11 4021	XXX
2 0010	0111	12 4100	XX
3 0011	0110	13 4101	XX
4 0100	0101	14 4110	XX
5 0101	0100	15 4111	XX
6 0110	0011		
7 0111	0010		
8 1000	0001		
9 1001	0000		

A'ACD	wxyz	A'ACD	wxyz
0 0000	1001	10 4020	XX
1 0001	1000	11 4021	XX
2 0010	0111	12 4100	XX
3 0011	0110	13 4101	XX
4 0100	0101	14 4110	XX
5 0101	0100	15 4111	XX
6 0110	0011		
7 0111	0010		
8 1000	0001		
9 1001	0000		

A'ACD	wxyz	A'ACD	wxyz
0 0000	1001	10 4020	XX
1 0001	1000	11 4021	XX
2 0010	0111	12 4100	XX
3 0011	0110	13 4101	XX
4 0100	0101	14 4110	XX
5 0101	0100	15 4111	XX
6 0110	0011		
7 0111	0010		
8 1000	0001		
9 1001	0000		

AB	CD	C'D'	C'D	CD	CD'
A'B'				1	1
AB'				1	1
AB				X	X

$$y = c$$

$$Z = A'AC'D' + A'AC'D + A'AC'C'D' + A'AC'D + A'AC'D'$$

A'B'	CD	C'D'	C'D	CD	CD'
A'B'				1	1
AB'				1	1
AB				X	X

$$Z = D'$$

$$\begin{aligned} w &= A'AC'C'D' + A'AC'D \\ w &= A'AC'(C'D + D) = A'AC'C' \# \end{aligned}$$

$$x = A'AC'D' + A'AC'D + A'AC'C'D' + A'AC'D'$$

A'B'	CD	C'D'	C'D	CD	CD'
A'B'				1	1
AB'				1	1
AB				X	X

$$\therefore x = BC' + BC' \#$$

A'B'	CD	C'D'	C'D	CD	CD'
A'B'				1	1
AB'				1	1
AB				X	X

$$\therefore x = BC' + BC' \#$$

$$y = A'AC'C'D' + A'AC'D + A'AC'D' + A'AC'D$$

Binary to Excess-3

$$4+8+2 \rightarrow 111$$

1's complement
N \rightarrow 111
0 \rightarrow 000

1's complement
N \rightarrow 111
0 \rightarrow 000

$$2 = 5$$

Design a combinational CKT that converts a decimal digit from the 2021 code to BCD.

$$\begin{array}{c} 8021 \\ \hline 2424 \\ \text{BCD} \end{array}$$

$$A = wxyz^2 + wx^2yz$$

$$\begin{aligned} d &= wx'y^2z + w^2x^2y^2z + wxy^2y^2z + w^2xy^2z \\ &\quad + w^2xy^2z \end{aligned}$$

wx^2y^2z	wxy^2y^2z	w^2x^2y^2z	w^2xy^2z	w^2x^2y^2z
wx^2y^2z	x	x	x	x
wxy^2y^2z		1	1	
w^2x^2y^2z				x

wx^2y^2z	wxy^2y^2z	w^2x^2y^2z	w^2xy^2z	w^2x^2y^2z
wx^2y^2z	x	x	x	x
wxy^2y^2z		1	1	
w^2x^2y^2z				x

$$1 = w^2x^2y^2z + wxy^2y^2z$$

$$B = w^2x^2y^2z + wxy^2y^2z + w^2xy^2z$$

$$C = w^2x^2y^2z + w^2xy^2z$$

$$D = w^2x^2y^2z + w^2xy^2z + w^2xy^2z$$

$$E = w^2x^2y^2z + w^2xy^2z + w^2xy^2z$$

$$F = w^2x^2y^2z + w^2xy^2z + w^2xy^2z$$

$$G = w^2x^2y^2z + w^2xy^2z + w^2xy^2z$$

$$H = w^2x^2y^2z + w^2xy^2z + w^2xy^2z$$

$$I = w^2x^2y^2z + w^2xy^2z + w^2xy^2z$$

$$J = w^2x^2y^2z + w^2xy^2z + w^2xy^2z$$

$$P = 2$$

* Analysis procedure

Analysis of combinational circuit is reverse process of design of a combinational

- anal circuit.

