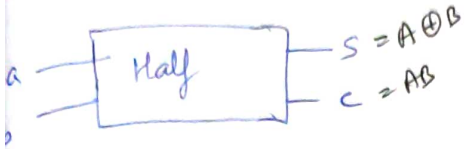
Unit-2

Half adder

It is a combination circuit which adds only 2 bits, and produces called sum and carry.



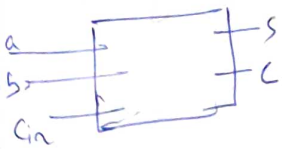
I/P		O/P	
a	b	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Draw ~~stair~~ using Basic gate & NAND gate also

Full adder

It is a combinational ckt which adds 3 bits ~~giving~~ x, y, and C_{in} from another circuit.

Full adder produces 2 outputs sum & carry

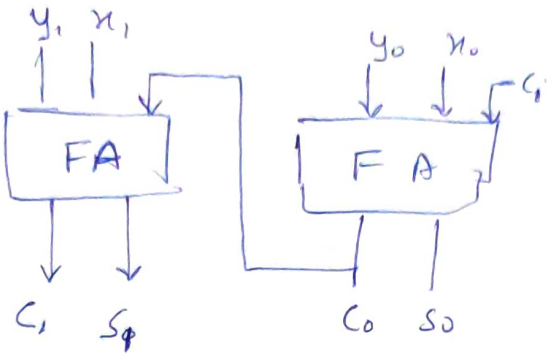


I/P			O/P	
a	b	C	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Add 2 numbers of 2 digit (Parallel adder)

Num1 = ~~x_0 y~~ x_1 x_0

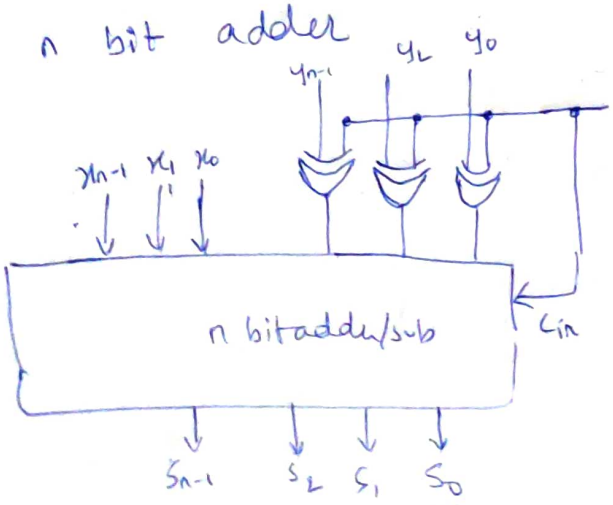
Num 2 = y_1 y_0



Parallel adder adds any 2 number, binary digit if there are 2 number with 4 digits each. It requires 4 full adder connected in cascade. Such adder circuits are called as parallel adder (or Ripple carry adder).

Q) difference btw ~~Sequencial~~ & combination logic gates.
Sequencial

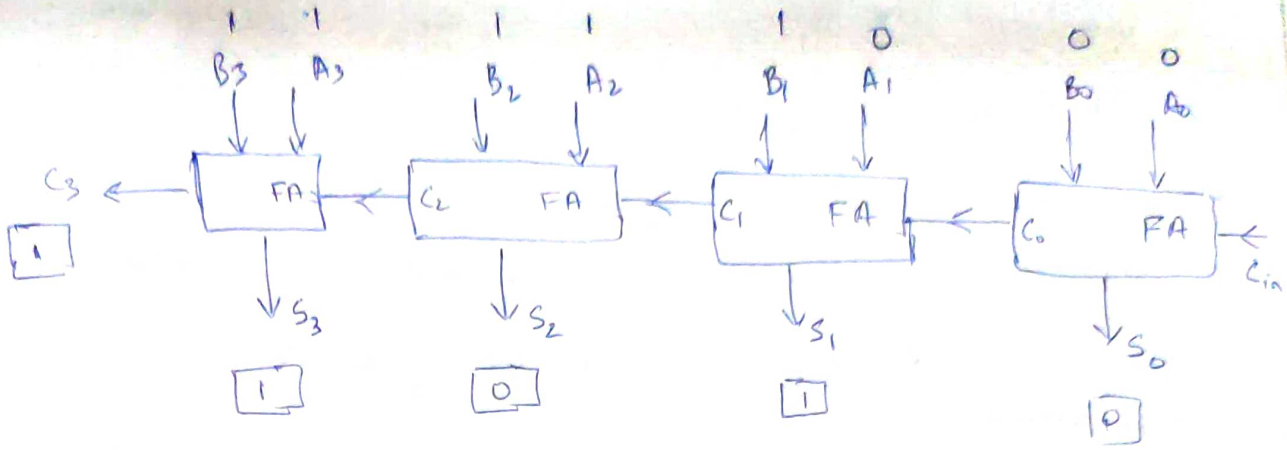
a) Advantages & disadvantages of sequencial & combination gates



