Digital Design : 2021-22
Lecture 15 : Decoders an Encoders

By Dr. Sanjay Vidhyadharan

Decoder

A Decoder is a combinational circuit that converts binary information from ‘n’ input lines to 2^n output lines

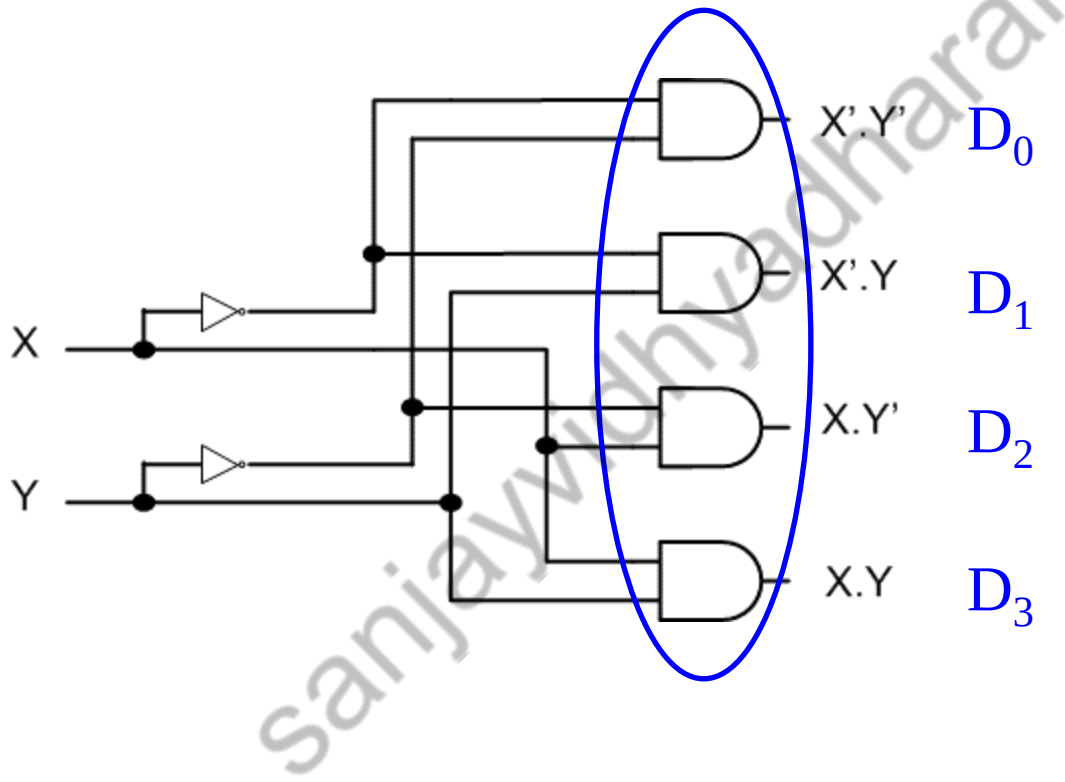
An Example: 2 to 4 Line Decoder

X	Y
0	0
0	1
1	0
1	1

D_0	D_1	D_2	D_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Decoder

An Example: 2 to 4 Line Decoder



Minterms
(0, 1, 2, 3)

Minterm Generating circuit

Decoder

An Example: 2 to 4 Line Decoder with Enable

E	X	Y
1	0	0
1	0	1
1	1	0
1	1	1
0	X	X

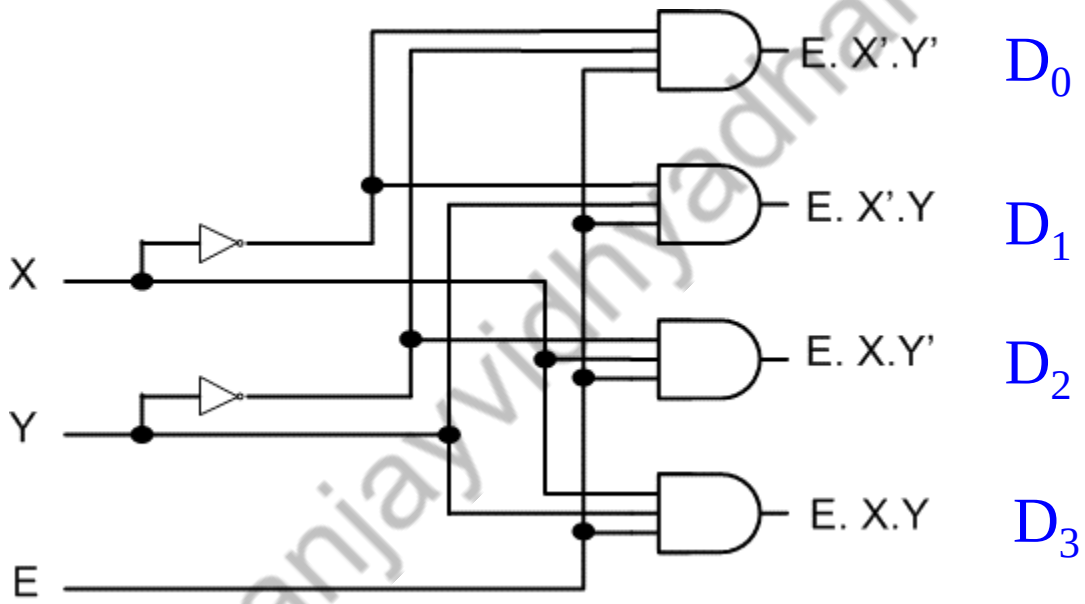
D_0	D_1	D_2	D_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
0	0	0	0