1. Starts with logic diagram and analysis with finding Boolean function or expression with truth table of the given diagram. The procedure for analysis are

1. labelling arbitrary (any) symbols that all the function of input variables. Draw the Boolean function for each gate.

Other

2. repeat the process step ② until the output of the CKT is obtained in terms of input variables only.

* Example

T₁

T₂

T₃

T₄

T₅

T₆

T₇

T₈

T₉

T₁₀

T₁₁

T₁₂

T₁₃

T₁₄

F₁

F₂

F₃

F₄

F₅

F₆

F₇

F₈

F₉

F₁₀

F₁₁

F₁₂

F₁₃

F₁₄

F₁₅

F₁₆

F₁₇

F₁₈

F₁₉

F₂₀

F₂₁

F₂₂

F₂₃

F₂₄

F₂₅

F₂₆

F₂₇

F₂₈

F₂₉

F₃₀

Time	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Time

From the analysis we can see that it has three binary input A, B, C and two outputs F_1 and F_2 .

Boolean function of each gate output.

$$T_1 = A \cdot B \cdot C$$

$$T_2 = A + B + C$$

$$F_2 = A \cdot B + A \cdot C + B \cdot C$$

$$F_2' = \overline{A \cdot B + A \cdot C + B \cdot C}$$

similarly

$$T_3 = T_2 \quad F_2' = T_2 \cdot (A \cdot B + A \cdot C + B \cdot C)'$$

$$F_1 = T_1 + T_3$$

$$F_1 = A \cdot B \cdot C + T_3 = A \cdot B \cdot C + T_2 \cdot (A \cdot B + A \cdot C + B \cdot C)'$$

$$F_1 = A \cdot B \cdot C + (A + B + C) \cdot (A \cdot B + A \cdot C + B \cdot C)'$$

Therefore output

$$F_1 = A \cdot B \cdot C + (A + B + C) + (A \cdot B + A \cdot C + B \cdot C)'$$

$$F_2 = A \cdot B + A \cdot C + B \cdot C$$

A	B	C	F_2	F_2'	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	1	1	1
0	1	0	0	1	0	1	1	1
0	1	1	1	0	0	1	0	0
1	0	0	0	1	0	1	1	1
1	0	1	1	0	0	1	0	0
1	1	0	1	0	0	1	0	0
1	1	1	1	0	1	1	0	1

* Multilevel NAND and NOR Circuits

Combinational CKts are more frequently constructed with NAND or NOR Gates because they are most commonly used and readily available in integrated CKts. There are two types of multilevel CKts. They are

1. Multilevel NAND Circuits :→

Implementation of NAND gate for the Boolean function can be done by following steps

