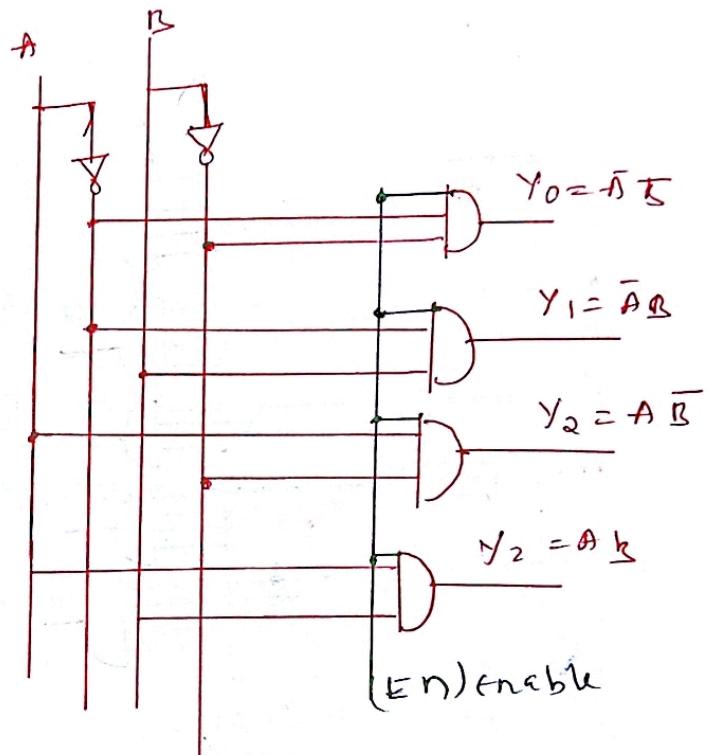
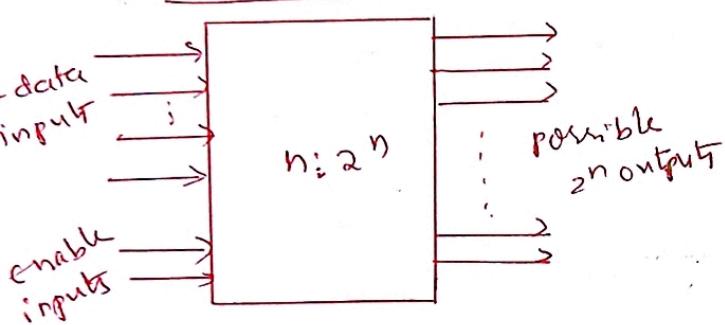


1266

→ Decoders

→ General structure of decoder

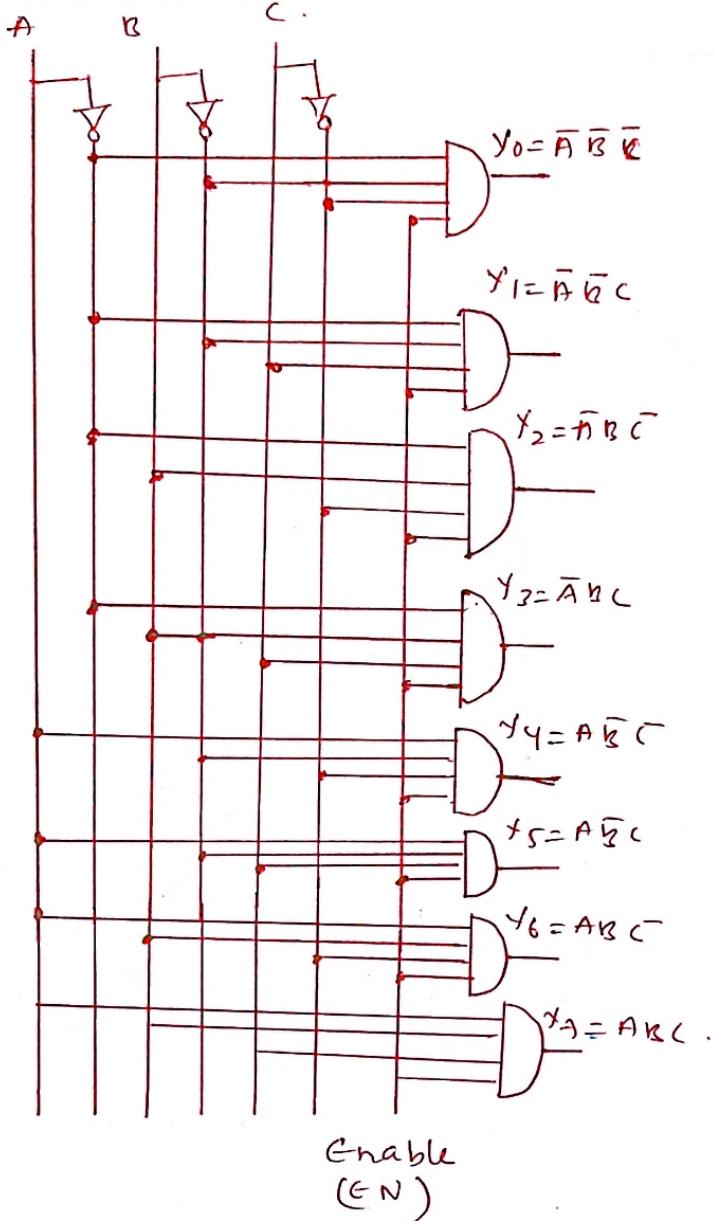


Inputs			Outputs			
E_n	A	B	Y_3	Y_2	Y_1	Y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

→ A decoder is a combinational circuit that connects binary information from n input lines to a maximum of 2^n unique output lines.

