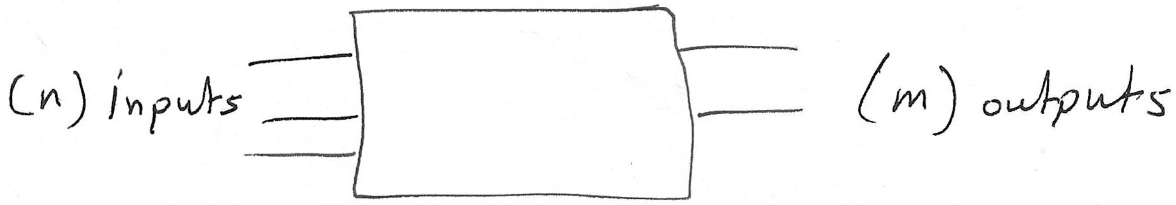
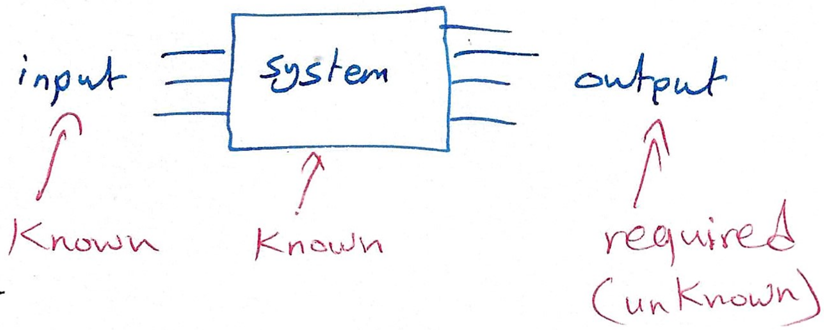
Chapter 4 :- Combinational logic Design

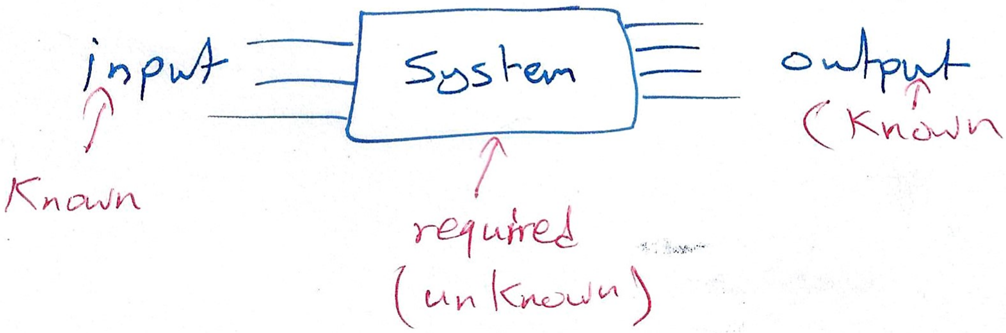
Combination Circuit :- A block of logic gates where each output is determined from present Combination of inputs.



Analysis :-

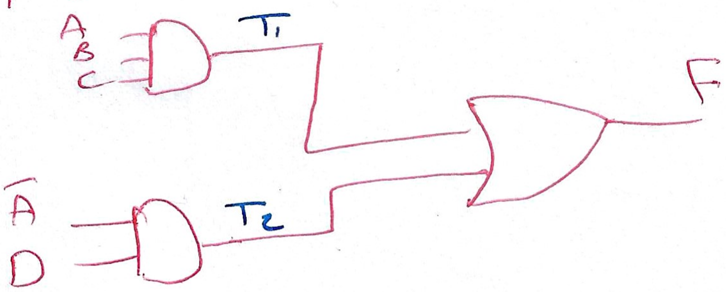


Design :-



* Analysis :-

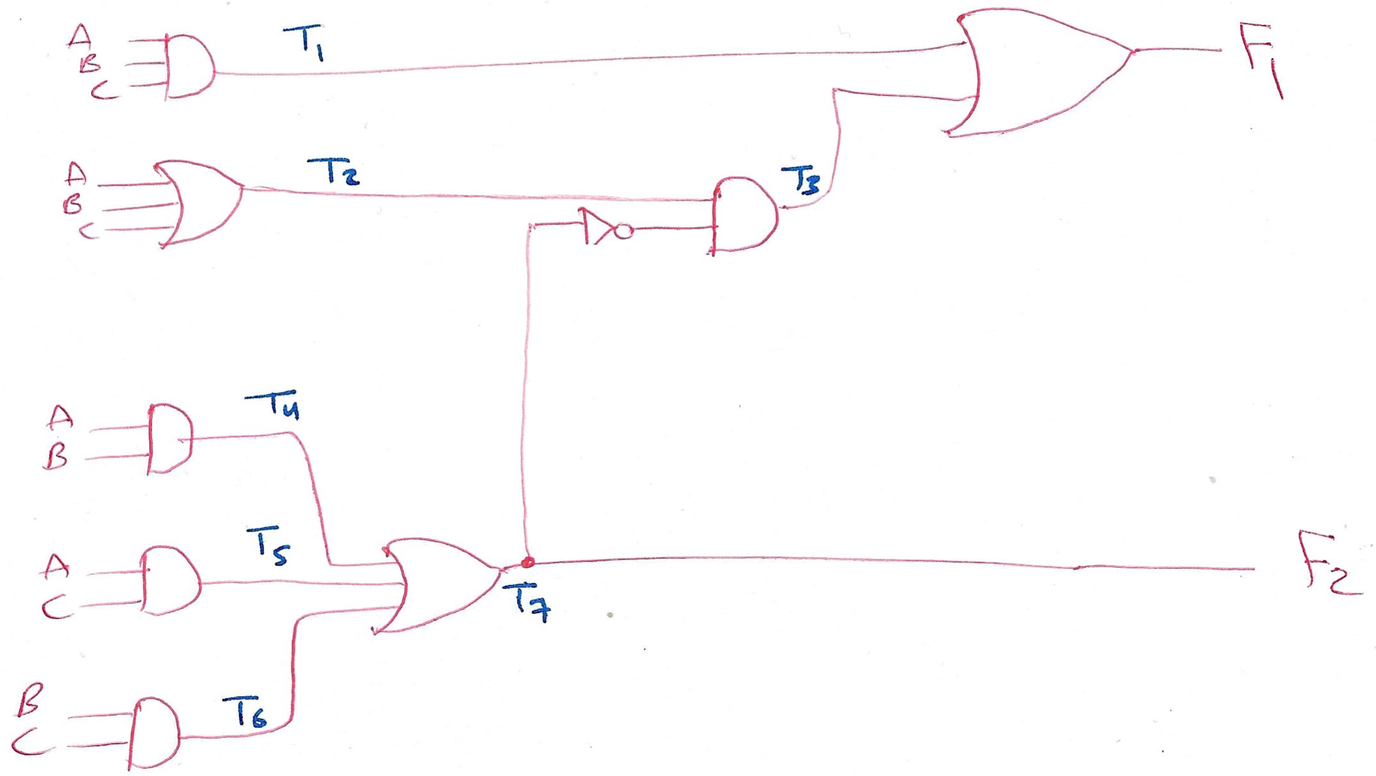
Example :-



$F = T_1 + T_2$

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

Example:-



$$F_1 = T_1 + T_3$$
$$= (A.B.C) + (T_2 . \overline{T_7})$$
$$= A.B.C + [(A+B+C) . (\overline{T_4 + T_5 + T_6})]$$
$$= A.B.C + ((A+B+C) . (\overline{T_4} . \overline{T_5} . \overline{T_6}))$$
$$= A.B.C + [(A+B+C) . ((\bar{A}+\bar{B}) . (\bar{A}+\bar{C}) . (\bar{B}+\bar{C}))]$$
$$= A.B.C + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$$

$$F_2 = T_7 = T_4 + T_5 + T_6$$
$$= A.B + A.C + B.C$$

* Design :-

* Design Procedures :-

1. Specification :-
   1. read the problem carefully
   2. Determine the number of inputs and outputs
   3. Assign symbols to inputs and outputs
   4. what the circuit should do

2. Formulation :-
   1. convert the specification into truth table
   2. write the expression as SOM / POM

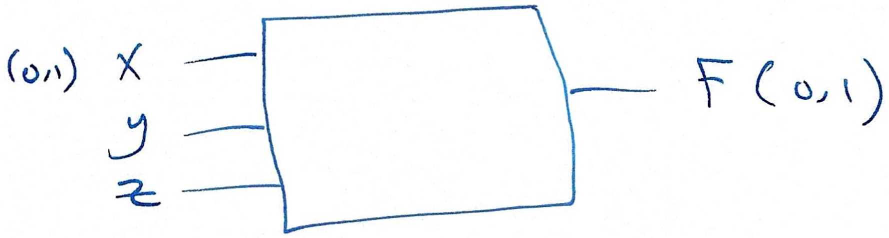
3. Logic minimization :-
   1. minimize the expression using Kmap

4. Technology Mapping :-
   → Draw the circuit

5. Verification :-
   → test the circuit

Example: Design a Combinational Circuit that takes 3 bits input number and checks whether the number is prime or not.

(1)



(2)

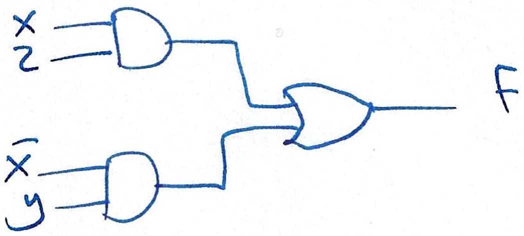
x	y	z	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

(3)

x \ yz	00	01	11	10
0			1	1
1		1	1	

$$F = xz + \bar{x}y$$

(4)



(5)

$(0.0)+(1.0) \stackrel{?}{=} 0$ ✓

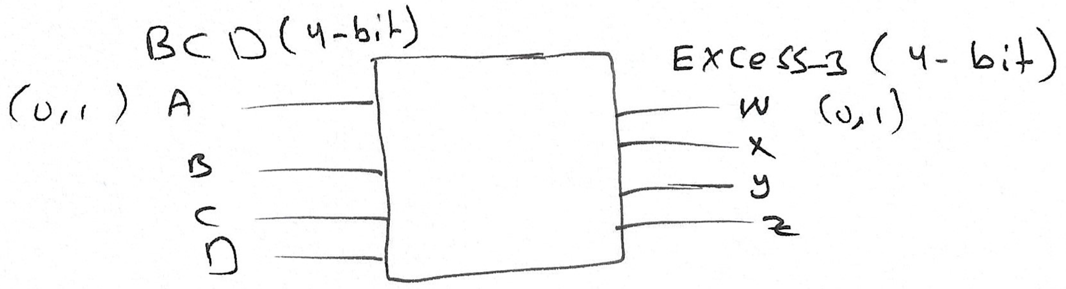
x	y	z	F
0	0	0	0

$(1.1)+(0.0) \stackrel{?}{=} 1$ ✓

x	y	z	F
1	0	1	1

Example:- Design a BCD to Excess-3 Code Converter.

