Sarala. T
Asst. Prof.
E&CE Dept
SVCE.

Module - 2:

Analysis & Design of Combinational Logic:

General approach to Combinational logic Design:

To design combinational logic design steps are there:

→ Develop a statement describing the problem to be solved.

→ Based on the problem statement, construct a truth table that clearly establishes the relationship b/w the input & the output variables.

→ use K-map or Quine-McCluskey technique to simplify the functions in deriving the O/P Equations. The best solution will require the fewest gates & gate inputs.

→ Arrange the simplified equations to suit the logic primitive type to be used in realizing the circuit. using NAND-NOR, or AND-OR logic as required.

→ Draw the final logic diagram.

→ Document the design by identifying variables that indicates assertion levels & if possible Provide a truth table.

Example:

1) Design a combinational circuit that will multiply two - two bit binary values.

Soln:

Four I/p variables (A_1, A_0, B_1, B_0) & Four O/p variables (P_3, P_2, P_1, P_0) are needed. The four O/p variables are necessary because the maximum product of two - two bit values ($3_{10} \times 3_{10} = 9_{10}$) requires four bits.

Construct truth table.

Inputs				outputs.				
A_1	A_0	B_1	B_0	P_3	P_2	P_1	P_0	
0	0	0	0	0	0	0	0	$0_{10} \times 0_{10}$
0	0	0	1	0	0	0	0	$0_{10} \times 1_{10}$
0	0	1	0	0	0	0	0	$0_{10} \times 2_{10}$
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	1	1	
1	1	0	1	0	1	1	0	
1	1	1	0	0	1	1	0	
1	1	1	1	1	0	0	1	$3_{10} \times 3_{10} = 9_{10}$

The O/P SOP equations are.

$P_3 = f(A_1, A_0, B_1, B_0) = \Sigma(15)$

$P_2 = f(A_1, A_0, B_1, B_0) = \Sigma(10, 11, 14)$

$P_1 = f(A_1, A_0, B_1, B_0) = \Sigma(6, 7, 9, 11, 13, 14)$

$P_0 = f(A_1, A_0, B_1, B_0) = \Sigma(5, 7, 13, 15)$

The individual Simplified equations are.

$P_3 = A_1 A_0 B_1 B_0$

$P_2 = A_1 A_0' B_1 B_0' + A_1' A_0 B_1 B_0 + A_1 A_0 B_1 B_0'$

$= A_1 B_1 B_0' (A_0' + A_0) + A_1 A_0' B_1 B_0$

$= A_1 B_1 B_0' + A_1 A_0' B_1 B_0$

$= A_1 B_1 (B_0' + A_0' B_0)$ ∵ absorption law.

$P_2 = A_1 B_1 B_0' + A_1 B_1 A_0'$

$P_1 = A_1' A_0 B_1 B_0' + A_1' A_0 B_1 B_0 + A_1 A_0' B_1' B_0 + A_1 A_0' B_1 B_0'$
$+ A_1 A_0 B_1' B_0 + A_1 A_0 B_1 B_0'$

$= A_1' A_0 B_0 (B_1' + B_1) + A_1 A_0' B_0 (B_1' + B_1)$
$+ A_1 A_0 B_1' B_0 + A_1 A_0 B_1 B_0'$

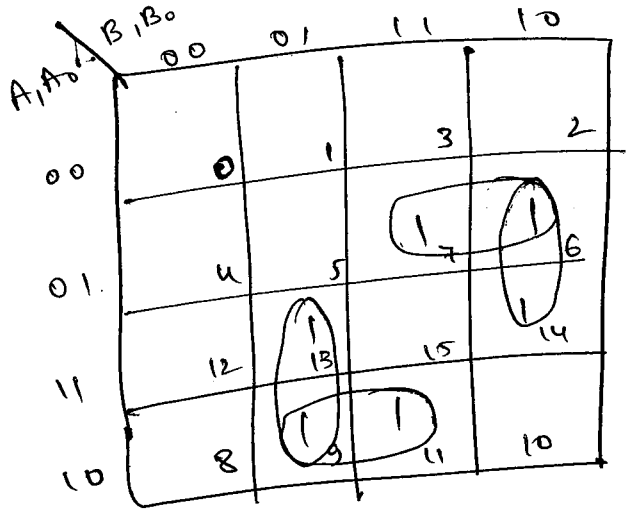
$= A_1' A_0 B_0 + A_1 A_0' B_0 + A_1 A_0 B_1' B_0 + A_1 A_0 B_1 B_0'$

$= A_0 B_0 (A_1' + A_1 B_1') + A_1 A_0' B_0 + A_1 A_0 B_1 B_0'$

$= A_0 B_0$

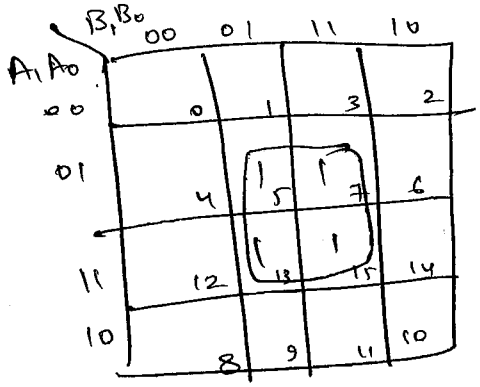
$P_1 = A_1'A_0B_1(B_0' + B_0) + A_1B_1'B_0(A_0 + A_0') + A_1B_1B_0'(A_0 + A_0')$

$= A_1'A_0B_1 + A_1B_1'B_0 + A_1B_1B_0' + A_1A_0'B_0($



$P_1 = B_1'B_0A_1 + A_1A_0'B_0 + A_1'A_0B_1 + B_1B_0'A_0$

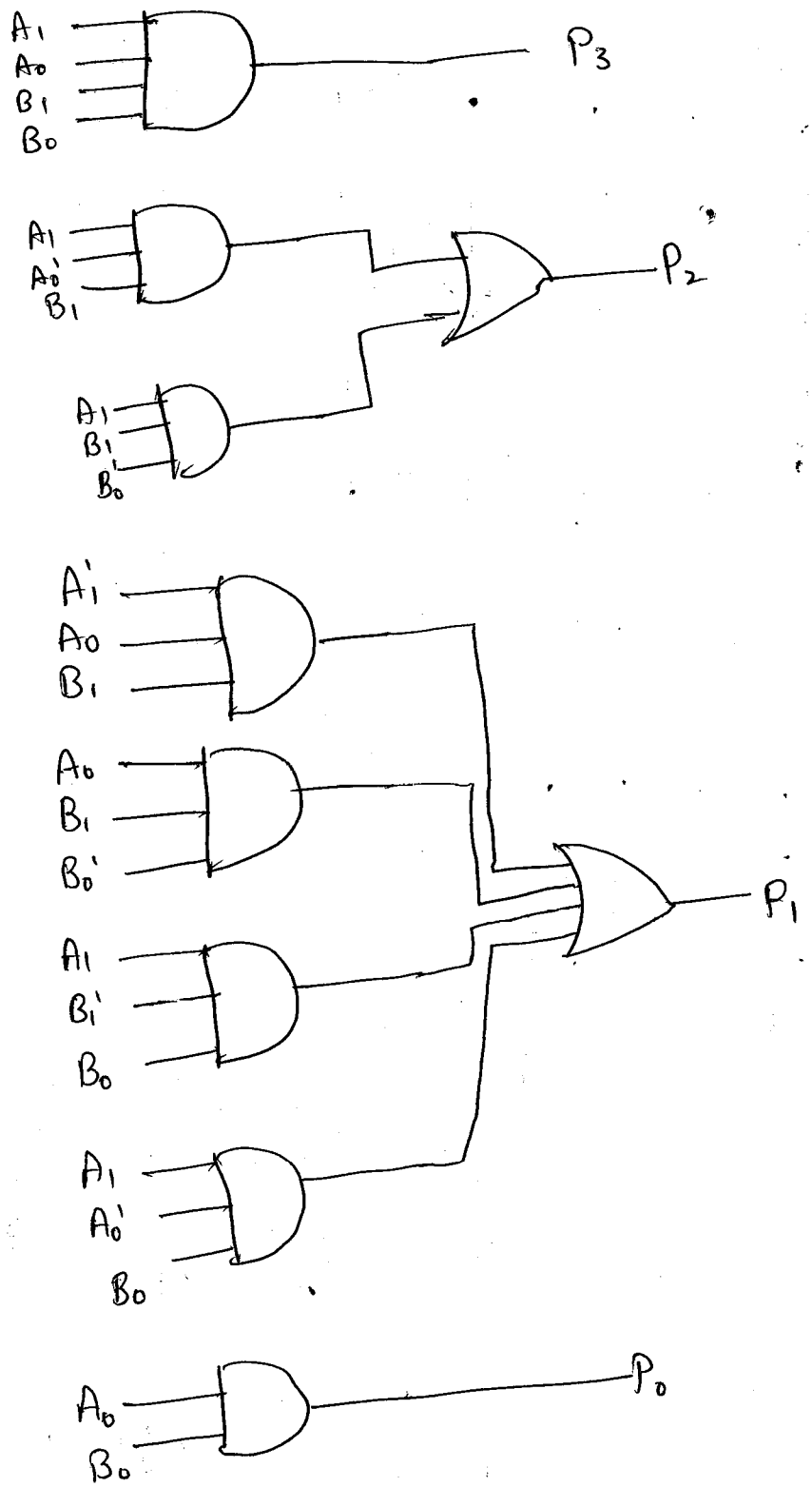
$P_0 =$



$P_0 = A_0B_0$

→ Realize the simplified equations using two-level AND-OR network.

→ Draw the mixed logic diagram that satisfies the equations.


A1
A0
B1
B0
P3
A1
A0'
B1
A1
B1
B0'
P2
A1'
A0
B1
A0
B1
B0'
A1
B1'
B0
A1
A0'
B0
P1
A0
B0
P0

Example 2:

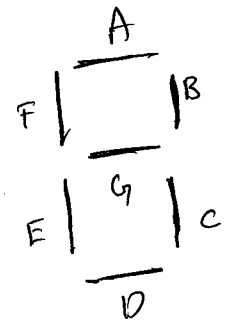
2) Design a combinational circuit that will accept a 2421 BCD code & drive a TIL-312 Seven segment display.