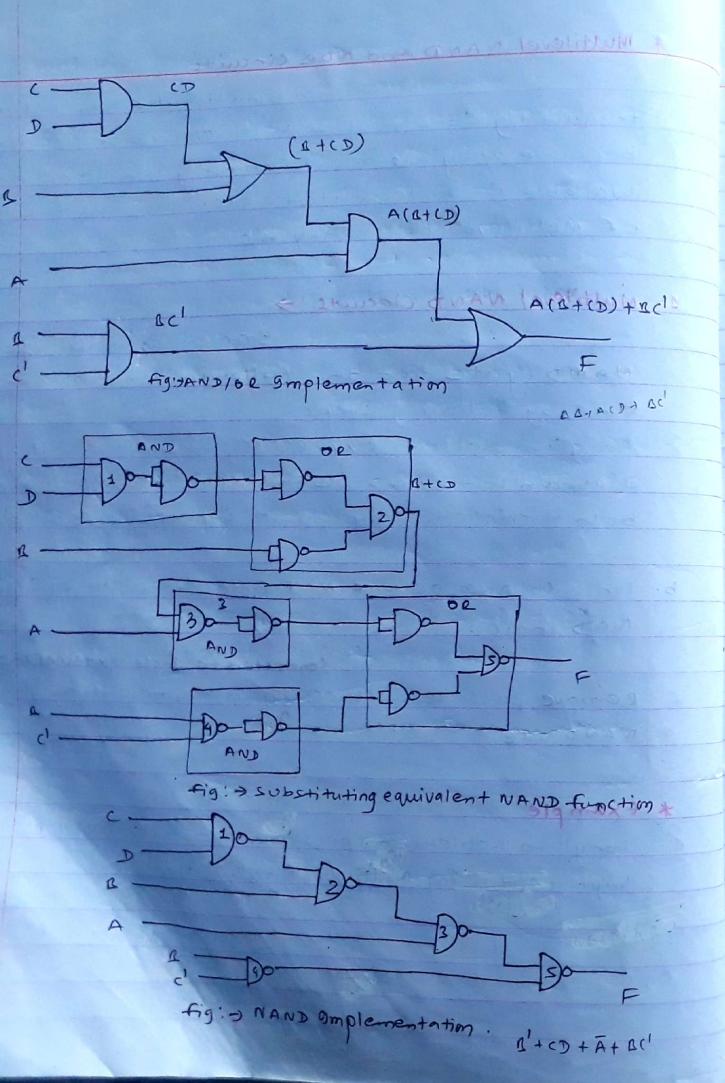
- a. For given Boolean expression, draw the logical diagram with AND, OR and NOT Gates. Assume both normal and complement inputs are available.
- b. For each AND, OR and NOT Gate, draw the equivalent NAND logic.
- c. Remove cascade inverters and represent by single NAND Gate.

Connected to single external input

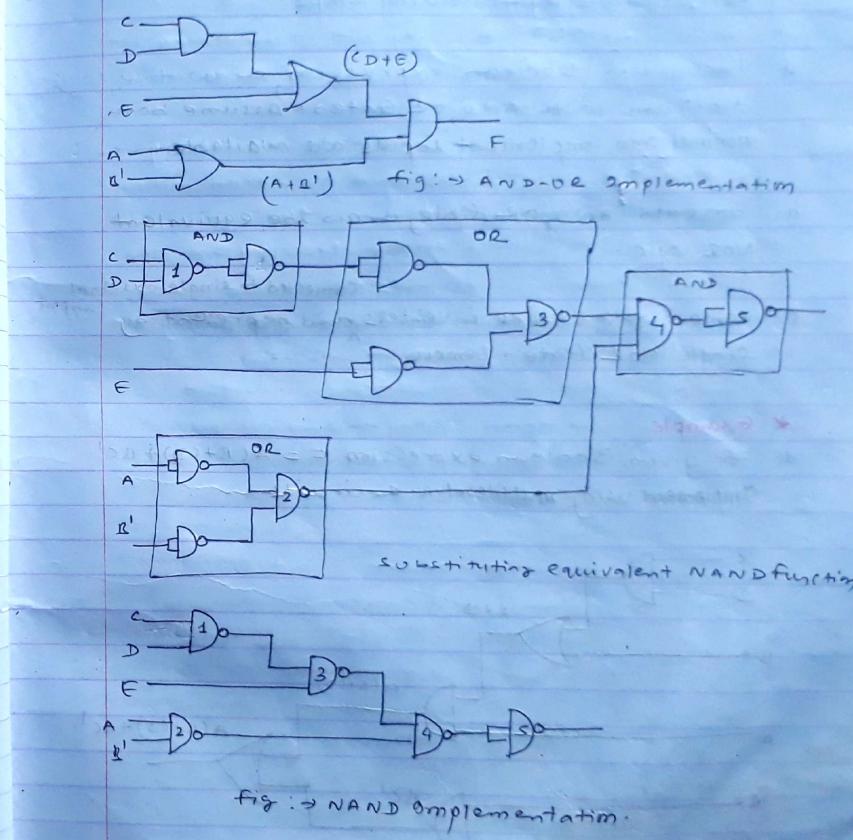
remove cascade inverters and represent by single NAND Gate.

* Example

1. For given Boolean expression $F = A(B+C'D) + AC'$ implement using Multilevel NAND Circuits



2. For given Boolean function $F = (A + C') \cdot ((CD + E))$. Implement using Multilevel NAND Circuits.



2. Multi level NOR circuits:

Implementation for multilevel NOR gate for the Boolean function can be done by following steps.