(1)

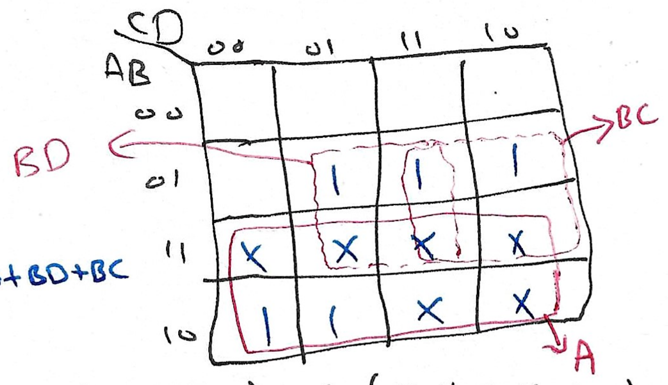


(2)

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	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
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	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

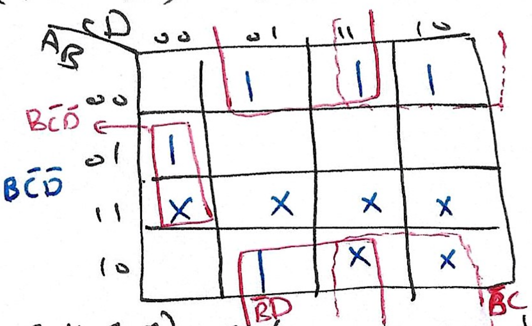
(3)

$W = \sum_m(5,6,7,8,9) + \sum_d(10,11,12,13,14,15)$



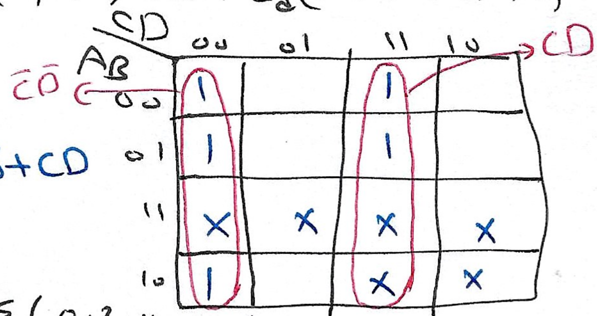
$W = A + BD + BC$

$X = \sum_m(1,2,3,4,9) + \sum_d(10,11,12,13,14,15)$



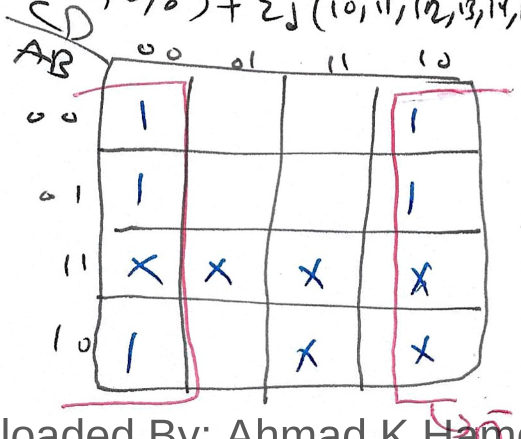
$X = \bar{B}C + \bar{B}D + B\bar{C}\bar{D}$

$y = \sum_m(0,3,4,7,8) + \sum_d(10,11,12,13,14,15)$



$y = \bar{C}\bar{D} + CD$

$z = \sum_m(0,2,4,6,8) + \sum_d(10,11,12,13,14,15)$



$z = \bar{D}$

$W = A + BC + BD$

$W = A + B \cdot (C + D)$

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

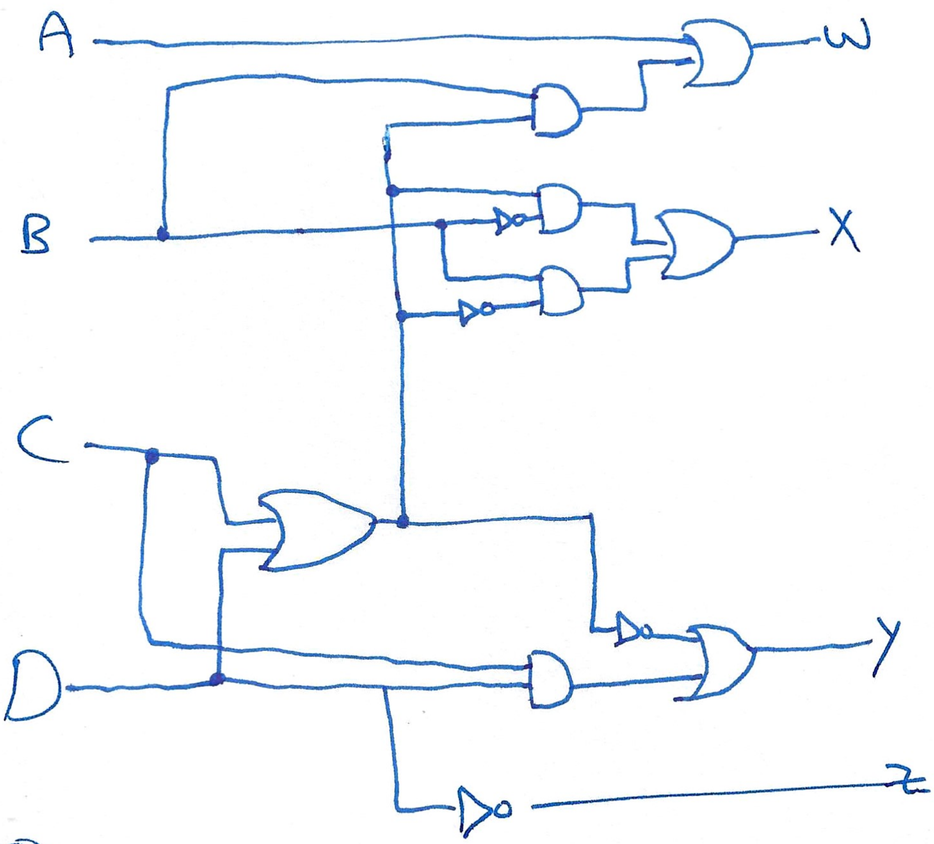
$X = \bar{B} \cdot (C + D) + B \cdot \overline{(C + D)}$

$Y = \bar{C}\bar{D} + CD$

$Y = CD + \overline{(C + D)}$

$Z = \bar{D}$

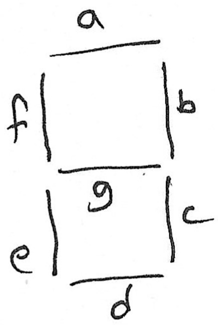
(4)

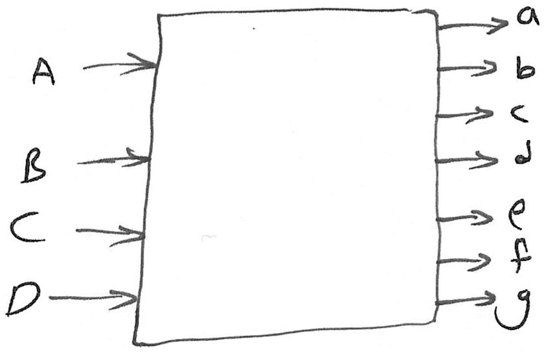


(5)

Example: Design a BCD to 7-segment Decoder

(1)

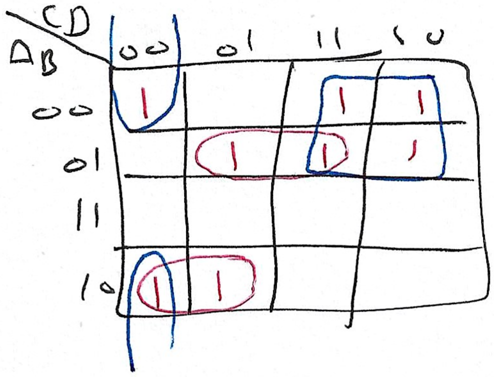




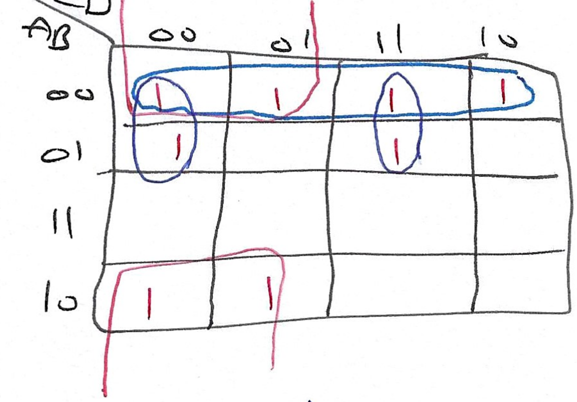
(2)

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	1	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
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

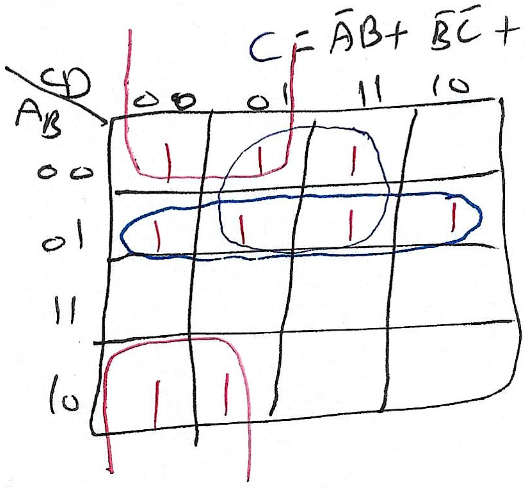
$a = \bar{A}C + \bar{A}BD + A\bar{B}\bar{C} + \bar{B}\bar{C}\bar{D}$



