# SNS COLLEGE OF TECHNOLOGY

**Coimbatore-35**
**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

## 19EC306– DIGITAL ELECTRONICS

II YEAR/ III SEMESTER

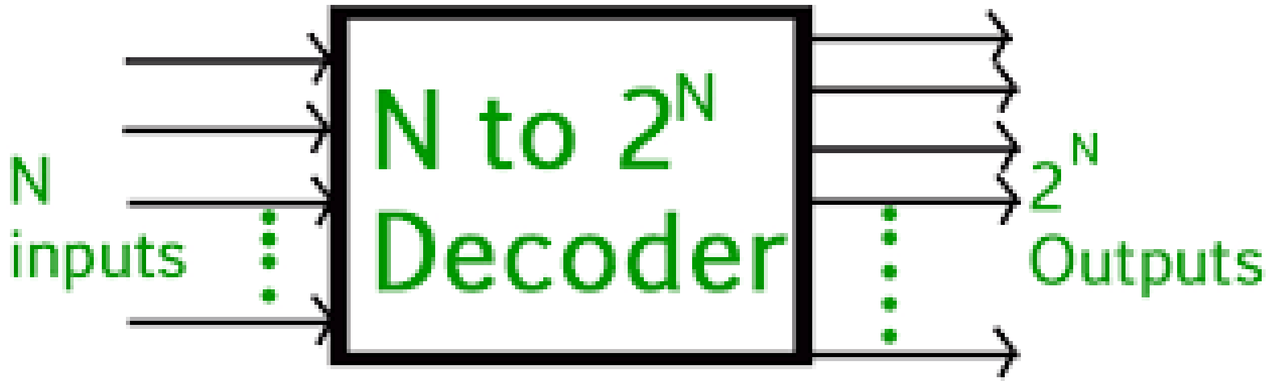
1

## UNIT 2 – **COMBINATIONAL CIRCUITS**

## TOPIC – DECODER AND ENCODER

# WHAT IS A DECODER?

- **Decoder** is a combinational logic circuit that converts binary information from the n coded inputs to a maximum of $2^n$ unique outputs.



# DECODER

A decoder has

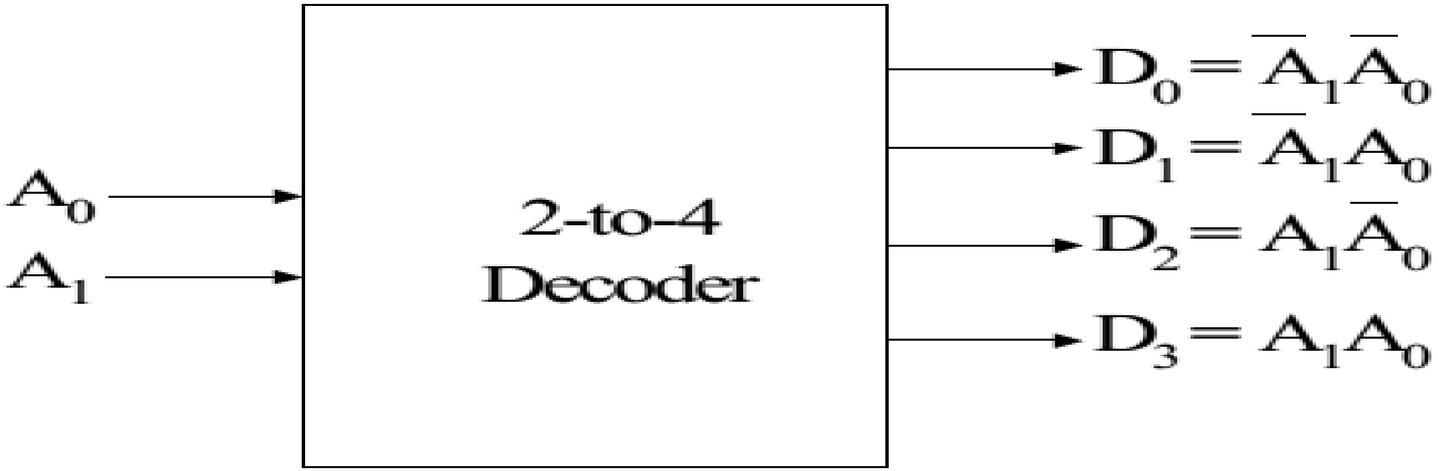
- *n* inputs
- $2^n$ outputs

- A decoder selects one of $2^n$ outputs by decoding the binary value on the *n* inputs.
- The decoder generates all of the minterms of the *n* input variables.

