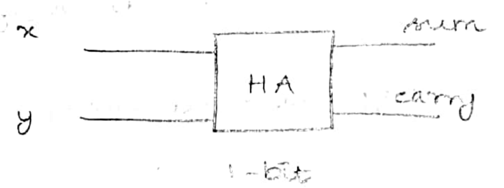
Unit : II Combinational circuits

Digital Logic circuits

Combinational	Sequential
output depends only on present input	Flip flops
→ Half adder	Register
→ Full adder	Counter
→ Ripple carry adder	Shift register
→ Carry look ahead adder	ALU
→ Multiplexer	Memory
→ Demultiplexer	Control unit
→ Coder	
→ Decoder	
→ Comparitor	

Half adder :

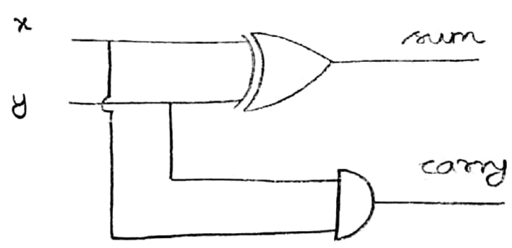


x	y	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

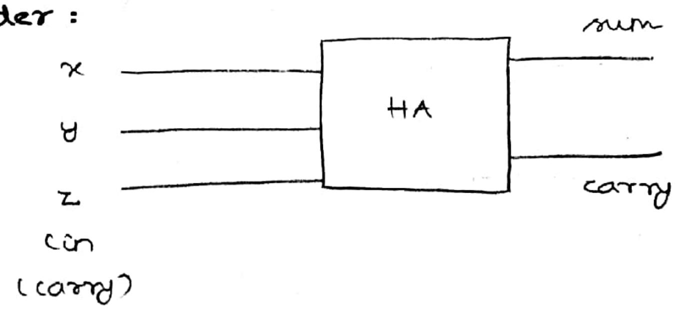
sum of minterms

$$\text{sum} = x\bar{y} + \bar{x}y = x \oplus y$$

$$\text{carry} = xy$$



Full adder :

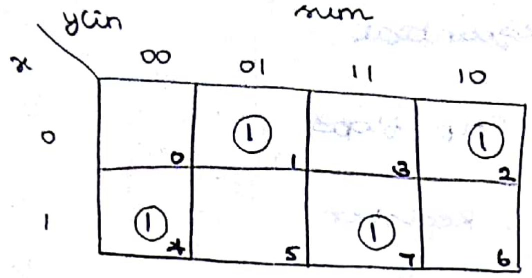


$sum = \bar{x}\bar{y}c_{in} + \bar{x}y\bar{c}_{in} + x\bar{y}\bar{c}_{in} + xyc_{in}$

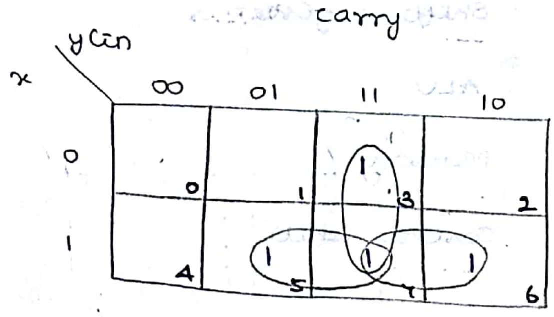
$carry = xc_{in} + xy + yc_{in}$

	x	y	cin	sum	carry
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

sum



carry



$carry = xy + (x+\bar{x})yc_{in} + (y+\bar{y})xc_{in}$

$A + AB = A$

$= xy + xyc_{in} + \bar{x}yc_{in} + xyc_{in} + x\bar{y}c_{in}$

$= xy + (\bar{x}y + x\bar{y})c_{in}$

$= xy + c_{in}(x \oplus y)$

$carry = xy + c_{in}(x \oplus y)$

$sum = xy(\quad + \bar{c}_{in}) + \bar{x}\bar{y}c_{in} +$

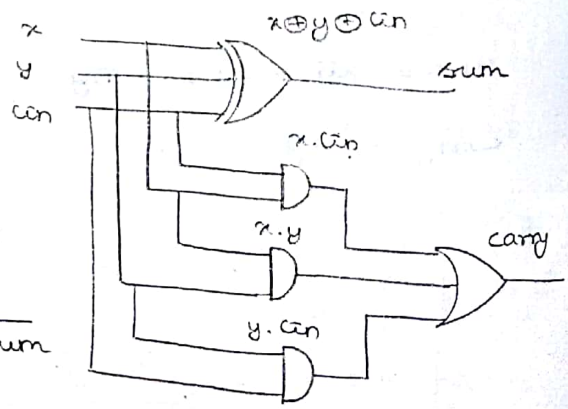
$= \bar{c}_{in}(\bar{x}y + x\bar{y}) + c_{in}(xy + \bar{x}\bar{y})$

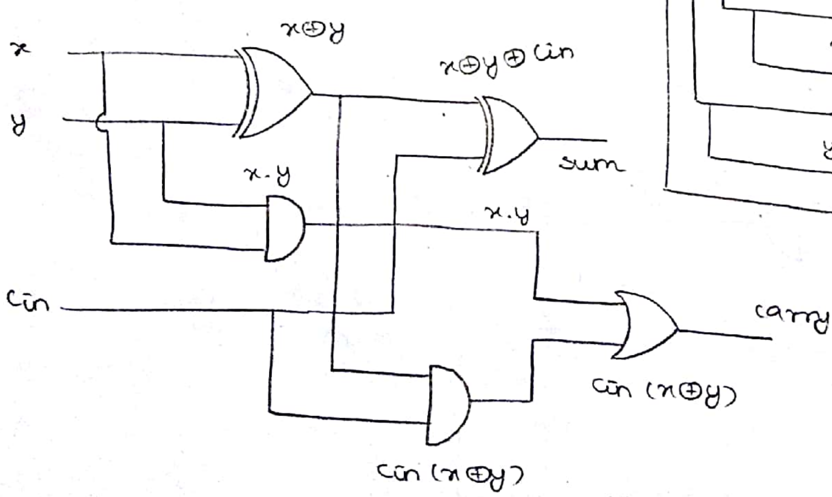
$= \bar{c}_{in}(x \oplus y) + c_{in}(x \odot y)$

$= \bar{c}_{in}(A) + c_{in}(\bar{A})$

$= \bar{c}_{in} \oplus A$

$sum = \bar{c}_{in} \oplus x \oplus y$





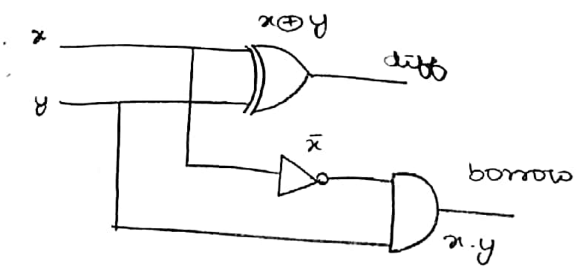
Half subtractor :

x	y	diff	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

	x	y	cin	diff	borrow
0	0	0	0	0	0
1	0	0	1	1	1
2	0	1	0	1	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	0
6	1	1	0	0	0
7	1	1	1	1	1

sum $= \bar{x}y + \bar{y}x = x \oplus y$

borrow $= \bar{x}y$



difference

x \ y cin	00	01	11	10
0	0	1 (1)	3	1 (2)
1	1 (4)	5	1 (7)	6

Full subtractor :

borrow

x \ y cin	00	01	11	10
0	0	1 (1)	1 (3)	1 (2)
1	4	5	1 (7)	6

difference $= \bar{x}\bar{y}\,cin + \bar{x}y\,\overline{cin} + x\bar{y}\,\overline{cin} + xy\,cin$

borrow $= \bar{x}\,cin + \bar{x}y + y\,cin$

difference $= cin(xy + \bar{x}\bar{y}) + \overline{cin}(\bar{x}y + x\bar{y})$

$= cin(x \odot y) + \overline{cin}(x \oplus y)$

$= cin\,\bar{A} + \overline{cin}\,A$

$= cin \oplus A$

$= cin \oplus x \oplus y$

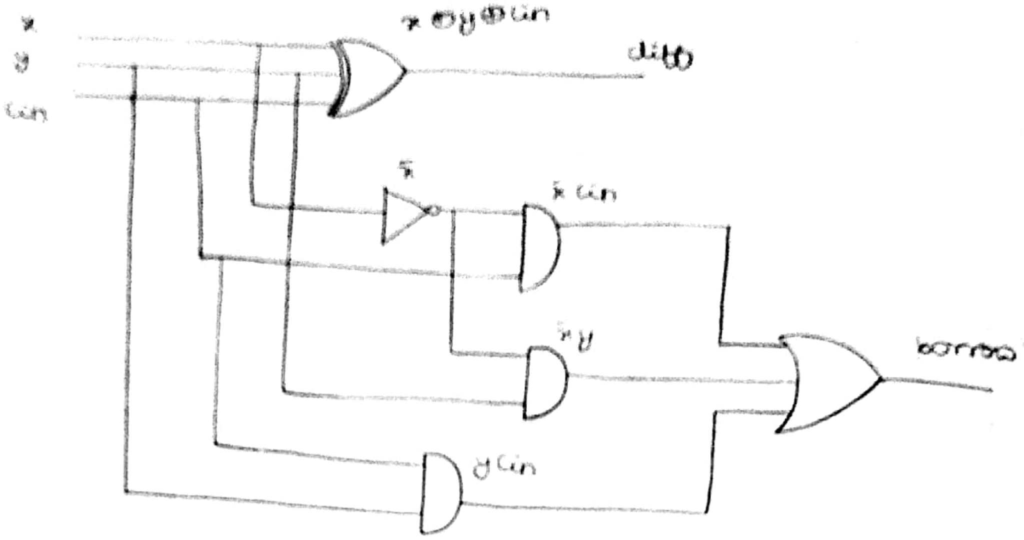
borrow $= \bar{x}\,cin + \bar{x}y + y\,cin$