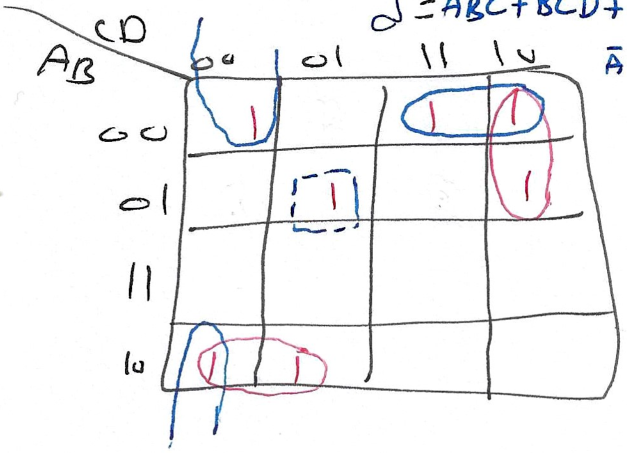
$b = \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{C}\bar{D} + \bar{A}CD$



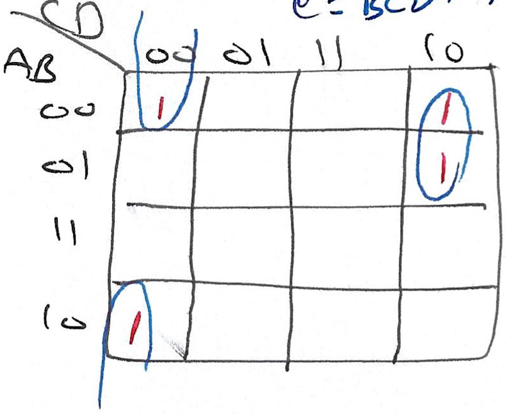
$c = \bar{A}B + \bar{B}\bar{C} + \bar{A}D$



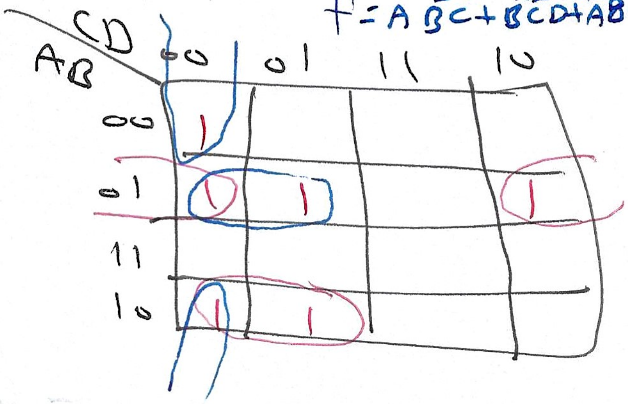
$d = A\bar{B}\bar{C} + \bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C + \bar{A}C\bar{D} + \bar{A}B\bar{C}D$



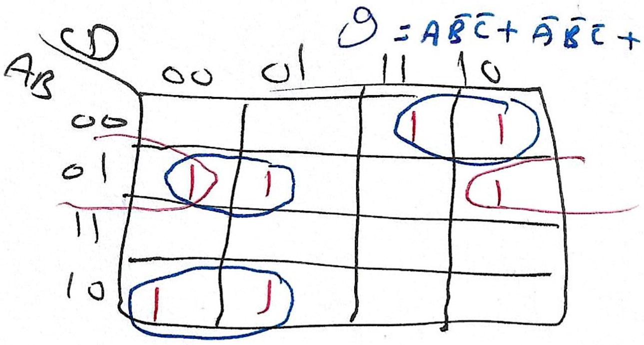
$e = \bar{B}\bar{C}\bar{D} + \bar{A}C\bar{D}$



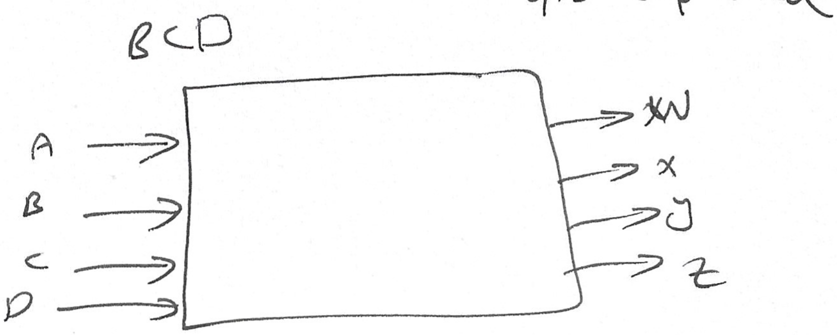
$f = A\bar{B}\bar{C} + \bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C} + \bar{A}B\bar{D}$



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



Example:- Design a combinational circuit that generate the 9's complement of a BCD digit?



9's complement of 0 = 9 − 0 = 9
1 = 9 − 1 = 8
⋮

A	B	C	D	W	X	Y	Z
0	0	0	0	1	0	0	1
0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	1
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

$W = \bar{A}\bar{B}\bar{C}$

AB \ CD	00	01	11	10
00	1	1		
01				
11	X	X	X	X
10			X	X

$X = B\bar{C} + \bar{B}C$

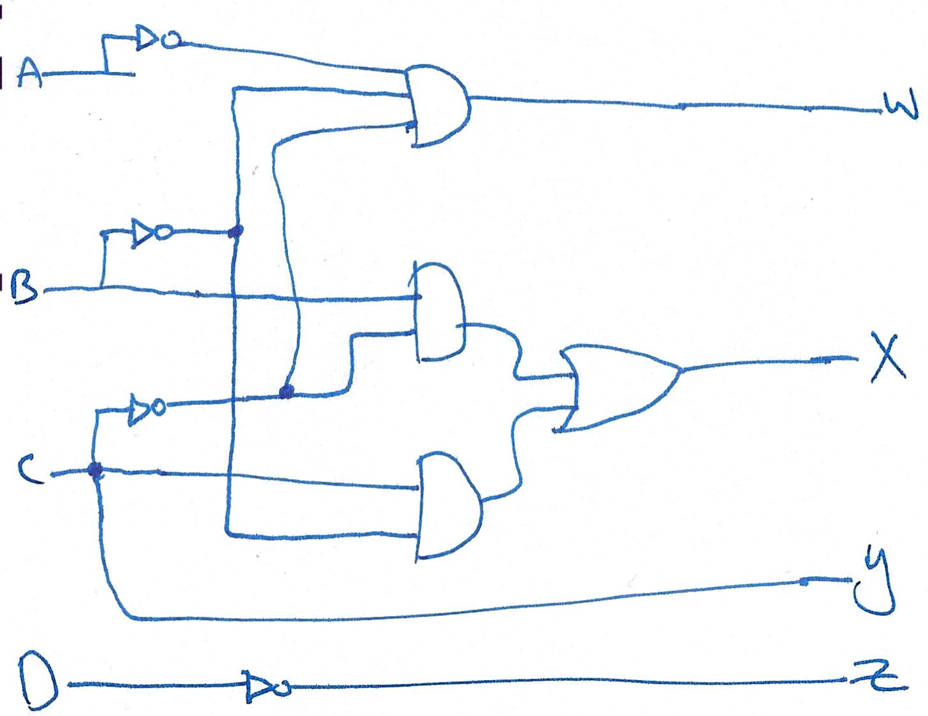
AB \ CD	00	01	11	10
00			1	1
01	1	1		
11	X	X	X	X
10			X	X

$y = C$

AB \ CD	00	01	11	10
00			1	1
01			1	1
11	X	X	X	X
10			X	X

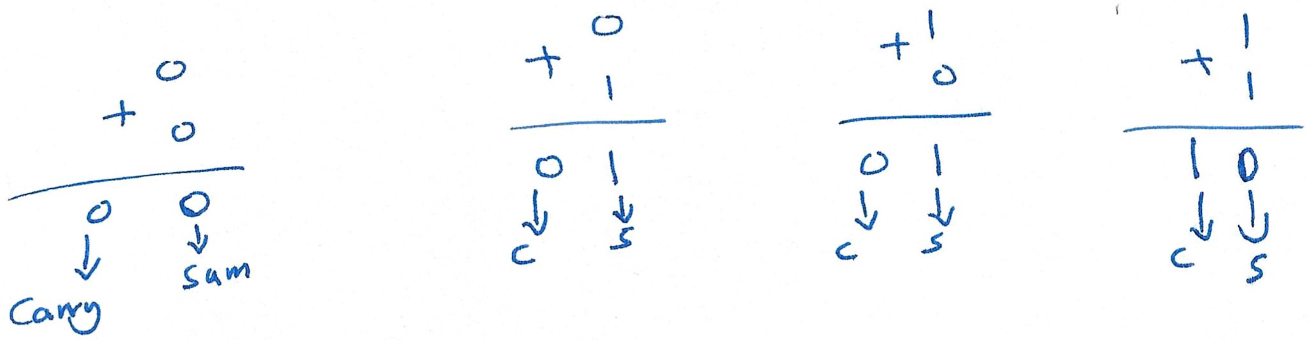
$z = \bar{D}$

AB \ CD	00	01	11	10
00	1			1
01	1			1
11	X	X	X	X
10	1		X	X



* Binary adder and subtractor:-

* addition :-

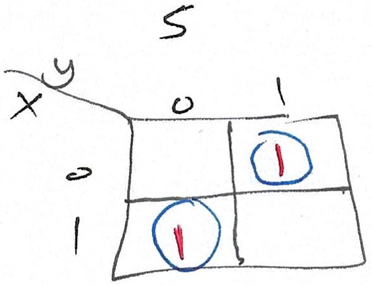


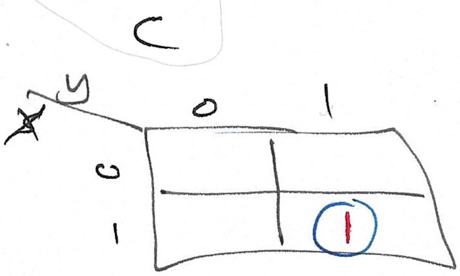
Half Adder (HA):- Combinational circuit that perform addition of two bits

block diagram

Truth table

x	y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

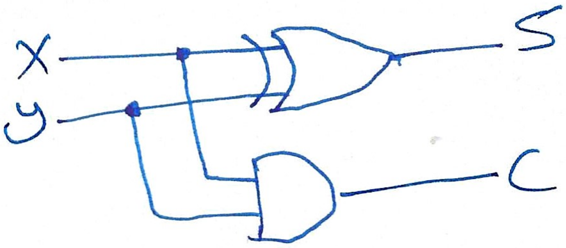




$S = \bar{x}y + x\bar{y}$

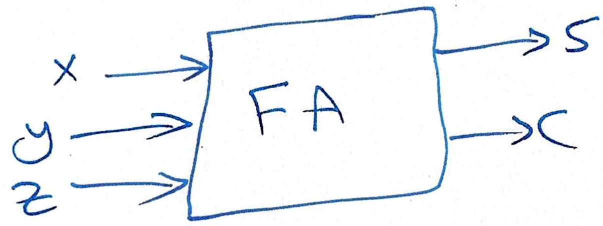
$= x \oplus y$

$C = x \cdot y$



Circuit diagram

Full Adder :- Combinational circuit perform addition of 3-bits (FA)