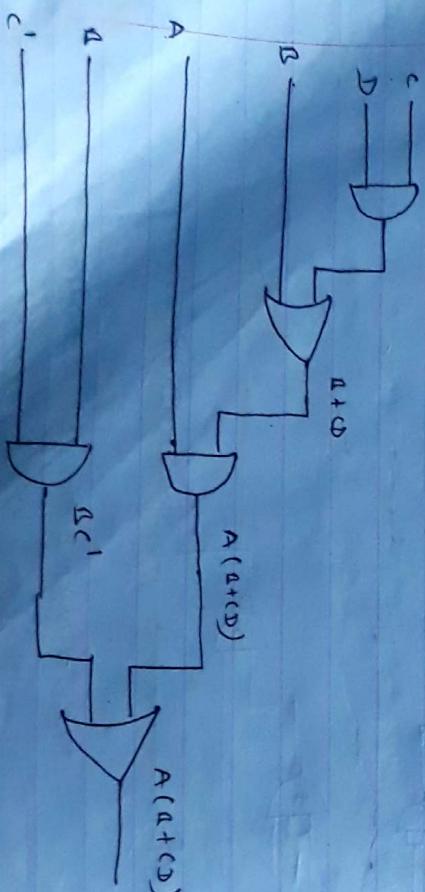
- For given Boolean expression draw the logic diagram with AND, OR Gates. Assume both normal and complement inputs are available.

- For each AND, OR Gate, draw the equivalent NOR logic.

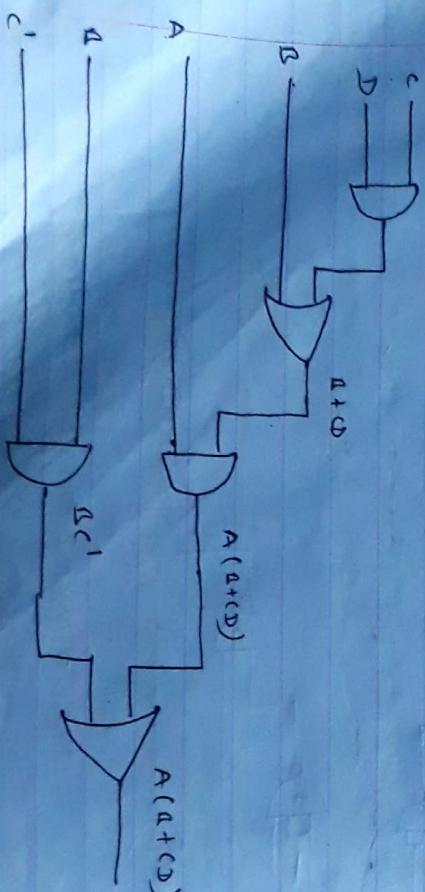
- Remove the cascade inverters and represent by single NOR gate.

* Example

- For given Boolean expression $F = A(B+C'D) + AC'$ implement using multilevel NOR circuits.

