CSE1003 Digital Logic and Design
Module 4
Combinational Circuits II
L2

Dr. S.Hemamalini
Professor
School of Electrical Engineering
VIT Chennai

Contents

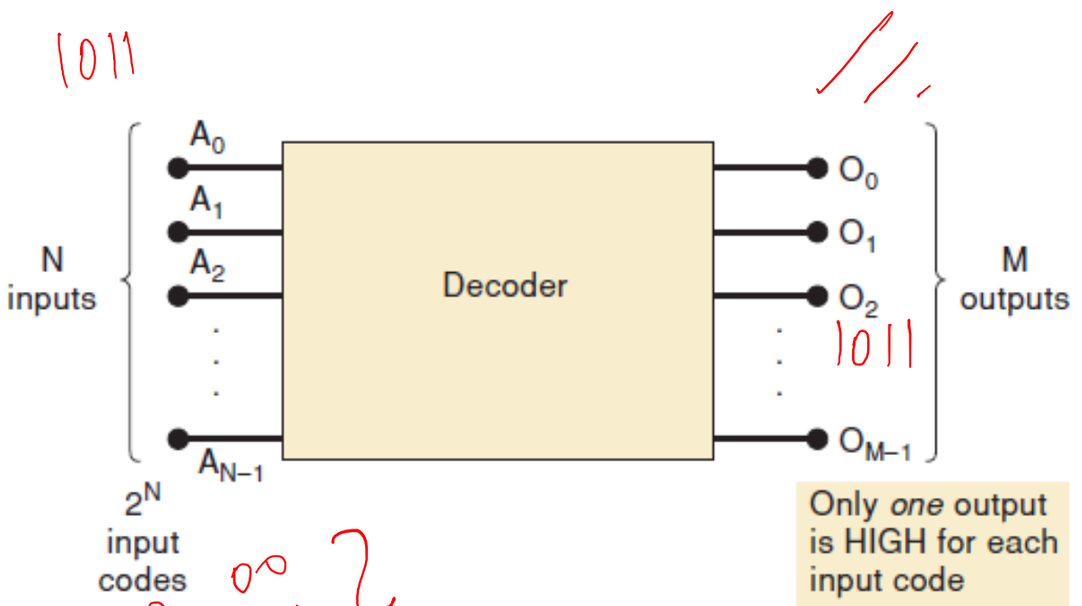
6 hrs

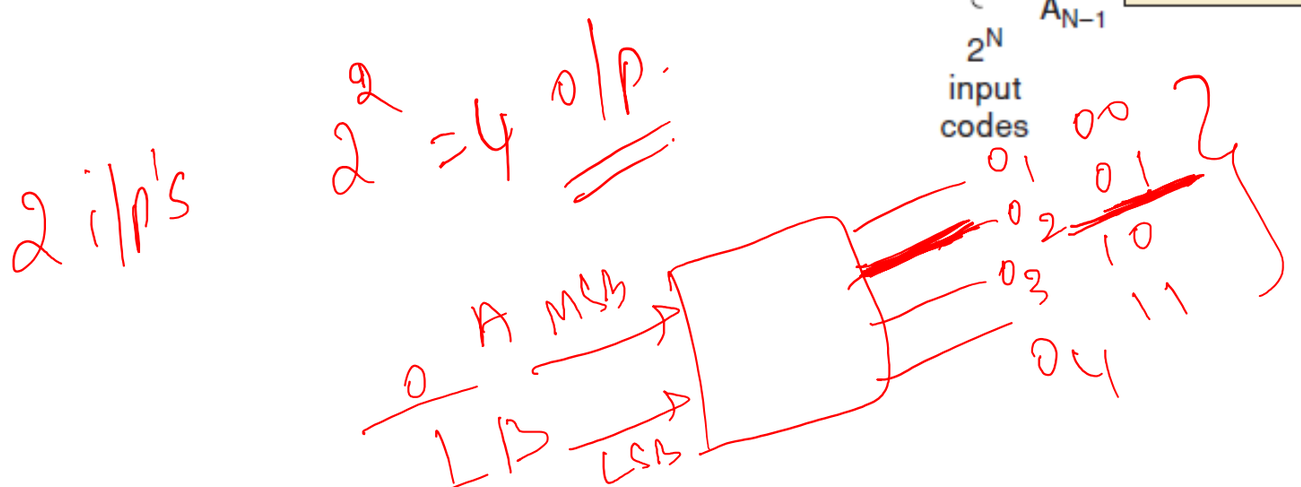
- Binary Parallel Adder - Look ahead carry
- Magnitude Comparator
- **Decoders**
- **Encoders**
- Multiplexers
- Demultiplexers

- CO4: Analyze the operation of medium complexity standard combinational circuits like the encoder, decoder, multiplexer, demultiplexer.