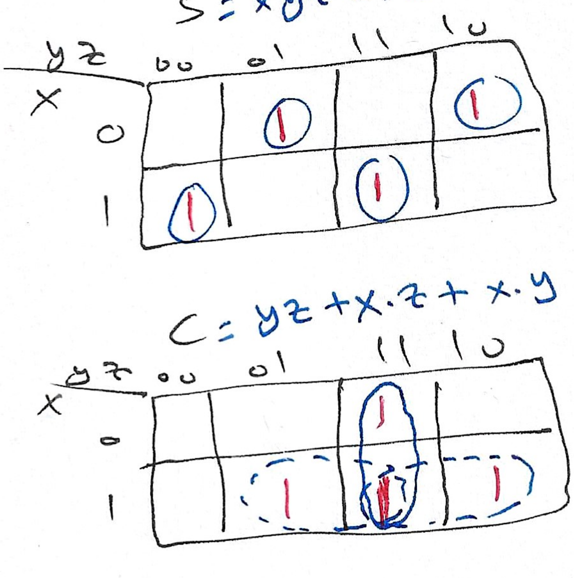


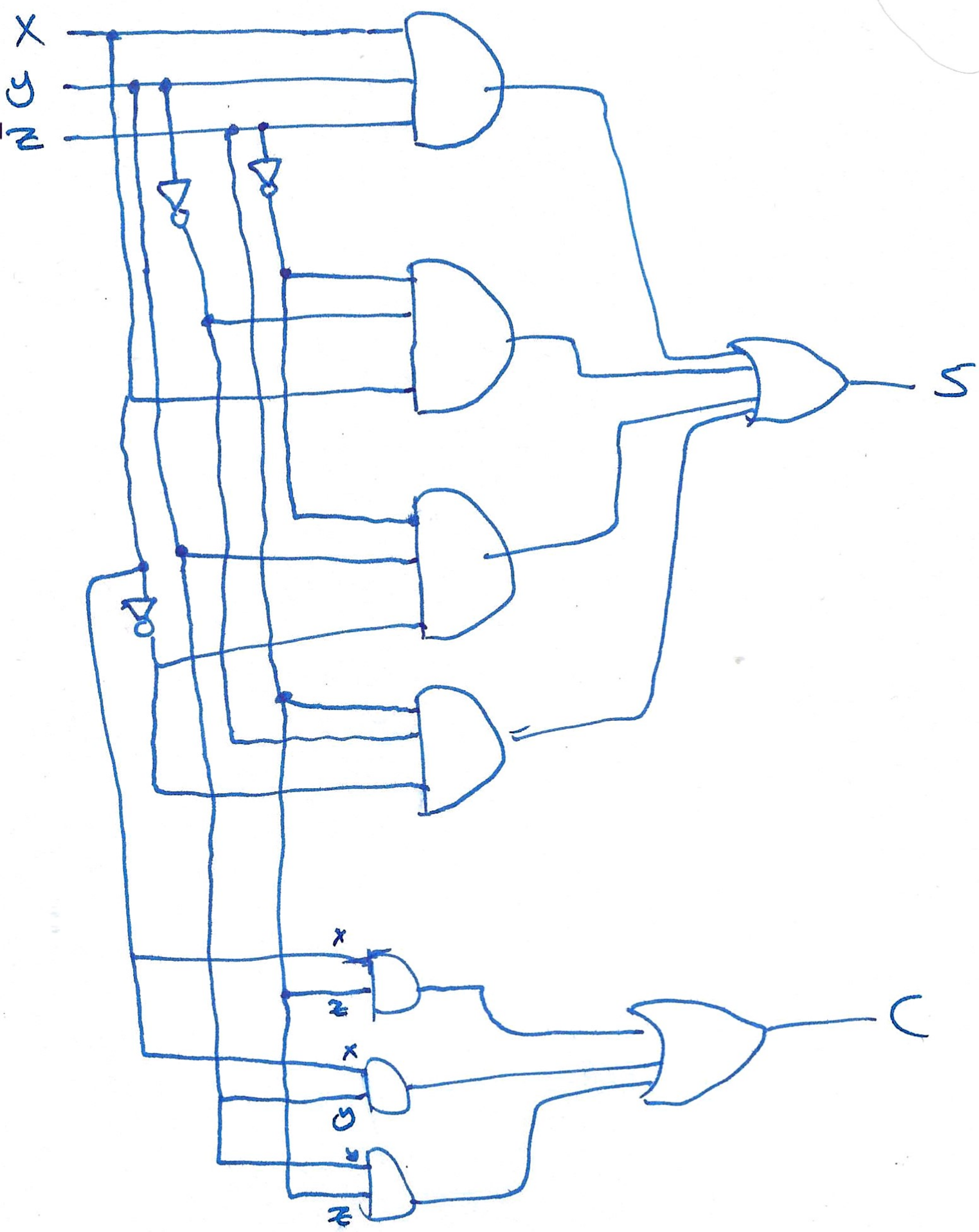
block diagram

x	y	z	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

$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$

$C = yz + x \cdot z + x \cdot y$





$$S = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xyz$$

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

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

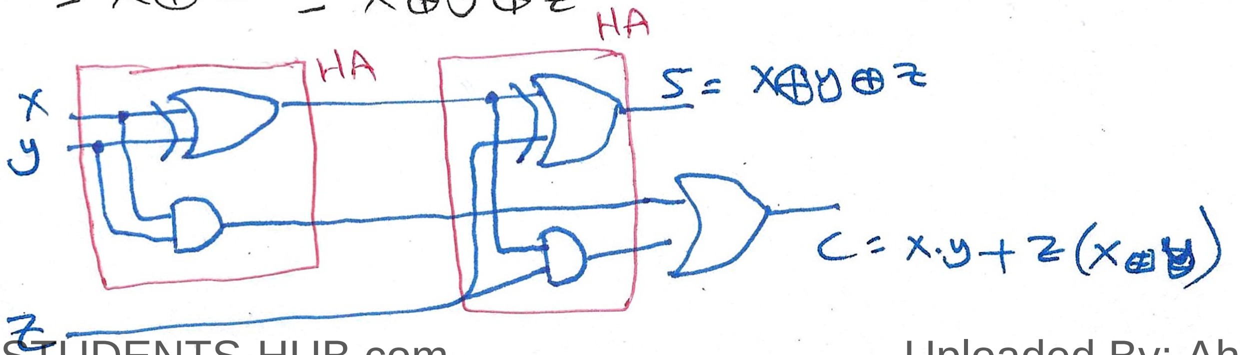
$$= \bar{x}w + x\bar{w}$$

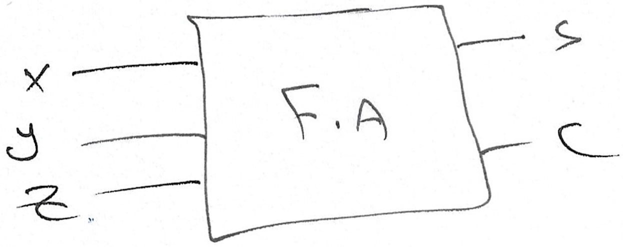
$$= x \oplus w = x \oplus y \oplus z$$

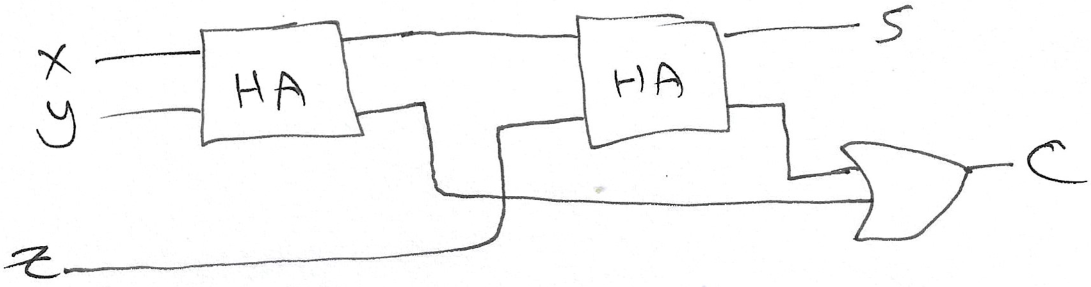
$$C = \bar{x}yz + x\bar{y}z + xyz + xy\bar{z}$$