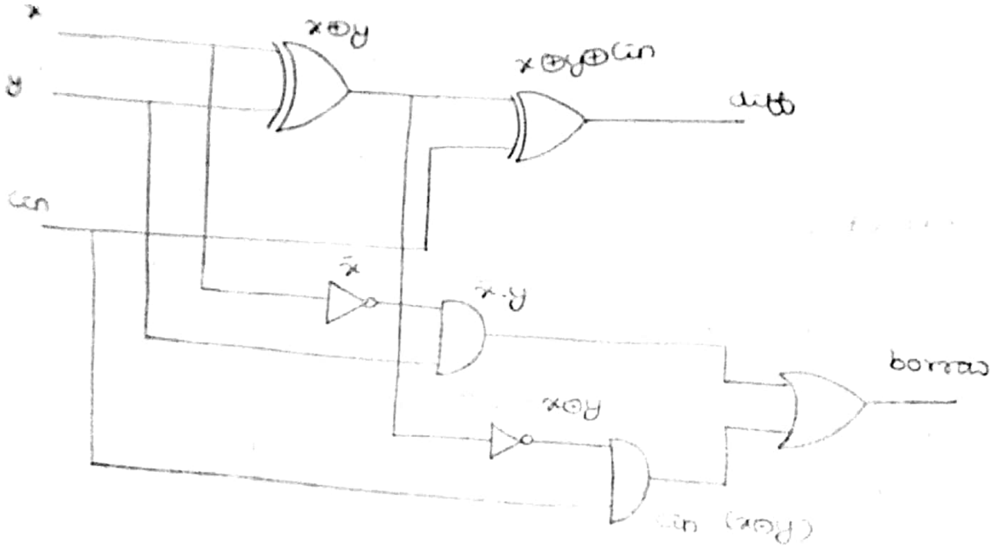
$= \bar{x}y + \bar{x}(cin)(y+\bar{y}) + y\,cin(x+\bar{x})$

$= \bar{x}y + \bar{x}y\,cin + \bar{x}\bar{y}\,cin + xy\,cin + \bar{x}y\,cin$

$= \bar{x}y + cin(\bar{x}\bar{y} + xy)$

$= \bar{x}y + C_{in}(x \odot y)$





Code converters

1. BCD to Binary
2. BCD to Excess - 3
3. Binary to gray
4. Gray to binary
5. Binary to BCD

BCD				BINARY			
A_3	A_2	A_1	A_0	B_3	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
9 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

B_3

A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	x	x	x	x
10	1	1	x	x

$B_3 = A_3$

E_3

A_3A_2 \ A_1A_0	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

$E_3 = A_3 + A_2A_0 + A_2A_1$

	BCD				Excess - 3			
	A_3	A_2	A_1	A_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
	0	1	0	0	0	1	1	1
9	0	1	0	1	1	0	0	0
	0	1	1	0	1	0	0	1
	0	1	1	1	1	0	1	0
	1	0	0	0	1	0	1	1
	1	0	0	1	1	1	D	0

E_2

A_3A_2 \ A_1A_0	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	x	x	x	x
10	0	1	x	x

$E_2 = A_3A_0 + \bar{A}_2A_1$

$A_2\bar{A}_1\bar{A}_0 + A_3A_0$

$= A_3A_0 + \bar{A}_2A_1 +$

$A_2\bar{A}_1\bar{A}_0$

E_1

A_3A_2 \ A_1A_0	00	01	11	10
00	1 (0)	0 (1)	1 (3)	0 (2)
01	1 (4)	0 (5)	1 (7)	0 (6)
11	x (12)	x (13)	x (15)	x (14)
10	1 (8)	0 (9)	x (11)	x (10)

E_0

A_3A_2 \ A_1A_0	00	01	11	10
00	1 (0)	0 (1)	0 (3)	1 (2)
01	1 (4)	0 (5)	0 (7)	1 (6)
11	x (12)	x (13)	x (15)	x (14)
10	1 (8)	0 (9)	x (11)	x (10)

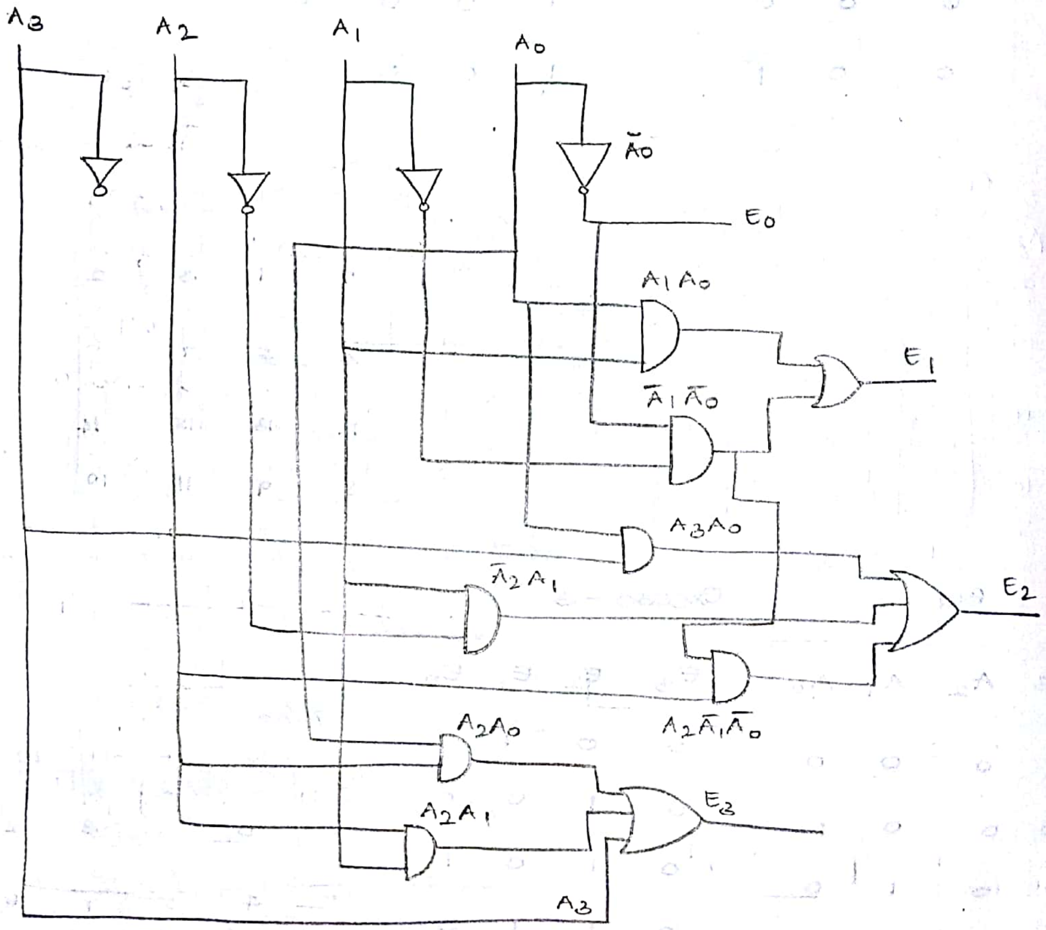
$E_1 = \bar{A}_1\bar{A}_0 + A_1A_0$

$= A_1 \odot A_0$

$E_0 = \bar{A}_0$

$E_3 = A_3 + A_2A_0 + A_2A_1$

$E_2 = A_3A_0 + \bar{A}_2A_1 + A_2\bar{A}_1\bar{A}_0$



$$B_0 = \bar{G}_3\bar{G}_2(G_1 \oplus G_0) + \bar{G}_3G_2(G_1 \odot G_0) + G_3G_2(G_1 \oplus G_0) + G_3\bar{G}_2(G_1 \odot G_0)$$

$$= (G_1 \oplus G_0)(\bar{G}_3\bar{G}_2 + G_3G_2) + (G_1 \odot G_0)(\bar{G}_3G_2 + G_3\bar{G}_2)$$

$$= (G_1 \oplus G_0)(G_3 \odot G_2) + (G_1 \odot G_0)(G_3 \oplus G_2)$$

$$= G_1 \oplus G_0 \oplus G_3 \oplus G_2 = G_0 + B_1$$

	Gray				Binary			
	G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	~~0~~1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	~~0~~1	0	0	1	0	0
5	0	1	~~0~~1	1	0	1	0	~~0~~1
	~~0~~	~~0~~1	~~1~~	~~0~~				
	~~0~~	~~1~~	~~1~~	~~1~~				
6	0	1	0	1	0	1	1	0
7	0	1	0	0	0	1	1	1
8	~~0~~1	1	0	0	~~0~~1	0	0	0
9	1	1	0	1	1	0	0	1
10	1	1	1	1	1	0	1	0
11	1	1	1	0	1	0	1	1
12	1	0	1	0	1	1	0	0
13	1	0	1	1	1	1	0	1
14	1	0	0	1	1	1	1	0
15	1	0	0	0	1	1	1	1

$B_3 = G_3$

$B_2 = \bar{G}_3 G_2 + G_3 \bar{G}_2 = G_3 \oplus G_2$

$B_1 = \bar{G}_3 \bar{G}_2 G_1 + \bar{G}_3 G_2 \bar{G}_1 + G_3 G_2 G_1 + G_3 \bar{G}_2 \bar{G}_1$

$= \bar{G}_2 (\overset{A}{G_1 \oplus G_3}) + G_2 (\overset{\bar{A}}{G_1 \odot G_3})$

$= G_2 \oplus G_1 \oplus G_3$

$= B_2 \oplus G_1$

$B_0 = \bar{G}_3 \bar{G}_2 \bar{G}_1 G_0 + \bar{G}_3 \bar{G}_2 G_1 \bar{G}_0 + \bar{G}_3 G_2 \bar{G}_1 \bar{G}_0 + G_3 G_2 G_1 G_0 + G_3 G_2 \bar{G}_1 G_0 + G_3 G_2 G_1 \bar{G}_0 + G_3 \bar{G}_2 \bar{G}_1 \bar{G}_0 + G_3 \bar{G}_2 G_1 G_0$

B_3

A_3A_2 \ A_1A_0	00	01	11	10
00	0 (0)	0 (1)	0 (3)	0 (2)
01	0 (4)	0 (5)	0 (7)	0 (6)
11	1 (12)	1 (13)	1 (15)	1 (14)
10	1 (8)	1 (9)	1 (11)	1 (10)