Decoders

- A **decoder** is a logic circuit that accepts a set of inputs that represents a binary number and activates only the output that corresponds to that input number.
- *N* inputs and *M* outputs
- 2^N possible input combinations or codes



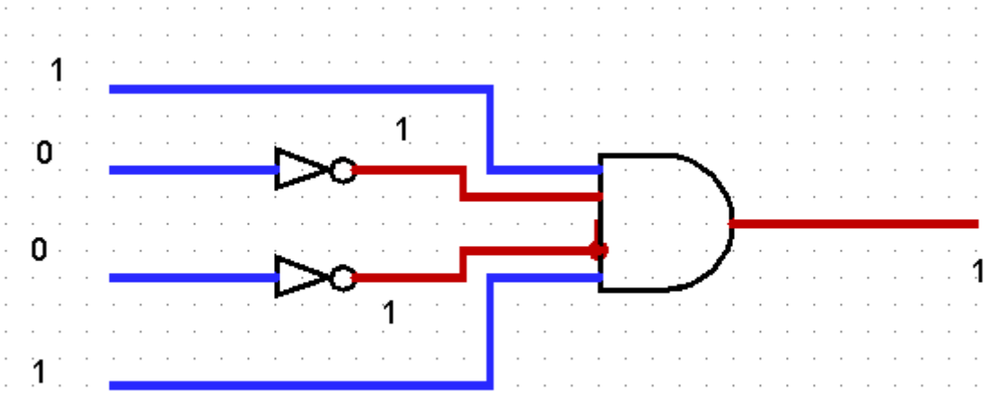


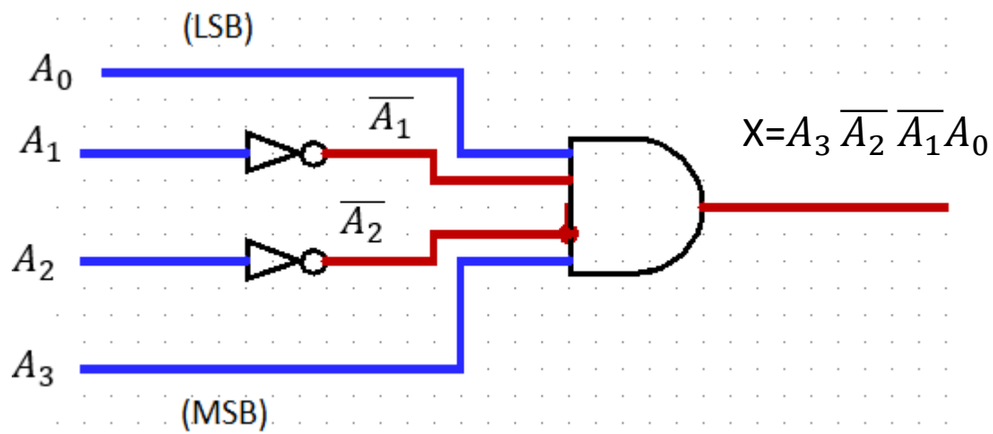
Decoders

I/P: $A_3 A_2 A_1 A_0$