C=0 ; S = a + b

C=1 ; S = a + (b's 2 compliment)
= a − b

2 Numbers 4 bits Each / Ripple / parallel carry adder



Advantage:

- easy to construct.

disadvatage:

-)
-) There might be delay in carry from 1 stage to another stage.
-) If number is large, say 16 bits then 16 full adders shd be cascaded which may introduce more propagation delay.

A MSB $A_3\ A_2\ A_1\ A_0$ LSB 1 1 0 0

$B_3\ B_2\ B_1\ B_0$ 1 1 1 0

1 1 0 1 0

Subtractor

① A 1 0 0 1

B − 1 0 0 0 → 1's → 0 1 1 1

+ 1

2's ↓

1 0 0 0

1 0 0 1

+ 1 0 0 0

1 0 0 0 1

② 1 0 0 1 → 1 0 0 1

1 0 1 1 → 1's → 0 1 0 0

+ 1

0 1 0 1 → 2's → + 0 1 0 1

1 1 1 0 → 1's → 0 0 0 1

+ 1

0 0 1 0

In digital systems subtraction is done using 2's compliment method.

eg: B subtract B from A

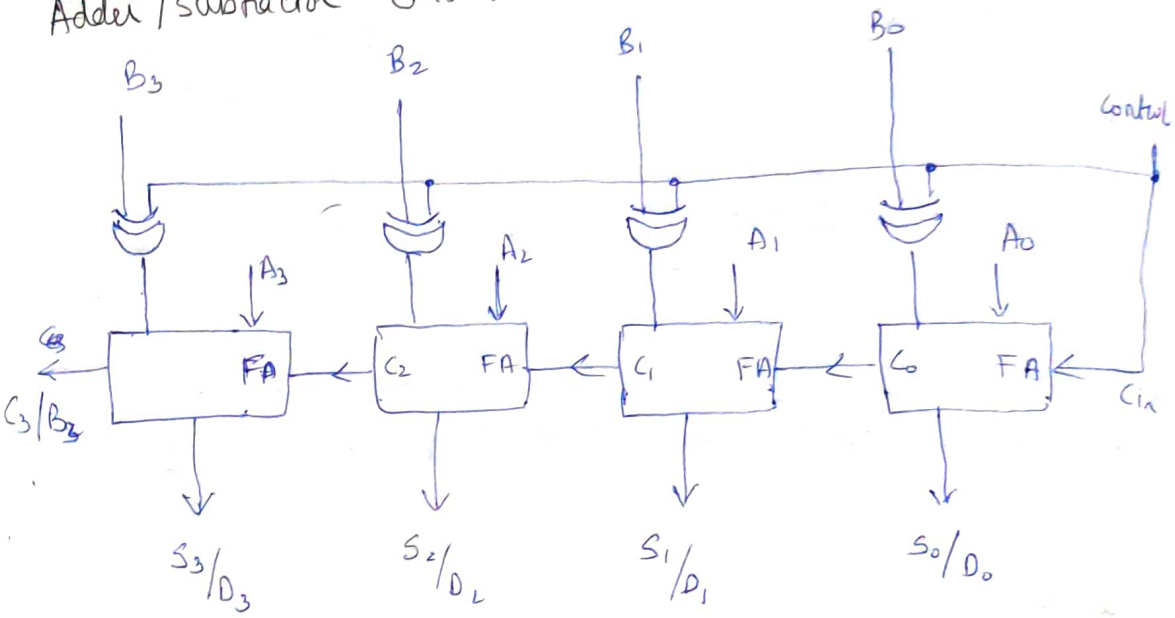
→ generate 1's compliment of B

→ generate 2's compliment of B by adding 1 in LSB.