  - Exactly one output will be active for each combination of the inputs.

What does “active” mean?

# DECODER



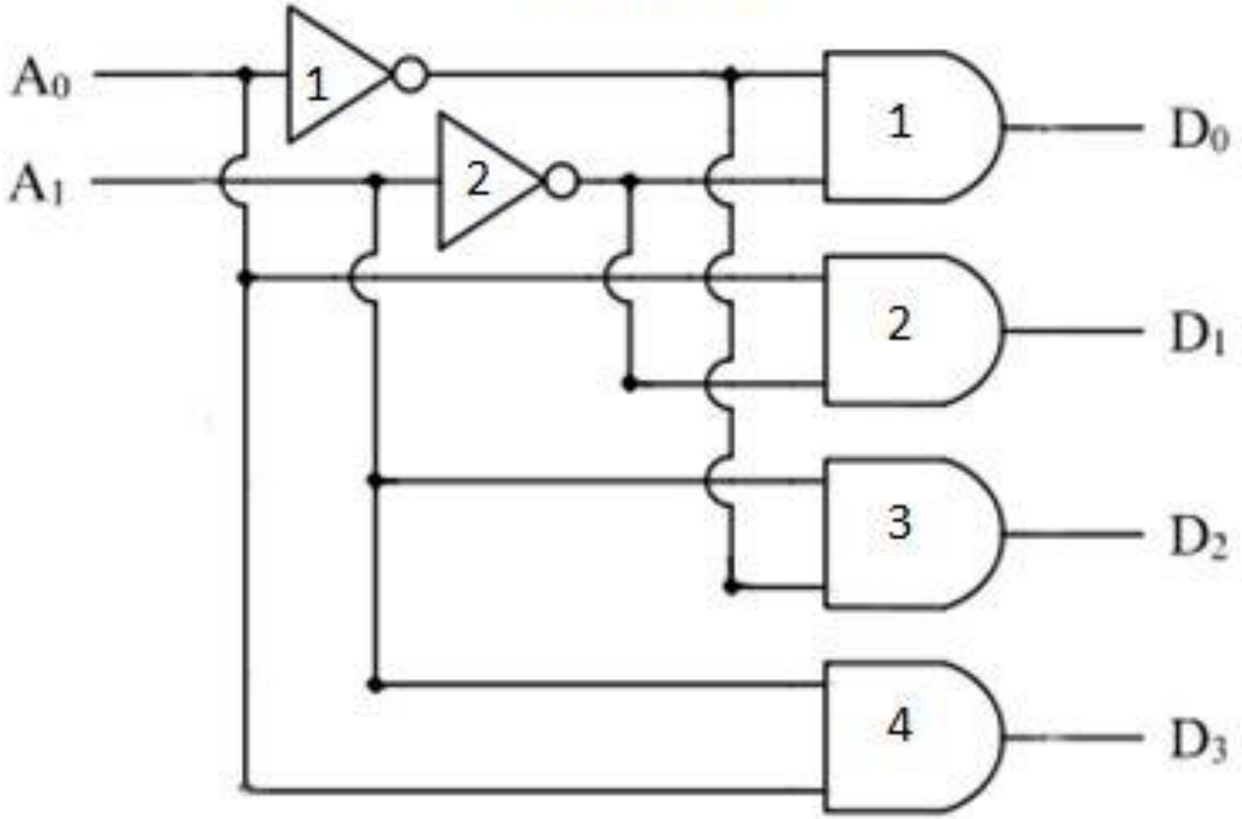

A 2-to-4 decoder without enable

| Decimal # | Input | | Output | | | |
|---|---|---|---|---|---|---|
| | $A_1$ | $A_0$ | $D_0$ | $D_1$ | $D_2$ | $D_3$ |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 1 | 0 |
| 3 | 1 | 1 | 0 | 0 | 0 | 1 |

Truth table for 2-to-4 decoder

# DECODER

**Logic Diagram**




**Truth Table**

| $A_1$ | $A_0$ | $D_3$ | $D_2$ | $D_1$ | $D_0$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