Decoder with enable pin is also called as Demultiplexer

X and Y – Select Lines E- Input line

Decoder

An Example: 2 to 4 Line Decoder with Enable



Decoder

An Example: 3 to 8 Line Decoder

X	Y	Z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1

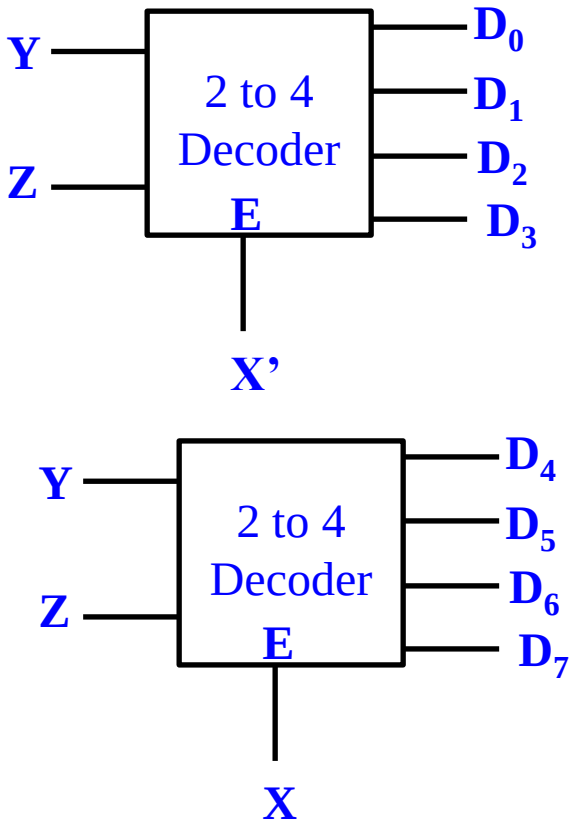
Direct Implementation – Home Assignment

Decoder

Implementing a 3 to 8 decoder using two 2 to 4 decoders with enable pin

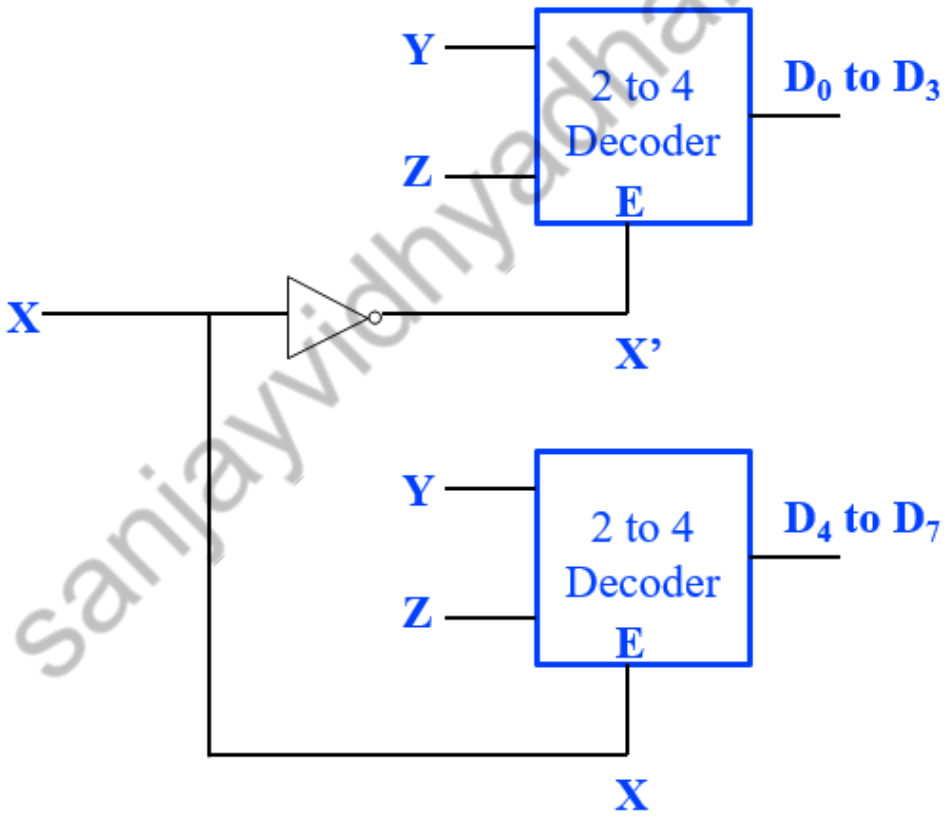
X	Y	Z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1