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

$$= z(x \oplus y) + xy$$





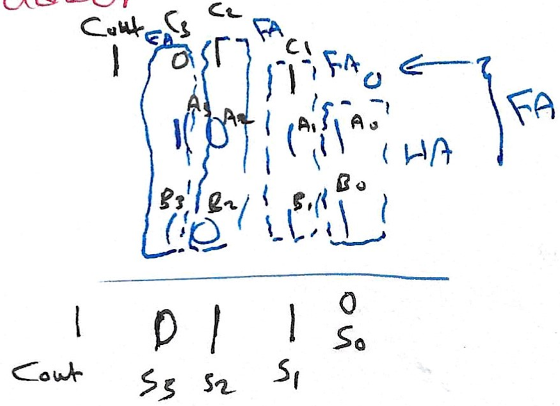


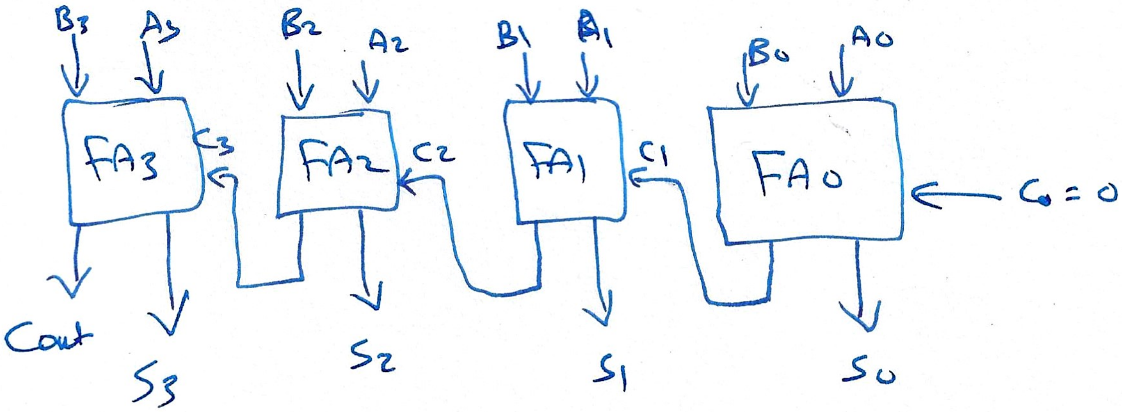
F.A

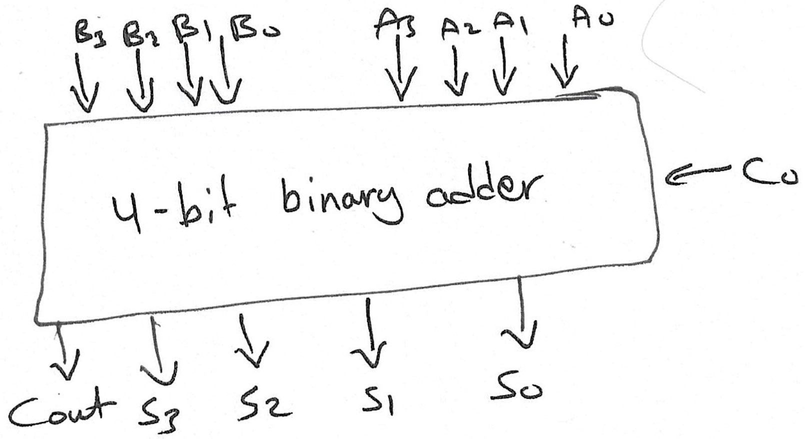
Binary Adder: Addition Any number of bits

Example: 4-bit binary adder

$$\begin{array}{r} A \\ +\ B \end{array} \qquad \begin{array}{rrrr} A_3 & A_2 & A_1 & A_0 \\ +\ B_3 & B_2 & B_1 & B_0 \end{array}$$





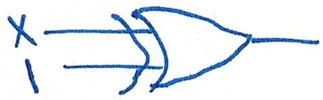


*Binary Subtraction:-

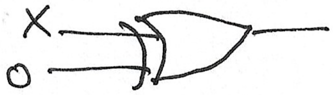
$A - B = A + 2\text{'s complement of } B$

$= A + 1\text{'s complement of } B + 1$

Note:



$F = 1 \oplus X = \overline{X}$

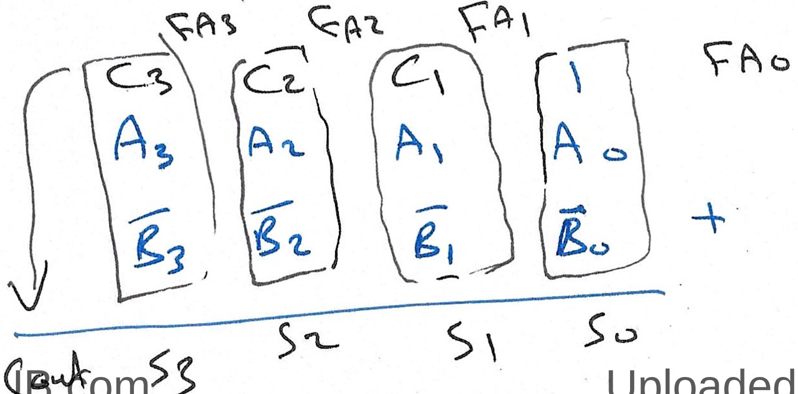


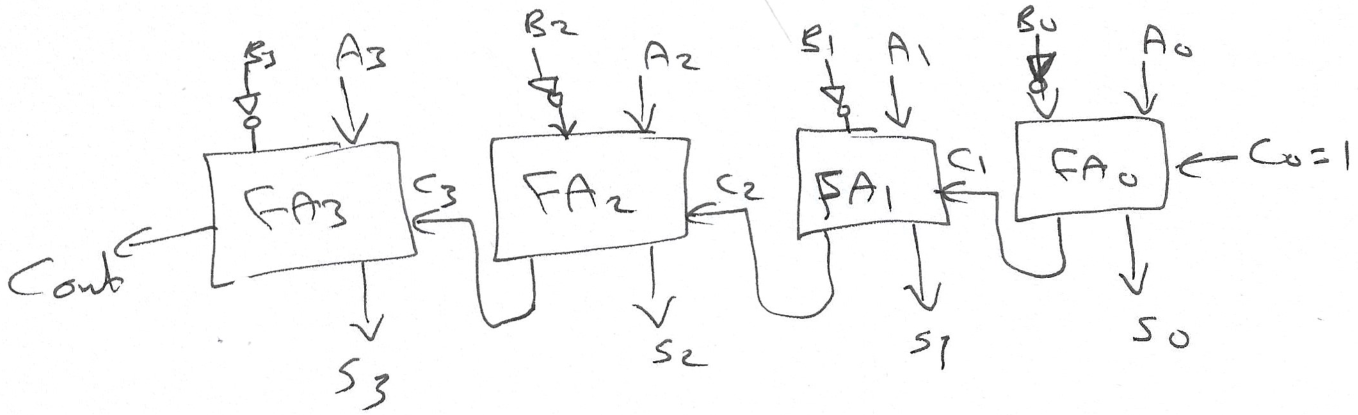
$F = 0 \oplus X = X$

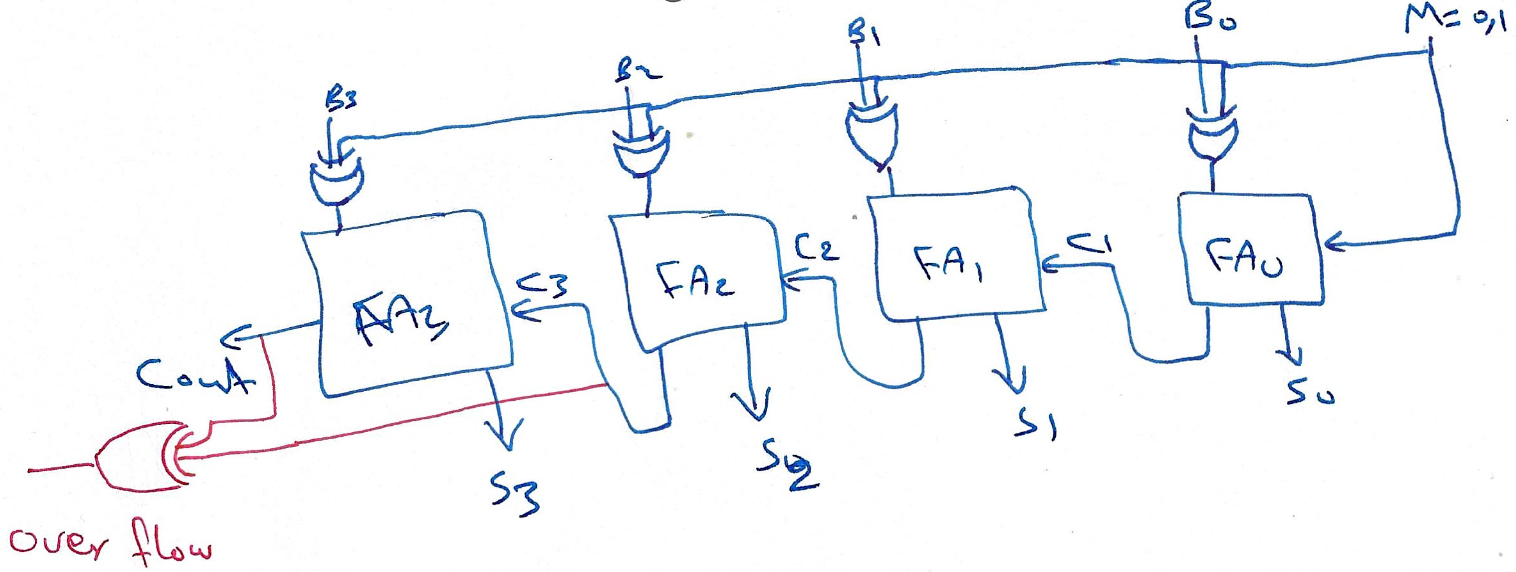
$A - B = A + 2\text{'s complement of } B$