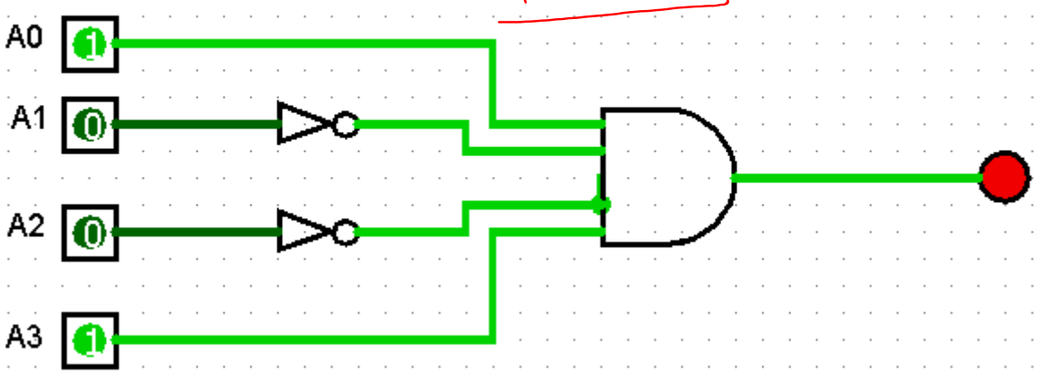
$2^3\ 2^2\ 2^1\ 2^0$

BASIC BINARY DECODER

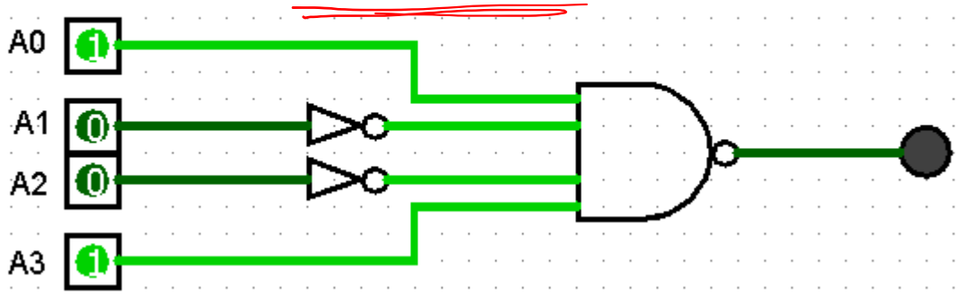




ACTIVE HIGH OUTPUT

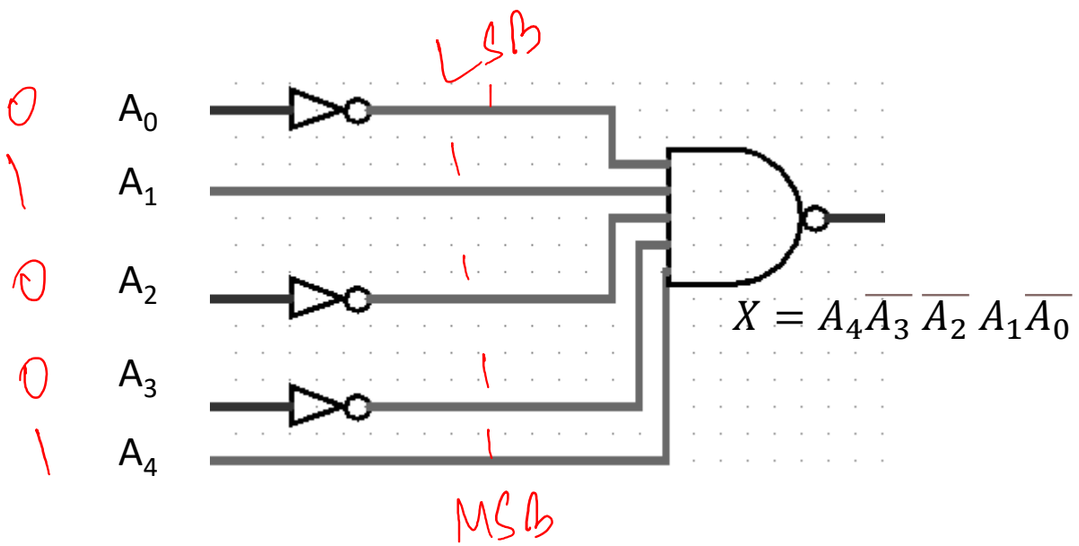


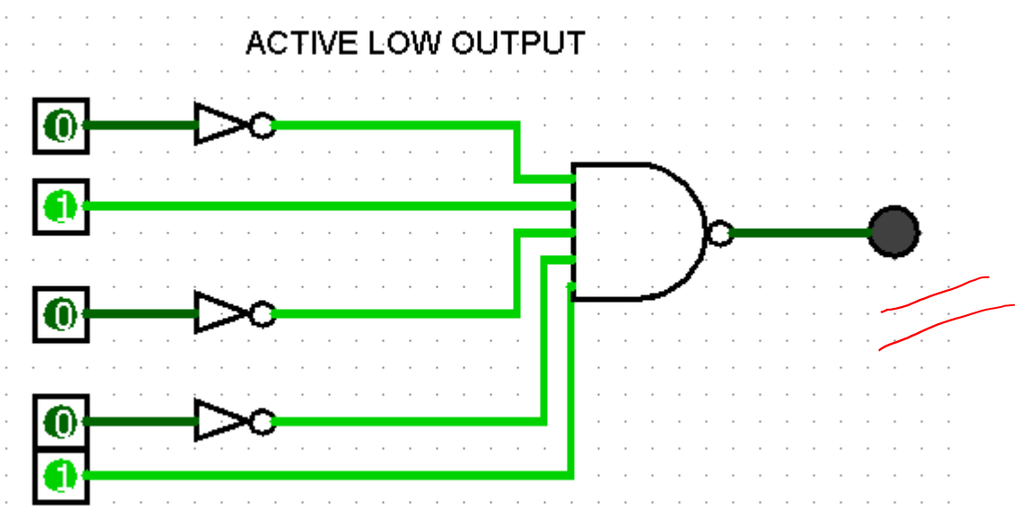
ACTIVE LOW OUTPUT



Decoders

- Develop the logic required to detect the binary code 10010 and produce an active LOW output.