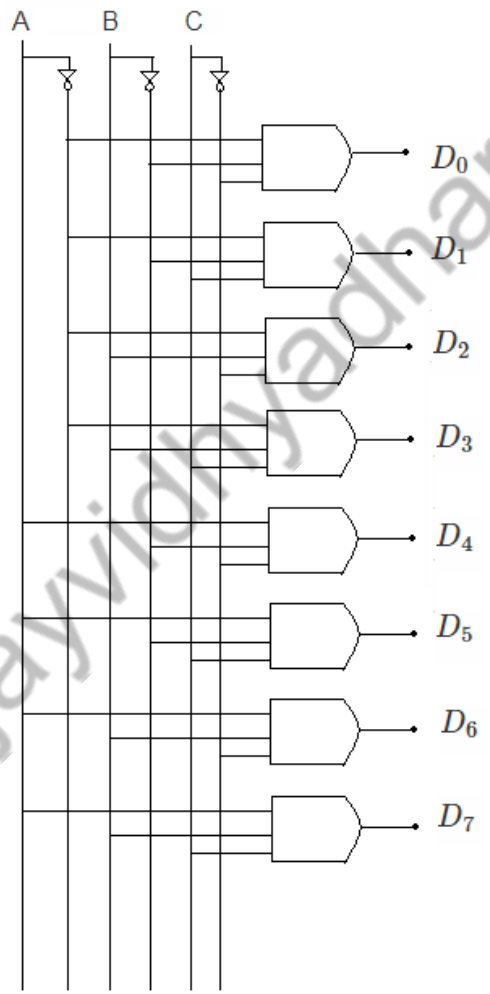
Decoder

Implementing a 3 to 8 decoder using two 2 to 4 decoders with enable pin



Decoder

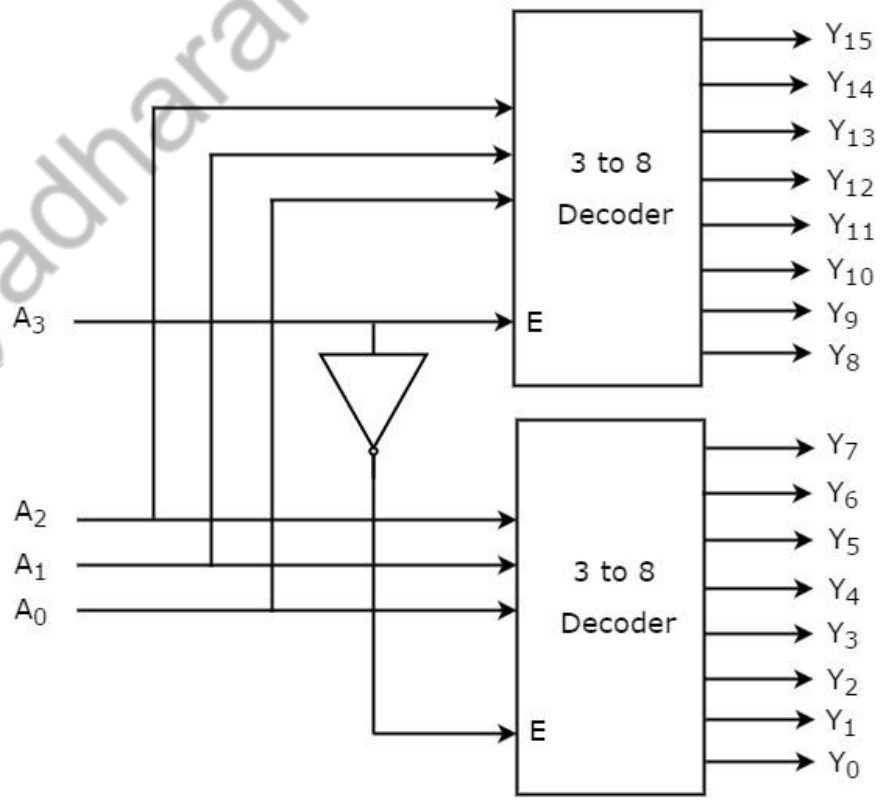
3 to 8 decoder



Decoder

4 to 16 decoder

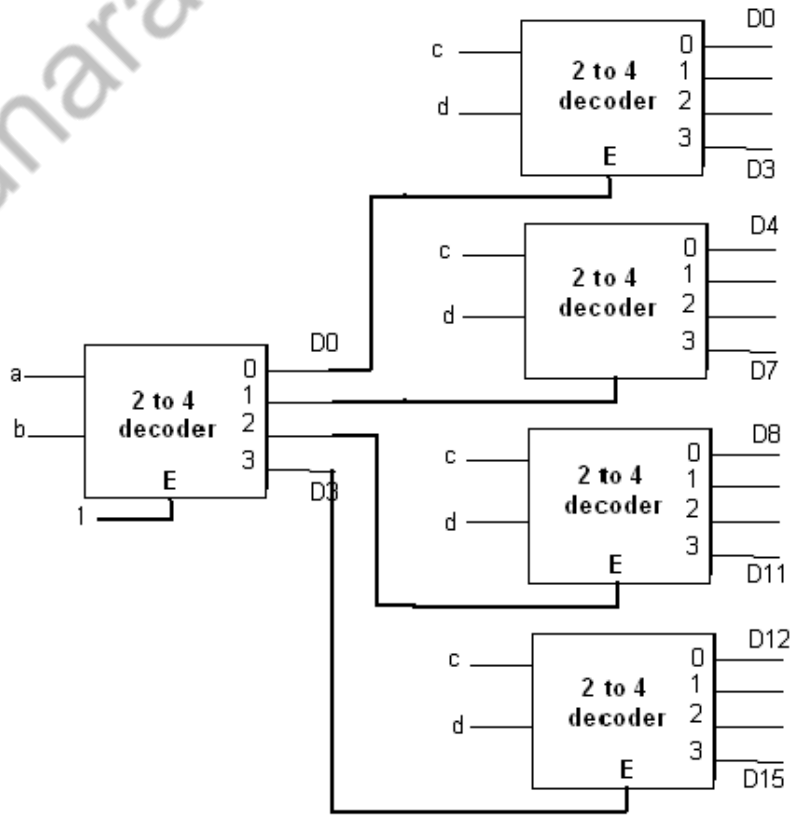
INPUTS				OUTPUTS															
A_3	A_2	A_1	A_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	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	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Decoder