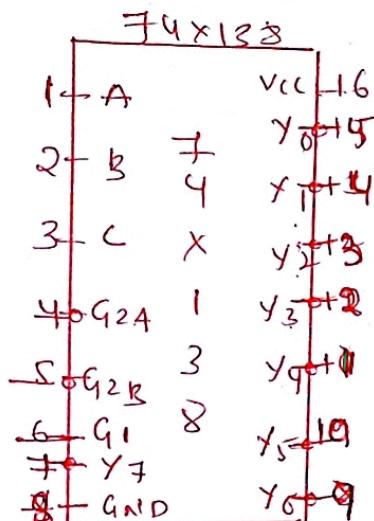
→ Application - binary \rightarrow octal conversion
 (2) Generating addresses for peripheral devices for the communication

→ 3x8 decoder



Inputs			Outputs								
EN	A	B	C	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

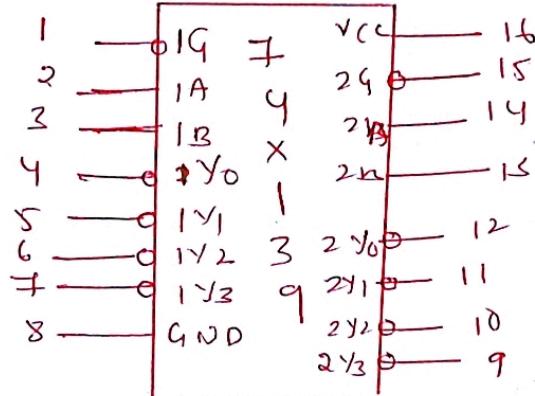
→ 74x138 3 to 8 decoder



Input				Outputs									
G _{2A}	G _{2B}	Y ₁	C	B	A	Y ₇	Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁	Y ₀
1	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	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	1	1	1	1	1	1	1	0
0	0	1	0	0	1	1	1	1	1	1	1	0	1
0	0	1	0	1	0	1	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	0	1	1	1
0	0	1	1	0	0	1	1	0	1	1	1	1	1
0	0	1	1	0	1	1	0	1	1	1	1	1	1
0	0	1	1	1	0	1	1	1	1	1	1	1	1
0	0	1	1	1	1	0	1	1	1	1	1	1	1

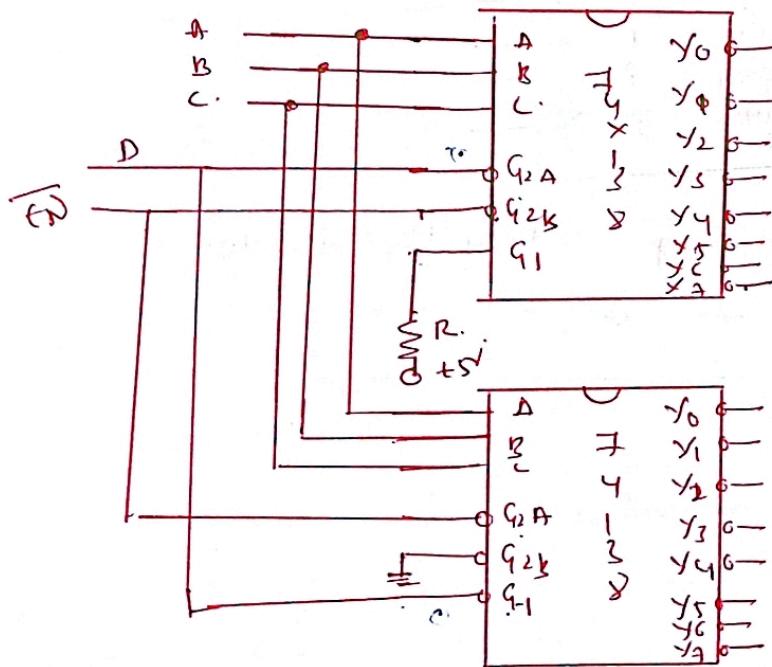
→ 74x139 dual

2 to 4 decoder



Inputs			Outputs			
Y ₃	Y ₂	Y ₁	Y ₃	Y ₂	Y ₁	Y ₀
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

→ Cascading binary decoders :- 4x16 using two 3x8 decoders.