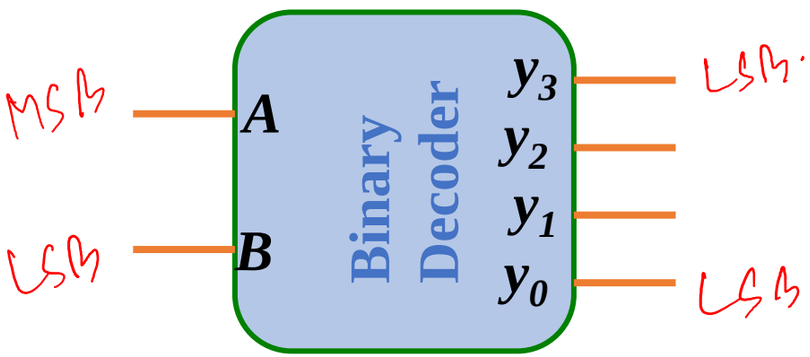




Decoders

2-to-4 Line Decoder

Block Diagram



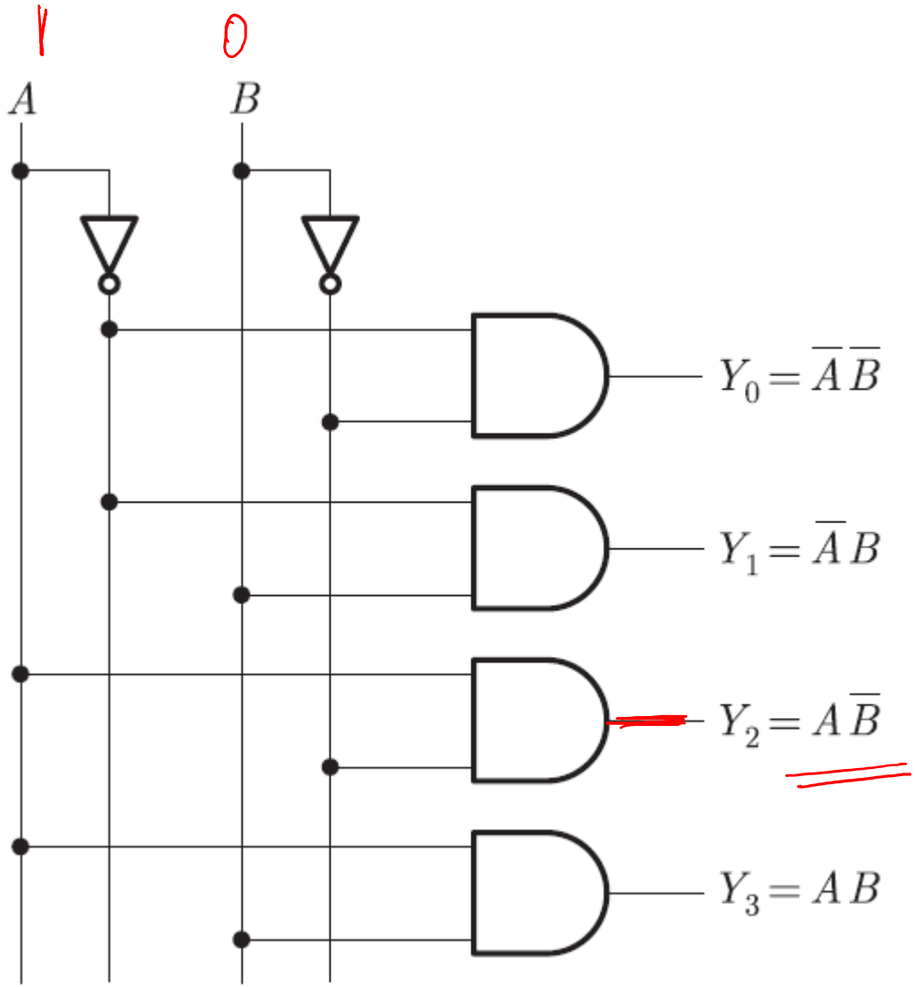
Truth Table

A	B	Y_3	Y_2	Y_1	Y_0	
0	0	0	0	0	1	m_0
0	1	0	0	1	0	m_1
1	0	0	1	0	0	m_2
1	1	1	0	0	0	m_3