**Equations**

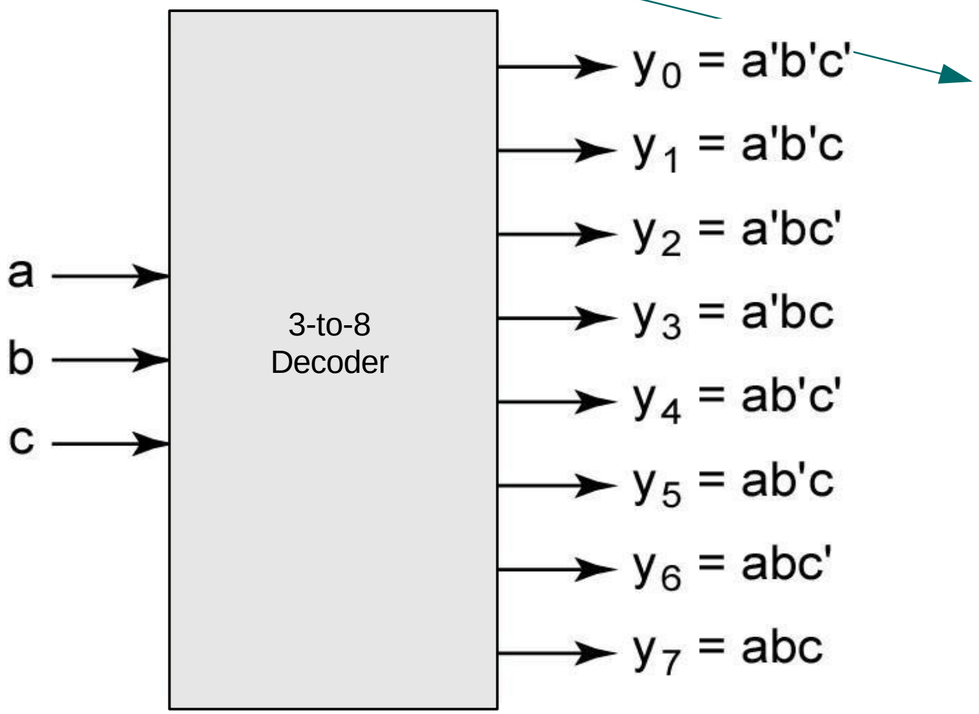
$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = \overline{A_1} \cdot A_0$$

$$D_2 = A_1 \cdot \overline{A_0}$$

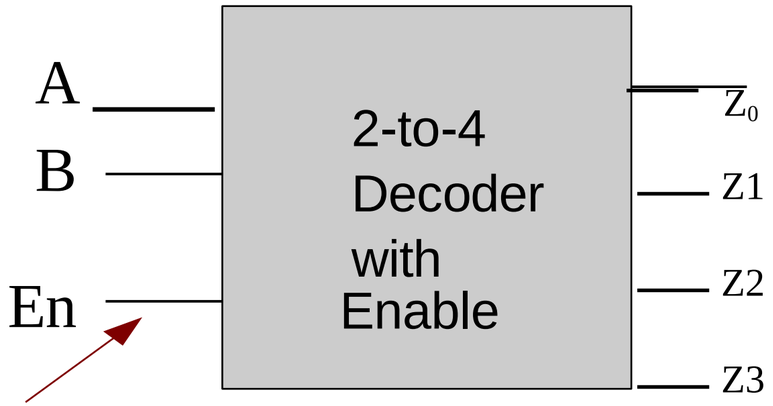
$$D_3 = A_1 \cdot A_0$$

# DECODERS




| 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 |
| 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 |

# Decoder with Enable



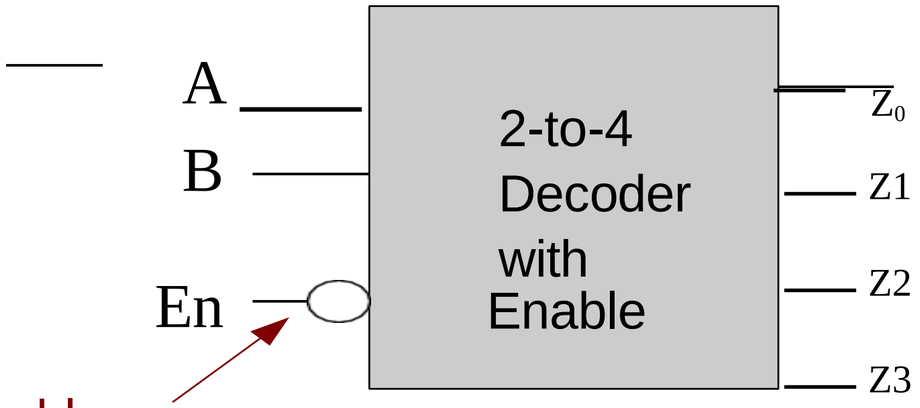

active-high enable

| En | A | B | $Z_0$ | $Z_1$ | $Z_2$ | $Z_3$ |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | x | x | 0 | 0 | 0 | 0 |

enabled

disabled

# Decoder with Enable




active-Low enable

enabled

disabled

| En | A | B | $Z_0$ | $Z_1$ | $Z_2$ | $Z_3$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | x | x | 0 | 0 | 0 | 0 |