D C B A

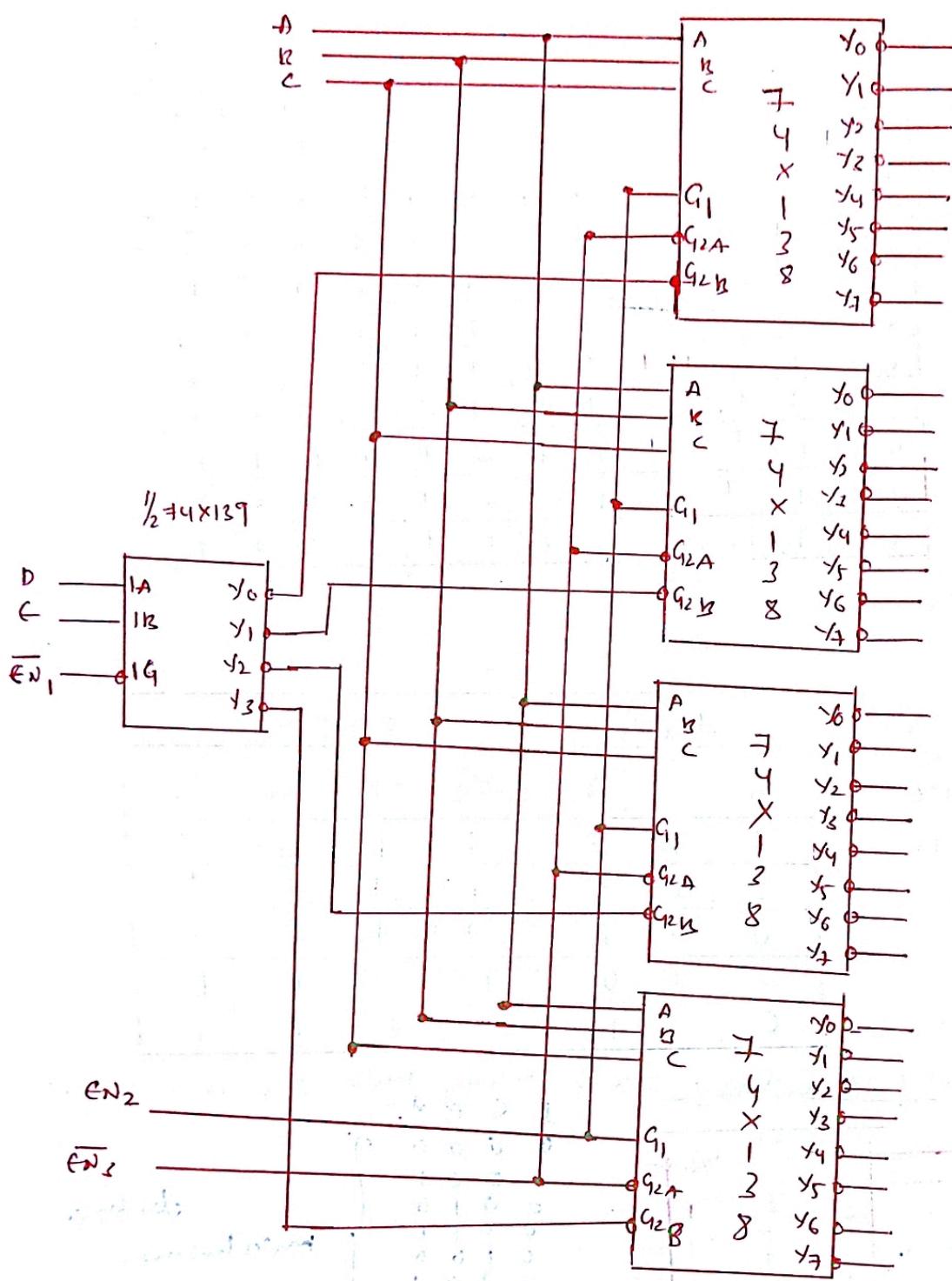
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1

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

for b=0
minterms

for b=1
minterms

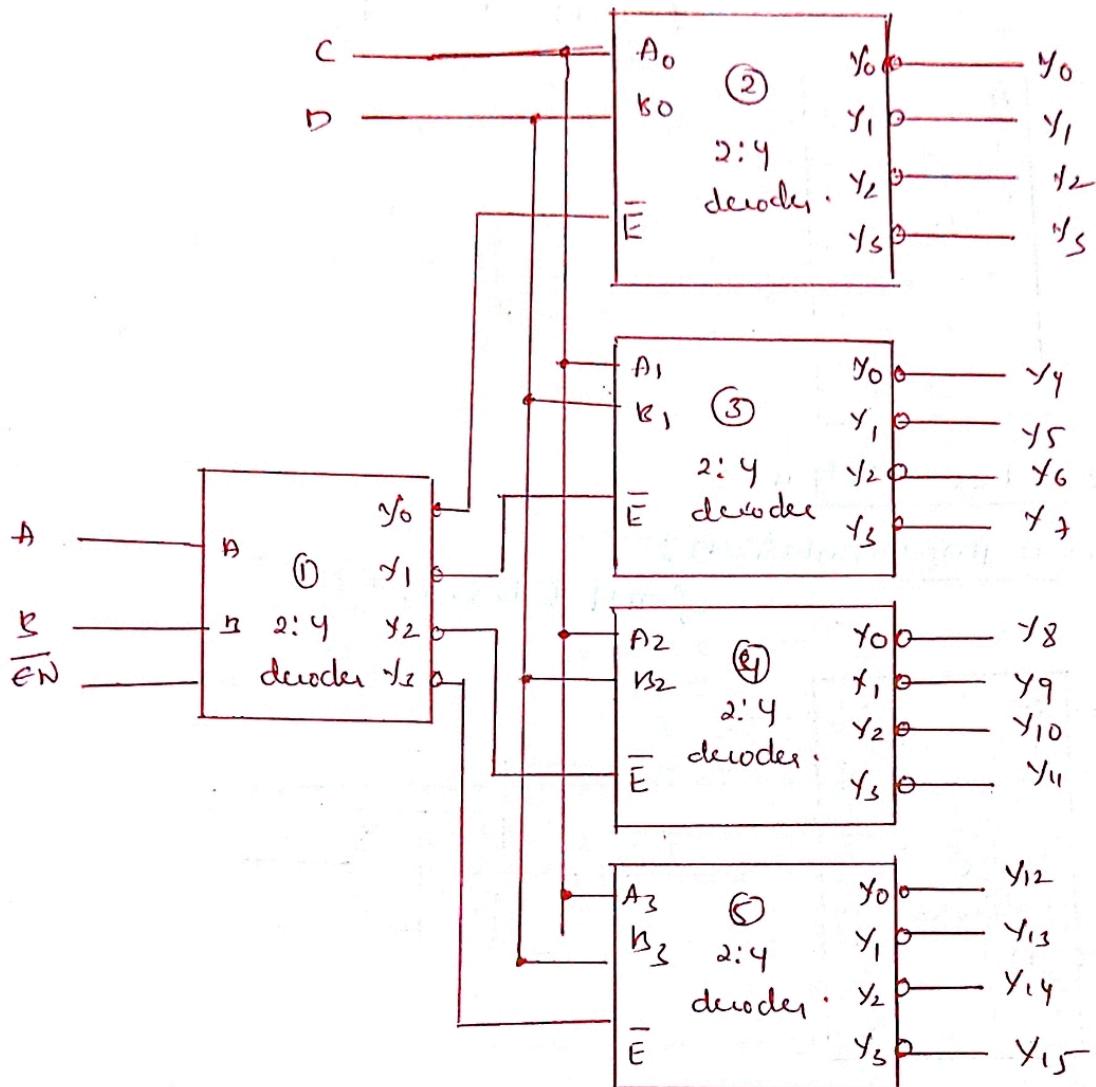
→ Design 5 to 32 decoder using one 2 to 4 decoder and four 3 to 8 decoder ICs.