B_2

A_3A_2 \ A_1A_0	00	01	11	10
00	(0)	(1)	(3)	(2)
01	1 (4)	1 (5)	1 (7)	1 (6)
11	D (12)	D (13)	D (15)	D (14)
10	1 (8)	1 (9)	1 (11)	1 (10)

B_1

A_3A_2 \ A_1A_0	00	01	11	10
00	(0)	(1)	1 (3)	1 (2)
01	1 (4)	1 (5)	D (7)	D (6)
11	D (12)	D (13)	1 (15)	1 (14)
10	1 (8)	1 (9)	(11)	(10)

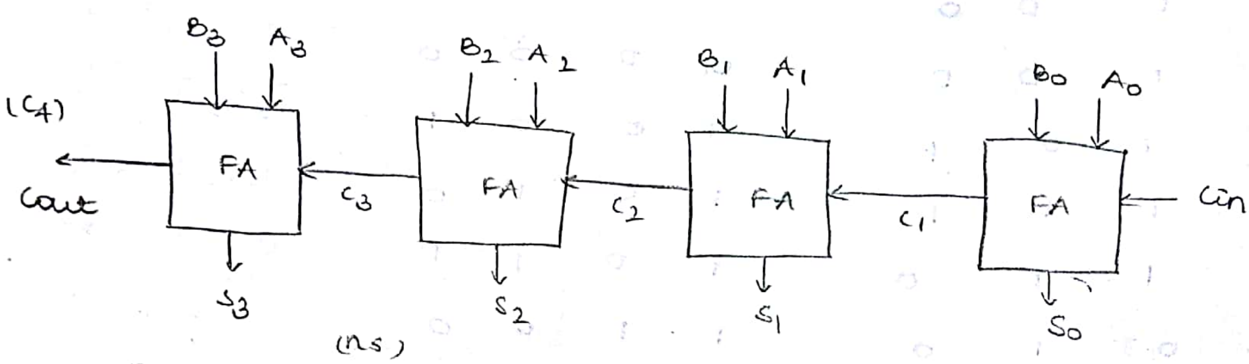
B_0

A_3A_2 \ A_1A_0	00	01	11	10
00	(0)	1 (1)	D (3)	1 (2)
01	1 (4)	D (5)	1 (7)	(6)
11	(12)	1 (13)	D (15)	1 (14)
10	1 (8)	D (9)	1 (11)	(10)

Binary parallel adders

1. Ripple carry adder (carry propagate adder)
2. Carry look ahead adder

RCA :



(ns)

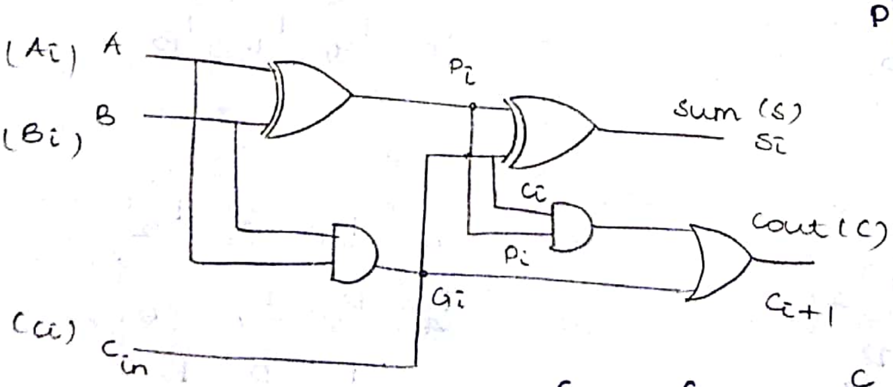
Propagation delay : Time interval b/w input reaching input terminal and producing output at output terminal.

Longest path of carry generation

⇓

Disadvantage

Carry look ahead adder :



$P_i \Rightarrow$ propagate fun

$= A_i \oplus B_i$

$G_i \Rightarrow$ generate fun

$= A_i \cdot B_i$

$C_{i+1} = C_i P_i + G_i$

$C_1 = G_0 + P_0 C_0$

C_0 - input carry

$C_2 = G_1 + P_1 C_1$

$C_2 = G_1 + P_1 \; (G_0 + P_0 C_0)$

$= G_1 + P_1 G_0 + P_1 P_0 C_0$

$C_3 = C_2 P_2 + G_2$

$= G_2 + P_2 G_1 + P_1 P_2 G_0 + P_1 P_2 P_0 \cancel{G_0} \, C_0$

$C_4 = G_3 + C_3 P_3$

$C_4 = G_3 + G_2 P_3 + P_2 P_3 G_1 + P_1 P_2 P_3 G_0 + P_1 P_2 P_0 P_3 \cancel{G_0} \, C_0$

8 - bit : CLA

$C_n = G_{n-1} + P_{n-1} G_{n-2} + P_{n-1} P_{n-2} G_{n-3} + \dots (P_{n-1} P_{n-2} \dots P_1 P_0 I_0)$

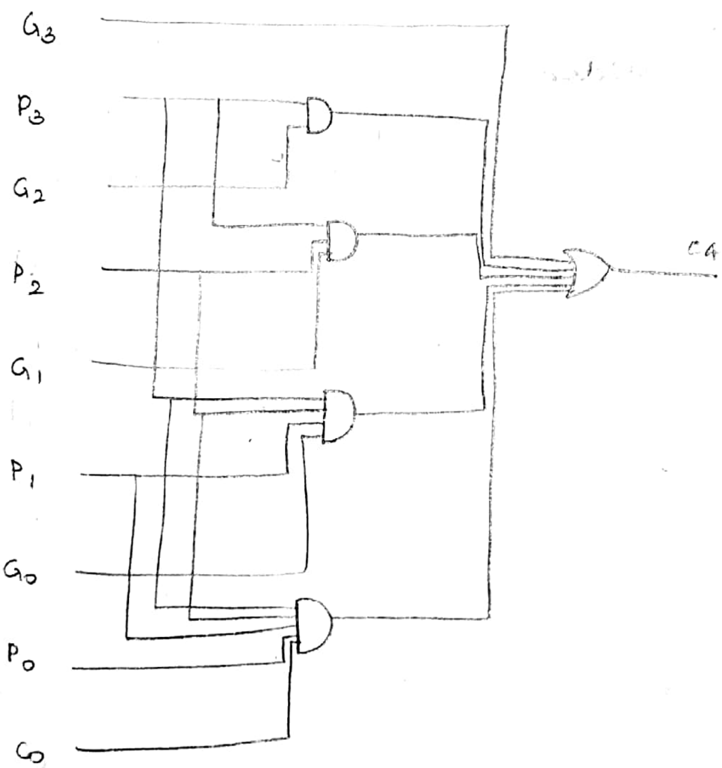
$C_8 = C_6 \; G_7 + P_7 G_6 + P_7 G_{\cancel{6}5} P_6 + P_7 P_6 P_5 G_4 + P_7 P_6 P_5 P_4 G_3 +$

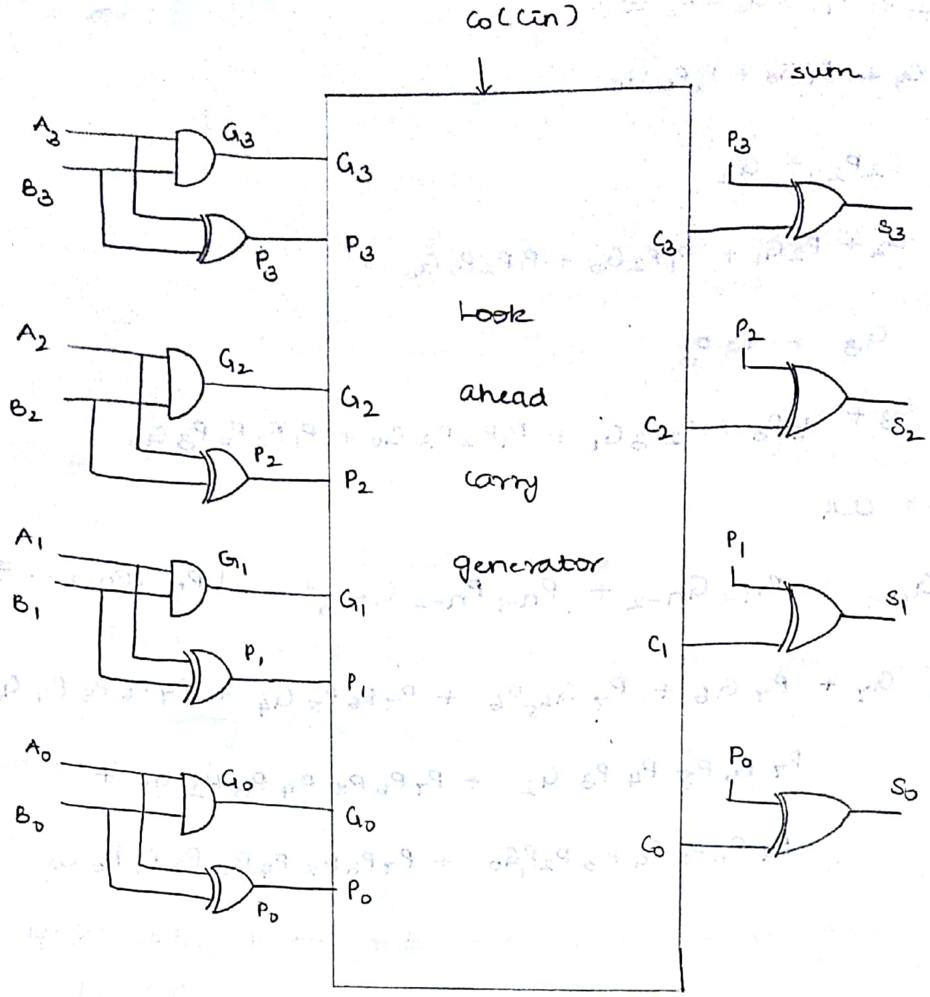
$P_7 P_6 P_5 P_4 P_3 G_2 + P_7 P_6 P_5 P_4 P_3 P_2 G_1 +$

$P_7 P_6 P_5 P_4 P_3 P_2 P_1 G_0 + P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0 C_0$