# WHY ENCODERS?

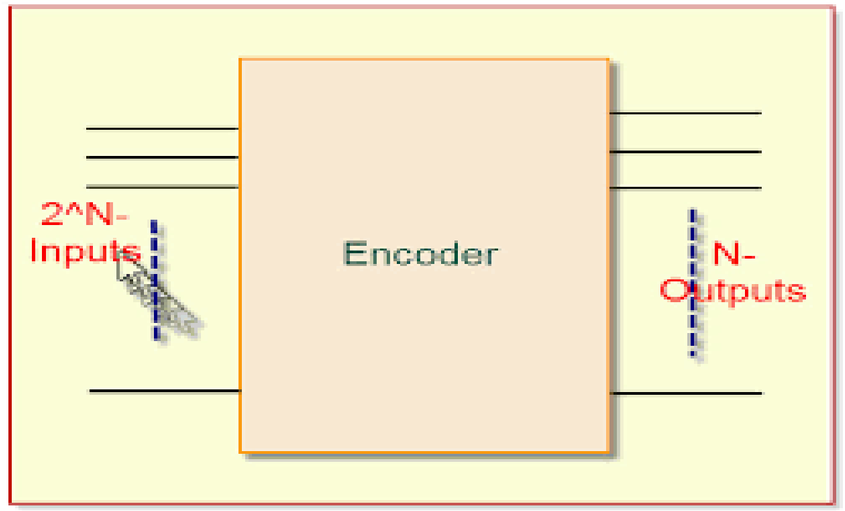
An encoder has

- $2^n$ inputs
- *n* outputs

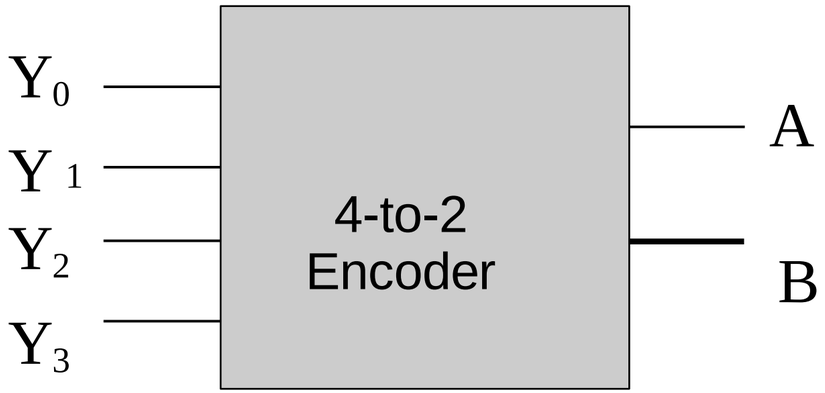
Outputs the binary value of the selected (or active) input.

Performs the inverse operation of a decoder. Issues

- What if more than one input is active?
- What if no inputs are active?