Logical expressions

$$Y_0 = \overline{A}\,\overline{B} \quad \text{and} \quad Y_1 = \overline{A}B$$

$$Y_2 = A\overline{B} \quad \text{and} \quad Y_3 = AB$$

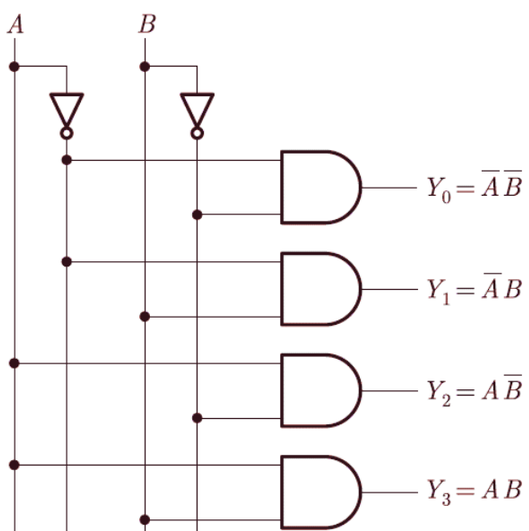


Logic Diagram

Decoders

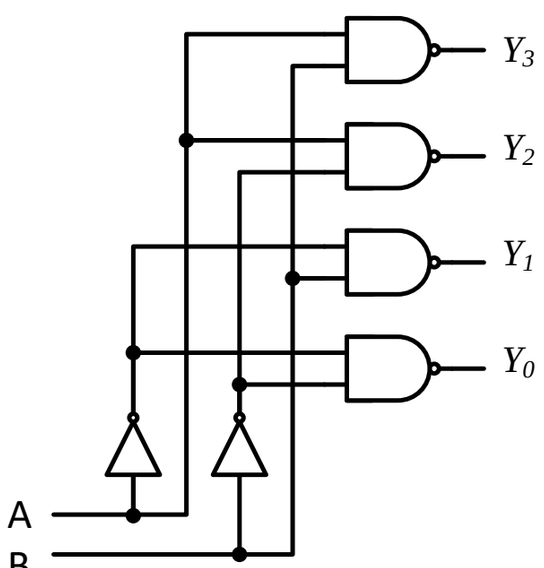
Active-High

A	B	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



Active-Low

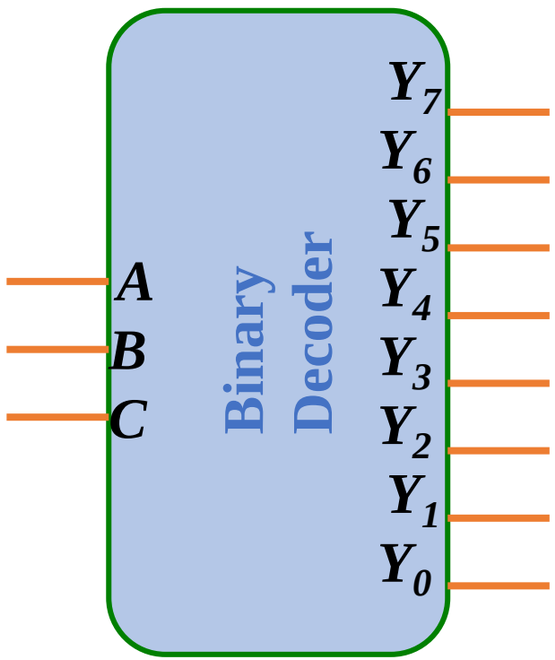
A	B	Y_3	Y_2	Y_1	Y_0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1