Soln: The 2421 code is shown in table

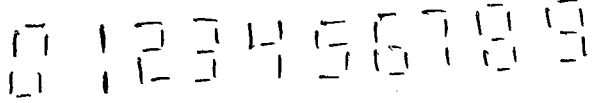
TIL-312 is a common-anode, red light emitting diode display package.

2421 code.

Decimal	2 w	4 x	2 y	1 z
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1



The LED is lit when the control connected to the cathode is a logic 0.



Generate a truth table illustrating active low o/p's to turn on each of the seven segment LED's.

Decimal	min/max term decimal	Inputs 2 4 2 1 w x y z	outputs A	B	C	D	E	F	G	#
0	0	0 0 0 0	0	0	0	0	0	0	1	0
1	1	0 0 0 1	1	1	1	1	0	0	1	1
2	8	1 0 0 0	0	0	1	0	0	1	0	2
3	9	1 0 0 1	0	0	0	0	1	1	0	3
4	10	1 0 1 0	1	0	0	1	1	0	0	4
5	11	1 0 1 1	0	1	0	0	1	0	0	5
6	12	1 1 0 0	0	1	0	0	0	0	0	6
7	13	1 1 0 1	0	0	0	1	1	1	1	7
8	14	1 1 1 0	0	0	0	0	0	0	0	8
9	15	1 1 1 1	0	0	0	0	1	0	0	9

$A = \pi(0, 8, 9, 11, 12, 13, 14, 15)$

$B = \pi(0, 8, 9, 10, 13, 14, 15)$

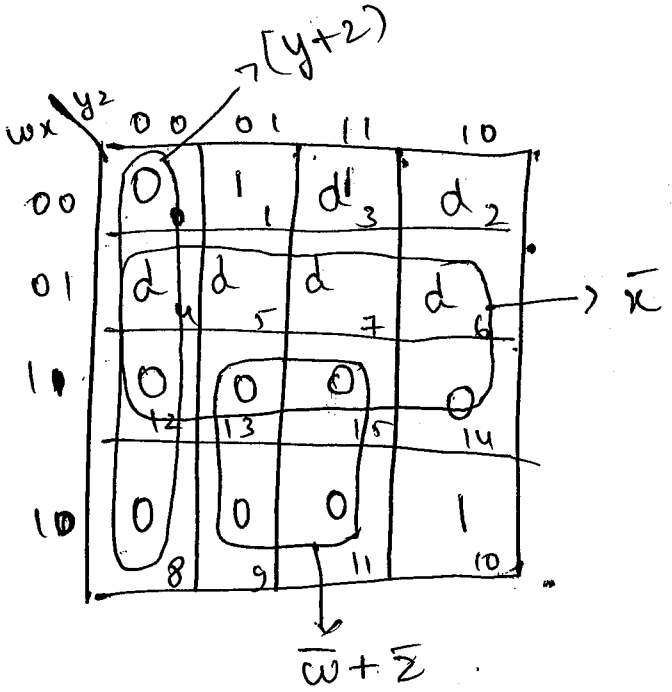
$C = \pi(0, 9, 10, 11, 12, 13, 14, 15)$

$D = \pi(0, 8, 9, 11, 12, 14, 15)$

$E = \pi(0, 1, 8, 12, 14)$

$F = \pi(0, 1, 10, 11, 12, 14, 15)$

$G = \pi(8, 9, 10, 11, 12, 14, 15)$



$A = \Sigma(1,10)$

$B = \Sigma(11,12)$

$C = \Sigma(8)$

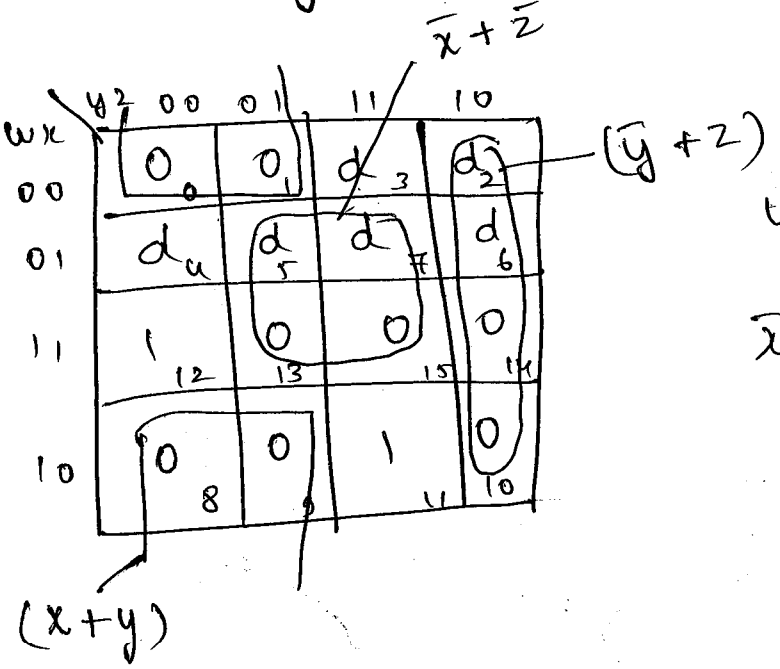
$D = \Sigma(1,10,13)$

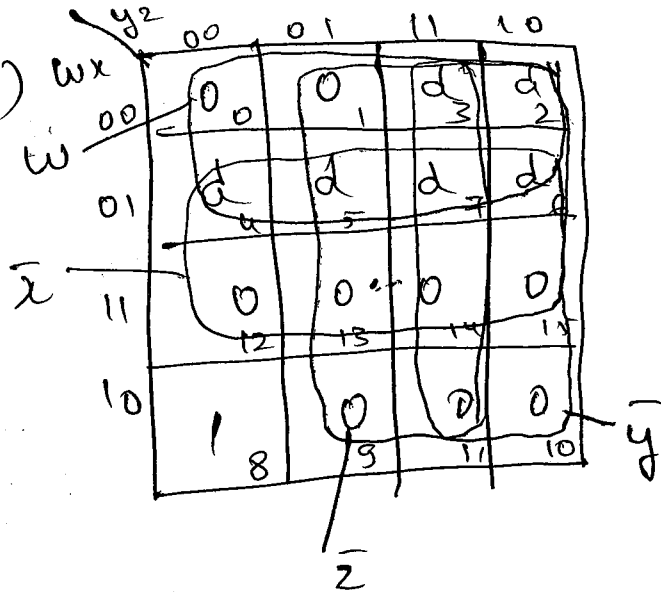
$E = \Sigma(1,9,10,11,13,15)$

$F = \Sigma(1,8,9,13)$

$G = \Sigma(0,1,13)$

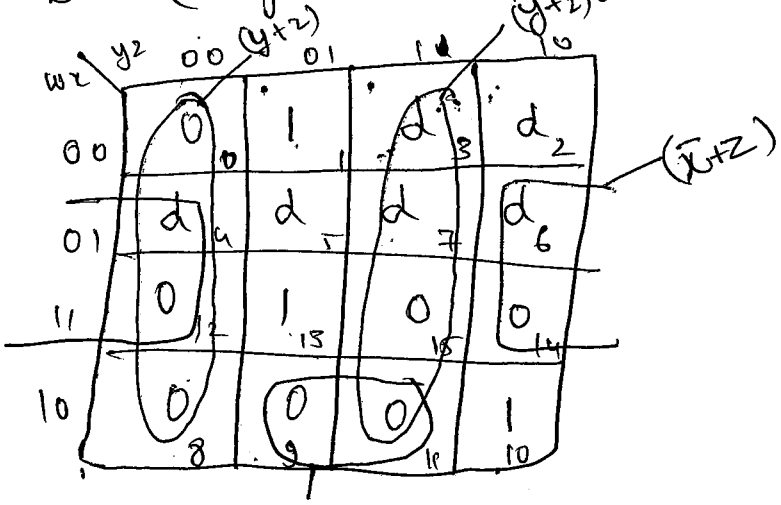
$A = (y+z)(\bar{w}+\bar{z})(\bar{x})$

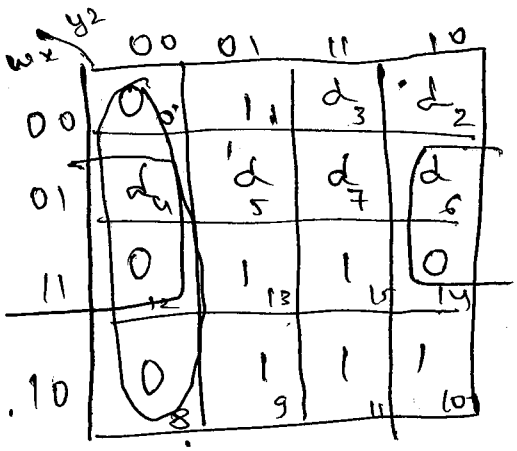




$B = (x+y)(\bar{x}+\bar{z})(\bar{y}+z)$

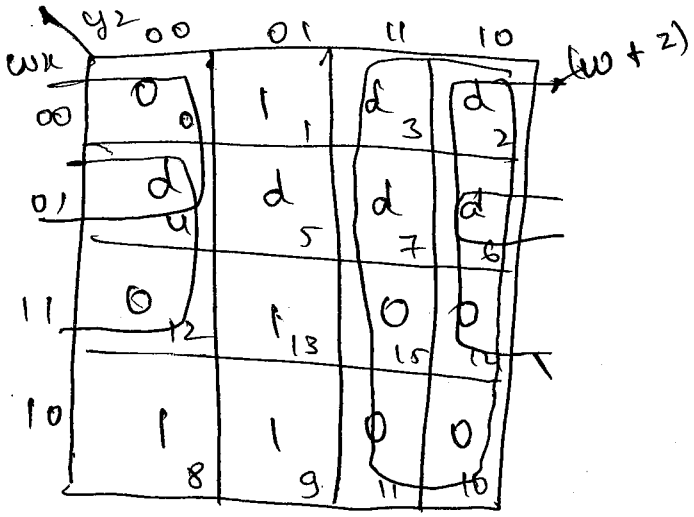
$C = w\bar{x}\bar{y}z$

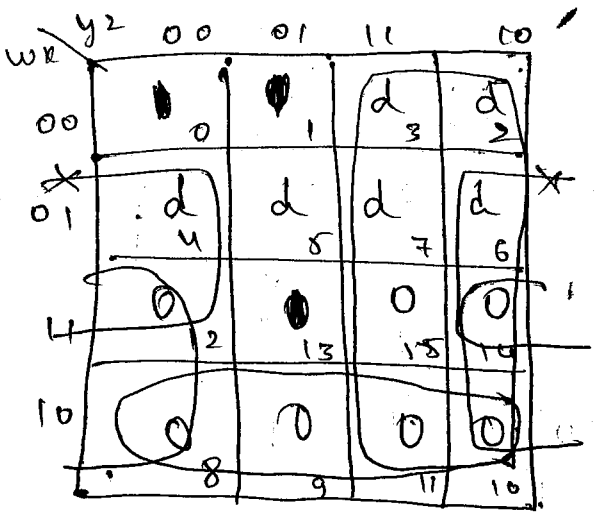




$E = (y+z)(\bar{x}+z)$

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





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

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

For active low SOP ~~Equals to~~. Equations are.