# Encoders

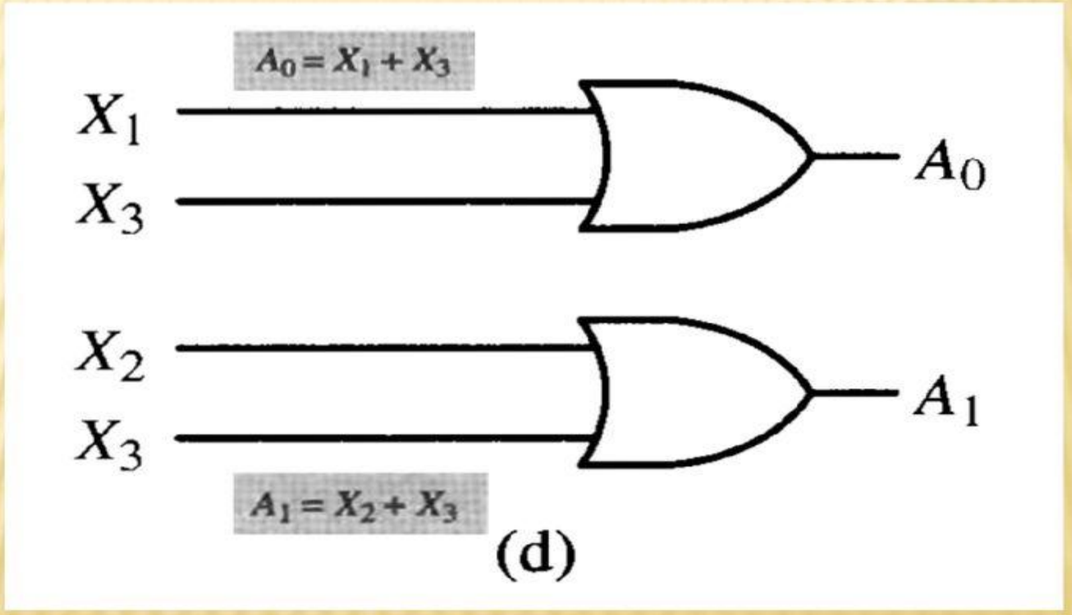



| $Y_0$ | $Y_1$ | $Y_2$ | $Y_3$ | A | B |
|---|---|---|---|---|---|
| 1 | 0 | $0_{10}$ | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |

# Encoders

## 4 TO 2 LINE ENCODER

Logic diagram

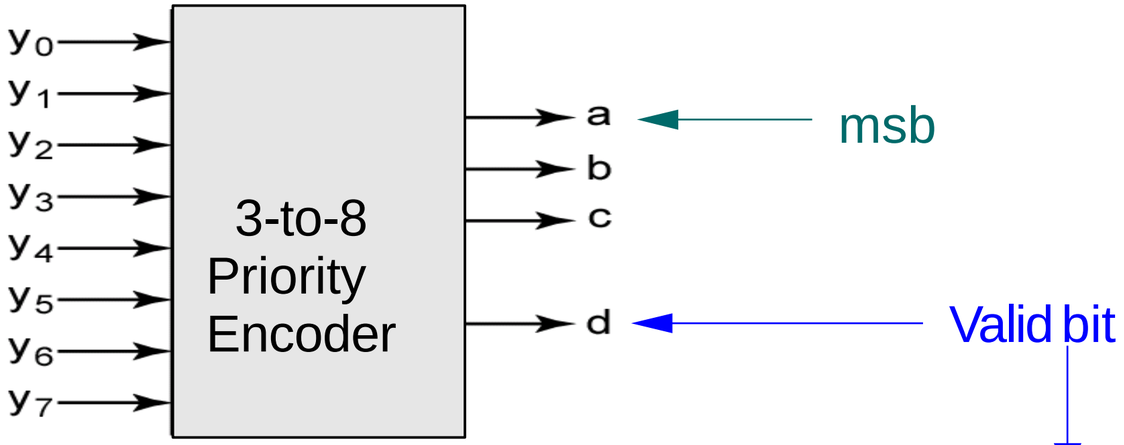


# Priority Encoders

- If more than one input is active, the higher-order input has priority over the lower-order input.
  - The higher value is encoded on the output
- A valid indicator, d, is included to indicate whether or not the output is valid.
  - Output is invalid when no inputs are active
    - d = 0
  - Output is valid when at least one input is active
    - d = 1

12

# Priority Encoders




| $y_0$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | $y_6$ | $y_7$ | a | b | c | d |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| X | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| X | X | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| X | X | X | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| X | X | X | X | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| X | X | X | X | X | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| X | X | X | X | X | X | 1 | 0 | 1 | 1 | 0 | 1 |
| X | X | X | X | X | X | X | 1 | 1 | 1 | 1 | 1 |

# Using an *n*-output Decoder

- *n*-output decoder to realize a logic circuit for a function with *n* minterms.
- Each minterm of the function can be mapped to an output of the decoder.
- For each row in the truth table, for the function, where the output is 1, sum (or "OR") the corresponding outputs of the decoder.

14

- That is, for each minterm in the minterm expansion of the function, OR the corresponding outputs of the decoder.

Leave remaining outputs of the decoder unconnected.

# Using an *n*-output Decoder

**Example**

- Using a 3-to-8 decoder, design a logic circuit to realize the following Boolean function

- F(A,B,C) = □m(2, 3, 5, 6, 7)

# Using an *n*-output Decoder

**Example**

- Using a 2-to-2 decoder, design a logic circuit to realize the following Boolean function

16

$$F(A,B,C) = \Sigma m(0, 1, 4, 6, 7)$$

# ASSESSMENT

1. What is a Encoder?
2. Device which converts an input device state into a binary representation of ones or zeros is termed as
    1. **Encoder**
    2. Decoder
    3. Multiplexer
    4. Data selector

17

3. A decoder converts n inputs to _____ outputs.($2^n$)
4. ---------- are building blocks of encoders.(Ans - OR gate)
5. Draw the block diagram of 2x4 decoder.

# THANK YOU