→ add number A with two's compliment of B is obtain final answer.

→ If may be in true form, else take 2's compliment

Adder / subtractor circuit



If control = 0, then Addition

control = 1, subtraction

eg: This is Adder/subtractor ckt

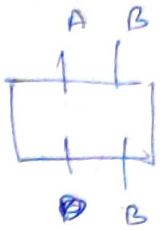
if control is 1 the X-OR gates do 1's complement and 1 is added in LSB to get 2's compliment.

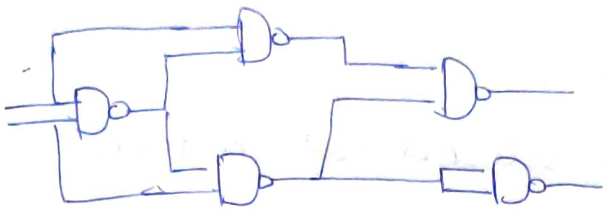
during subtraction the final carry is discarded.

HW: Half subtractor

Half subtractor is a combination circuit which subtracts B from A.

Full subtractor subtracts B from A also borrows in from A. It generate differ & borrow.



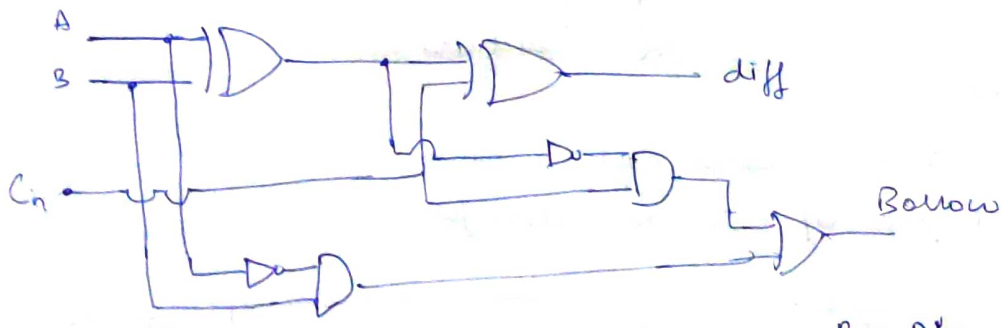


A	B	diff	borw
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Full subtractor subtracts generates di

Full Subtractor

A	B	Cin	diff	Bor
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

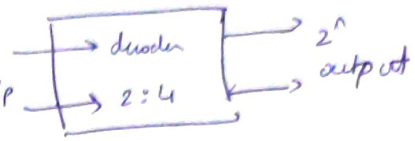


$B = A'C_{in} + A'B + BC_{in}$

Decoder

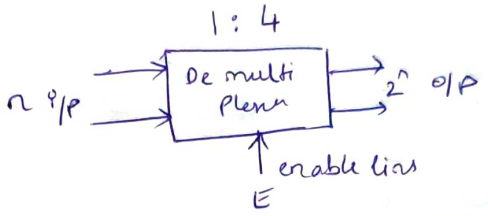
decoder is a combination ckt which has n i/p and 2^n o/p decodes it selects & assigns code one of the input and assigns it to anyone of the output.

eg: If there are 2 inputs unique codes n i/p only 1 is selected at a time



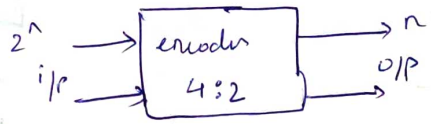
De multiplexer

anyone of i/p code is selected at a time.



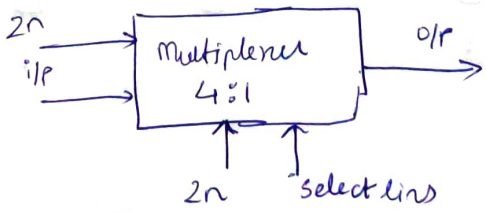
Encoder

The function of encoder is opp of decoder assigns a code for any 1 of the input at a time. There is special encoder called as priority encoder.