$A = \Sigma(1, 10)$

$B = \Sigma(4, 12)$

$C = \Sigma(8)$

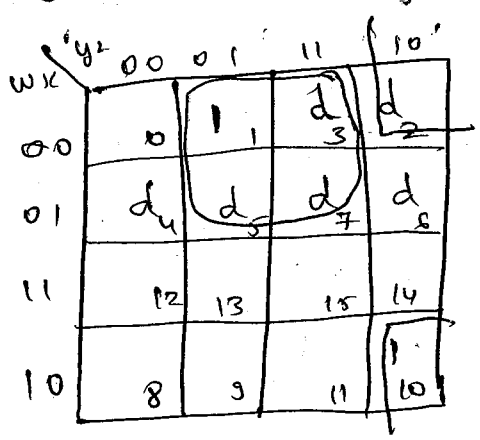
$D = \Sigma(1, 10, 13)$

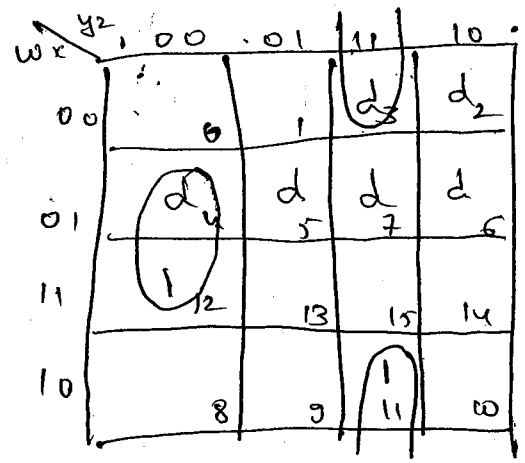
$E = \Sigma(1, 9, 10, 11, 13, 15)$

$F = \Sigma(1, 8, 9, 13)$

$G = \Sigma(0, 1, 13)$

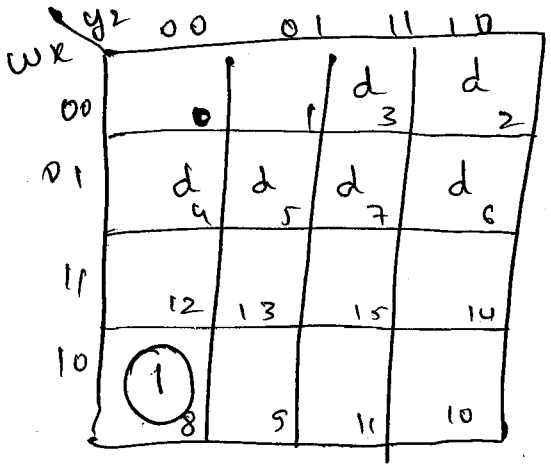
don't care terms are $\Sigma d(2, 3, 4, 5, 6, 7)$



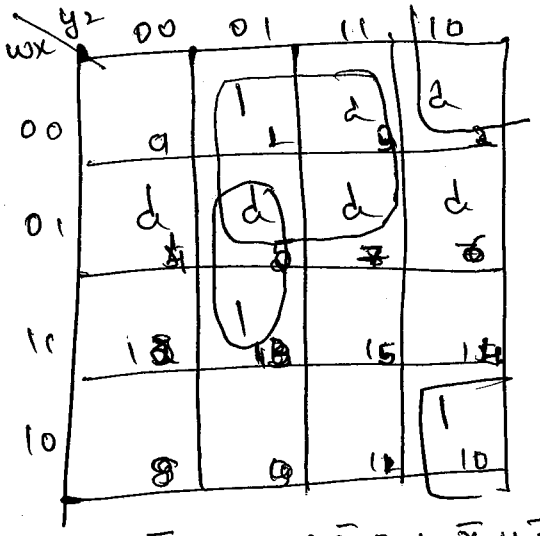


$A = z + \bar{x}y\bar{z}$

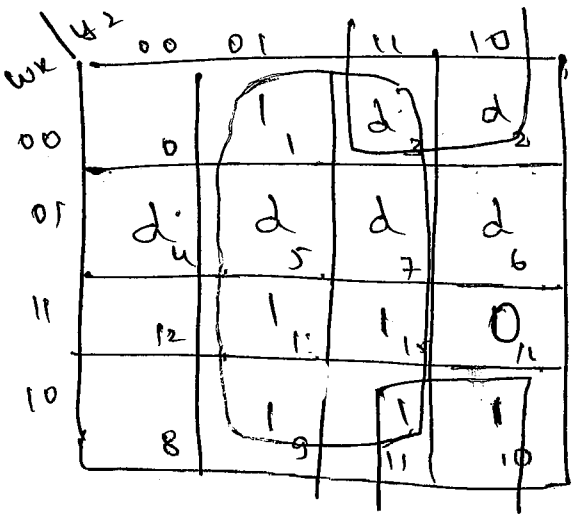
$B = x\bar{y}\bar{z} + \bar{x}y2$



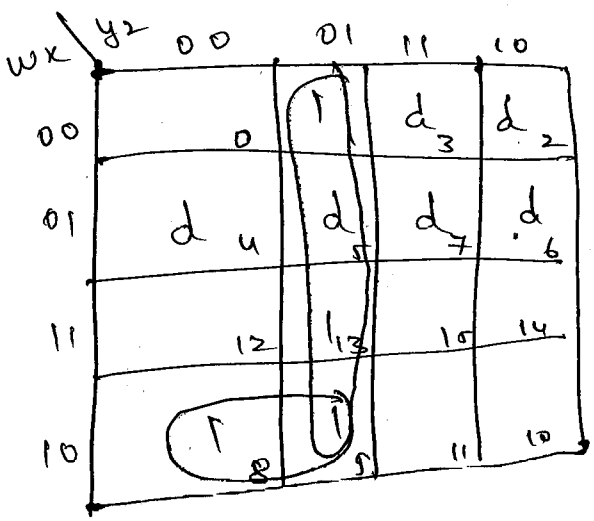
$C = w\bar{x}\bar{y}\bar{z}$



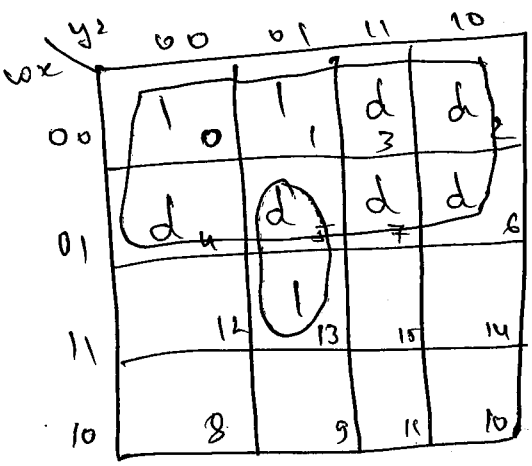
$D = \bar{w}z + x\bar{y}z + \bar{x}y\bar{z}$



$E = z + y\bar{z}$



$F = \bar{y}z + w\bar{x}\bar{y}$



$G = \bar{w} + x\bar{y}z$

→ The Initial requirement to realize the circuit, using NAND gates can be by using SOP equations.

→ using Demorgans Theorem the conversion of the Preceding eqn yields a set of NAND functions.

$$A = \overline{(\overline{\bar{w}\,z})\;(\overline{\bar{x}\,y\,\bar{z}})}$$ Nand Gate realization.