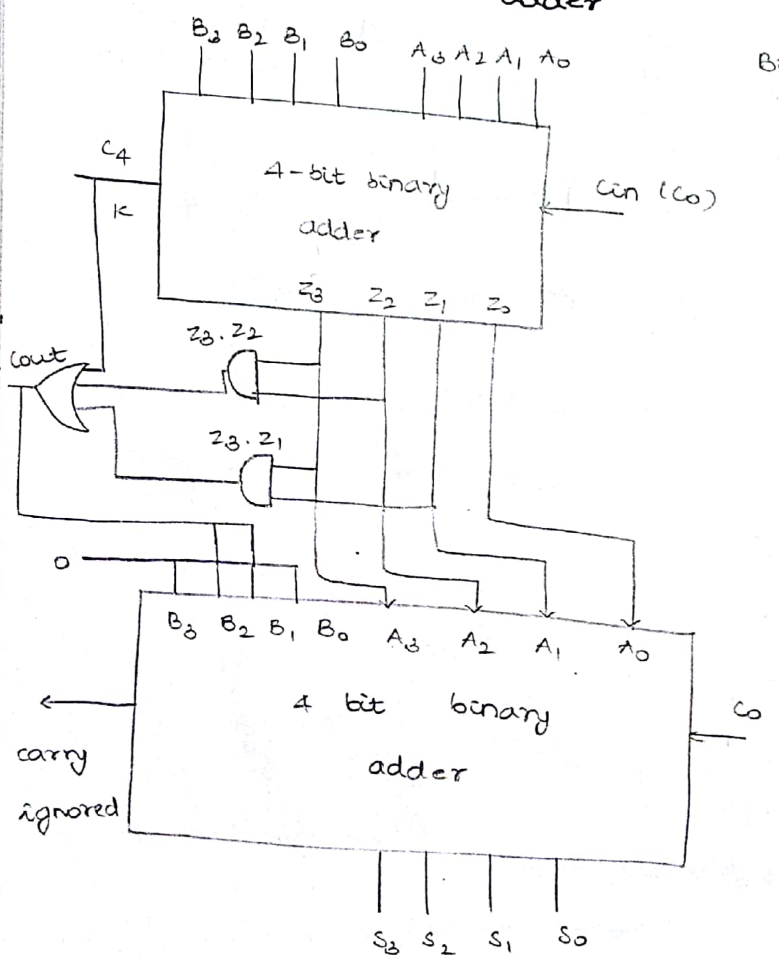
Lookahead carry generator → 4 gate delay (fixed)

whatever may be the no. of bits





BCD adder



Binary

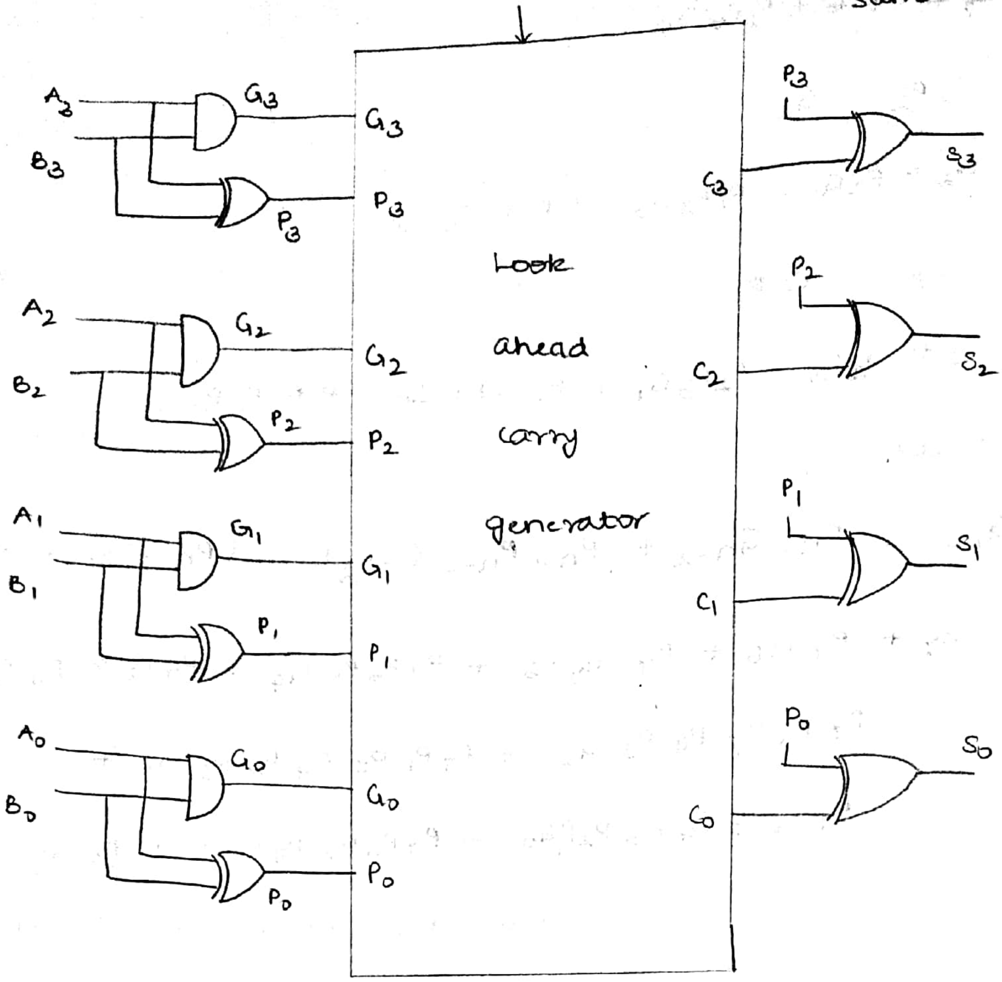
```
  1001
  0111
 -----
 10000
 -----
   16
```

BCD

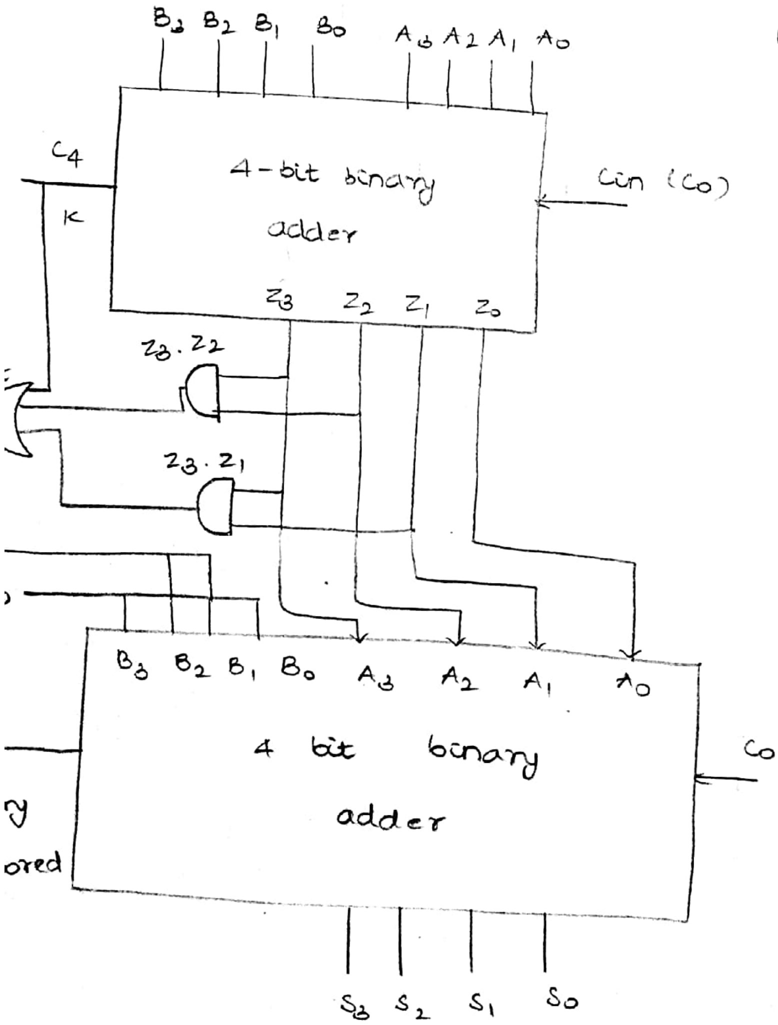
```
  1001
  0111
 -----
 10000
  0110
 -----
 10110
 ↓ └┬┘
 1  6
```

0111 1110 → will check ⇓ if >9 add 6 ⇓ result

↓ next check this ⇓ if this too >9 add 6 to previous result



BCD adder



Binary

$$\begin{array}{r} 1001 \\ 0111 \\ \hline 10000 \\ \hline 16 \end{array}$$

BCD

$$\begin{array}{r} 1001 \\ 0111 \\ \hline 10000 \\ 0110 \\ \hline 10110 \end{array}$$

1 6

0111 1110 → will check ⇓ if >9 add 6 ⇓ result

↓ next check this ⇓ if this too >9 add 6 to previous result

Decimal	Binary sum K	z_3	z_2	z_1	z_0	BCD sum Cout	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	0	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

Cout

z_3z_2 \ z_1z_0	00	01	11	10
00	0 (0)	0 (1)	0 (3)	0 (2)
01	0 (4)	0 (5)	0 (7)	0 (6)
11	1 (12)	1 (13)	1 (15)	1 (14)
10	0 (8)	0 (9)	1 (11)	1 (10)

When binary sum is 0-15

$Cout = z_3z_2 + z_3z_1$

When binary sum is 0-19

$Cout = K + z_3z_2 + z_3z_1$

Magnitude comparator

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

A > B

A_1A_0 \ B_1B_0	00	01	11	10
00	0	1	3	2
01	1 4	5	7	6
11	1 12	1 13	15	1 14
10	1 8	1 9	11	10

$F = A_1\bar{B}_1 + \bar{B}_1\bar{B}_0A_0 + A_1A_0\bar{B}_0$

A = B

A_1A_0 \ B_1B_0	00	01	11	10
00	1 0	1	3	2
01	4	1 5	7	6
11	12	13	1 15	14
10	8	9	11	1 10

$F = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1\bar{B}_0$

$A < B$

$A_1A_0 \backslash B_1B_0$	00	01	11	10
00	0	1 1	1 3	1 2
01	4	5	1 7	1 6
11	12	13	15	14
10	8	9	1 11	10