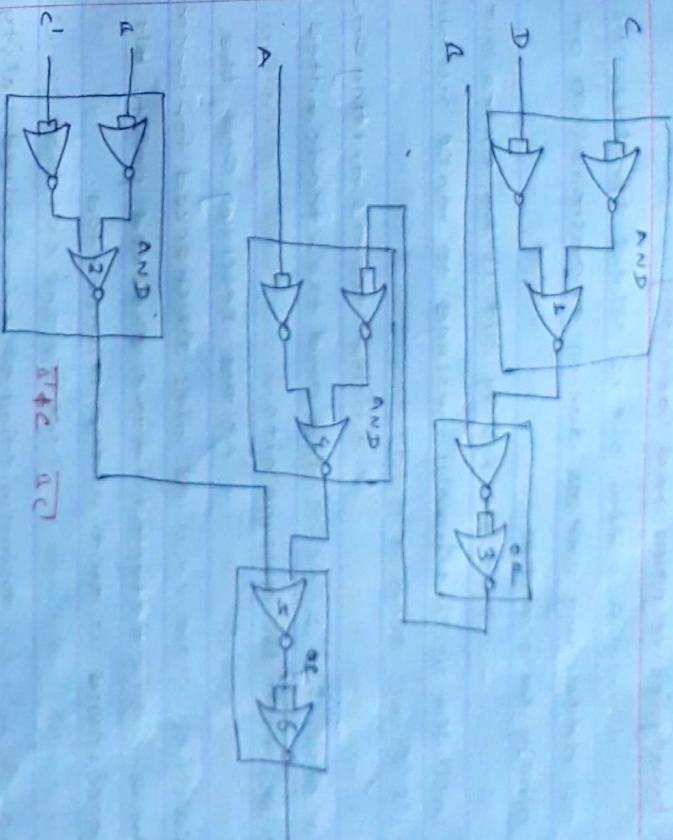


\rightarrow AND/OR Implementation



\rightarrow NOR Implementation

\rightarrow Substituting equivalent NOR ckt



\rightarrow Substituting equivalent NOR ckt

* Parity Generation and checking

A parity bit is used for the purpose of detecting errors during transmission of binary information.

A parity bit is an extra bit

included with a binary message to make number of 1's either even or odd.

At the transmitting end

a parity bit is generated and then transmitted along with the message bits.

At the receiving end, the

parity of whole received bits are checked for errors.

An error is detected if the checked parity bit doesn't correspond with the one transmitted.

Therefore the circuit that generates parity bit in transmitting end is called parity generator and the circuit that checks the parity bit at receiving end is called parity checker.

Design a combination circuit for 3-bit parity generator

and the circuit of a 4-bit parity checker using odd parity.

If x_1, x_2, x_3 are the three bit message transmitted

with the parity bit. The truth table for odd parity generator is

x_1	x_2	x_3	Parity bit generated by
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$\begin{aligned} p &= x_1'x_2' + x_1'x_2 + x_1x_2' + x_1x_2 \\ p &= x_1(x_2' + x_2) + x_1'(x_2 + x_2') \\ p &= x_1(y \oplus 2) + x_1'(y \oplus 2) \\ p &= x_1 \odot y \oplus 2 \end{aligned}$$

$$x_1 \odot x_2 = x_1 \oplus x_2$$

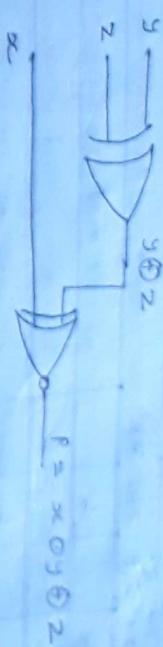


Fig: → 3 bit parity generator

x_1	x_2	x_3	Parity bit generated by	x_1	x_2	x_3	$p = x_1 \odot x_2 \oplus 2$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1
0	1	0	1	1	0	0	1
0	1	1	0	1	1	1	0
1	0	0	1	0	1	0	1
1	0	1	0	1	0	1	0
1	1	0	0	0	1	1	0
1	1	1	1	1	1	1	1

y_2	$y_2' \oplus 2$	$y_2' \oplus 2 + y_2$	$y_2' \oplus 2 + y_2 + y_2'$	$y_2' \oplus 2 + y_2 + y_2' + y_2$
0	1	1	0	1
1	0	0	1	0
0	1	0	1	1
1	0	1	0	0

→ three bit

message and the parity bit are transmitted to destination. Therefore, the receiver receives eight information. The output of the parity checker is known as error occurs i.e. when the number of 1's in four input is even. Therefore, truth table for the parity checker circuit

$$c = x'y'z'p + x'y'z'p' + x'y'z'p + x'y'z'p' + x'y'z'p + x'y'z'p' + x'y'z'p + x'y'z'p' + x'y'z'p + x'y'z'p$$

	x'y'z'p	x'y'z'p'								
00	00	01	11	10	11	10	00	01	11	10
00	00	01	11	10	11	10	00	01	11	10
01	01	11	10	00	01	11	01	11	10	00
11	11	10	00	01	11	10	11	10	00	01
10	10	00	01	11	10	00	11	10	01	11
11	11	10	00	01	11	10	11	10	01	11
10	10	00	01	11	10	00	11	10	01	11