4 to 16 decoder

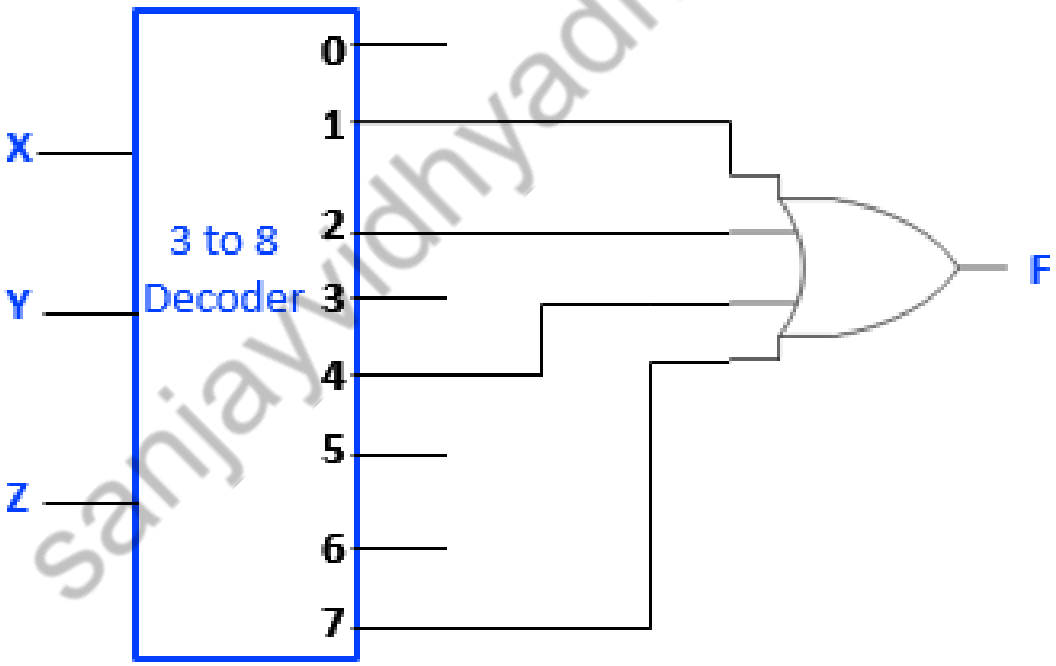
INPUTS				OUTPUTS															
A_3	A_2	A_1	A_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	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	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Decoder

Implementing function using Decoders

$F = \sum (1, 2, 4, 7) = X'Y'Z + X'YZ' + XY'Z' + XYZ$



Encoder

Encoder - combinational circuit that performs inverse operation of decoder

Has 2^n or fewer input lines and n output lines

An Example:

$X = D_2 + D_3$

$Y = D_1 + D_3$

D_0	D_1	D_2	D_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

X	Y
0	0
0	1
1	0
1	1

Encoder

$X = D_2 + D_3$

$Y = D_1 + D_3$

D_0	D_1	D_2	D_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

X	Y
0	0
0	1
1	0
1	1

Limitations:

If two ones appear i.e. if D_1 and D_2 are '1' then both X and Y will be '1' resulting in output 11

If all the inputs are zeros then output is 00 – which indicates that D_0 is 1

Priority Encoder

Encoder circuits must establish input priority

If high priority is given to inputs with higher subscripts
If both D_1 and D_2 are '1' then resulting o/p should be 10

If all the inputs are zeros then output is 00 – which indicates that D_0 is 1

This ambiguity can be resolved by providing one more output.