→ Design 4 to 16 decoder using 2 to 4 line decoder.

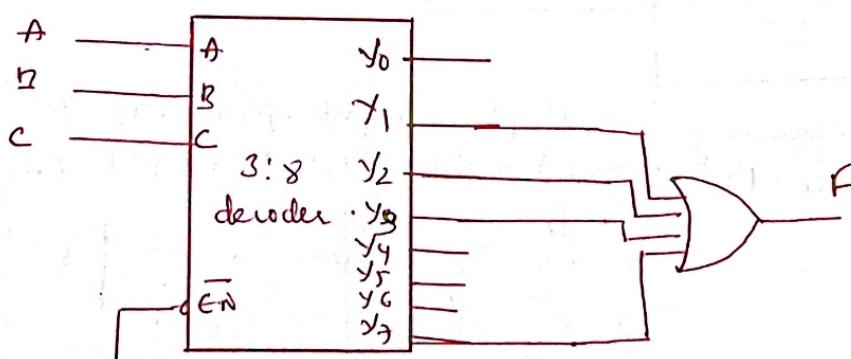
0	0	3	1
1	0	0	1
0	1	0	1
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	1

(4)



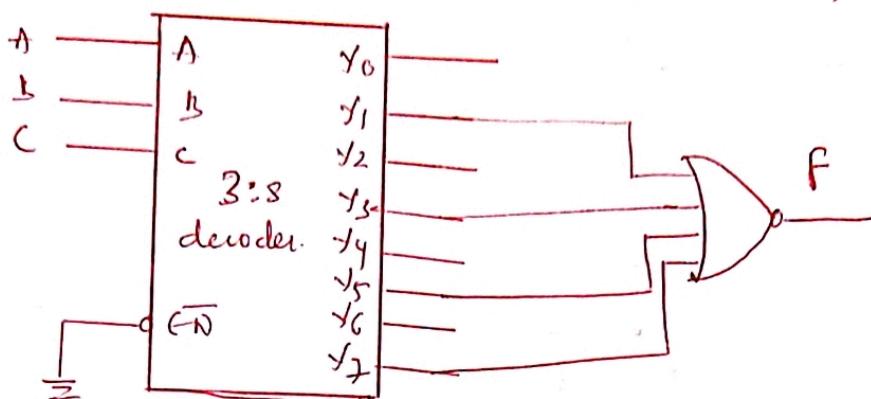
→ Realization of multiple output function using binary decoder :-

→ For active high output:-
SOP function implementation :- $F = \Sigma m(1, 2, 3, 7)$



→ POS function implementation:

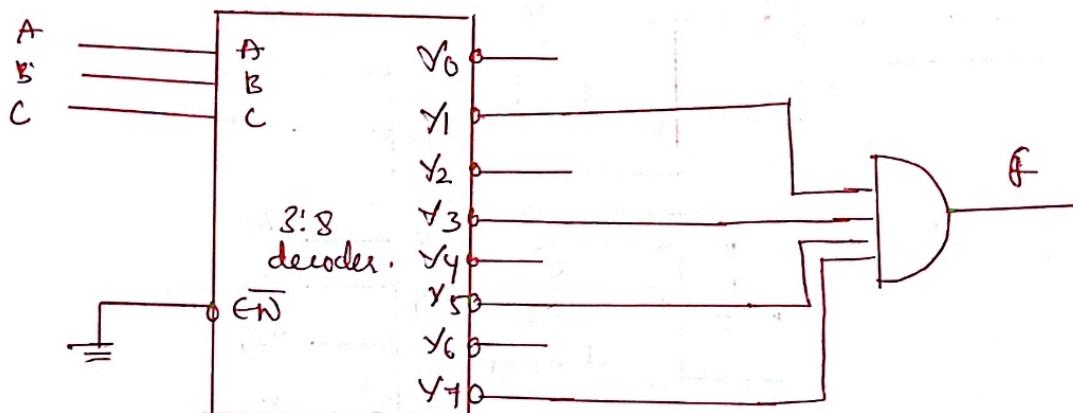
$$f = \prod M(1, 3, 5, 7)$$



→ For active Low output:

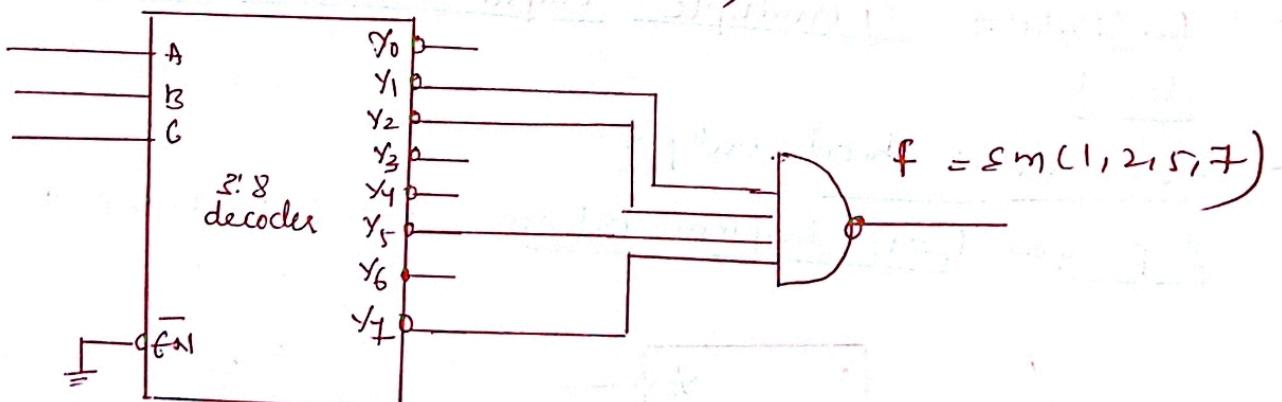
→ POS function implementation:

$$f = \prod M(1, 3, 5, 7)$$

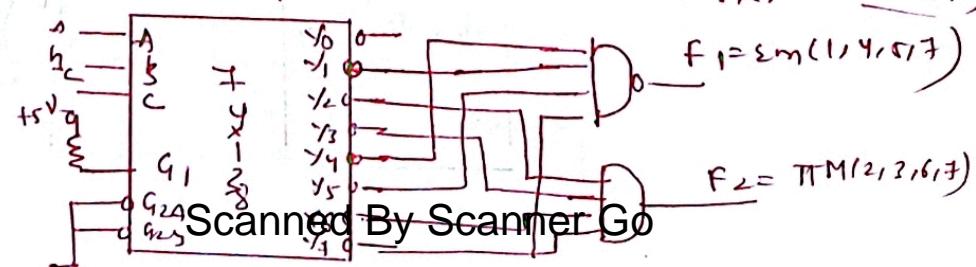


→ SOP function implementation:

$$f = \sum m(1, 2, 5, 7)$$



→ Implement following multiple output function using 74LS138 and external gates. $F_1(A, B, C) = \sum m(1, 4, 5, 7) \quad \sum m(2, 3, 6, 7)$

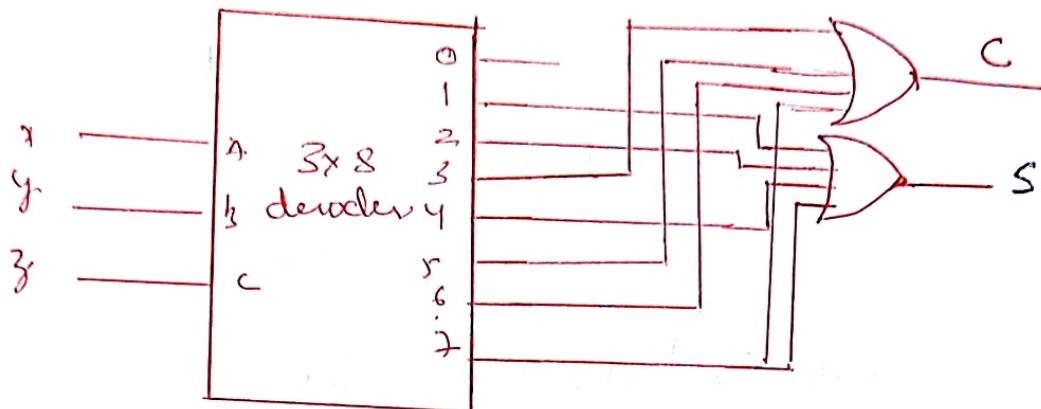


(5)

→ Implement a full-adder circuit with a decoder and two OR gates.

$$S(x, y, z) = \Sigma(1, 2, 4, 7)$$

$$C(x, y, z) = \Sigma(3, 5, 6, 7)$$

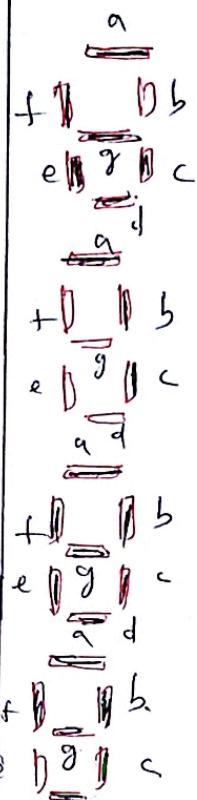


→ BCD to seven segment decoder: (LED | LCD decoders / drivers)