$A = B$

$F = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + A_1A_0B_1B_0 +$

$\quad \bar{a} \quad b \quad \bar{b} \quad a$

$A_1B_1\,\bar{A}_0\bar{B}_0 + \bar{A}_0\bar{B}_0 A_1B_1$

$= \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + A_1A_0B_1B_0 +$

$\bar{A}_1\bar{B}_1 \oplus A_0B_0$

$F = \bar{A}_1B_1 + \bar{A}_1\bar{A}_0B_0 + \bar{A}_0B_1B_0$

$A > B = A_1\bar{B}_1 + \bar{B}_1\bar{B}_0A_0 + A_1A_0\bar{B}_0$

$= A_1\bar{B}_1 + A_0A_1\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0B_1 + A_1A_0\bar{B}_0\bar{B}_1$

$\quad a \quad b \qquad a \quad c$

$A_0\bar{B}_0\bar{A}_1\bar{B}_1 \qquad A_0\bar{B}_0A_1B_1$

$= A_1\bar{B}_1 + A_0A_1\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1\bar{B}_0 + A_1A_0\bar{B}_0B_1$

$= A_1\bar{B}_1 + A_0A_1\bar{B}_1\bar{B}_0 + A_0\bar{B}_0(\bar{A}_1\bar{B}_1 + A_1B_1)$

$x + xy = x$ ↓ absorption law

$= \underbrace{A_1\bar{B}_1}_{x} + A_0\underbrace{A_1\bar{B}_1}_{x}\underbrace{\bar{B}_0}_{y} + A_0\bar{B}_0(A_1 \odot B_1)$

$A > B = A_1\bar{B}_1 + A_0\bar{B}_0(A_1 \odot B_1) = A_1\bar{B}_1 + x_1A_0\bar{B}_0$

$\bar{A}_0B_0\,\bar{A}_1\bar{B}_1 \qquad \bar{A}_0B_0\,A_1B_1$

$A < B = \bar{A}_1B_1 + \bar{A}_1\bar{A}_0B_0B_1 + \bar{A}_1\bar{A}_0B_0\bar{B}_1 + \bar{A}_0B_1B_0A_1 +$

$\bar{A}_0B_1B_0\bar{A}_1$

$= \bar{A}_1B_1 + \bar{A}_1\bar{A}_0B_0B_1 + \bar{A}_0B_0(\bar{A}_1\bar{B}_1 + A_1B_1)$

$= \bar{A}_1B_1 + \bar{A}_1\bar{A}_0B_0B_1 + \bar{A}_0B_0(A_1 \odot B_1)$

$= \bar{A}_1B_1 + \bar{A}_0B_0(A_1 \odot B_1) = \bar{A}_1B_1 + \bar{A}_0B_0x_1$

$F = \bar{A}_1\bar{B}_1(A_0B_0 + \bar{A}_0\bar{B}_0) + A_1B_1(A_0B_0 + \bar{A}_0\bar{B}_0)$

$= \bar{A}_1\bar{B}_1(A_0 \odot B_0) + A_1B_1(A_0 \odot B_0)$

$= (A_0 \odot B_0)(\bar{A}_1\bar{B}_1 + A_1B_1)$

$= (A_0 \odot B_0)(A_1 \odot B_1) = x_0 \cdot x_1$

Let $x_1 = A_1 \odot B_1$

$x_0 = A_0 \odot B_0$

$x_i = A_i \odot B_i$

4-bit :

$A > B = A_3\bar{B}_3 + x_3A_2\bar{B}_2 + x_3x_2A_1\bar{B}_1 + x_3x_2x_1A_0\bar{B}_0$

$A < B = \bar{A}_3B_3 + x_3\bar{A}_2B_2 + x_3x_2\bar{A}_1B_1 + x_3x_2x_1\bar{A}_0$

$A = B = x_3x_2x_1x_0$

8-bit :

$x_i = A_i \odot B_i$

$A > B = A_7\bar{B}_7 + x_7A_6\bar{B}_6 + x_7x_6A_5\bar{B}_5 + x_7x_6x_5A_4\bar{B}_4 +$
$x_7x_6x_5x_4A_3\bar{B}_3 + x_7x_6x_5x_4x_3A_2\bar{B}_2 + \cancel{x_3}\, x_7x_6x_5x_4x_3x_2$
$A_1\bar{B}_1 +$
$x_7x_6x_5x_4x_3x_2x_1A_0\bar{B}_0$

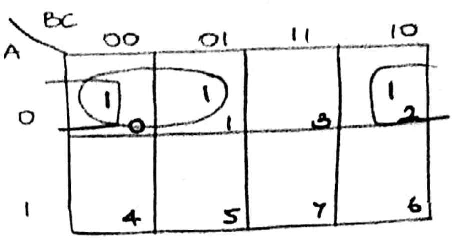
$A < B = \bar{A}_7B_7 + x_7\bar{A}_6B_6 + x_7x_6\bar{A}_5B_5 + x_7x_6x_5\bar{A}_4B_4 +$
$x_7x_6x_5x_4\bar{A}_3B_3 + x_7x_6x_5x_4x_3\bar{A}_2B_2 + x_7x_6x_5x_4x_3x_2\bar{A}_1B_1$
$x_7x_6x_5x_4x_3x_2x_1\bar{A}_0B_0$

$A = B = x_7x_6x_5x_4x_3x_2x_1x_0$

Design a combinational circuit with 3 input 1 output if output is 1 then Binary value of I/P is < 3 o/p is 0 otherwise.

	A	B	C	D
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

D

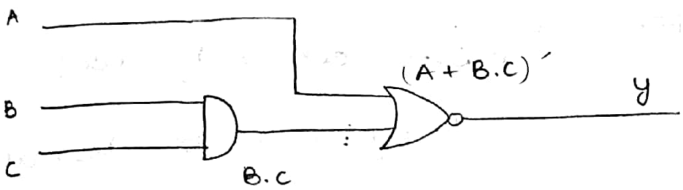


$F = \bar{A}\bar{B} + \bar{A}\bar{C}$

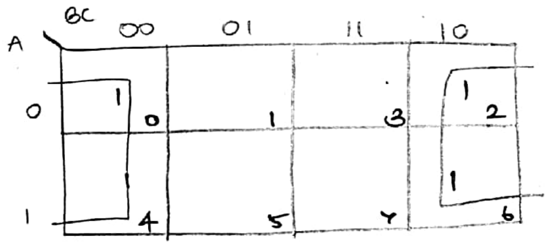
$= \bar{A}\,(\overline{\bar{B}+\bar{C}})$

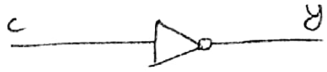
$= \overline{\bar{A}\;.\;\overline{B.C}}$

$y = \overline{(A + BC)}$



3 input one output, O/P is 1 when binary value of I/P is even number otherwise 0

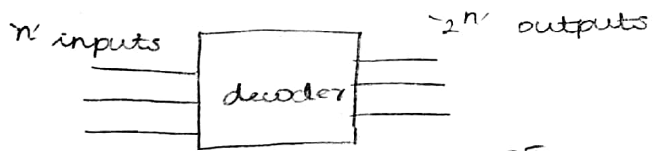


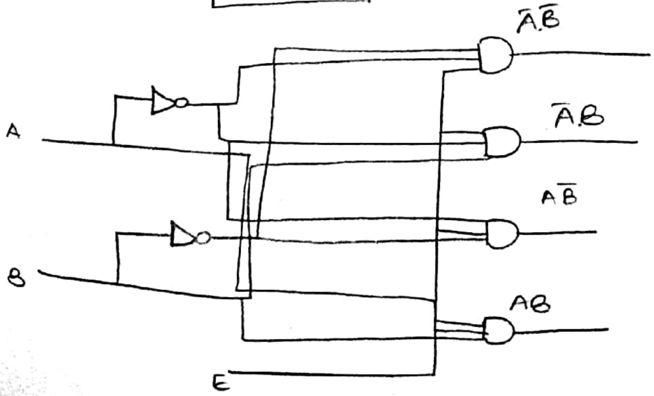


$y = \bar{C}$

2 to 4
2 : 4

Decoder





AB	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

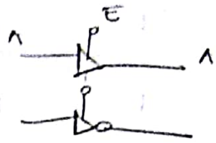
E	A	B	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Tristate device