Extra Output indicates whether at least one input is equal to 1

Priority Encoder

D_0	D_1	D_2	D_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

X	Y
0	0
0	1
1	0
1	1

Encoder

$X = D_2 + D_3$

$Y = D_1 + D_3$

D_0	D_1	D_2	D_3
1	0	0	0
X	1	0	0
X	X	1	0
X	X	X	1
0	0	0	0

X	Y	V
0	0	1
0	1	1
1	0	1
1	1	1
X	X	0

Priority Encoder

$X = D_2 + D_3$

$Y = D_1 D_2' + D_3$

$V = D_0 + D_1 + D_2 + D_3$

Priority Encoder

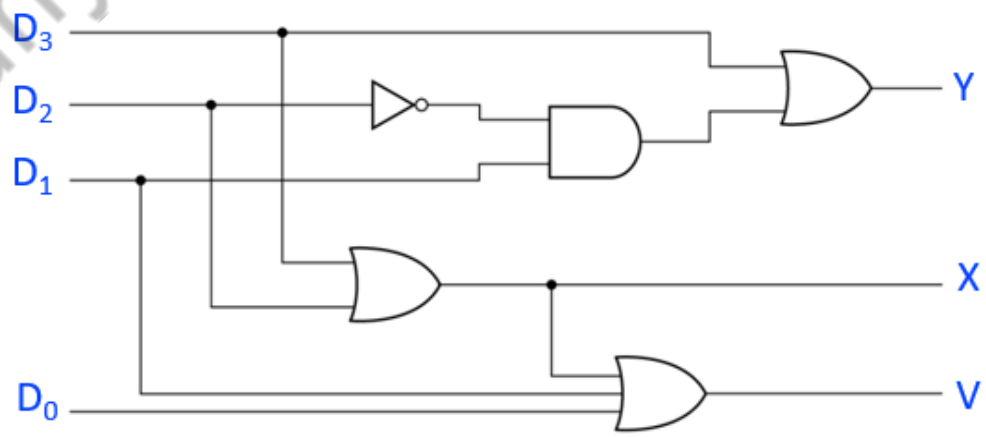
Priority Encoder

$X = D_2 + D_3$

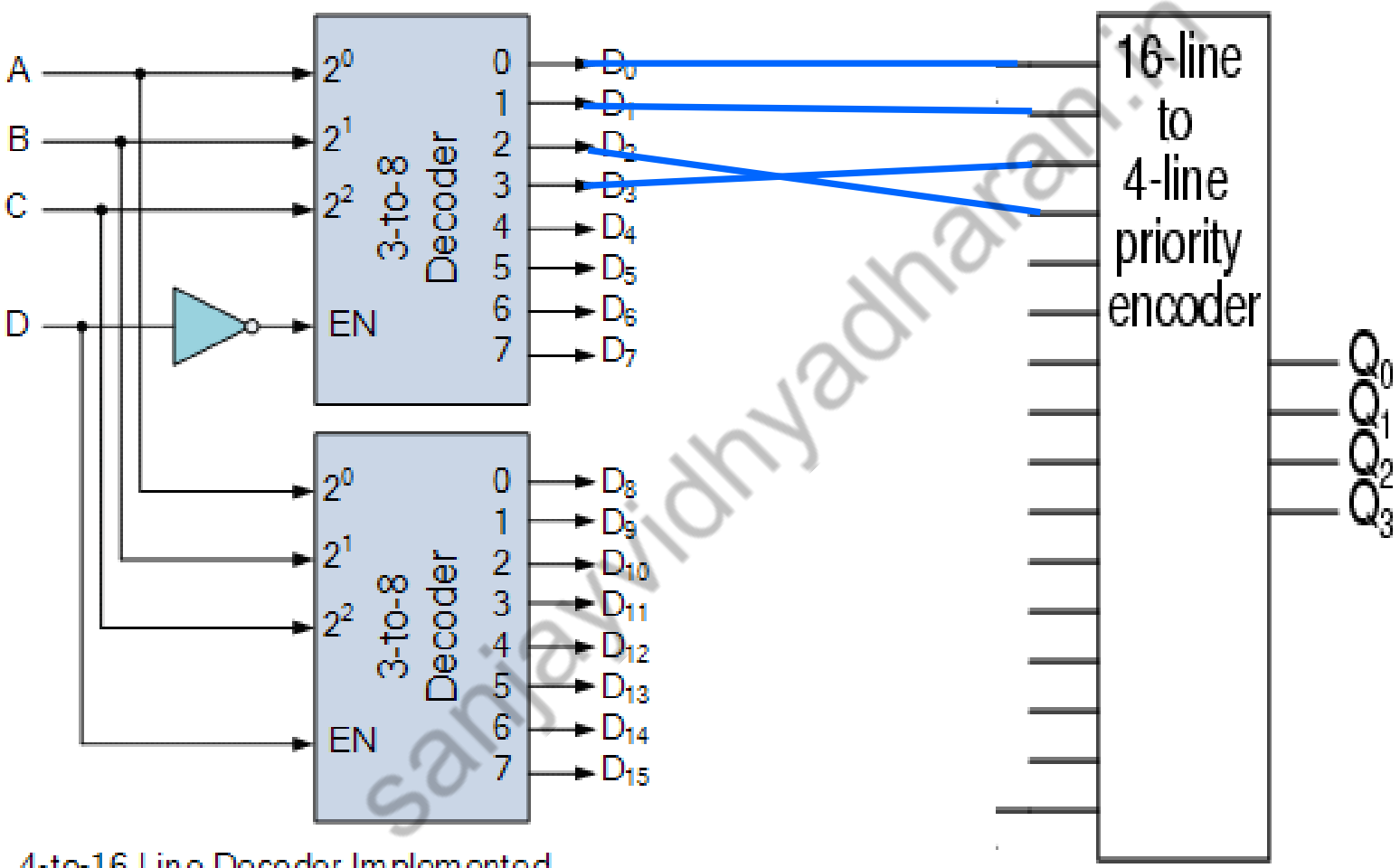
$Y = D_1D_2' + D_3$

$V = D_0 + D_1 + D_2 + D_3$

D_0	D_1	D_2	D_3	X	Y	V
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1
0	0	0	0	X	X	0