Digit	Segments activated	Display
0	a, b, c, d, e, f	
1	b, c	
2	a, b, d, e, g	
3	a, b, c, d, g	
4	b, c, -f, g	
5	a, c, d, -f, g	

6

a, c, d, e, f, g

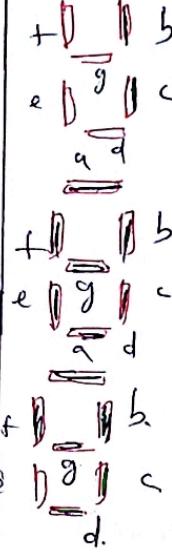


7

a, b, c

8

a, b, c, d, e, f, g

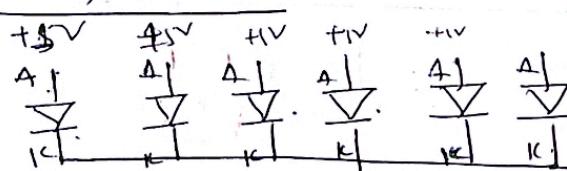


9

a, b, c, d, f, g.

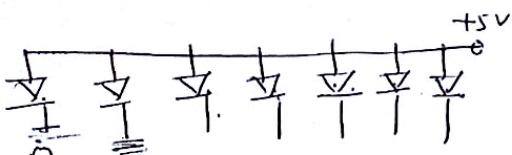


→ common cathode:



common-cathode

common anode:



common-anode

Digit	A B C D	a b c d e f g	Digit	A B C D	a b c d e f g
0	0 0 0 0	1 1 1 1 1 1 0	0	0 0 0 0	0 0 0 0 0 0 0 1
1	0 0 0 1	0 1 1 0 0 0 0	1	0 0 0 1	1 0 0 1 1 1 1 1
2	0 0 1 0	1 1 0 1 1 0 1	2	0 0 1 0	0 0 1 0 0 1 0
3	0 0 1 1	1 1 1 1 0 0 1	3	0 0 1 1	0 0 0 0 1 1 0
4	0 1 0 0	0 1 1 0 0 1 1	4	0 1 0 0	1 0 0 1 1 0 0
5	0 1 0 1	1 0 1 1 0 1 1	5	0 1 0 1	0 1 0 0 1 0 0
6	0 1 1 0	1 0 1 1 1 1 1	6	0 1 1 0	0 1 0 0 0 0 0
7	0 1 1 1	1 1 1 0 0 0 0	7	0 1 1 1	0 0 0 1 1 1
8	1 0 0 0	1 1 1 1 1 1 1	8	1 0 0 0	0 0 0 0 0 0 0
9	1 0 0 1	1 1 1 1 0 1 1	9	1 0 0 1	0 0 0 0 1 0 0

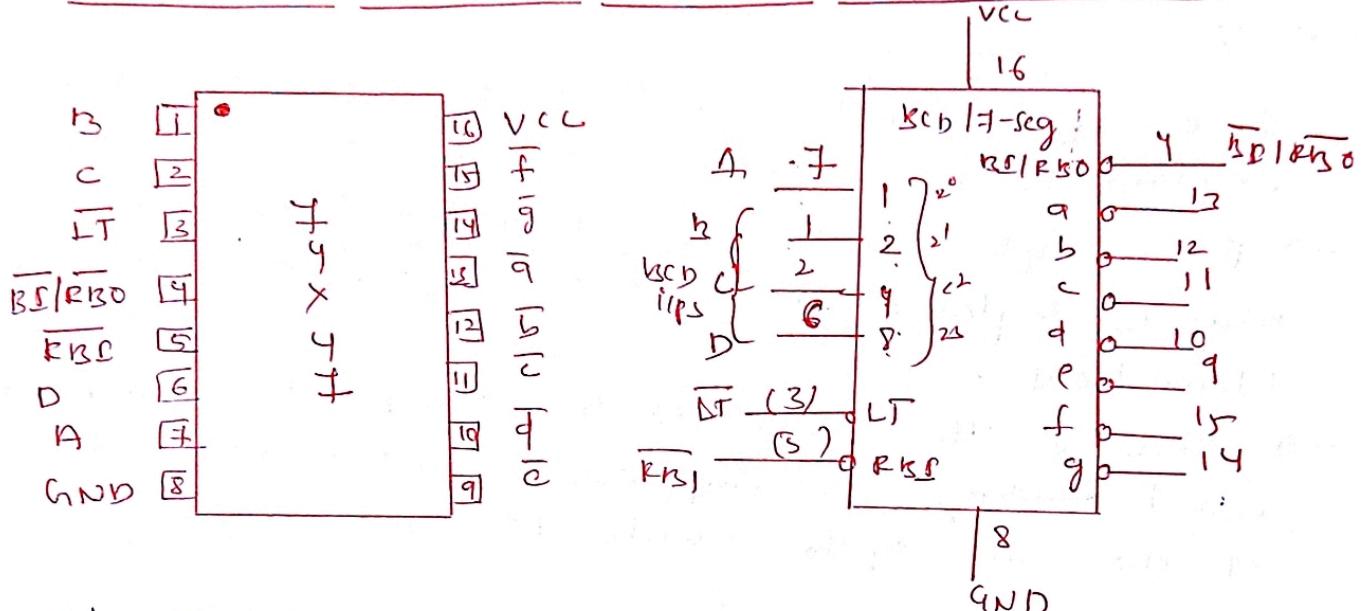
→ K-map simplification :- (common cathode)