Decoders

3-to-8 Line Decoder

Block Diagram

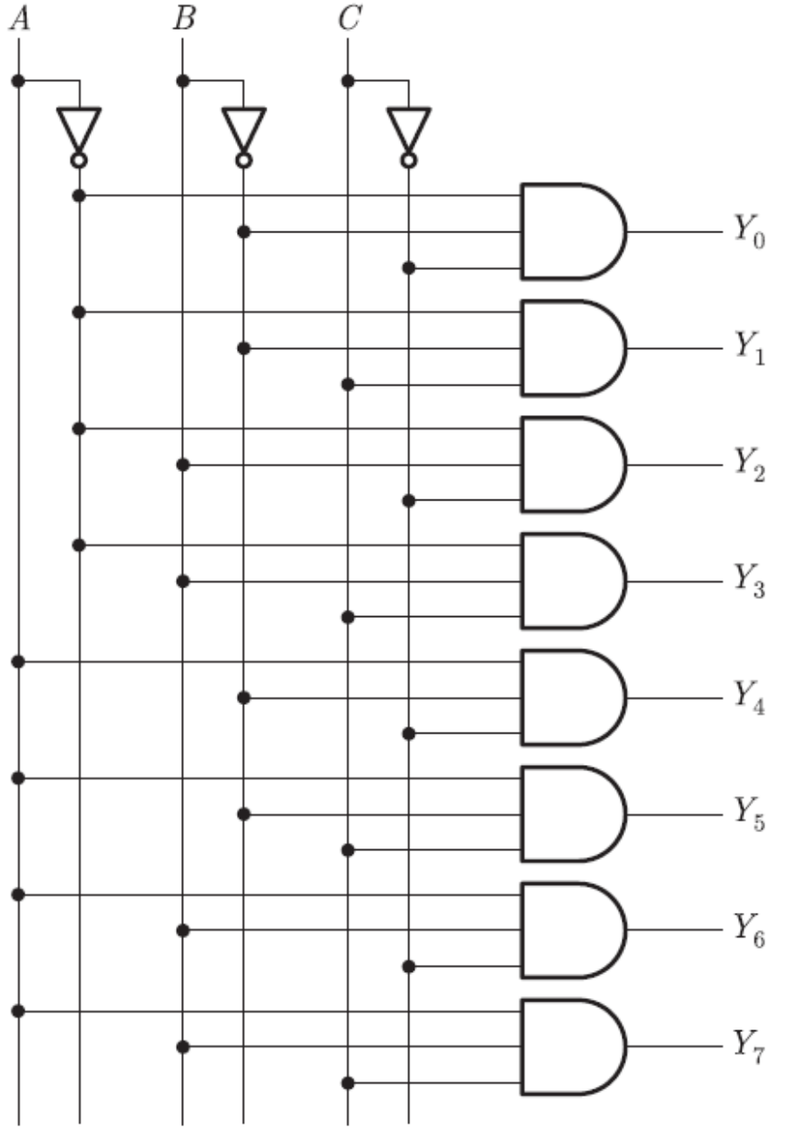


Truth Table

Inputs			Outputs								
A	B	C	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	
0	0	0	1	0	0	0	0	0	0	0	m_0
0	0	1	0	1	0	0	0	0	0	0	m_1
0	1	0	0	0	1	0	0	0	0	0	m_2
0	1	1	0	0	0	1	0	0	0	0	m_3
1	0	0	0	0	0	0	1	0	0	0	m_4
1	0	1	0	0	0	0	0	1	0	0	m_5
1	1	0	0	0	0	0	0	0	1	0	m_6
1	1	1	0	0	0	0	0	0	0	1	m_7

Logical expressions

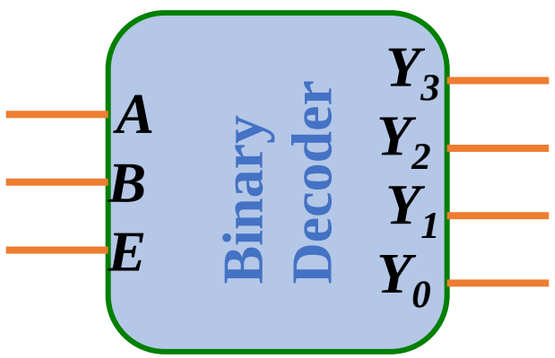
$$Y_0 = \overline{A}\,\overline{B}\,\overline{C};\ Y_1 = \overline{A}\,\overline{B}C;\ Y_2 = \overline{A}B\overline{C};\ Y_3 = \overline{A}BC$$
$$Y_4 = A\overline{B}\,\overline{C};\ Y_5 = A\overline{B}C;\ Y_6 = AB\overline{C};\ Y_7 = ABC$$



Logic Diagram

Decoder with ENABLE Input

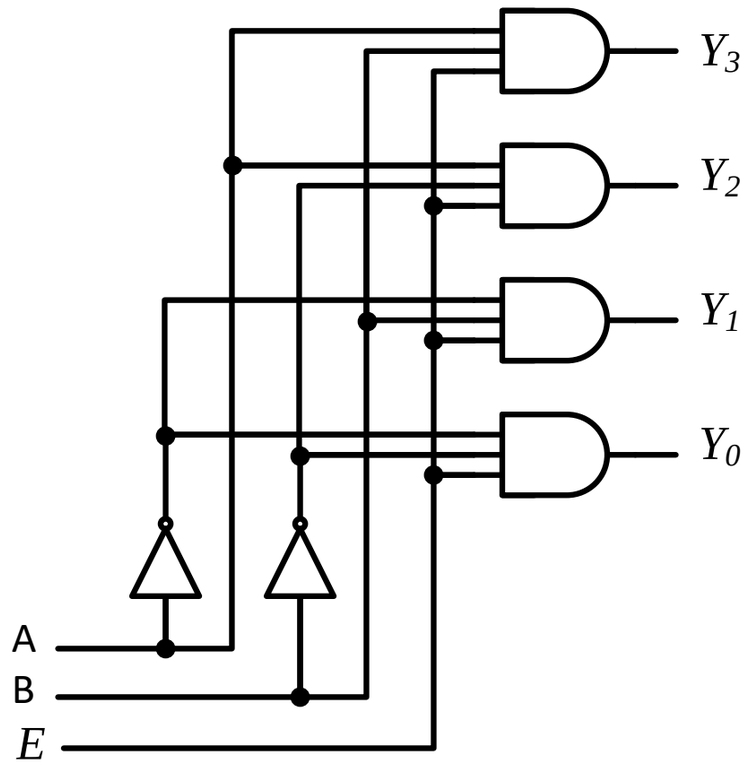
"*Enable*" Control → used to control the operation of the decoder