$$c = x'y(zp + z'p') + x'y(z'p + zp') + x'y(z'p + zp) + xy(zp + z'p')$$

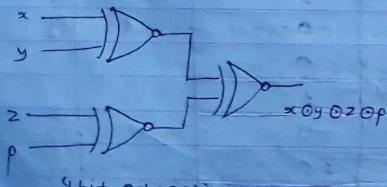
$$c = x'y(z \oplus p) + x'y(z \oplus p) + x'y(z \oplus p) + xy(z \oplus p)$$

$$c = (z \oplus p)(xy + x'y) + (z \oplus p)(x'y + xy)$$

$$c = (z \oplus y)(z \oplus p) + (z \oplus y)(z \oplus p)$$

$$c = (z \oplus y)(z \oplus p) + (\overline{z \oplus y})(\overline{z \oplus p})$$

$$c = (z \oplus y) \oplus z \oplus p$$



4 bit odd parity checker

2. Design a ckt for 3-bit parity generation & 4-bit parity checker using even parity.

If x, y, z are the three bit message transmitted with the parity bit. The truth table for even parity genera-

- tor is

x	y	z	parity bit (generator)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$p = x'y'z + x'y'z' + x'y'z + x'y'z'$$

x	y	z	y	z
0	0	0	1	1
0	0	1	1	1

$$p = z(xy + x'y) + z'(x'y + xy)$$

$$p = z(z \oplus y) + z'(z \oplus y)$$

$$p = z(z \oplus y) + z'(z \oplus y)$$

$$p = z \oplus y$$

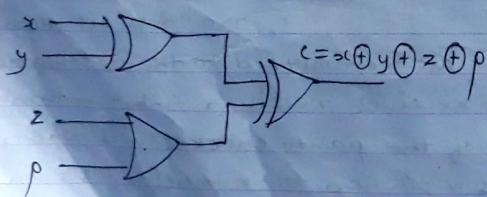
$$p = z(x \oplus y) + z'(x \oplus y)$$

$$p = z(x \oplus y)' + z'(x \oplus y)$$

$$p = z \oplus x \oplus y = x \oplus y \oplus z$$

The three bit message and the parity bit are transmitted to the destination. Therefore the receiver receives 4-bit information. The output c of the parity checker is 1 when an error occurs i.e. when the number of 1's in four input is odd. Therefore the truth table for even parity checker ckt is

$x \ y \ z \ p$	c	Parity error check
0 000	0	
1 000	1	
2 0010	2	
3 0011	0 ✓	
4 0100	1	
5 0101	0	
6 0110	0 ✓	
7 0111	1	
8 1000	1	
9 1001	0	
10 1010	0	$c = x'y'(z'p + z'p') + x'y(zp + z'p') + xy'(zp + z'p')$
11 1011	1	$+ xy(z'p + zp')$
12 1100	0	$c = x'y(z \oplus p) + x'y(z \otimes p) + xy(z \otimes p) + xy(z \oplus p)$
13 1101	1	$c = (z \oplus p)(xy + x'y) + z \otimes p(xy + x'y)$
14 1110	1	$c = (z \oplus p)(x \otimes y) + z \otimes p(x \oplus y)$
15 1111	0	$c = (x \oplus y)(z \oplus p) + (x \oplus y)(z \oplus p)$
		$c = x \oplus y \oplus z \oplus p$



2. Design even parity generator when a 3-bit message contains cyclic code.

$x \ y \ z$	Cyclic code	parity bit generated (P)
0 000	000	0
1 001	100	1
2 010	001	1
3 011	101	0
4 100	010	1
5 101	110	0
6 110	011	0
7 111	111	1

$$p = xy'z' + x'y'z + x'y'z' + xyz$$

$$p = y'(xz' + x'z) + y(xz + x'z') \Rightarrow x \otimes 02$$

$$p = y'(x \oplus z) + y(\overline{x \oplus z})$$

$$p = y \oplus x \oplus z$$