$$a = A + C + BD + \bar{B}\bar{D} ; b = \bar{B} + \bar{C}\bar{D} + CD ; c = \bar{B} + \bar{C} + b$$

$$d = \bar{B}\bar{D} + CD + \bar{B}\bar{C}D + ; e = \bar{B}\bar{D} + CD ; f = A + \bar{C}\bar{D} + \bar{B}\bar{C} + \bar{B}\bar{D}$$

$$g = A + \bar{B}\bar{C} + \bar{B}c + CD$$

→ 74LS47 BCD-to-7-segment decoder/driver



→ The 74LS47 is an example of an IC device that decodes a BCD input and drives a 7-segment display.

→ In addition to its decoding and segment drive capability, the 74LS47 has several additional features as indicated by the \bar{F} , \bar{RBS} , \bar{BS}/\bar{RBO} functions in the logic symbol of fig.(a).

→ As indicated by the bubbles on the logic symbol, all of the outputs (a through g) are active-low as are the LT (lamp test), \bar{RBS} (ripple blanking input), and \bar{BS}/\bar{RBO} (blanking input/ripple blanking output functions).

→ The outputs can drive a common-anode 7-segment display directly.

→ In addition to decoding a BCD input and producing the appropriate 7-segment outputs, the 74LS47 has lamp test and zero suppression capability.

→ Lamp test:-

When a Low is applied to the \overline{ET} input and the $\overline{BS}|\overline{RBO}$ is HIGH, all of the seven segments in the display are turned on.

→ Lamp test is used to verify that no segments are burned out.

→ Zero suppression:-

Zero suppression is a feature used for multidigit displays to blank out unnecessary zeros.

→ For example, in a 6-digit display the number 6.4 may be displayed as 006.400 if the zeros are not blanked out.

→ Blanking the zeros at the front of a number is called leading zero suppression and blanking the zeros at the back of the number is called trailing zero suppression.

→ Only non-essential zeros are blanked.

→ With zero suppression, the number 030.080 will be displayed as 30.08 (the essential zeros remain).

→ Zero suppression in the 74LS47 is accomplished using the \overline{BS} and $\overline{BS}|\overline{RBO}$ functions.

→ \overline{BS} is the ripple blanking input and \overline{RBO} is the ripple blanking output on the 74LS47; these are used for zero suppression.

→ \overline{BS} is the ~~ripple~~ blanking input that shares the same pin with \overline{RBO} ; in other words, the $\overline{BS}|\overline{RBO}$ pin can be used as an input or an output.

→ When used as a \overline{BS} (blanking input), all segment outputs are HIGH (non-active) when \overline{BS} is Low, which overrides all other inputs.

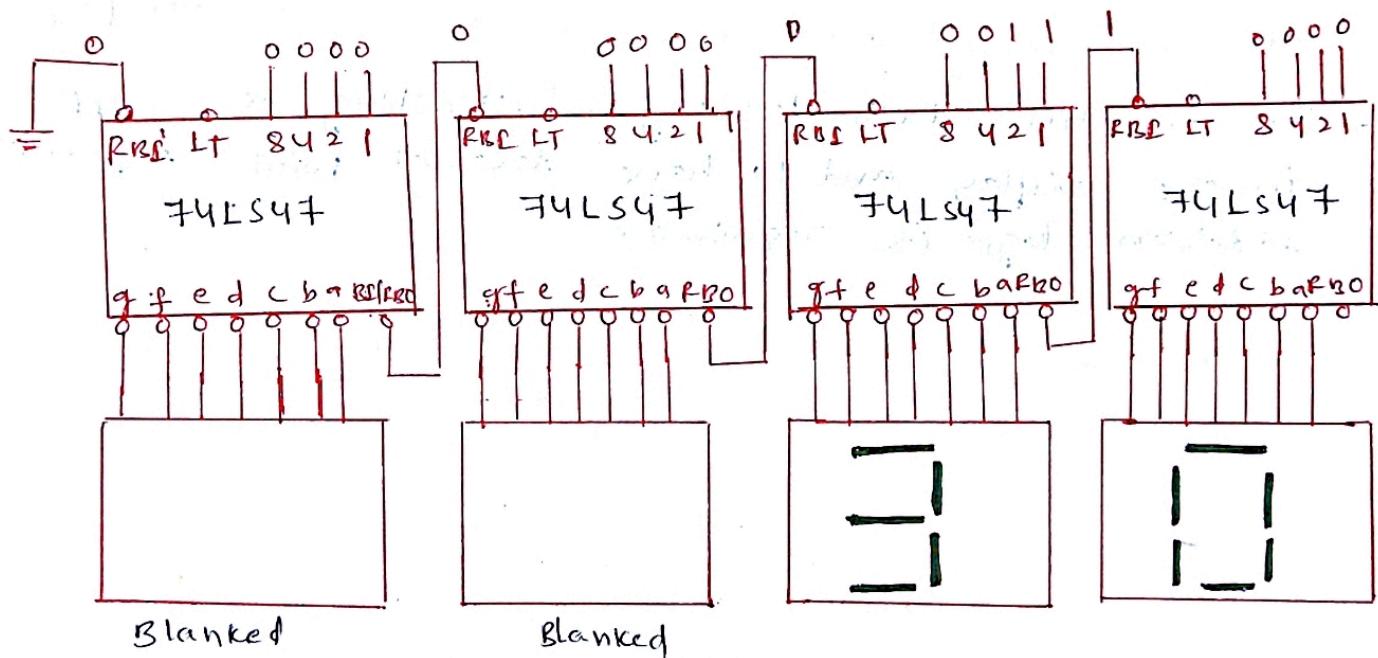
→ All of the segment outputs of decoder are non-active (high) if a zero code (0000) is on its inputs.