Active High
En = 1

Active Low
E = 0

E	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

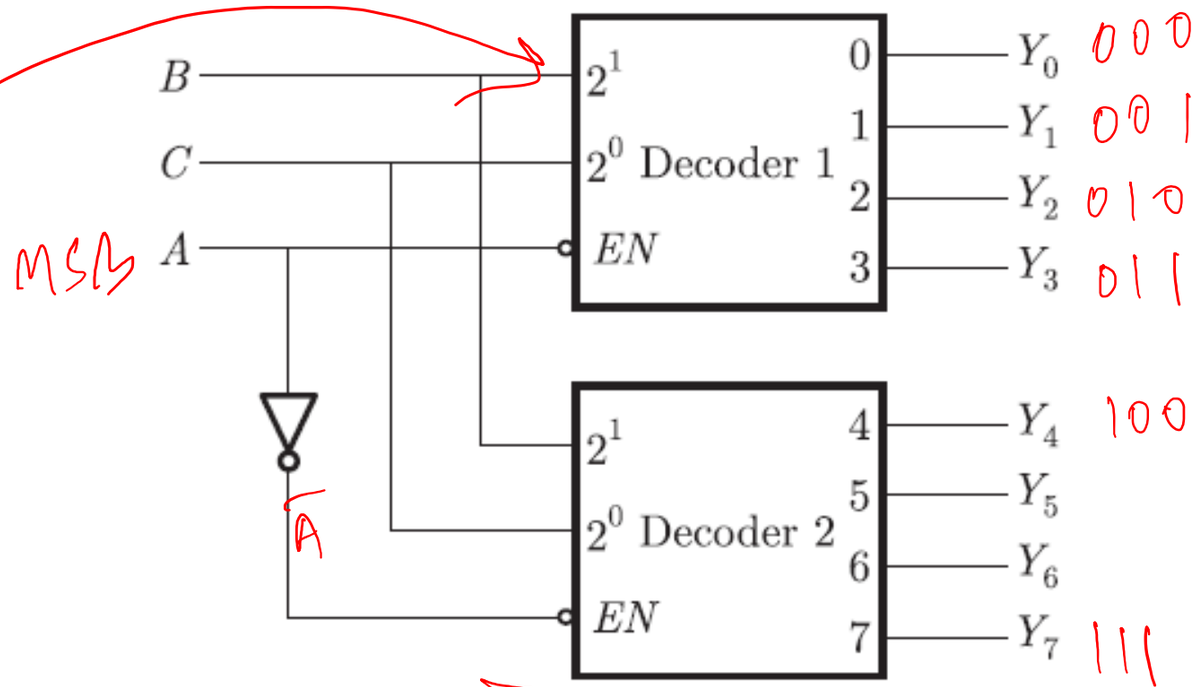


3-to-8 decoder using 2-to-4 decoder

74LS139

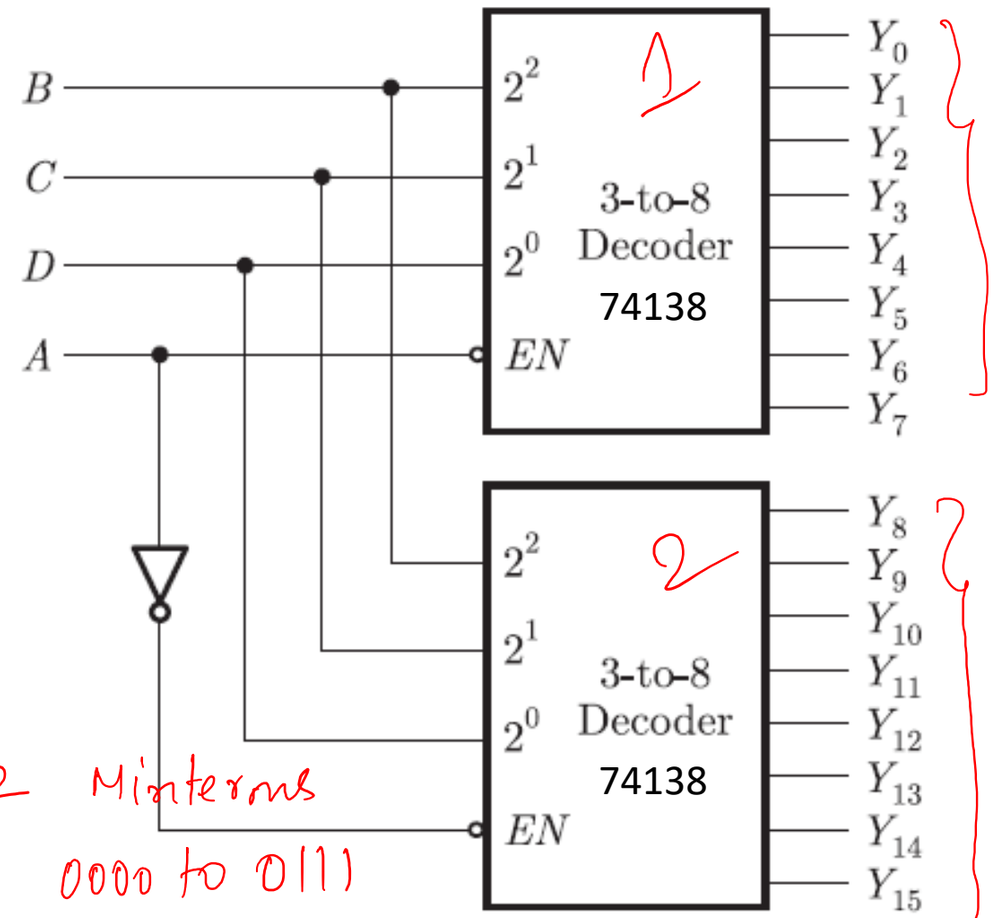
Expansion

A B C	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	1
0 0 1	0	0	0	0	0	0	1	0
0 1 0	0	0	0	0	0	1	0	0
0 1 1	0	0	0	0	1	0	0	0
1 0 0	0	0	0	1	0	0	0	0
1 0 1	0	0	1	0	0	0	0	0
1 1 0	0	1	0	0	0	0	0	0
1 1 1	1	0	0	0	0	0	0	0