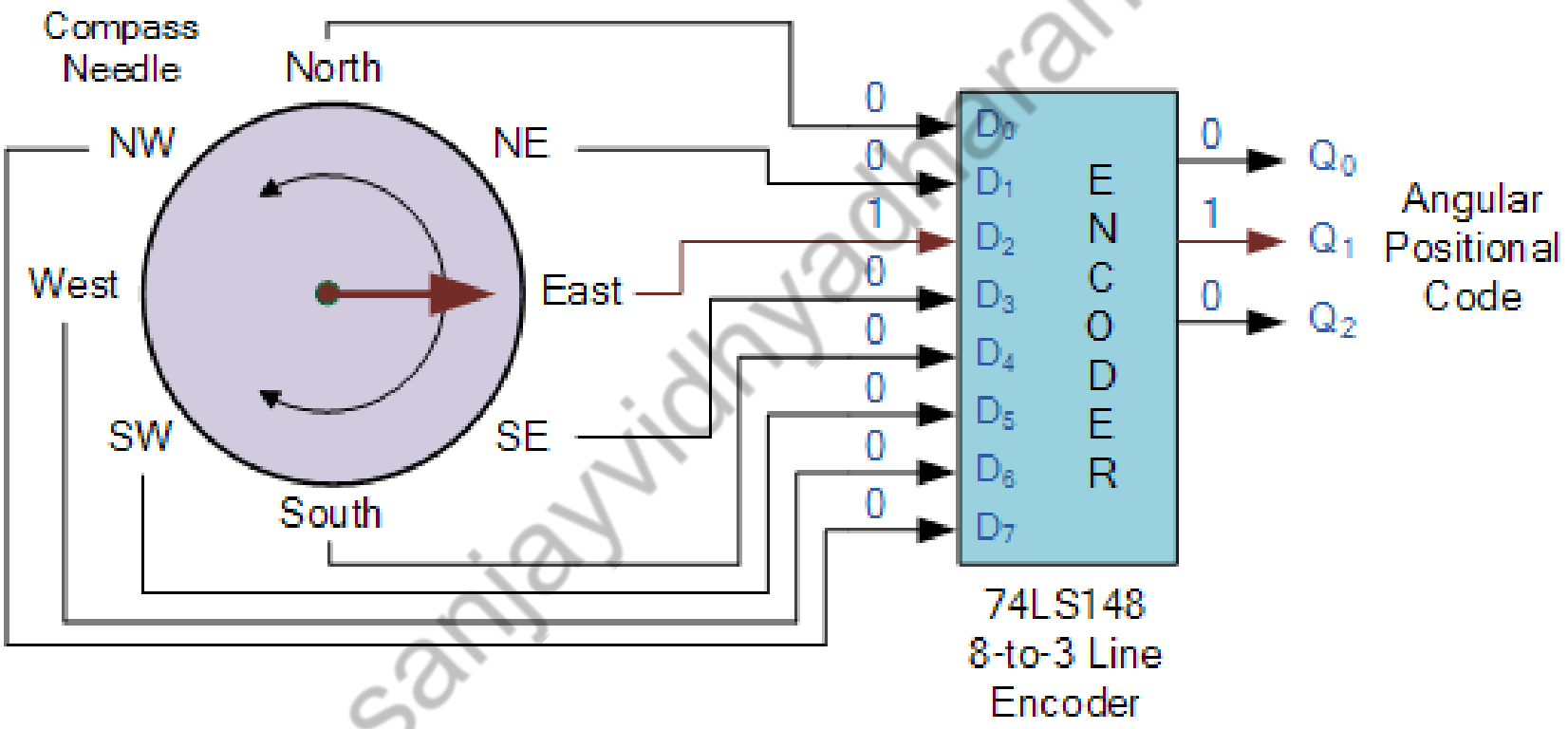


Binary to Gray Using Encoder-Decoder

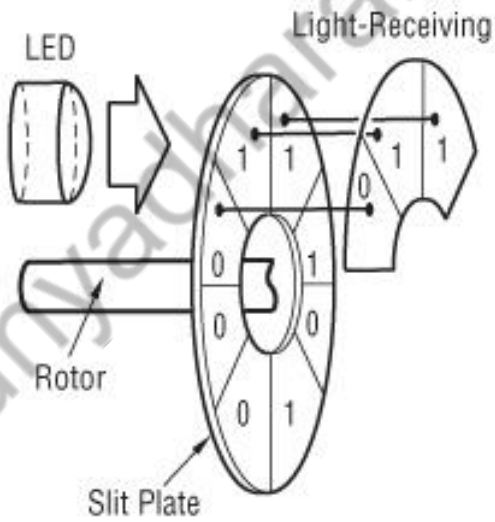


4-to-16 Line Decoder Implemented with two 3-to-8 Decoders