$$B = \overline{\left[\overline{(x\,y\,\bar{z})}\;\overline{(\bar{x}\,y\,z)}\right]}$$

$$C = \overline{\overline{(w\,\bar{x}\,\bar{y}\,\bar{z})}}$$

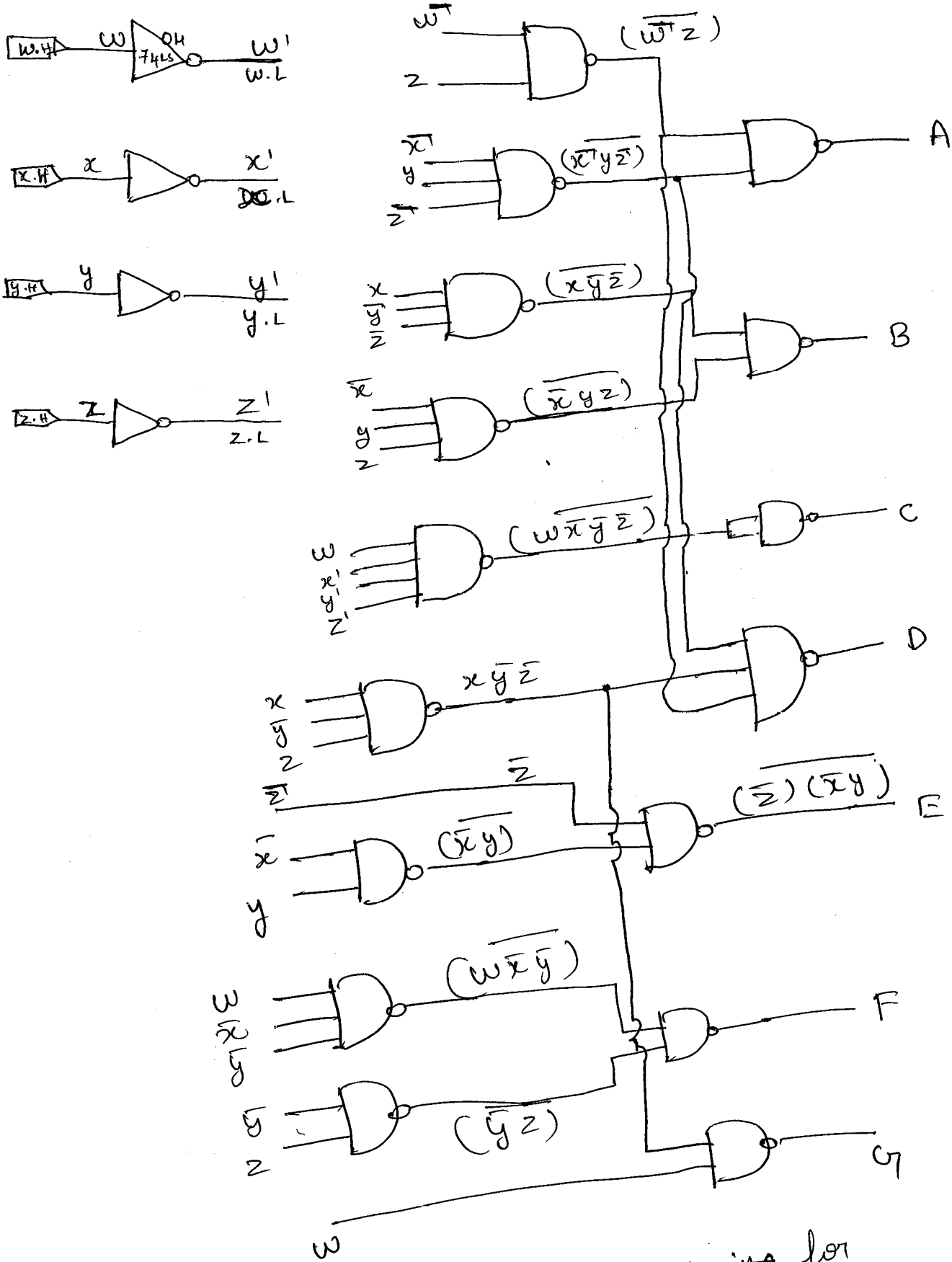
$$D = \overline{\left[\overline{(x\,\bar{y}\,z)}\;\overline{(\bar{x}\,y\,\bar{z})}\;\overline{(\bar{w}\,z)}\right]}$$

$$E = \overline{\left[\overline{(\bar{x}\,y)}\;\overline{(\bar{z})}\right]}$$

$$F = \overline{\left(\overline{(w\bar{x}\,\bar{y})}\;\overline{(\bar{y}\,\bar{z})}\right)}$$

$$G = \overline{\left[\overline{(w)}\;\overline{(x\,y'\,z')}\right]}$$

Logic diagram for 2421 BCD to 7 segment Display:



NAND gate realization using for 2421 BCD to 7 segment display.

Integrated circuits:

→ Small scale integrated circuits can be interconnected to realize combinational logic design:

Hierarchy of IC's based on no. of equivalent gates

→ SSI — 10 equivalent gates.

MSI — 100 gates

LSI — >100 <1000 gates

VLSI → >1000 gates

Digital integrated circuit logic families are:

→ Transistor transistor logic (TTL)

→ Emitter coupled logic (ECL)

→ complementary metal oxide semiconductor logic (CMOS)

The numbering system used to indicate the various subfamilies is partially indicated at the top of IC.

SN 54 74

SN → Prefix - for Texas instruments (manufacturer)

54 → Military operation temperature range (−55 to 125°C)

74 → Commercial temp range (0 to 70°C)

SN 74ALS00

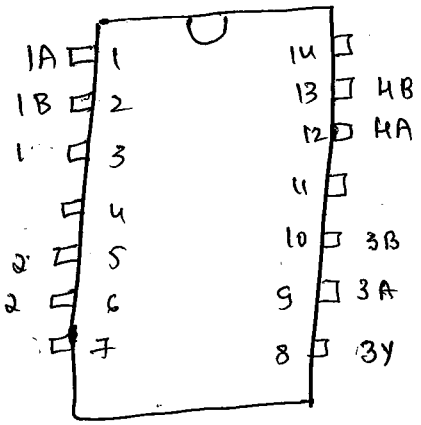
SN → Texas LS – low power schottky

74 – commercial temp range

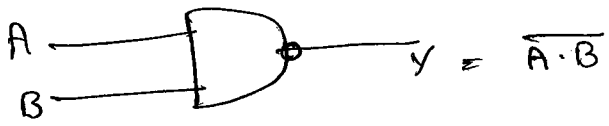
ALS – Advanced low power schottky (Increased speed & reduced power consumption)

00 – NAND operation.

DIP → Dual in-line packaging. (Epoxy plastic or ceramic)



SN74LS00



I/P A	I/P B	O/P Y
H	H	L
L	X	H
X	L	H

TTL circuit parameters:

V_{cc}, Power supply voltage: ~~Supp~~

I_{CCH} Power supply current. – 1

I_{CCL} Power supply current – 0

I_{OS} Short circuit O/p current – Amount of current driven to the circuit.

I_{OL} O/p current low – ~~low~~ low O/p current – O/p is 0.

I_{OH} O/p current high – high o/p current – O/p is 1

V_{OL} Voltage O/p low – max allowable O/p vtg. that represents '0'.

V_{OH} Vtg O/p High – ~~max~~ min allowable o/p vtg. that represents '1'.

I_{IH} current I/p high — I/p Vtg rep – 1

I_{IL} current ~~I~~/p ~~high~~ low – I/p vtg rep – 0

V_{IH} Voltage input High — min vtg to repn logical 1.

$V_{IH} = 2.0\ V$

V_{IL} Voltage Input low — max logical vtg to rep '0'.

$V_{IL} = 0.8\ V$

t_{PLH} Propagation delay time from o/p low to high.

t_{PHL} Propagation delay time from o/p high to low.

Fan out — Fan out is defined as the number of gate inputs that a single gate output can drive and still maintain voltage & current specifications.

TTL designation	function	I/p's
74XX00	Quad NAND	2
74XX02	Quad NOR	2
74XX04	Hex Inverter	1
74XX08	Quad AND	2
74XX10	tripple NAND	3
74XX11	tripple AND	3
74XX20	dual NAND	4
74XX21	Dual AND	4
74XX27	Tripple NOR	3
74XX30	NAND	8
74XX32	Quad OR	2
74XX86	Quad XOR	2

Decoders:

→ Decoders are a class of combinational logic circuits that convert a set of input variables representing a code into a set of output variables representing a different code.

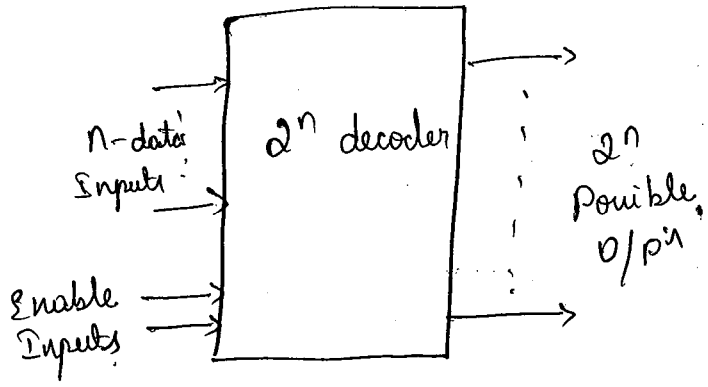
→ Encoded information is represented in n inputs producing 2^n outputs.

$n - 2^n$

→ O/P values 2^n ranges from $0 \rightarrow 2^n - 1$

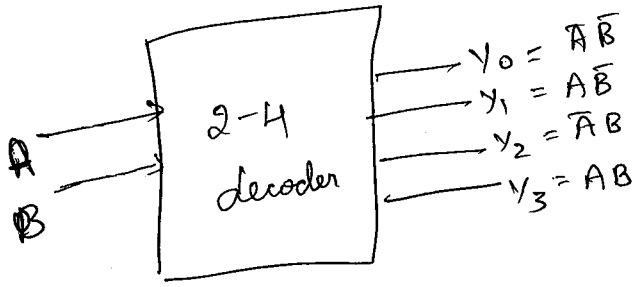
→ Decoders also have inputs to activate or enable decoded output based on data inputs.

→ model of Decoder.



2 - H line decoder.

I/P		O/P			
A	B	Y_0	Y_1	Y_2	Y_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



→ Decoders are available as integrated circuits

IC → 74XX139 → dual 2 to 4 decoder.

I/P

$n = 2 \quad \therefore 2^n = 2^2 = 4$ O/P.