4-to-16 Decoder using 3-to-8 Decoder

Binary Inputs				Decimal Output
A	B	C	D	Active Low
0	0	0	0	Y0
0	0	0	1	Y1
0	0	1	0	Y2
0	0	1	1	Y3
0	1	0	0	Y4
0	1	0	1	Y5
0	1	1	0	Y6
0	1	1	1	Y7
1	0	0	0	Y8
1	0	0	1	Y9
1	0	1	0	Y10
1	0	1	1	Y11
1	1	0	0	Y12
1	1	0	1	Y13
1	1	0	1	Y14
1	1	1	1	Y15



Enable

A	Decoder 1	Decoder 2	Minterms
0	Enabled	Disabled	0000 to 0111
1	Disabled	Enabled	1000 to 1111

IMPLEMENTATION OF LOGIC EXPRESSIONS USING DECODERS

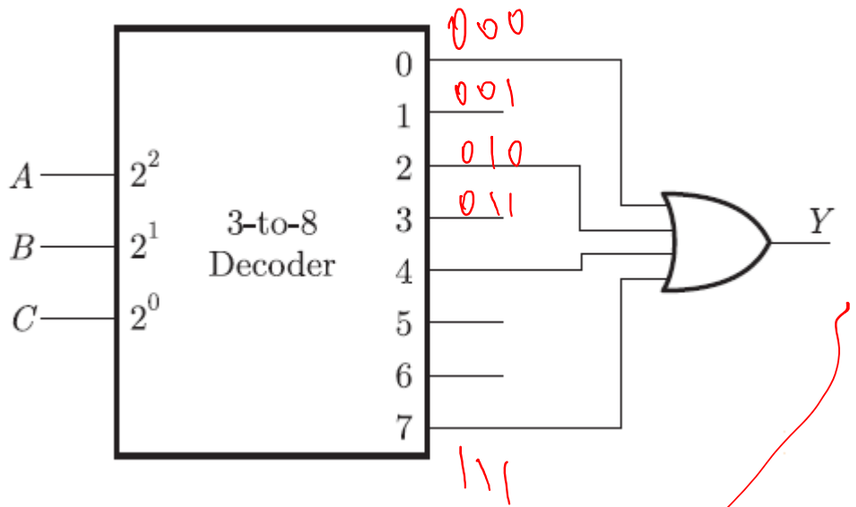
- Any combinational circuit with *n*-inputs and *m* outputs can be implemented with an *n*-to-2^n decoder and OR gates.

Procedure:

1. Express the given Boolean function in sum of minterms.

2. A decoder that generates all the minterms of the input variables is then chosen.

3. The inputs to each OR gate are selected from the decoder outputs according to the list of minterms of each function.



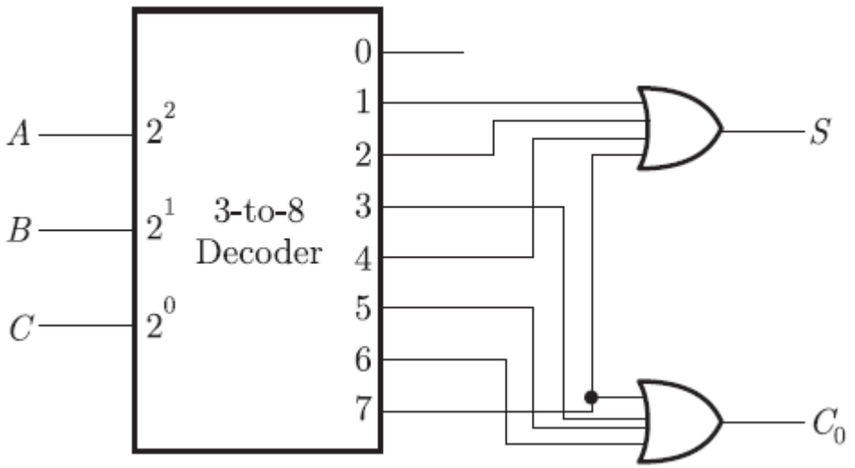
$$Y = A \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} + A \cdot B \cdot C + \overline{A} \cdot \overline{B} \cdot \overline{C}$$

$$Y = \Sigma m(0, 2, 4, 7)$$

Implement a full adder circuit using a 3-to-8 line decoder.

Each output is a minterm
All minterms are produced
Sum the required minterms

Inputs			Sum	Carry
A	B	C	S	C_0
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



Sum output, $S = \Sigma 1,2,4,7$

Carry output, $C_0 = \Sigma 3,5,6,7$