$= A + 1\text{'s complement of } B + 1$

$= A + \overline{B} + 1$





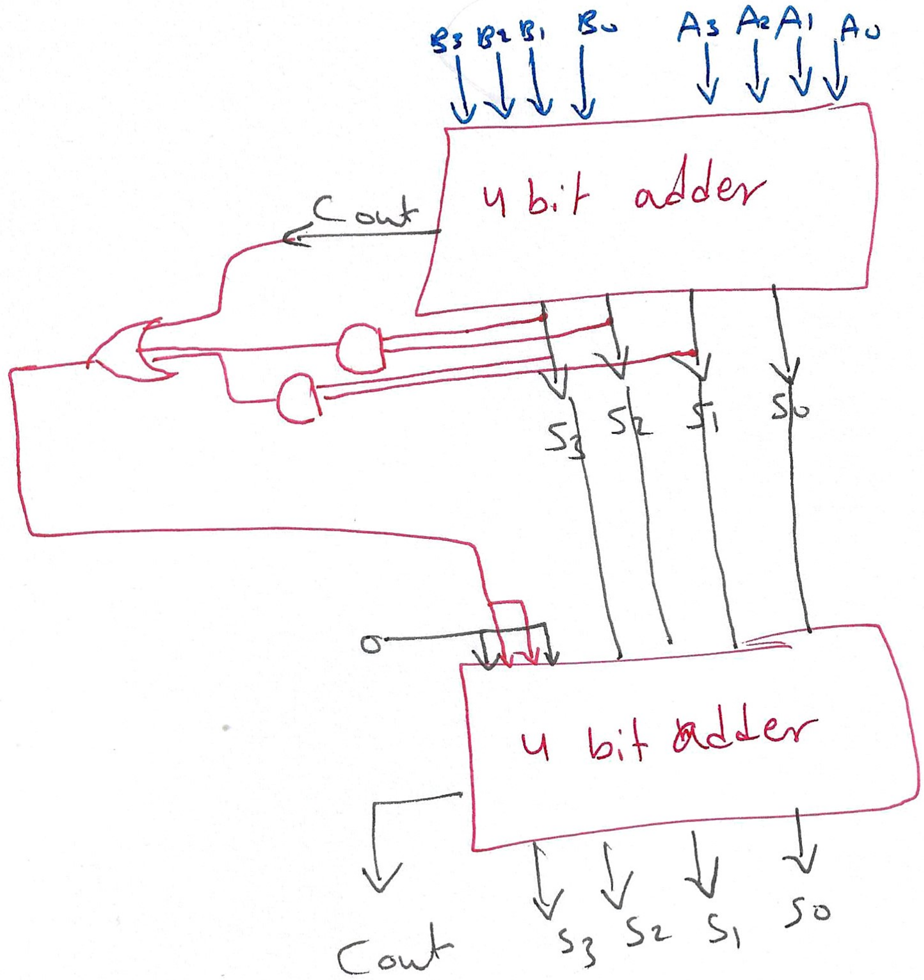


C_3	Cout	
0	1	overflow
1	0	
0	0	no overflow
1	1	

* BCD Adder

$$\begin{array}{r} 5 \\ +4 \end{array} \qquad \begin{array}{r} 101 \\ +\ 100 \\ \hline 1001 \end{array} \quad \text{Valid}$$

$$\begin{array}{r} +5 \\ 5 \end{array} \qquad \begin{array}{r} 01 \\ 101 \\ +\ 101 \\ \hline 1010 \\ +\ 110 \\ \hline 10000 \end{array} \quad \text{not Valid}$$

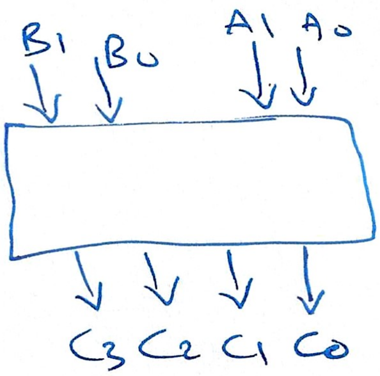


* Binary Multiplier:-

0 X 0 = 0
1 X 0 = 0
0 X 1 = 0
1 X 1 = 1

} AND operation

Example:- Design a 2-bit X 2 bit binary multiplier



" "
3 x 3 = 9 → 1001
4-bit

A_1	A_0	B_1	B_0	C_3	C_2	C_1	C_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

$C_0 = A_0B_0$

A_1A_0 \ B_1B_0	00	01	11	10
00				
01		1	1	
11		1	1	
10				

$C_1 = \bar{A}_1A_0B_0 + A_1\bar{A}_0B_0 + A_1\bar{B}_1B_0 + A_1A_0\bar{B}_1\bar{B}_0$

A_1A_0 \ B_1B_0	00	01	11	10
00				
01		1	1	
11		1		1
10		1	1	

$C_2 = A_1\bar{A}_0B_1 + A_1B_1\bar{B}_0$

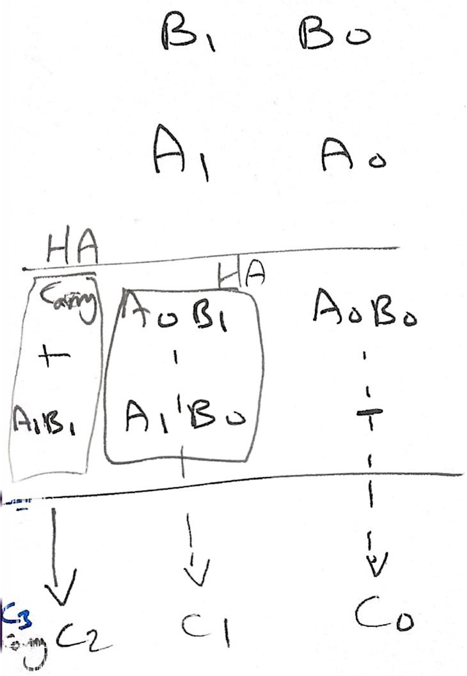
A_1A_0 \ B_1B_0	00	01	11	10
00				
01				
11				1
10			1	1

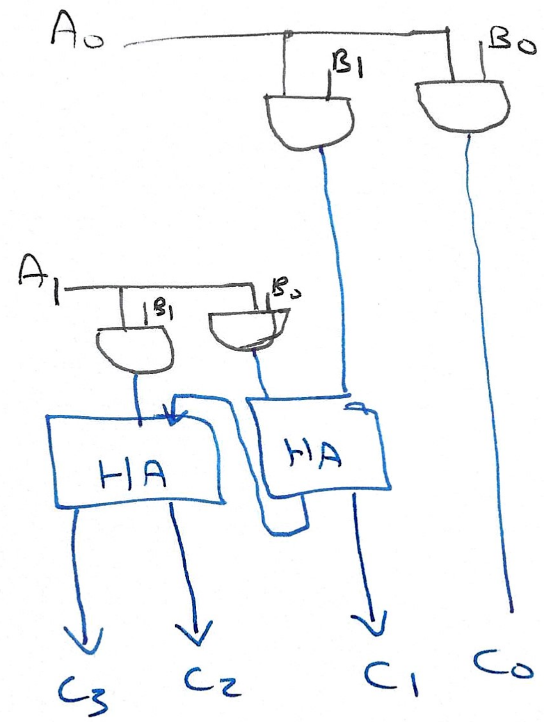
$C_3 = A_1A_0B_1B_0$

A_1A_0 \ B_1B_0	00	01	11	10
00				
01				
11			1	
10				

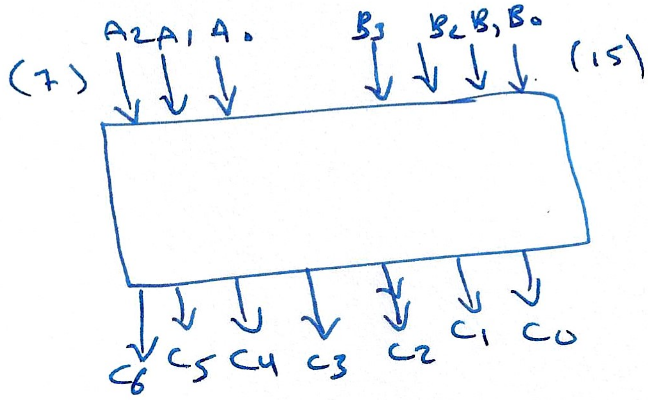
an efficient method

Example: 2-bit X 2-bit binary adder





Example: Design 4-bit X 3-bit multiplier

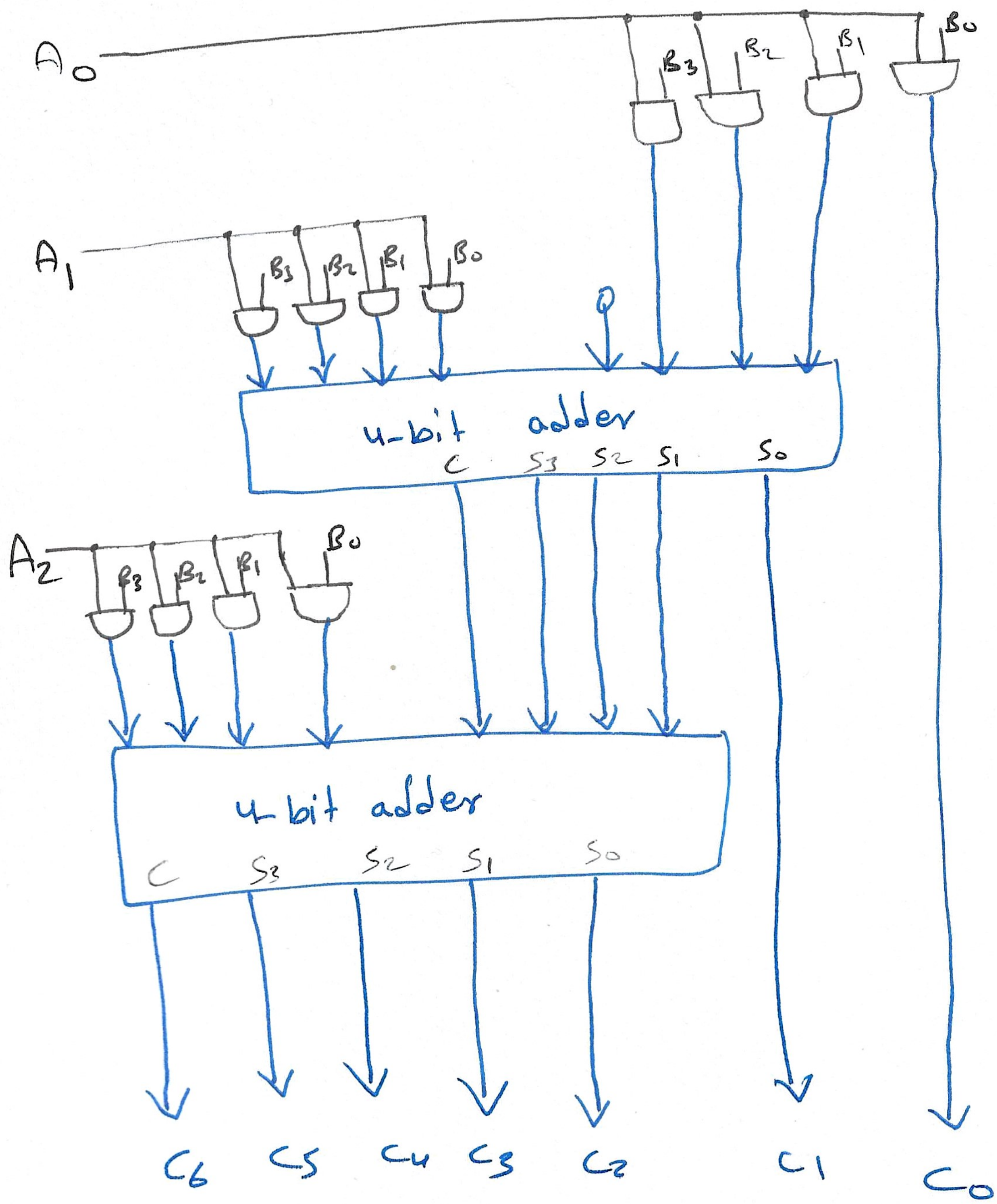


7*15 = 105

↓

7-bit

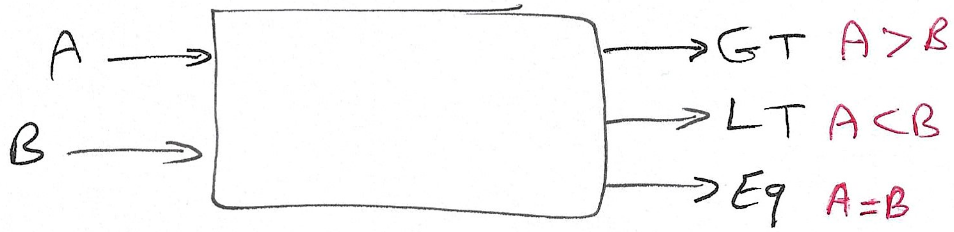
		B3	B2	B1	B0
			A2	A1	A0
		A0B3	A0B2	A0B1	A0B0
+	A1B3	A1B2	A1B1	A1B0	—
+ A2B3	A2B2	A2B1	A2B0	—	—