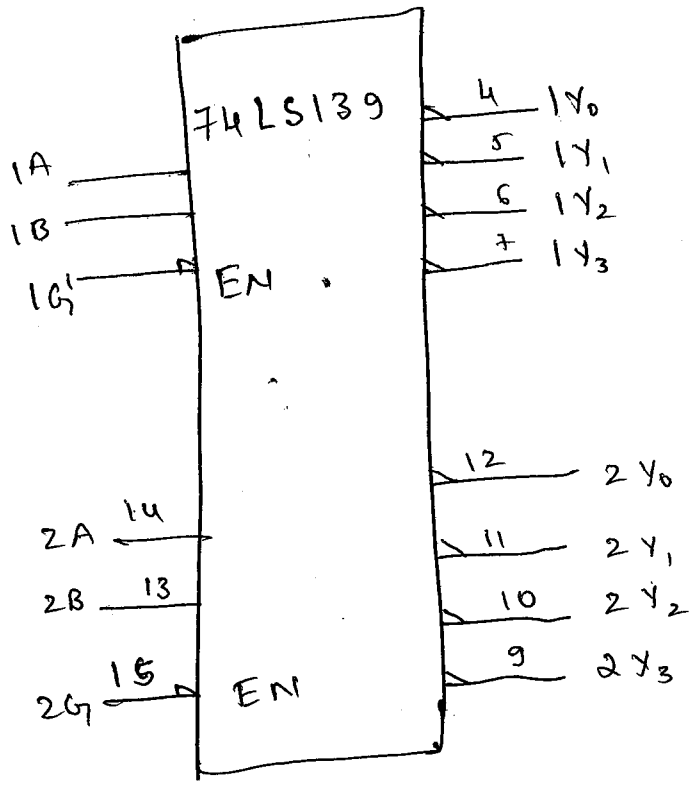
→ one & only one of the outputs Y_0 to Y_3, is active for a given output.

→ Enable G_1 in diagram is active low.

∴ If $G_1 = 0$ than only O/P's will be driven to O/P lines.

Truth table:

Inputs			outputs			
EN G_1'	Select A	B	Y_0	Y_1	Y_2	Y_3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L



IEEE Symbol.

Realize the Boolean functions.

Ex:

$f = \bar{A} + B$

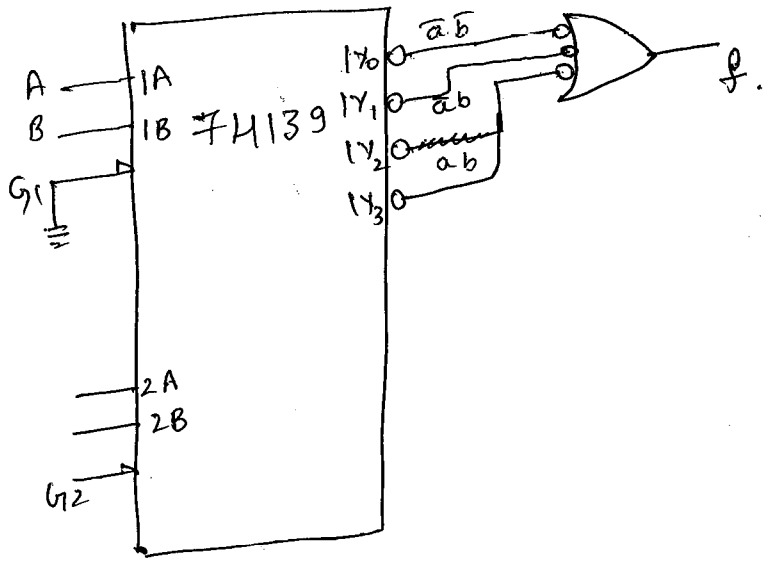
minterm canonical form.

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

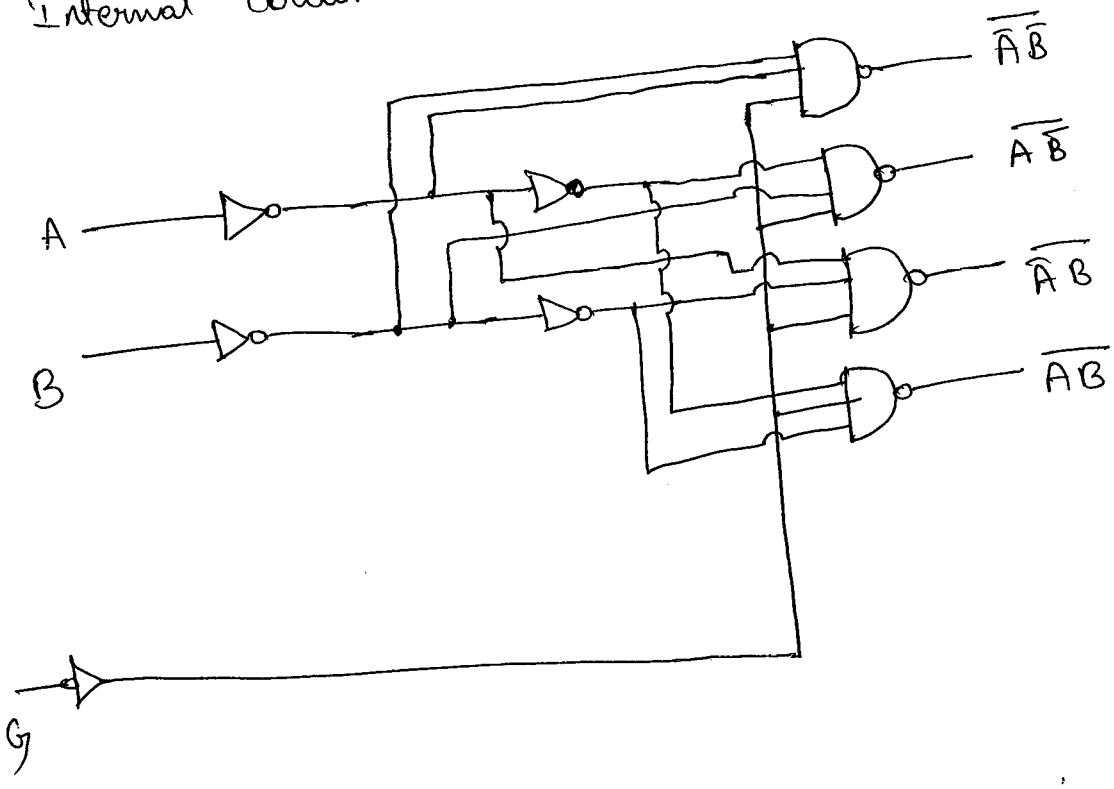
$= \bar{A}B + \bar{A}\bar{B} + AB + \bar{A}B$

$= 01 \quad 00 \quad 11$

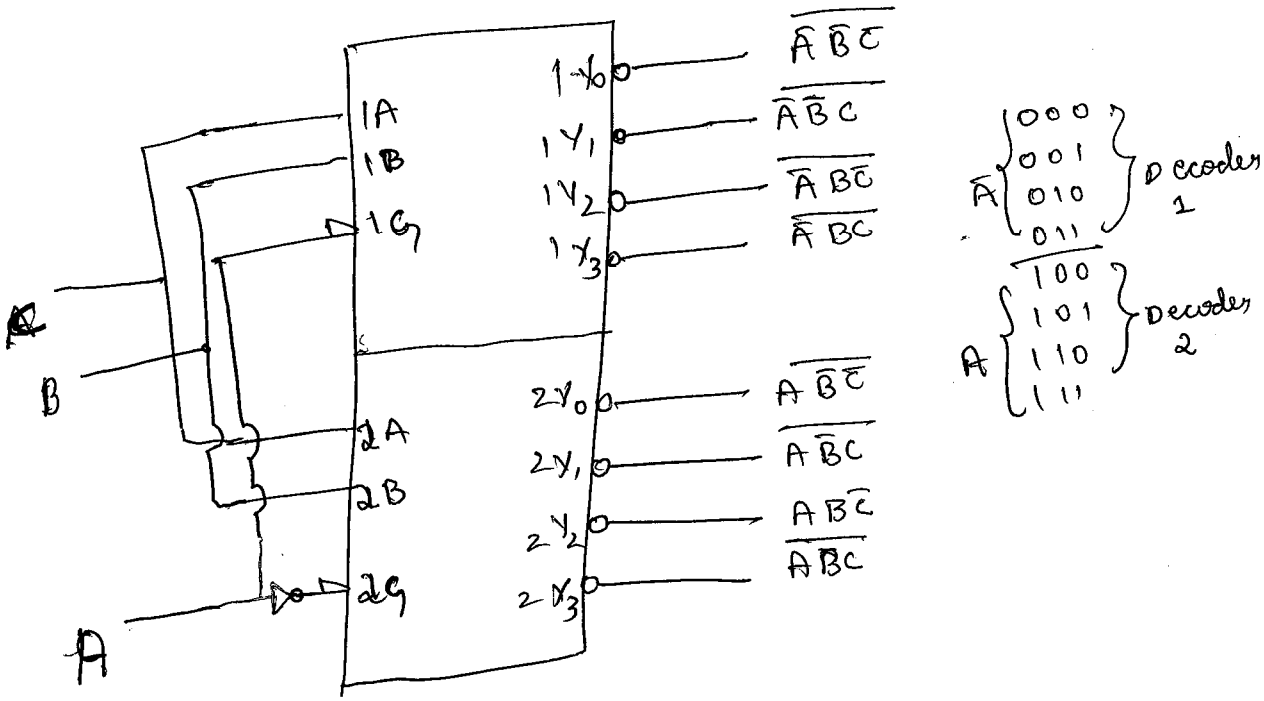
$f = \Sigma(1, 0, 3)$



Internal circuit -



from IC 74139 we can realize 3-8 decoder by using the enable input as 3rd Input.



for first 4 I/p values 1st 2-4 Decoder will be enabled & for other 4 I/p values 2nd decoder will be enabled.

Problem:

1) Realize the following Boolean functions using 74139

a) $f_1(a,b) = \Sigma(0,2)$

b) $f_2(a,b,c) = \Sigma(1,3,6,7)$

3 to 8 line decoder:

MSI - medium scale integrated.

The 74xx138 is a 3 to 8 MSI decoder IC. The three inputs are decoded to produce one of eight outputs. Three Enable inputs are provided, all of which must be active before decoding can occur.