→ This causes the display to be blank and produces

\overline{RBO} .

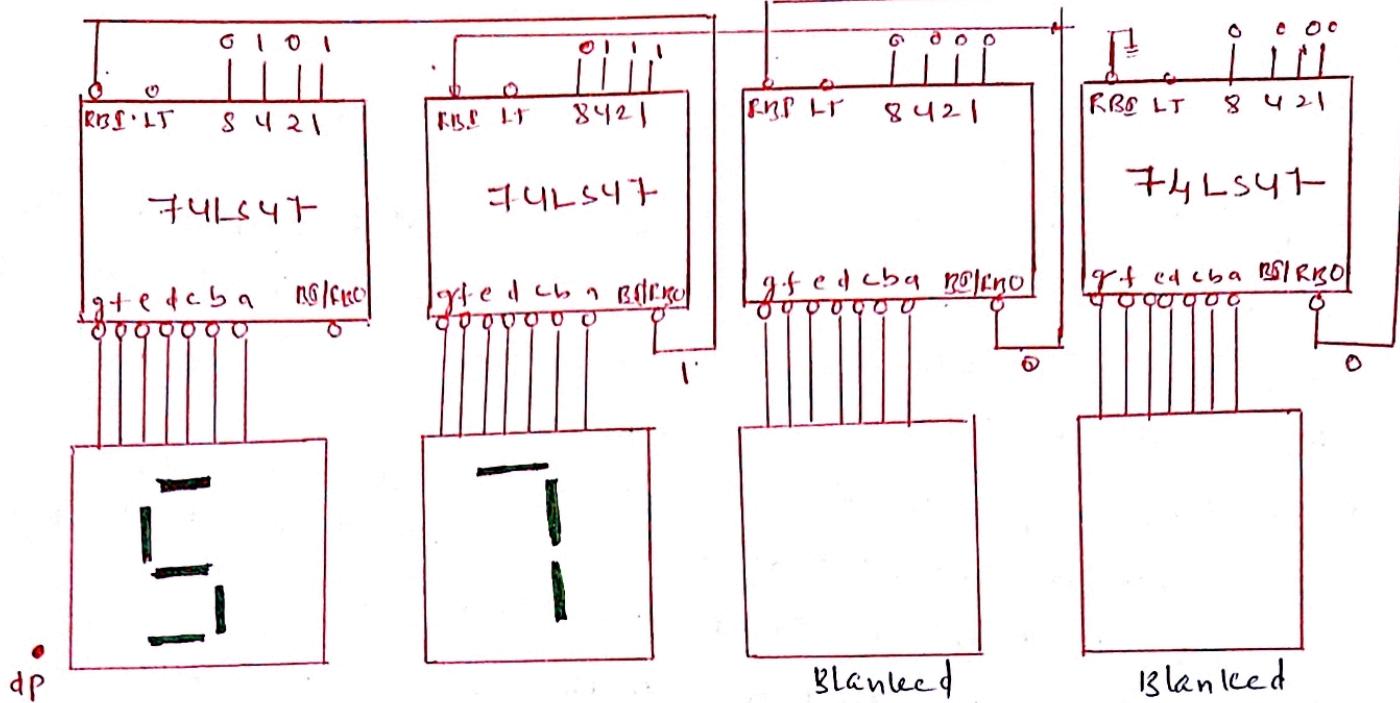
(7)

- The logic diagram illustrates leading zero suppression for a whole number.
- The highest-order digit position (leftmost) is always blanked if a zero code is on its BCD inputs because the \overline{RBO} of the most-significant decoder is made low by connecting it to ground.
- The \overline{RBO} of each decoder is connected to the \overline{RBO} of the next lowest-order decoder so that all zeros to the left of the first non-zero digit are blanked.
- For example, in part(a) of the figure the two highest-order digits are zeros and therefore are blanked.
- The remaining two digits, 3 and 0 are displayed.
-



(a) Illustration of leading zero suppression.

- The logic diagram in fig(b) illustrates tracking zero-suppression for a fractional number.
- The lowest-order digit (right-most) is always blanked if a zero code is on its BCD inputs because the \overline{RBO} is connected to ground.
- The \overline{RBO} of each decoder is connected to the \overline{RBO} of the next highest-order decoder so that all zeros to the right of the first non-zero digit are blanked.



(b) Illustration of trailing zero suppression

- In part (b) of figure , the two lowest - order digits are zeros and therefore are blanked .
- The remaining two digits , 5 and 7 are displayed .
- To combine both leading and trailing zero suppression in one display and to have decimal point capability, addition logic is required .