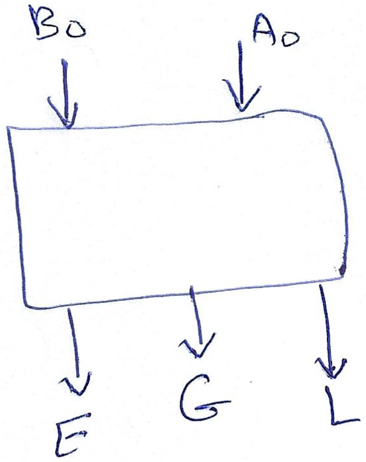
multiplicand → B → degree of adder

multiplier → A → number of adders

Magnitude Comparator:-



Example:- Design a 1 bit magnitude comparator.



A_0	B_0	G	L	E	
0	0	0	0	1	m_0
0	1	0	1	0	m_1
1	0	1	0	0	m_2
1	1	0	0	1	m_3

L

A_0 \ B_0	0	1
0		1
1		

G

A_0 \ B_0	0	1
0		
1	1	

E

A_0 \ B_0	0	1
0	1	
1		1

$L_0 = \bar{A}_0 B_0$

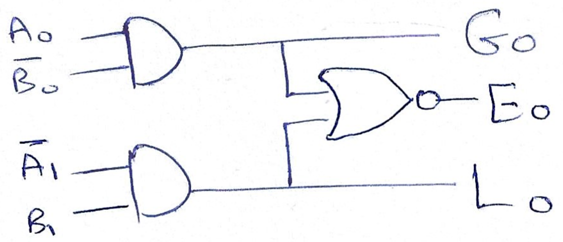
$G_0 = A_0 \bar{B}_0$

$E = \bar{A}_0 \bar{B}_0 + A_0 B_0$

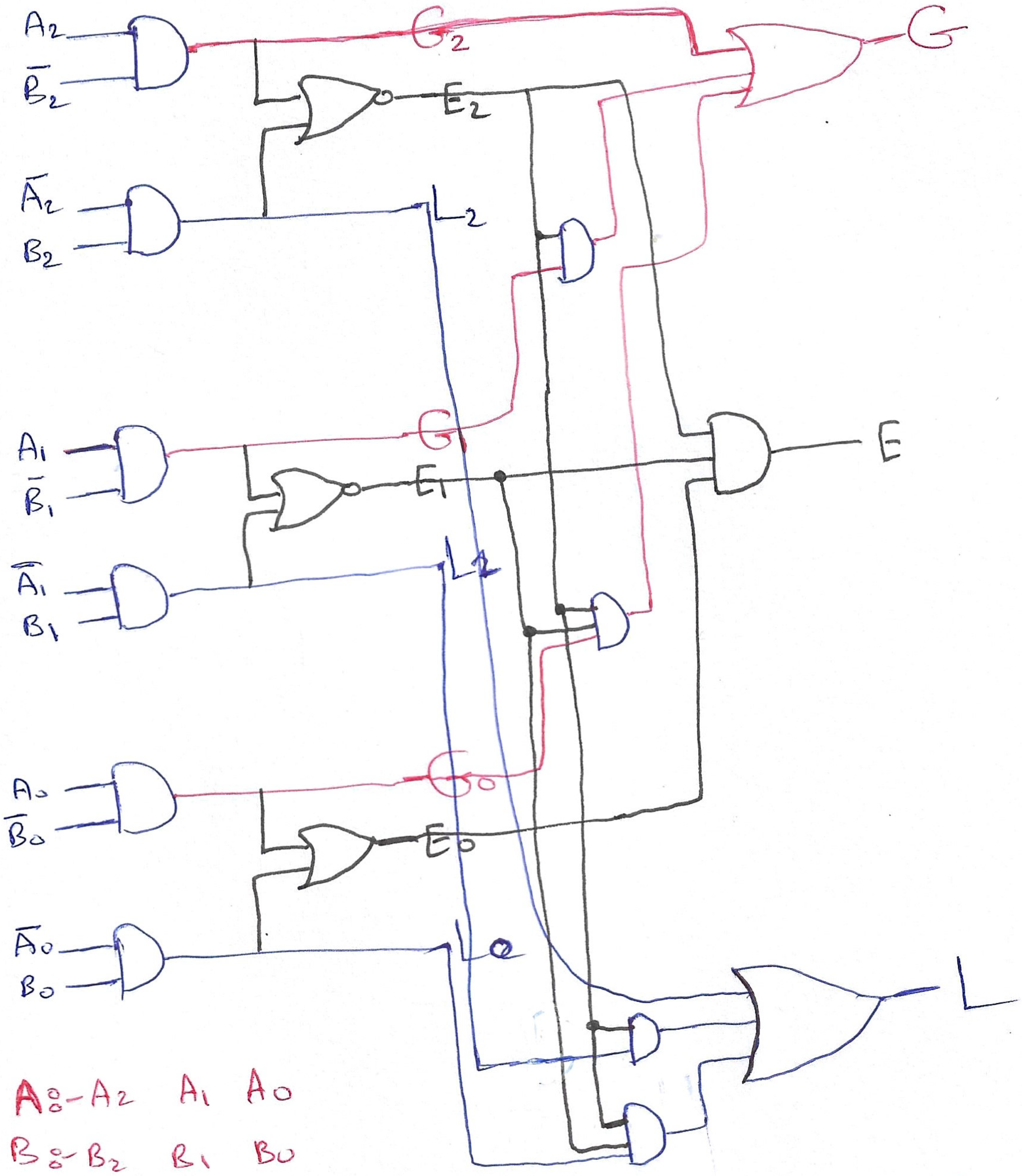
$$L_0 + G_0 = \bar{A} B_0 + A_0 \bar{B}_0 = A_0 \oplus B_0$$

$$E = \bar{A}_0 \bar{B}_0 + A_0 B_0 = \overline{(A \oplus B)}$$

$$= \overline{(L_0 + G_0)}$$



Example: Design a 3_bit magnitude Comparator



A : A_2 A_1 A_0

B : B_2 B_1 B_0

A = B : $\overset{E_2}{(A_2 == B_2)}$ && $\overset{E_1}{(A_1 == B_1)}$ && $\overset{E_0}{(A_0 == B_0)}$ = $E_2 \cdot E_1 \cdot E_0$

A > B : $(A_2 > B_2)$ || $(A_2 == B_2)$ && $(A_1 > B_1)$ || $(A_2 == B_2)$ && $(A_1 == B_1)$ && $(A_0 > B_0)$

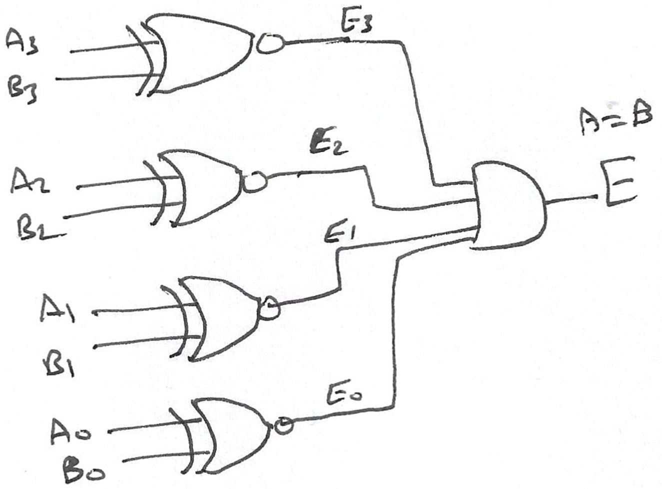
A < B : $(A_2 < B_2)$ || $(A_2 == B_2)$ && $(A_1 < B_1)$ || $(A_2 == B_2)$ && $(A_1 == B_1)$ && $(A_0 < B_0)$

Example: Design 4-bit X 4-bit equality comparator

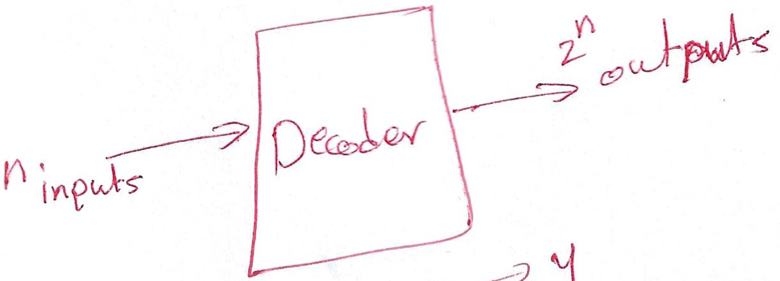
A: $A_3 A_2 A_1 A_0$

B: $B_3 B_2 B_1 B_0$

$A = B$: $(A_3 == B_3)$ && $(A_2 == B_2)$ && $(A_1 == B_1)$ && $(A_0 == B_0)$



* **Decoder**: Combinational Circuit that has n inputs and 2^n outputs.