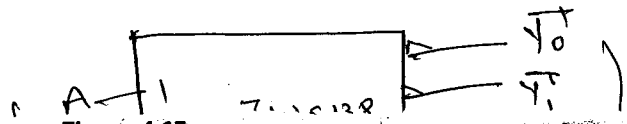
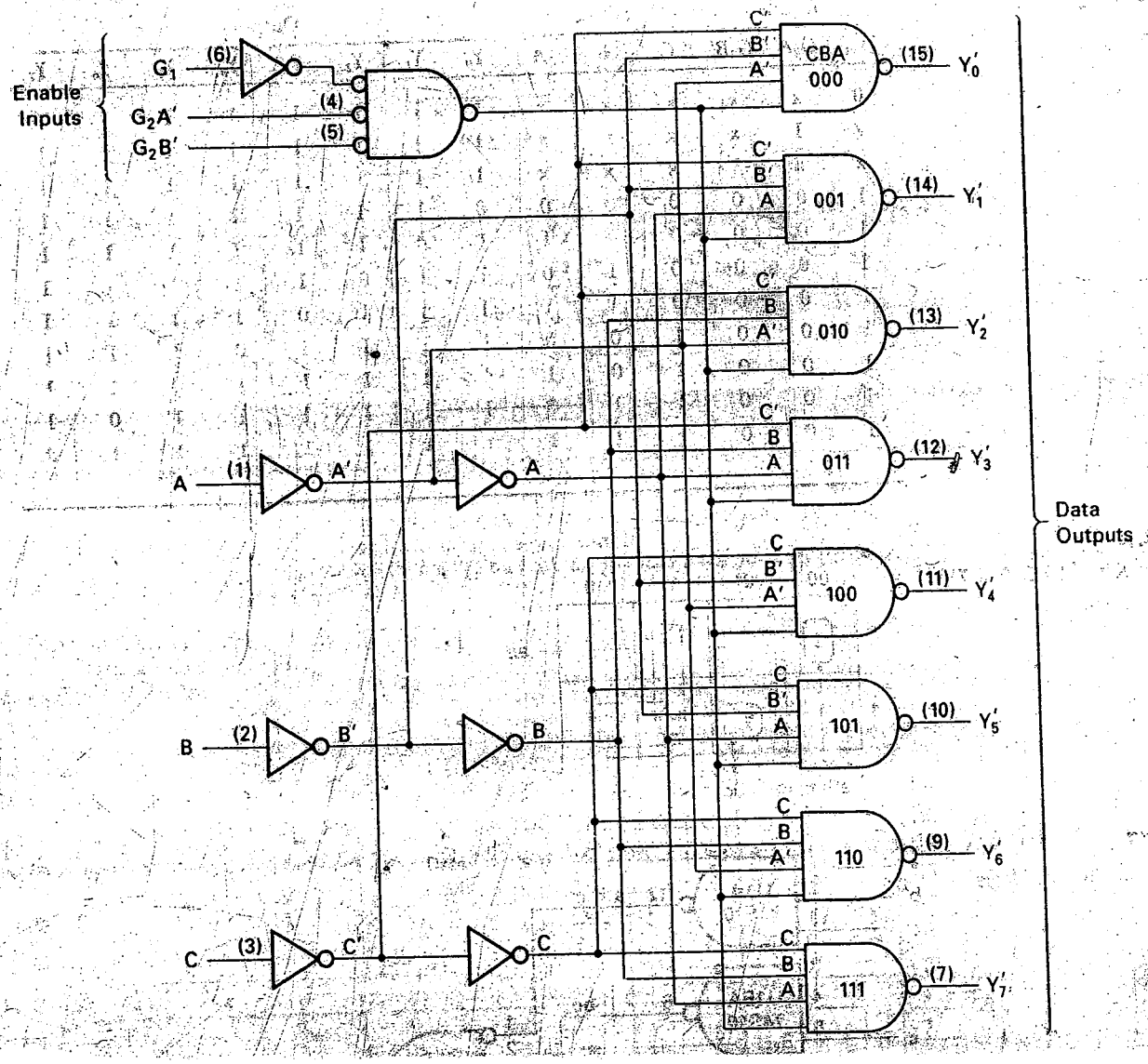


Figure 4.15
74xx138 decoder



Function table for 74xxx138 decoder

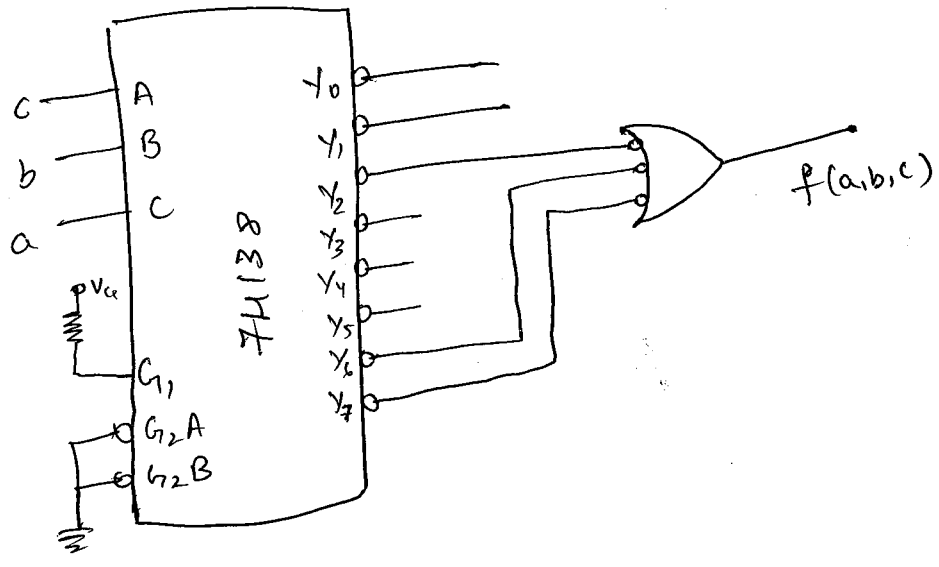
I/p						O/p's.							
Enable			Select										
G_1	$\overline{G_{2A}}$	$\overline{G_{2B}}$	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
0	x	x	x	x	x	1	1	1	1	1	1	1	1
x	1	x	x	x	x	1	1	1	1	1	1	1	1
x	x	1	x	x	x	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

A 3-8 decoder can be used straight away for the implimentation of logic expression three variables as illustrated in the examples below.

Problem:

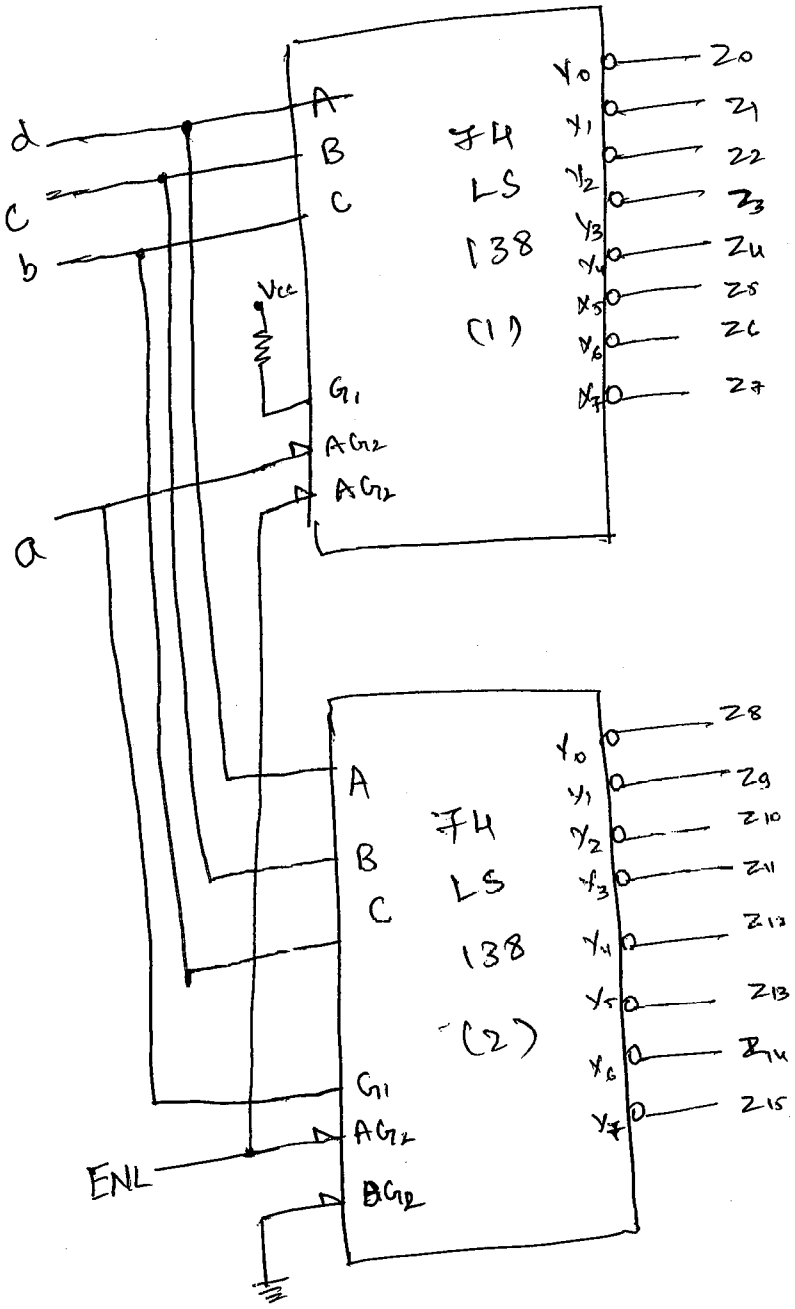
Implement the following function using 74138

$$f(a,b,c) = \Sigma(2,6,7)$$



2) ~~K map~~ Realize for $X = \Sigma(0,3,5,6,7)$

Realize 4-16 decoder using two 74XX138 decoders



The upper half of the 4 I/P table the most significant bit is 0. This enables 74138(1) & disables 74138(2)