Multiplexer

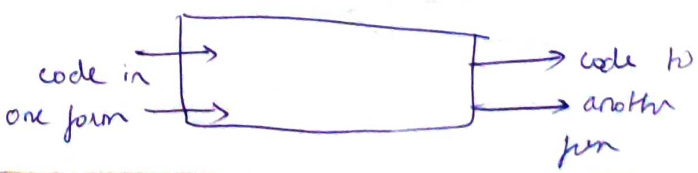
It is a combination ckt which directs any one of input to output depends on state of select line



code converters

code convert is a combi ckt which converts 1 form of code to another form

eg: BCD to 7 segment
gray to BCD



Decoder

$n = 2$

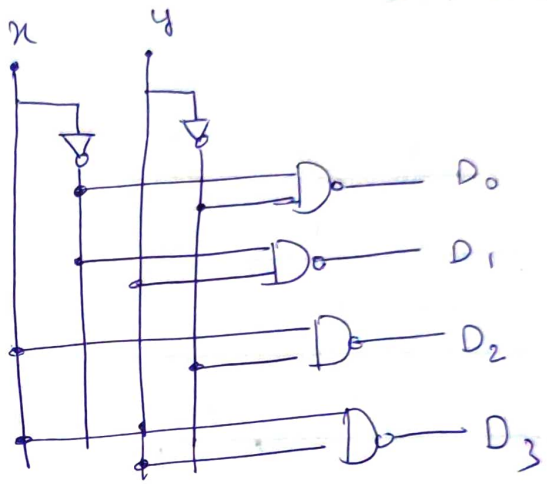
IP		OP			
x	y	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$D_0 = \bar{x}\,\bar{y}$

$D_1 = \bar{x}\,y$

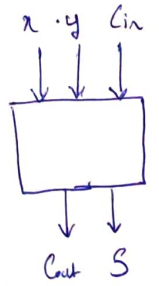
$D_2 = x\bar{y}$

$D_3 = x\,y$



Demultiplexer

implement a full adder using Decoder

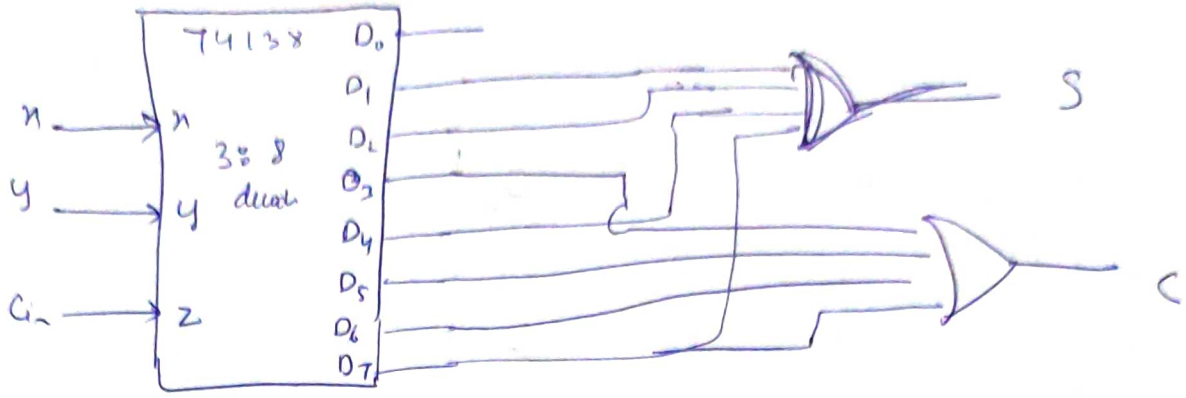


x	y	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = \Sigma m\,(1, 2, 4, 7)$

$C_o = \Sigma m\,(3, 5, 6, 7)$

$S = m(1,2,4,7)$ $C = m(3,5,6,7)$



o)

$$y = \bar{a}b + cd$$

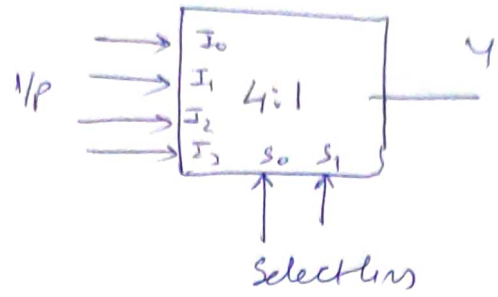
	$\bar{c}\bar{d}$	$\bar{c}d$	cd	$c\bar{d}$
$\bar{a}\bar{b}$	0 (0)	0 (1)	1 (3)	0 (2)
$\bar{a}b$	1 (4)	1 (5)	1 (7)	1 (6)
ab	0 (12)	0 (13)	1 (15)	0 (14)
$a\bar{b}$	0 (8)	0 (9)	1 (11)	0 (10)

$$Y = \Sigma m(4,5,6,7,3,15,11)$$

Applicati:

- higher order decoder can be constructed using lower order decoder
- boolean exp can be implement
- combination ckt adder, subtractor
- mostly integrated ckt have enable line, therefore they can be treated as demultiplexer also

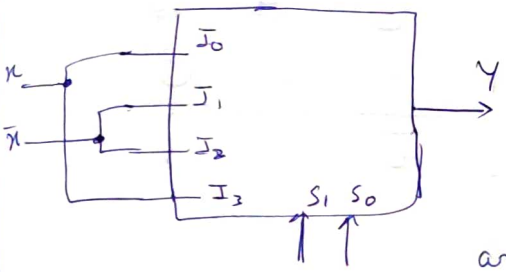
multiplexer



	S_1	S_0	y
m_0	0	0	I_0
m_1	0	1	I_1
m_2	1	0	I_2
m_3	1	1	I_3

	I_0	I_1	I_2	I_3
$\bar{x}$	0	(1)	(2)	3
x	(4)	5	6	(7)
	x	$\bar{x}$	$\bar{x}$	x

x	y	C_i	S	C
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

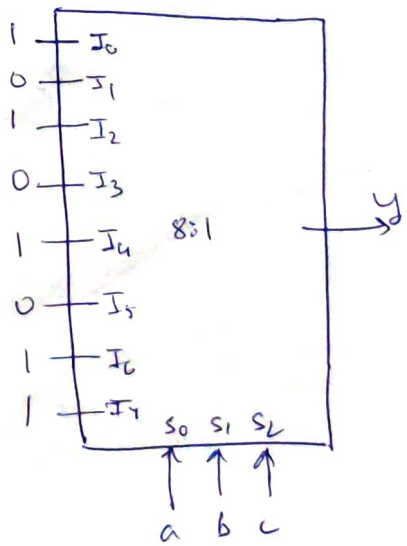


multiplexing is process of combining the different comm channels and transfer it over a single transmission medium called multiplexer

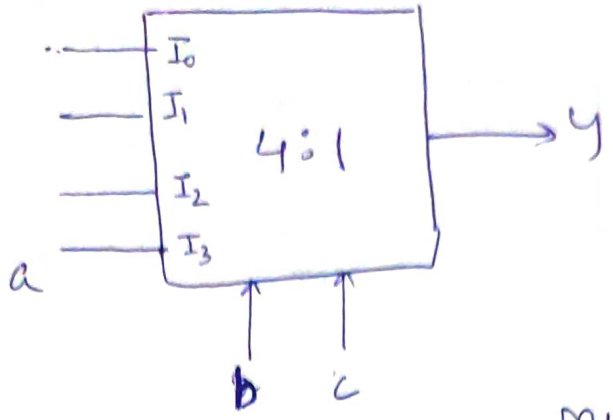
~~Implement Carry using MUX~~

Q) $y = \Sigma m(0,2,4,6,7)$ using 8:1 MUX

	a	b	c	y
m_0	0	0	0	1
m_1	0	0	1	0
m_2	0	1	0	1
m_3	0	1	1	0
m_4	1	0	0	1
m_5	1	0	1	0
m_6	1	1	0	1
m_7	1	1	1	1



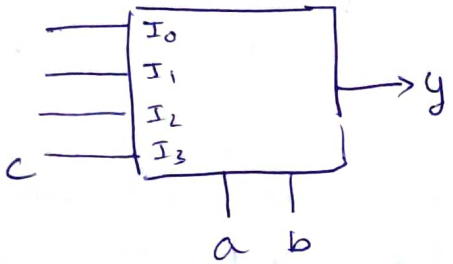
If a is input



MUX Reduction Table

	I_0	I_1	I_2	I_3
$\bar{a}$	0	1	2	3
a	4	5	6	7
	1	b	1	a

If c is input



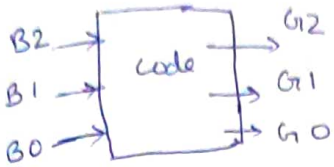
	I_0	I_1	I_2	I_3
$\bar{c}$	0	2	4	6
c	1	3	5	7
	$\bar{c}$	$\bar{c}$	$\bar{c}$	1