(minterms OR maxterms)

2 → 4

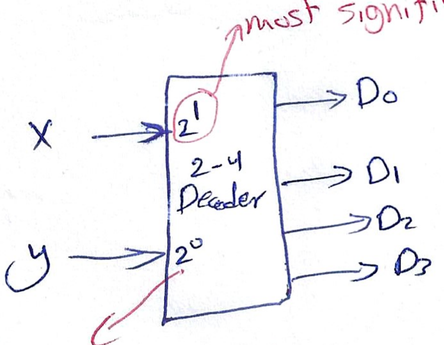
3 → 8

4 → 16

5 → 32

⋮

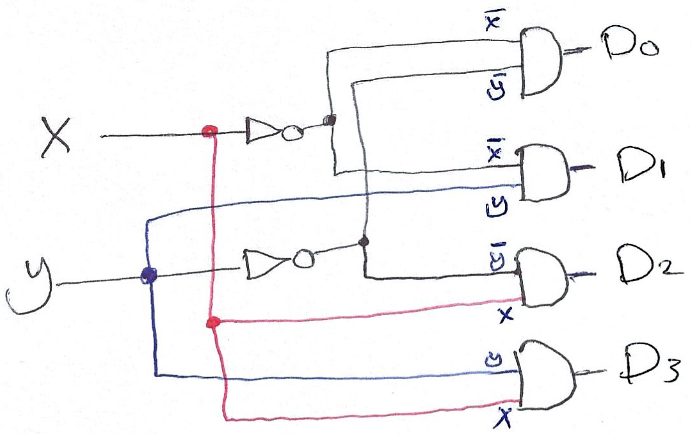
Example :- Design a 2→4 decoder



Block Diagram

	X	Y	D_0	D_1	D_2	D_3
m_0	0	0	1	0	0	0
m_1	0	1	0	1	0	0
m_2	1	0	0	0	1	0
m_3	1	1	0	0	0	1

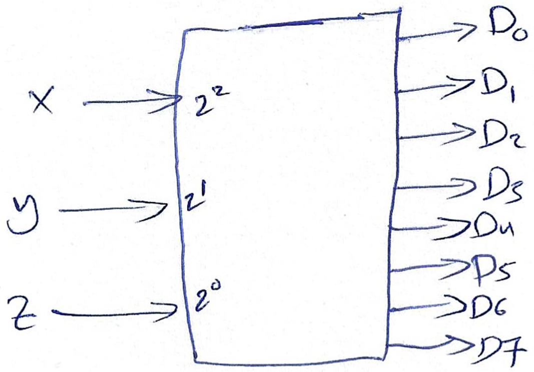
$D_0 = \bar{X}\bar{Y} = m_0$ $D_1 = \bar{X}Y = m_1$ $D_2 = X\bar{Y} = m_2$ $D_3 = XY = m_3$



outputs are mintermes

Circuit diagram

Example :- Design 3X8 Decoder



outputs are minterms

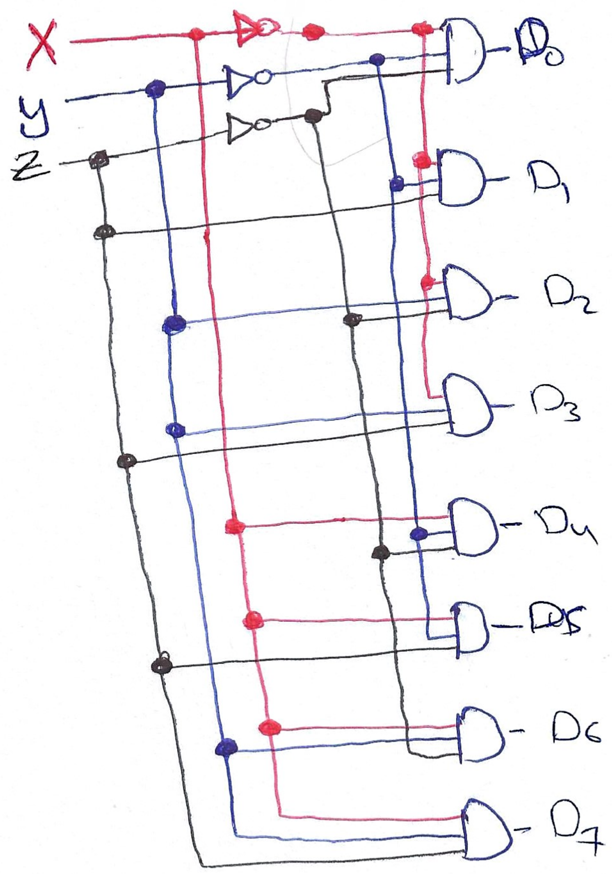
X	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$D_0 = \bar{x}\bar{y}\bar{z}$ $D_1 = \bar{x}\bar{y}z$

$D_2 = \bar{x}y\bar{z}$ $D_3 = \bar{x}yz$

$D_4 = x\bar{y}\bar{z}$ $D_5 = x\bar{y}z$

$D_6 = xy\bar{z}$ $D_7 = xyz$



Example: Implement the following function using decoder.

$F_1(x,y) = \sum(0,3)$, $F_2(x,y) = \sum(0,1,2)$, $F_3(x,y) = \prod(0,3)$

$F_1 = m_0 + m_3$, $F_2 = m_0 + m_1 + m_2$, $F_3 = \prod(0,3)$

OR gate
Sum of minterms

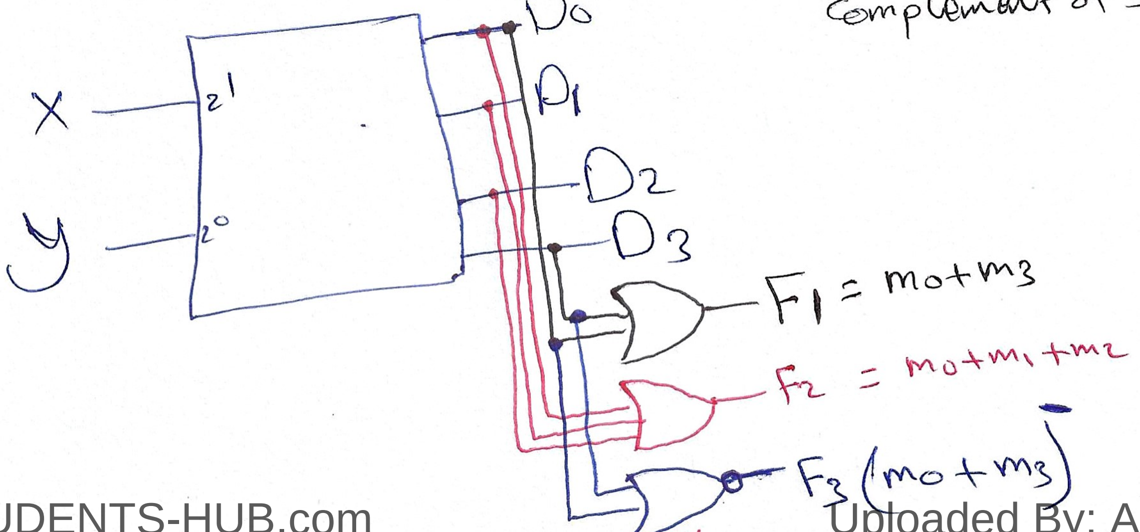
OR
gate Sum of minterms

$\bar{F}_3 = \sum(0,3)$

$\bar{F}_3 = m_0 + m_3$

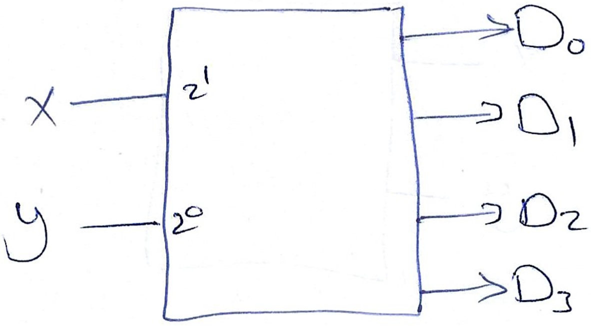
$\therefore F_3 = \bar{\bar{F}}_3 = \overline{(m_0 + m_3)}$

NOR gate
complement of sum of mintermes



Active High decoder

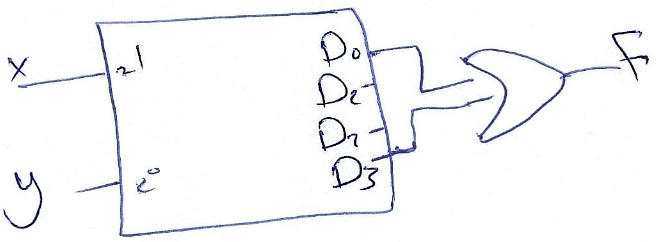
outputs are minterms



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

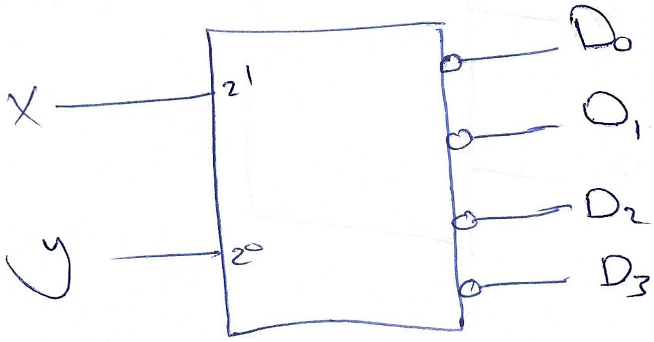
$D_0 = \bar{x}\bar{y}$, $D1 = \bar{x}y$, $D_2 = x\bar{y}$, $D_3 = xy$

$F(x,y) = \Sigma(0,3)$



Active low decoder

outputs are maxterms

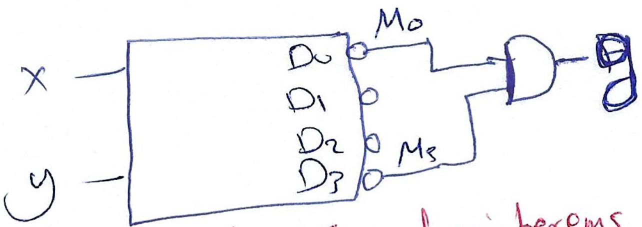


x	y	D_0	D_1	D_2	D_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

$D_0 = x+y = M_0 = (\bar{x}\cdot\bar{y})' = \overline{m_0}$

$D_1 = M_1 = x+\bar{y} = (\bar{x}\cdot y)' = \overline{m_1}$

$g(x,y) = \Pi(0,3)$



To implement a sum of minterms function using active low, we just use NAND

$F = \Sigma(0,3)$