low	0
high	1
high impedance	Z

Buffer → holds the input and creates a delay

O/P follows I/P

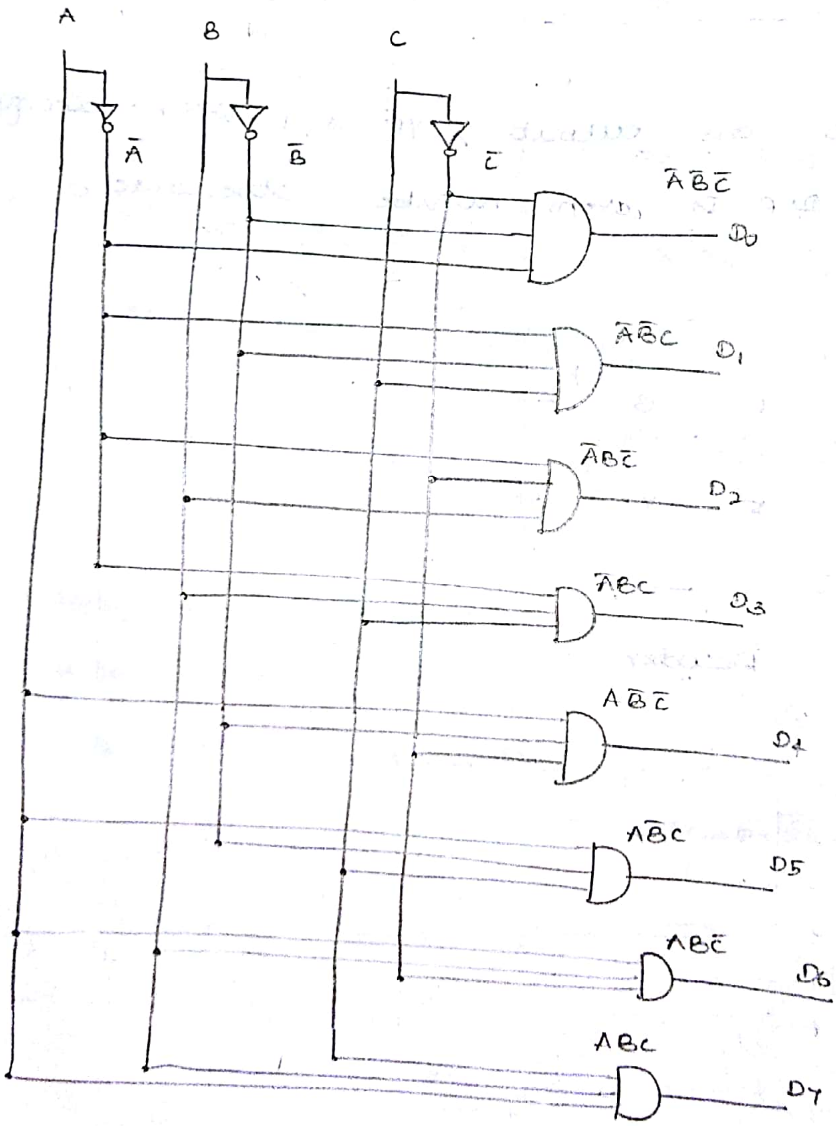


active low

3 to 8 74LS138 } Most commonly used

4 to 16

Eg: BCD to decimal



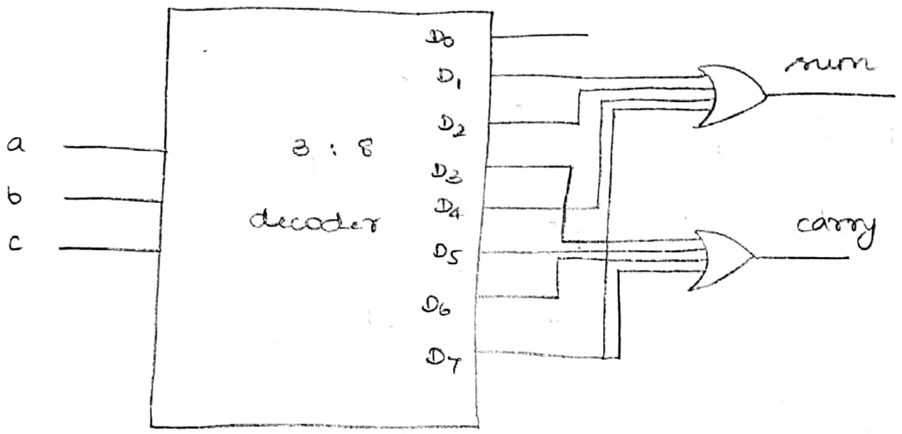
Scanned by CamScanner

A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_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

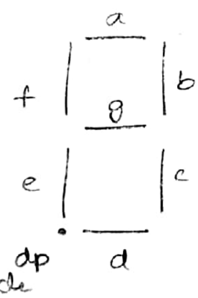
Full adder using decoder

sum = $\Sigma(1, 2, 4, 7)$

Cout = $\Sigma(3, 5, 6, 7)$

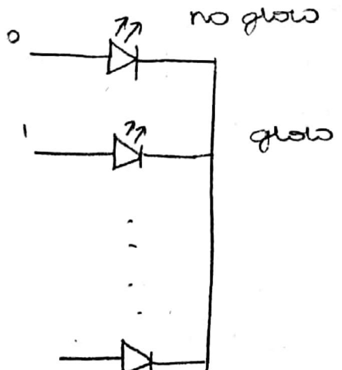


Seven segment

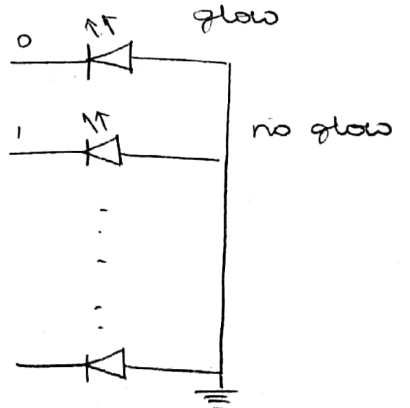


7 inputs

Common cathode

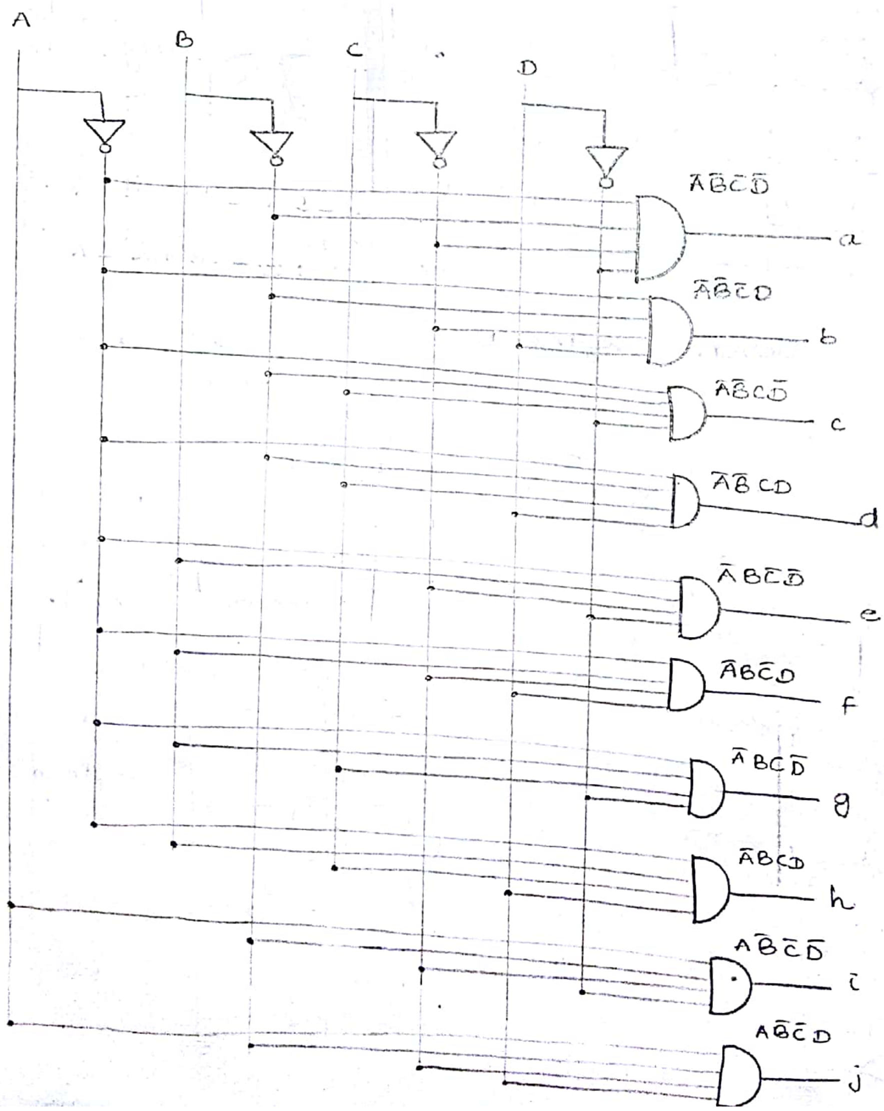


Common anode



BCD to seven segment

	BCD				seven segment									
	A	B	C	D	a	b	c	d	e	f	g	h	i	j
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5	0	1	0	1	0	0	0	0	0	1	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	1	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	0	0	0	1



BCD to seven segment

BCD				seven segment						
A	B	C	D	a	b	c	d	e	f	g
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	1
0	1	0	0	0	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	0	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

a

AB \ CD	00	01	11	10
00	1 (0)	(1)	1 (3)	1 (2)
01	(4)	1 (5)	1 (7)	1 (6)
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	1 (9)	X (11)	X (10)

$a = C + A + \bar{B}\bar{D} + BD$

$= C + A + B \odot D$

b

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	1 (3)	1 (2)
01	1 (4)	(5)	1 (7)	(6)
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	1 (9)	X (11)	X (10)

$b = \bar{B} + CD + \bar{C}\bar{D}$

$= \bar{B} + C \odot D$

c

AB \ CD	00	01	11	10
00	1 (0)	1 (1)	1 (3)	(2)
01	1 (4)	1 (5)	1 (7)	1 (6)
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	1 (9)	X (11)	X (10)

$c = \bar{C} + D + BC$

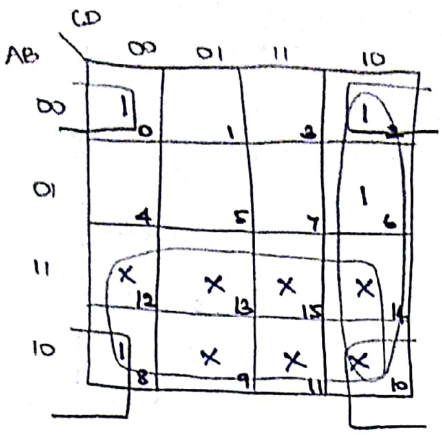
d

AB \ CD	00	01	11	10
00	1 (0)	(1)	1 (3)	1 (2)
01	(4)	1 (5)	(7)	1 (6)
11	X (12)	X (13)	X (15)	X (14)
10	1 (8)	1 (9)	X (11)	X (10)