Encoder Applications

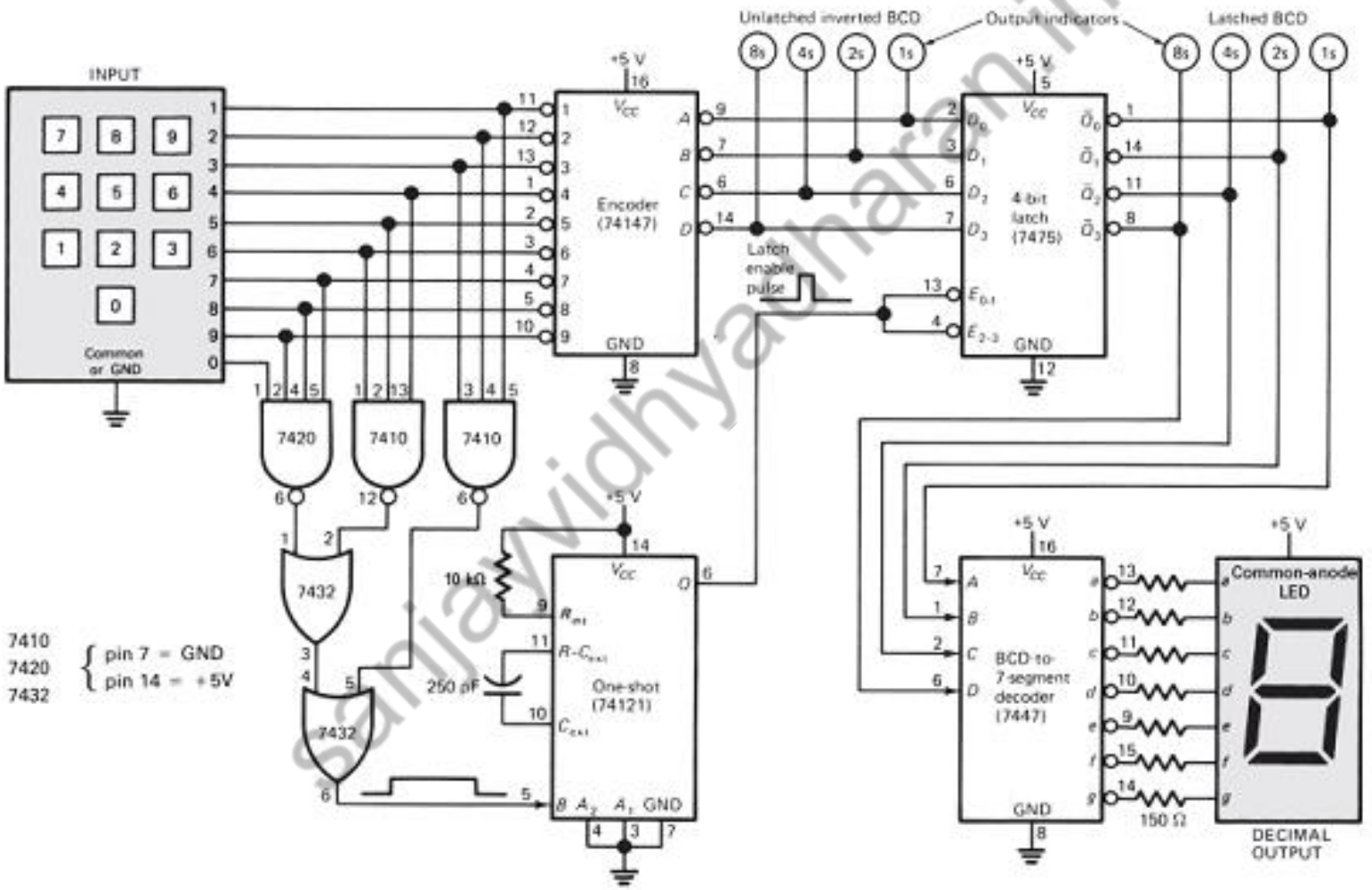


Encoder Applications

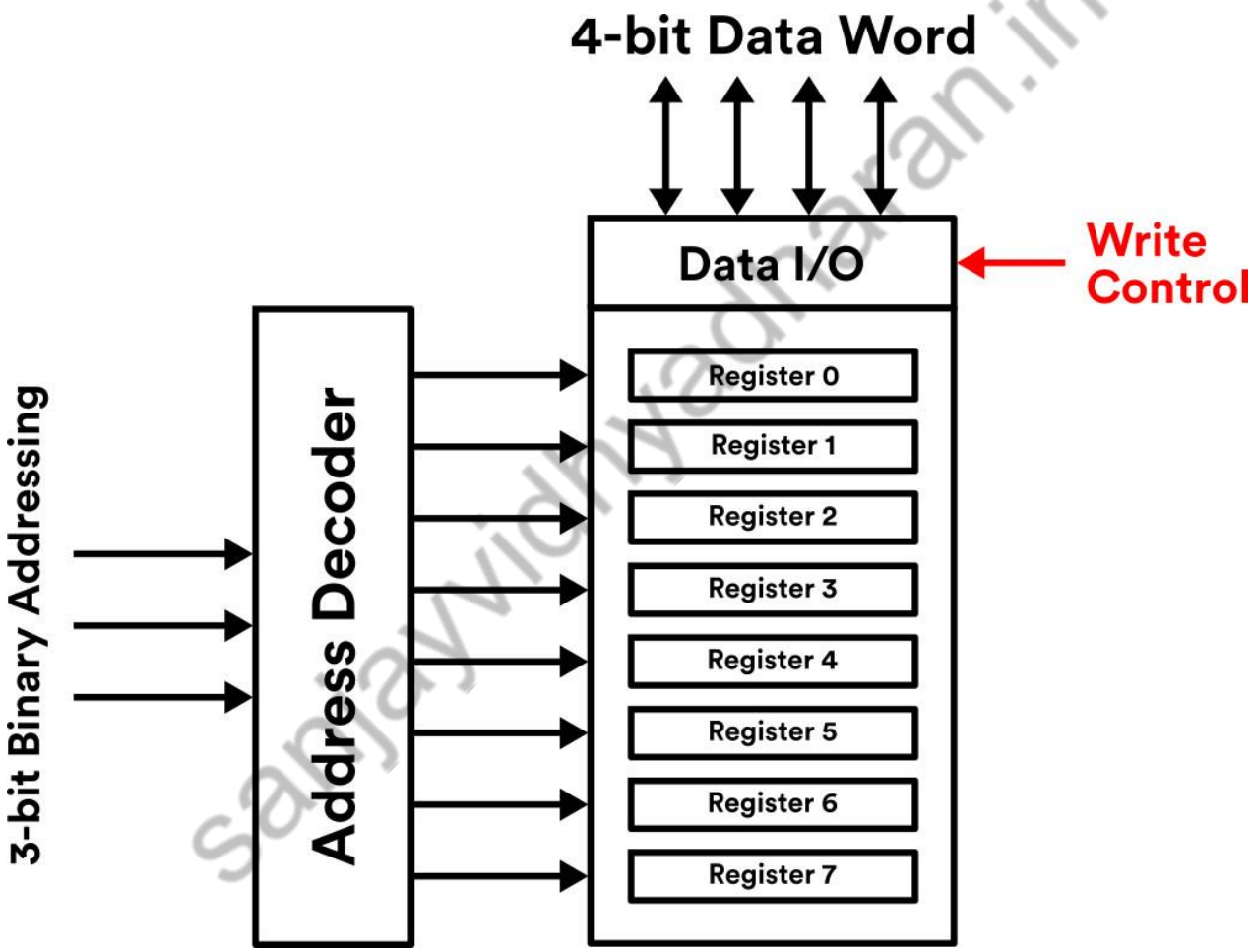


Sensor Output		Angle
011	➡	45°
001	➡	90°
000	➡	135°