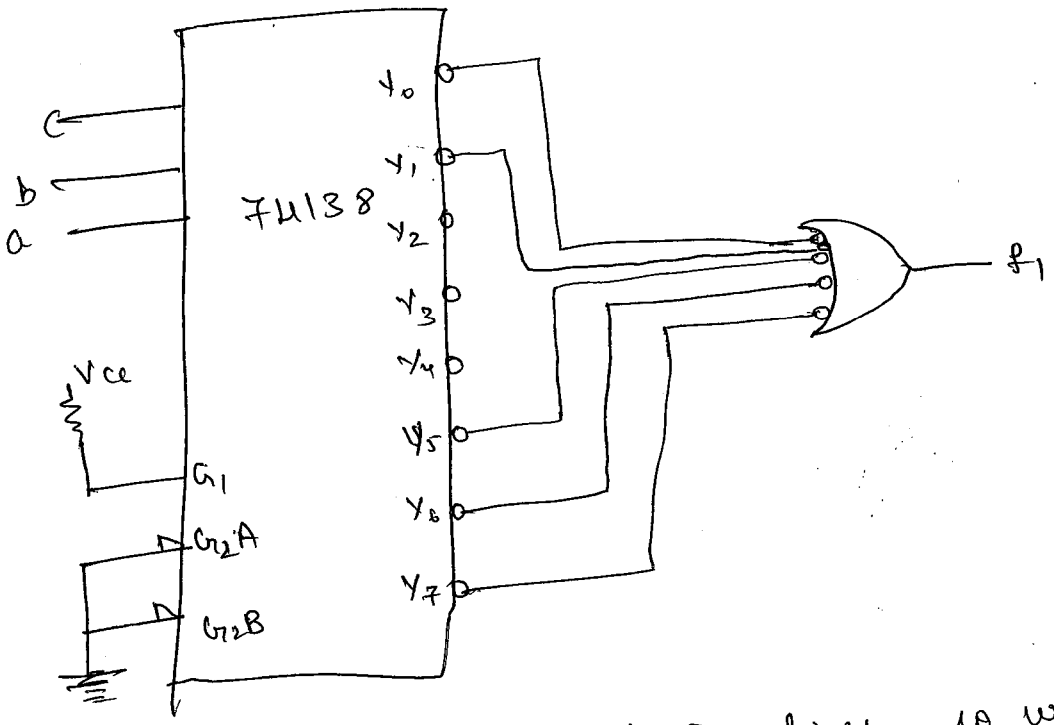
I/p's 0000 ⟶ 0111 ⟶ will be decoded in 74138(1)

I/p's 1000 ⟶ 1111 ⟶ will be decoded in 74138(2)

✱ Implement the following two pairs using 74138 decoder.

1) $f_1(a,b,c) = \Sigma(0,1,5,6,7)$

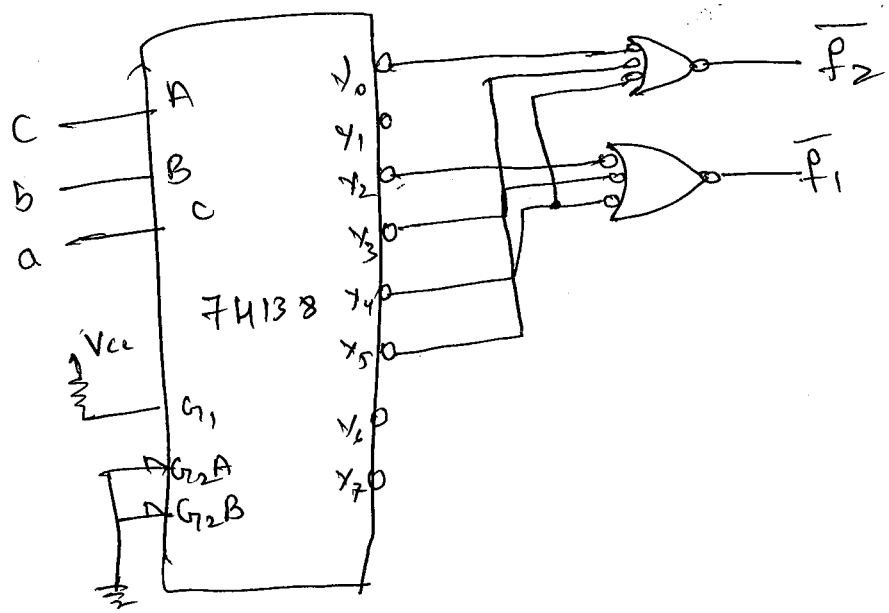
$f_2(a,b,c) = \Sigma(1,2,3,6,7)$



This will take higher no. of I/p lines so we can represent the I/p in maxterms.

$\overline{f_1}(a,b,c) = \Sigma'(2,3,4)$

$\overline{f_2}(a,b,c) = \Sigma(0,4,5)$



b) $f_1(a,b,c) = \Sigma(0,2,4)$

$f_2(a,b,c) = \Sigma(1,2,4,5,7)$

$\downarrow$

$\bar{f}_2(a,b,c) = \Sigma(0,3,6)$

c) $g(a,b,c) = \Pi(2,4,7)$

d) $f(w,x,y,z) = \Sigma(3,7,9)$

Implement the following using 2-4 decoder

a) $f_1 = \Sigma(0,4,5)$

b) $f_2 = \Pi(6,7)$

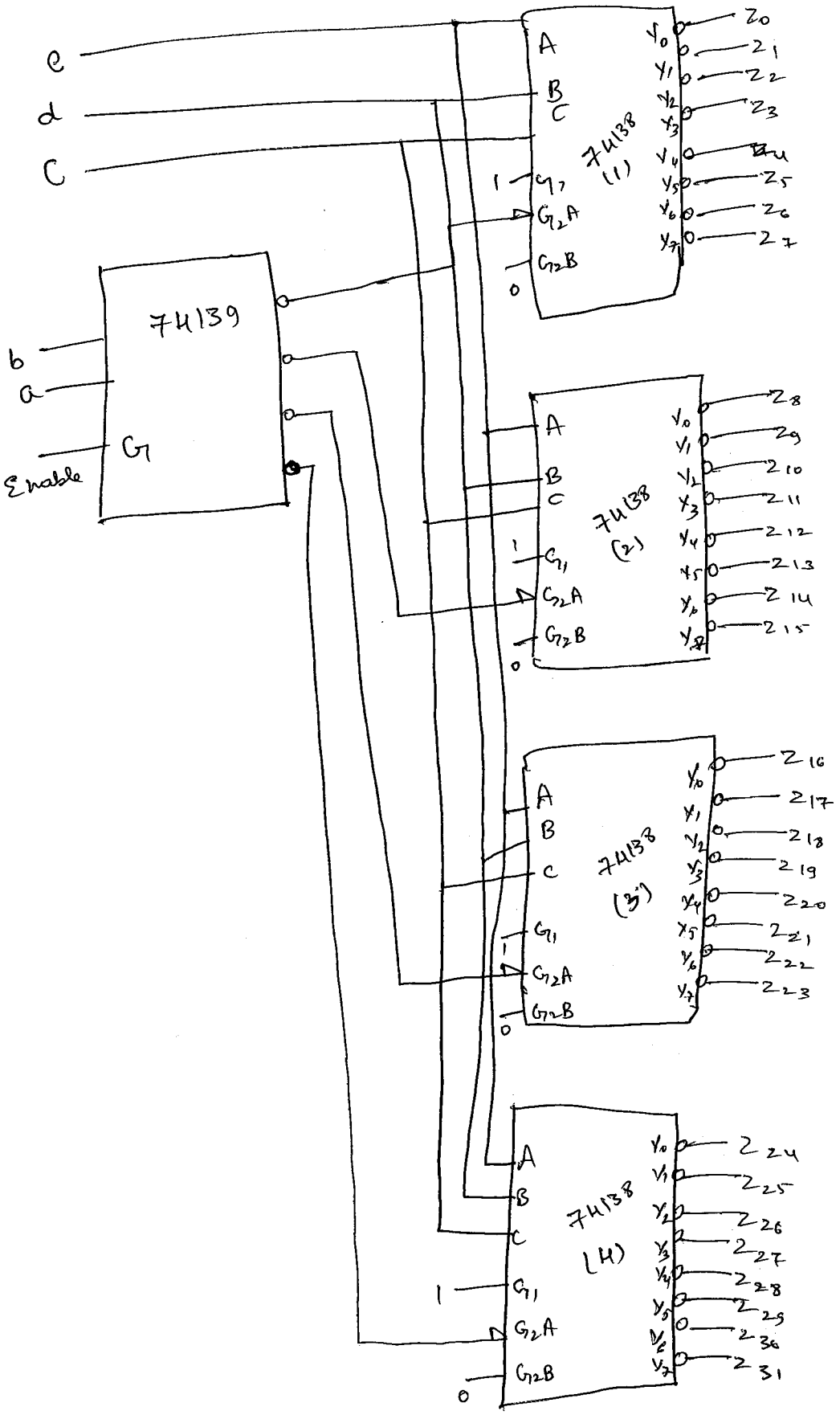
Implement the following using 3-8 decoder.

a) $f_1 = \Sigma(2,3,5,6)$

b) $f_2 = \Sigma(4,10,12)$

c) $f_3 = \Pi(5,7,13,15)$

Configure a 5 to 32 decoder using 4 → 3-8 decoder IC's & a 2 to 4 decoder IC's.



5-32 decoder from 3-8 & 2-4 decoders

Problems:

1) Implement the multiple function $f_1(a,b,c,d) = \Sigma(0,4,8,10,14,15)$ & $f_2(a,b,c,d) = \Sigma(3,7,9,13)$ using two 74138 (3-8) decoder.

2) Implement $f(x,y,z) = \Pi(1,2,4,6)$ using (3-8) decoder (74138)

3) $f(w,x,y,z) = \Sigma(1,4,8,13)$
   $g(w,x,y,z) = \Sigma(2,7,13,14)$

4) $R(w,x,y,z) = \Sigma(1,5,8,9,12,13)$

5) $R = (w,x,y,z) = w\bar{y} + \bar{y}z$

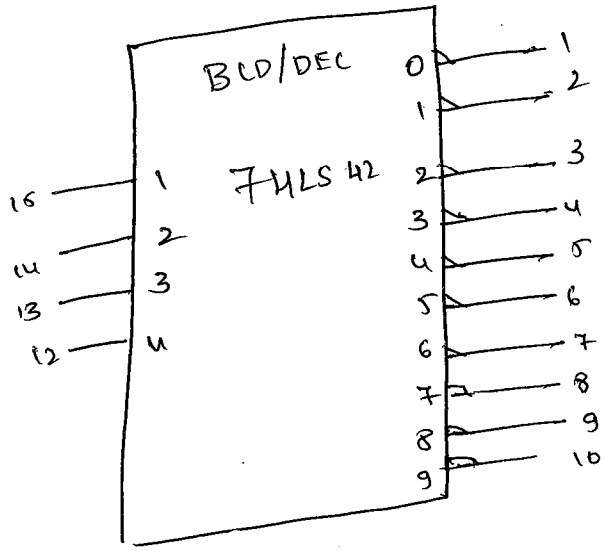
6) $A = f(x,y,z) = \Pi(0,1,3,5)$

7) $f(x,y,z) = \Sigma(0,1,2,4,5,6,7)$

BCD Decoders:

BCD decoders have four Inputs & 10 outputs. the four bit BCD input is decoded to one of 10 outputs.