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

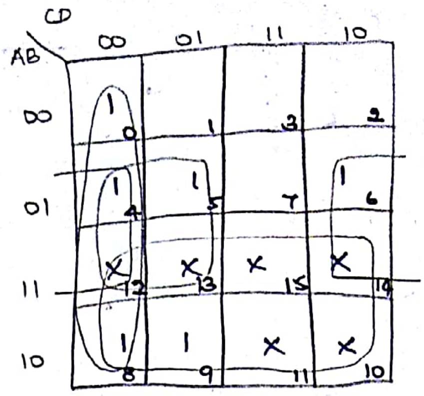
$= \bar{D}(\bar{B} + C) + \bar{B}C + B\bar{C}D + A$

e



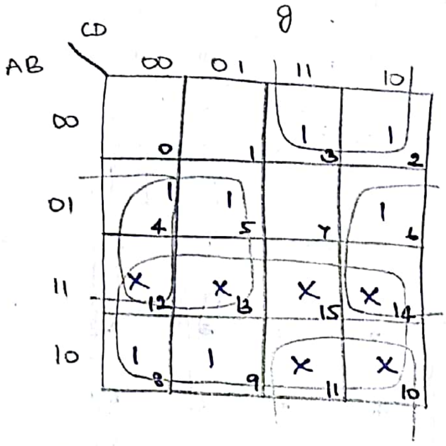
$e = A + \bar{B}\bar{D} + C\bar{D}$

f



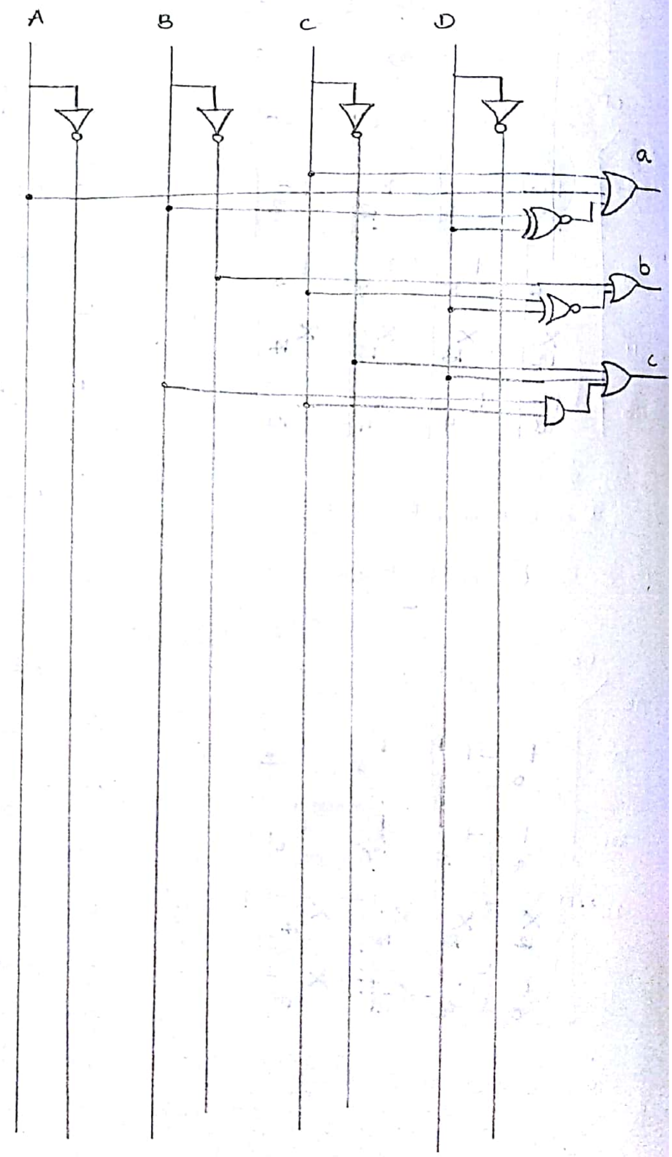
$f = A + \bar{C}\bar{D} + B\bar{C} + B\bar{D}$

g

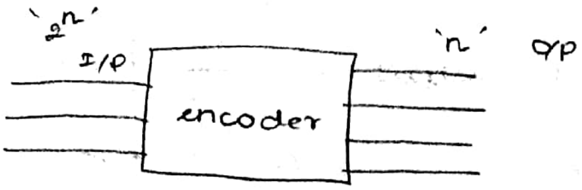


$g = A + B\bar{C} + B\bar{D} + \bar{B}C$

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

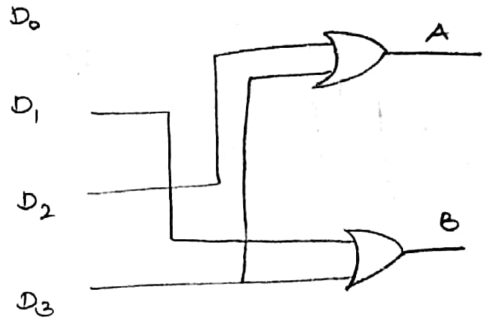


ENCODER



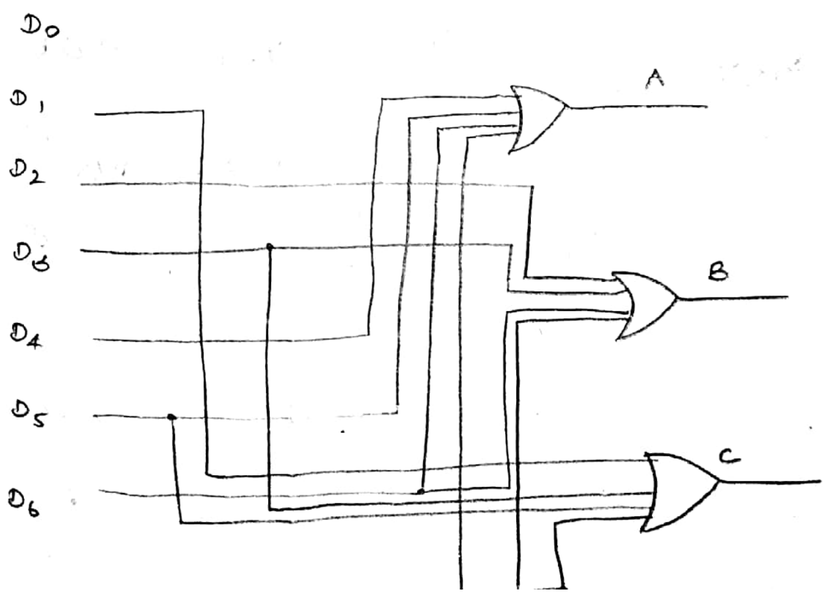
4:2 Encoder

D_0	D_1	D_2	D_3	A	B
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

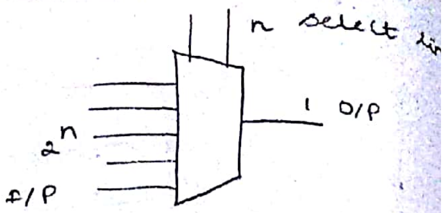


8:3 Encoder

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



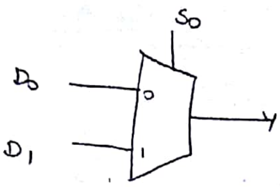
Multiplexer (Mux) → data selecter

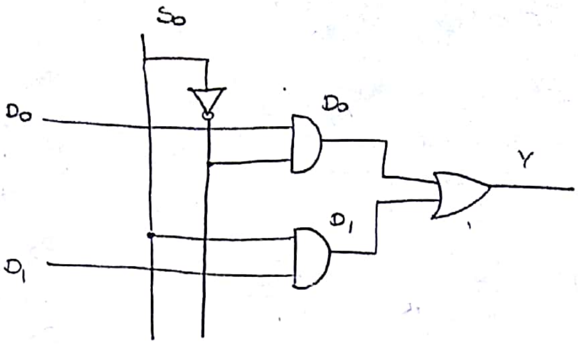


2^n inputs 1 output

n select lines

2 : 1 MUX



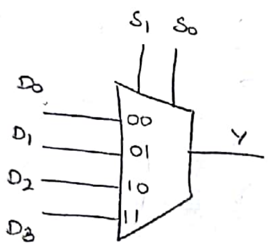


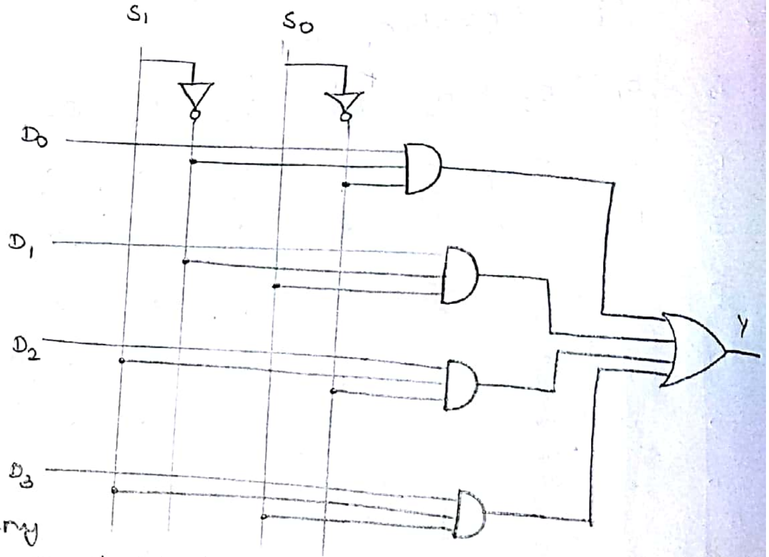
S_0	Y
0	D_0
1	D_1

$S_0 = 0$ $Y = D_0$

$S_0 = 1$ $Y = D_1$

4 : 1 MUX





Using multiplexer any digital circuit can be constructed

Design a full adder using multiplexer

designe for circuit to be with no. of inputs 'n' & choose 'n-1' select lines

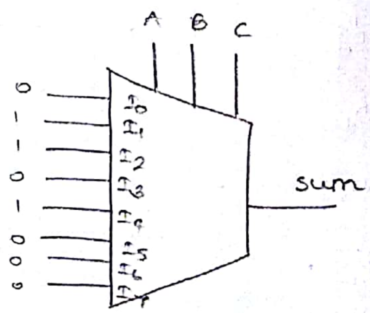
$sum = \Sigma(1,2,4,7)$

$carry = \Sigma(3,5,6,7)$

either 4:1 or 8:1

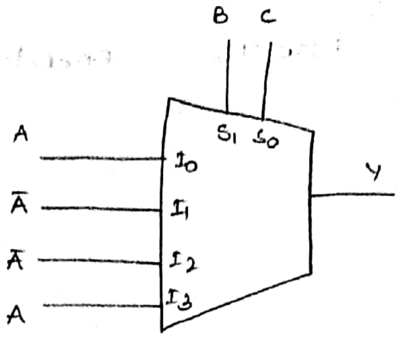
A	B	C	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