If b is input

	I_0	I_1	I_2	I_3
$\bar{b}$	0	1	4	5
b	2	3	6	7
	1	0	1	b

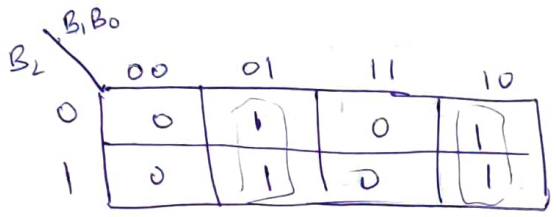
Design Binary to Grey code converter

3bits



I/P 2^2 B_2	2^1 B_1	2^0 B_0	O/P G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

G_0

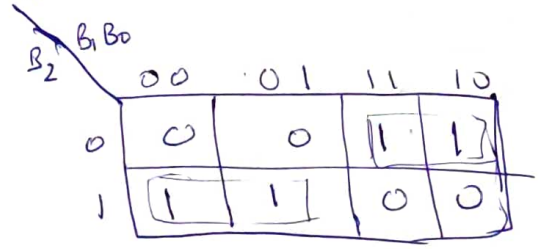


$G_2 = B_2$

$G_0 = \overline{B_1} B_0 + B_1 \overline{B_0}$

$G_0 = B_1 \oplus B_0$

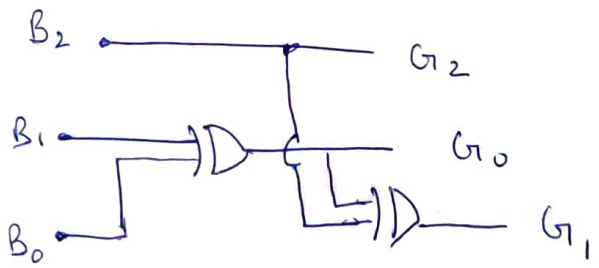
$G_1 =$



~~$B_1 = G_1$~~

$G_1 = B_1 \overline{B_2} + \overline{B_1} B_2$

$G_1 = B_1 \oplus B_2$



Similary a 4bit binary to grey can be done

grey to binary

G_2 \ G_1G_0	00	01	11	10
0	0	0	1	1
1	1	1	0	0

$$B_1 = \overline{G_2}\, G_1 + G_2\, \overline{G_1}$$

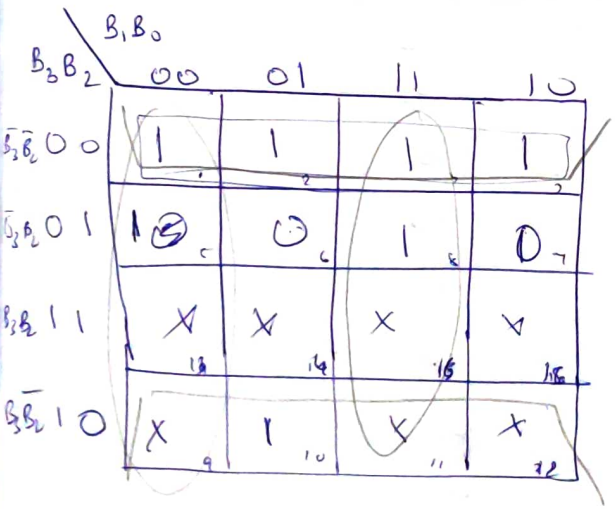
Binary to BCD

a	b	c	d	B_2	B_2	B_1	B_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	0	0	1	1
0	1	0	0	0	1	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	0	x	x	x	x
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