SN 74X X 42 is BCD to decimal decoder MSI Integrated circuit.



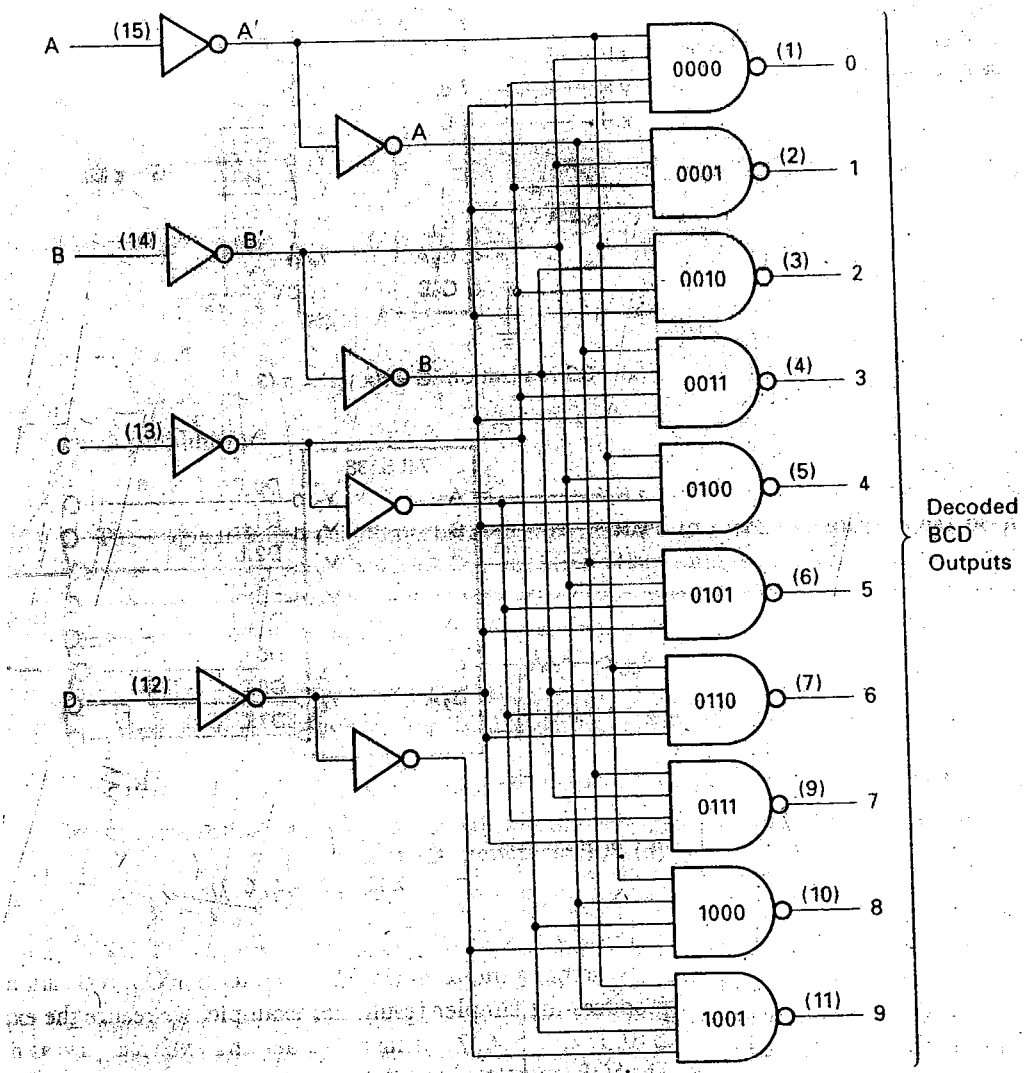
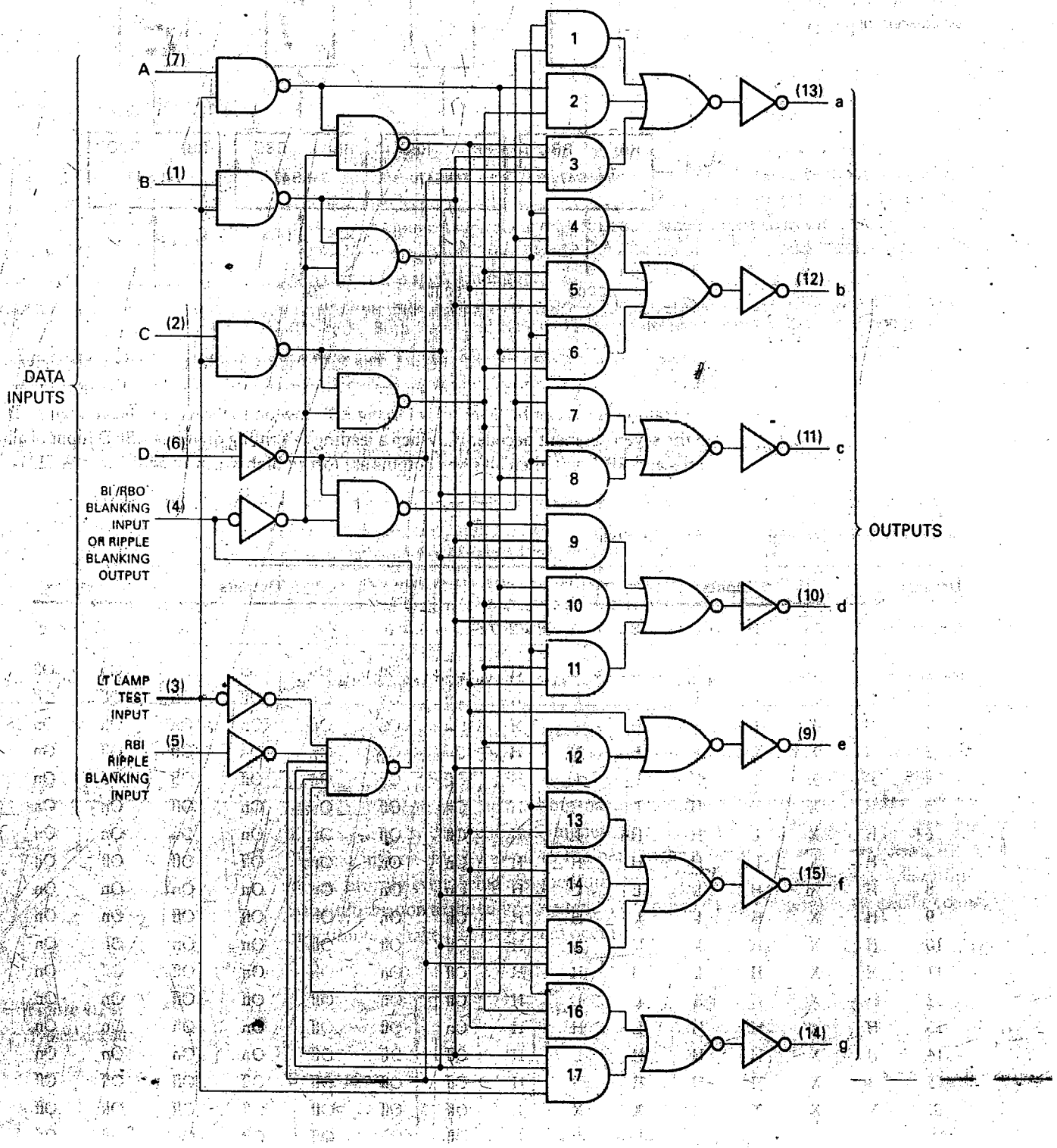
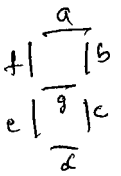


Figure 4.26
74xx47 BCD to seven-segment decoder/driver



7H××47 BCD to seven segment decoder truth table.



	Inputs						BI'/RBO'	Outputs						
Decimal	$\overline{LI}$	$\overline{RBI}$	D	C	B	A		a	b	c	d	e	f	g
0	H	H	L	L	L	L	H	on	on	on	on	on	on	off
1	H	X	L	L	L	H	H	off	on	on	off	off	off	off
2	H	X	L	L	H	L	H	on	on	off	on	on	off	on
3	H	X	L	L	H	H	H	on	on	on	on	off	off	on
4	H	X	L	H	L	L	H	off	on	on	off	off	on	on
5	H	X	L	H	L	H	H	on	off	on	on	off	on	on
6	H	X	L	H	H	L	H	off	off	on	on	on	on	on
7	H	X	L	H	H	H	H	on	on	on	off	off	off	off
8	H	X	H	L	L	L	H	on	on	on	on	on	on	on
9	H	X	H	L	L	H	H	on	on	on	off	off	on	on
10	H	X	H	L	H	L	H	off	off	off	on	on	off	on
11	H	X	H	L	H	H	H	off	off	on	on	off	off	on
12	H	X	H	H	L	L	H	off	on	off	off	off	on	on
13	H	X	H	H	L	H	H		off	off	on		on	on
14	H	X	H	H	H	L	H	off	off	off	on	on	on	on
15	H	X	H	H	H	H	H	off	off	off	off	off	off	off
BI	X	X	X	X	X	X	L	off	off	off	off	off	off	off
RBI	H	L	L	L	L	L	L	off	off	off	off	off	off	off
LT	L	X	X	X	X	X	H	on	on	on	on	on	on	on