8 : 1 MUX



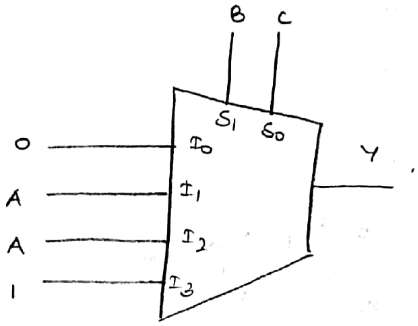
Implementation table for sum

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



Implementation table for carry

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



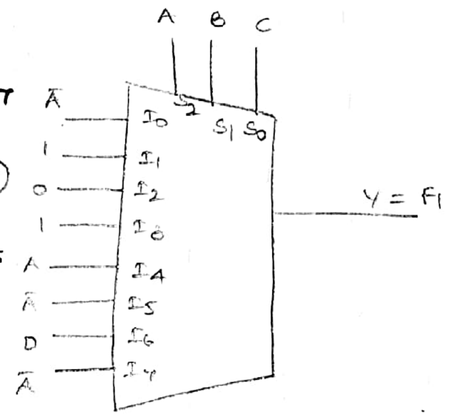
$F_1 = \Sigma(0,1,3,5,7,9,11,12)$

$F_2 = \Sigma(1,5,7,11,12,13,15)$

16 → 4 inputs → 3 select lines → 8 : 1 MUX

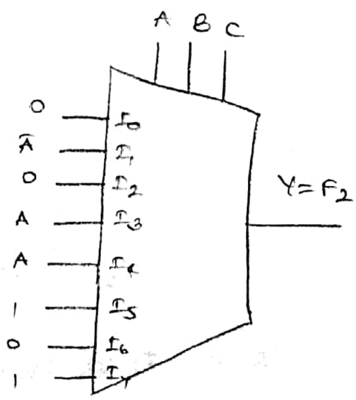
Implementation table for F_1

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



Implementation table for F_2

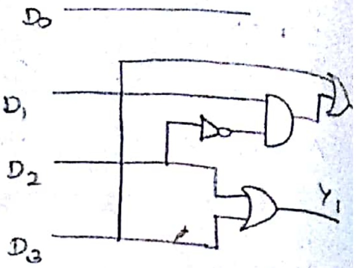
	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	0	$\bar{A}$	0	A	A	1	0	1



Priority Encoder

4 : 2

	D_0	D_1	D_2	D_3	Y_1	Y_0
8	1	0	0	0	0	0
4, 12	X	1	0	0	0	1
2, 6, 10, 14	X	X	1	0	1	0
1, 3, 5, 7, 9, 11, 13, 15	X	X	X	1	1	1



Y_1

D_0D_1 \ D_2D_3	00	01	11	10
00	0	1 (1)	1 (3)	1 (2)
01	4	1 (5)	1 (7)	1 (6)
11	12	1 (13)	1 (15)	1 (14)
10	8	1 (9)	1 (11)	1 (10)

$Y_1 = D_3 + D_2$

Y_0

D_0D_1 \ D_2D_3	00	01	11	10
00	0	1 (1)	1 (3)	2
01	1 (4)	1 (5)	1 (7)	6
11	1 (12)	1 (13)	1 (15)	14
10	8	1 (9)	1 (11)	10

$Y_0 = D_3 + D_1\overline{D_2}$

8 : 3

	128 D_0	64 D_1	32 D_2	16 D_3	8 D_4	4 D_5	2 D_6	1 D_7	Y_2	Y_1	Y_0
	1	0	0	0	0	0	0	0	0	0	0
	X	1	0	0	0	0	0	0	0	0	1
	X	X	1	0	0	0	0	0	0	1	0
	X	X	X	1	0	0	0	0	0	1	1
8	X	X	X	X	1	0	0	0	1	0	0
4	X	X	X	X	X	1	0	0	1	0	1
	X	X	X	X	X	X	1	0	1	1	0
	X	X	X	X	X	X	X	1	1	1	1

$Y_2 = \Sigma(4, 5, 6, 7)$

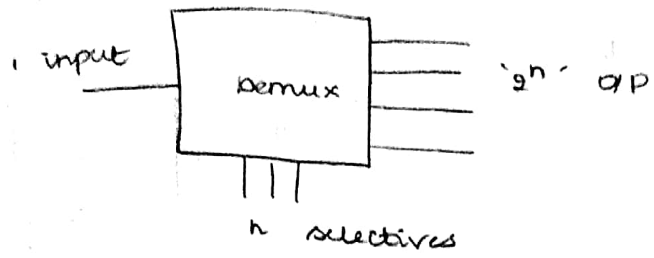
$Y_2 = D_4\overline{D_5}\,D_6\overline{D_7} + D_7 + D_6\overline{D_7} + D_5\overline{D_6}\,\overline{D_7}$

$Y_1 = D_7 + D_6\overline{D_7} + D_3\overline{D_4}\,\overline{D_5}\,\overline{D_6}\,\overline{D_7} + D_2\overline{D_3}\,\overline{D_4}\,\overline{D_5}\,\overline{D_6}\,\overline{D_7}$

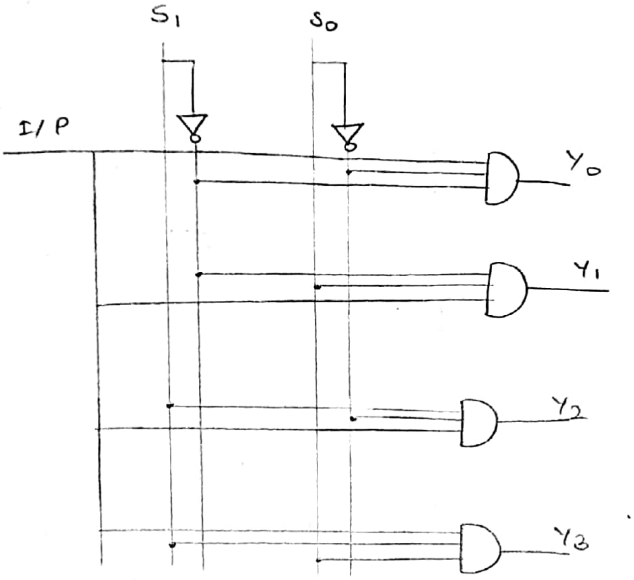
$Y_0 = D_7 + D_5\overline{D_6}\,\overline{D_7} + D_3\overline{D_4}\,\overline{D_5}\,\overline{D_6}\,\overline{D_7} + D_1\overline{D_2}\,\overline{D_3} \ldots \overline{D_7}$

Scanned by CamScanner

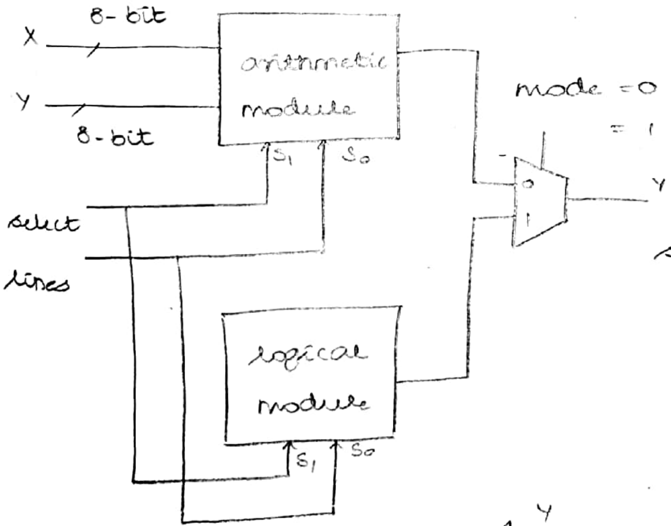
Demultiplexer



1:4 demux

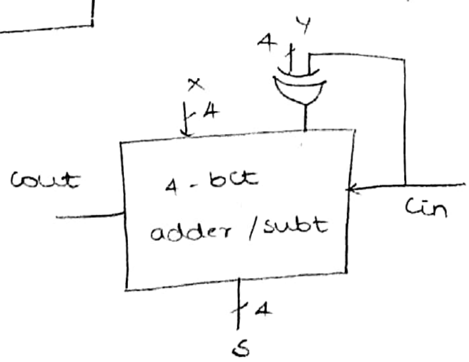


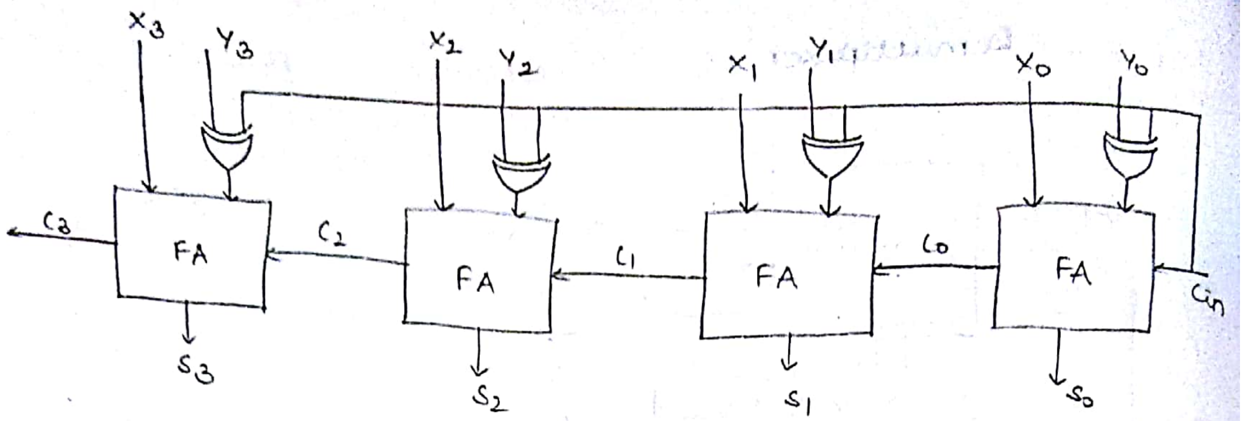
8-bit ALU



mode = 0 arithmetic
= 1 logic

select lines	arithmetic	logic
0 0	X+Y	AND
0 1	X−Y	OR
1 0	Y−X	XOR
1 1	–	NAND





$X + Y$ and $X - Y$

$Y \rightarrow$ adder

$\bar{Y} \rightarrow$ subtractor
$\downarrow$
1's complement

0	0	0	$\rightarrow Y \Rightarrow X + Y + 0$
0	1	1	
1	0	1	$\rightarrow \bar{Y} \Rightarrow X + (\bar{Y} + 1) = X + (\bar{Y}+1)$
1	1	0	

$(\bar{Y}+1)$
$\downarrow$
2's complement