100	➡	180°
010	➡	225°
101	➡	270°
110	➡	315°
111	➡	360°

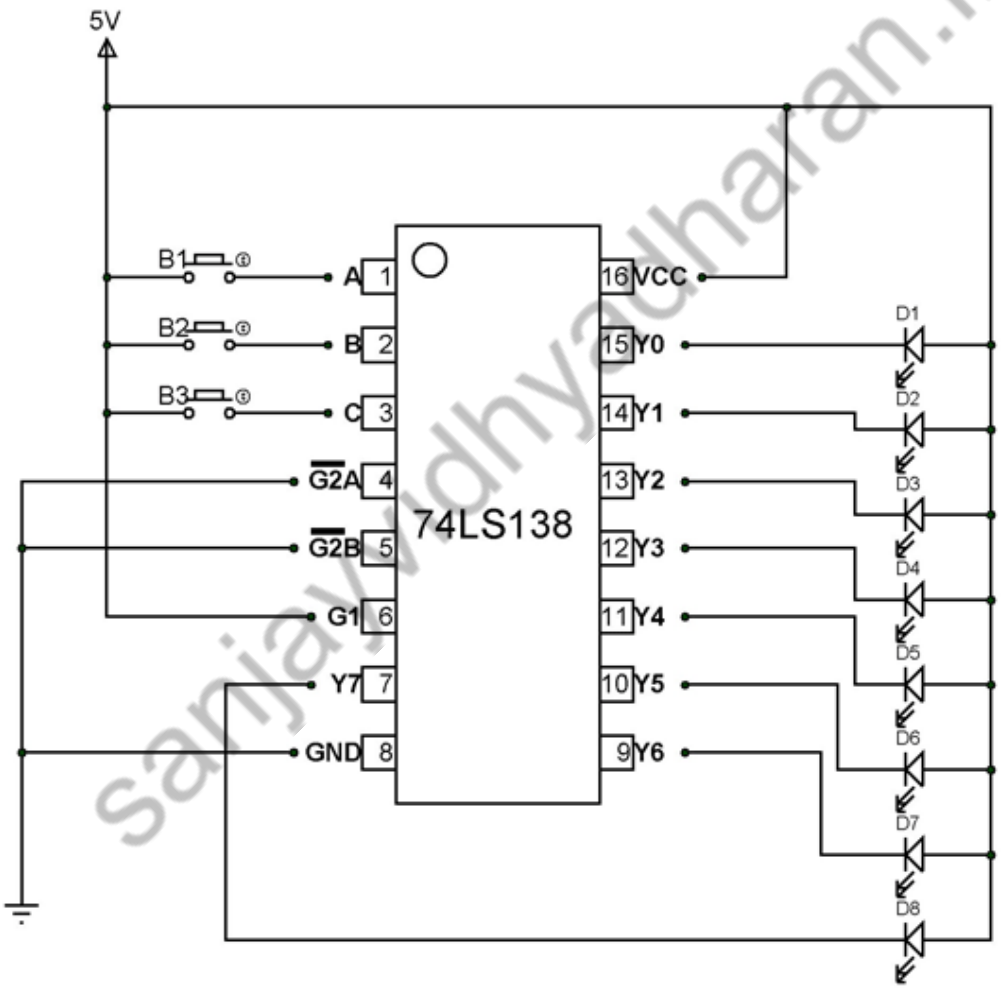
Encoder Applications



Decoder Applications



TTL Encoder and Decoder



Thank you