BCD to 7 segment

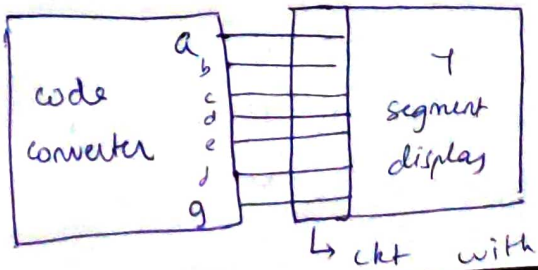
BCD i/p	a	b	c	d	e	f	g	h
0 0 0 0	1	1	1	1	1	1	0	
0 0 0 1	0	1	1	0	0	0	0	
0 0 1 0	1	1	0	1	1	0	1	
0 0 1 1	1	1	1	1	0	0	0	
0 1 0 0	1	1	1	0	0	1	1	
0 1 0 1	1	0	1	1	0	1	1	
0 1 1 0	1	0	1	1	1	1	1	
0 1 1 1	1	1	1	0	0	0	0	
1 0 0 0	1	1	1	1	1	1	0	
1 0 0 1	1	1	1	0	0	1	1	

For B



$$b = \bar{B}_2 + \bar{B}_1 \bar{B}_0 + B_1 B_0$$

Similarly K-MAP is plotted for output a, c, d, e, f, g.
Once the expression is obtained implementation is done using suitable logic gates

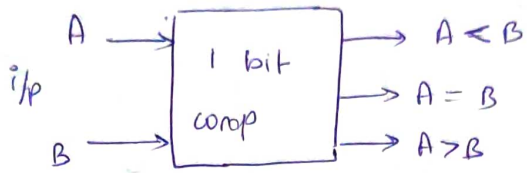


↳ ckt with resistor

COMPARATOR

Design 1 bit comparator

comparator is combi ckt which compares 2 numbers a&b and produce output of 2 numbers $a=b$, $a<b$ & $a>b$.



A	B	A = 0	A < B	A > B
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0
		XNOR	$\overline{A}.B$	$A\overline{B}$

2 bit

A_1 A_0	B_1 B_0	A = B	A < B	A > B
0 0	0 0	1	0	0
0 0	0 1	0	1	0
0 0	1 0	0	1	0
0 0	1 1	0	1	0
0 1	0 0	0	0	1
0 1	0 1	1	0	0
0 1	1 0	0	1	0
0 1	1 1	0	1	0
1 0	0 0	0	0	1
1 0	0 1	0	0	1
1 0	1 0	1	0	0
1 0	1 1	0	1	0
1 1	0 0	0	0	1
1 1	0 1	0	0	1
1 1	1 0	0	0	1
1 1	1 1	1	0	0

For A > B :-

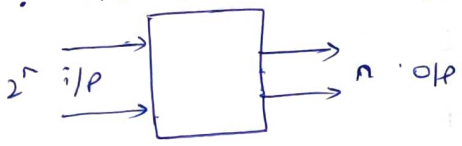
A_1A_0 \ B_1B_0	$\bar{B}_1\bar{B}_0$ (00)	$\bar{B}_1B_0$ (01)	B_1B_0 (11)	$B_1\bar{B}_0$ (10)
$\bar{A}_1\bar{A}_0$ (00)	0 (0)	0 (1)	0 (3)	0 (2)
$\bar{A}_1A_0$ (01)	1 (4)	0 (5)	0 (7)	0 (6)
A_1A_0 (11)	1 (12)	1 (13)	0 (15)	1 (14)
$A_1\bar{A}_0$ (10)	1 (8)	1 (9)	0 (11)	0 (10)

$$A>B = A_1\bar{B}_1 + A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0$$

similarly a n bit comparator can be designed

Encoder

It is a combinational circuit with 2^n inputs and n outputs. The function is opposite of decoder



for every single input it assigns a unique code

eg: Input I_3 code is 00
Input I_1 code is 10

This is called as 4:2 encoder

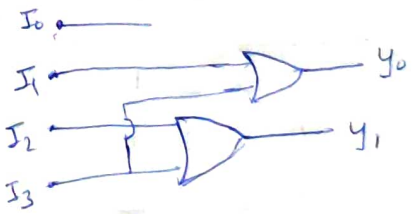
Similarly 8:3 & 16:4 encoder can be designed

Table

I_3	I_2	I_1	I_0	x	y
0	0	0	1	0	0
0	0	1	0	0	1
0	1	1	0	1	0
1	0	0	1	1	1

$$x = I_1 + I_0$$
$$y = I_2 + I_0$$

$$x = I_2 + I_3$$
$$y = I_1 + I_3$$



The same encoder can be converted to priority encoder

eg: If I_3 has to get highest priority the code assigned I_3 has highest value. I_0 ____

These encoders can be designed with simple OR gate.