# DIGITAL DESIGN

CS/ECE/EEE/INSTR F215

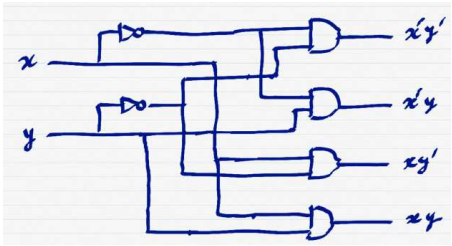
Sarang Dhongdi

# DECODER

## Decoder

- n inputs → (max) $2^n$ unique output lines
- n-to-m line decoder ($m \leq 2^n$)
- Generates minterms of n input variables
- General decoders
  - 2 to 4 line decoder
  - 3 to 8 line decoder
  - 4 to 16 line decoder

## Two-to-Four line Decoder

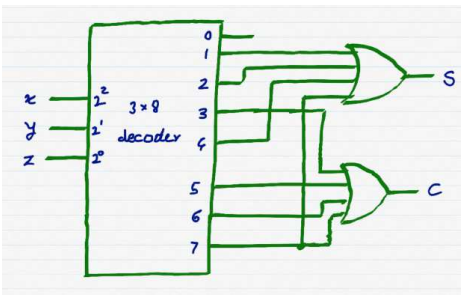



## Three-to-Eight line decoder

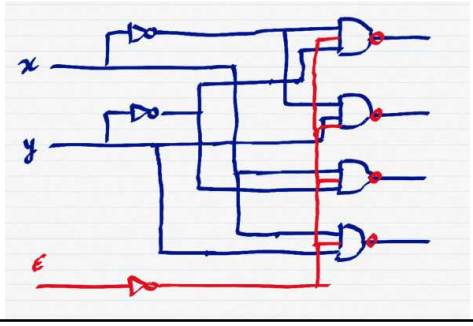
| Inputs | | |
|---|---|---|
| X | Y | Z |
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

| Outputs | | | | | | | |
|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 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 |

## Implement Full adder using 3:8 decoder



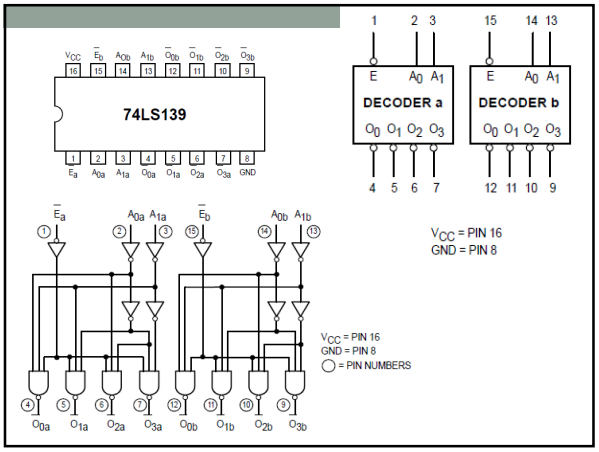
## 2-to-4 line Decoder with Enable input



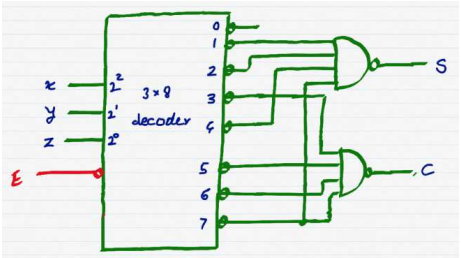

## 2-to-4 line Decoder with Enable input

| Inputs | | |
|---|---|---|
| E | A | B |
| 1 | X | X |
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |

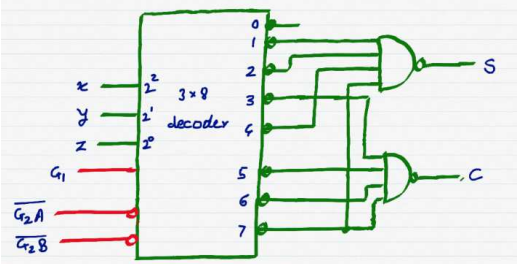
| Outputs | | | |
|---|---|---|---|
| D0 | D1 | D2 | D3 |
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |



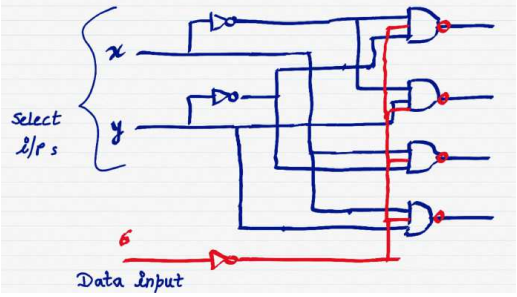

## Implement Full adder using 3:8 decoder, with enable i/p



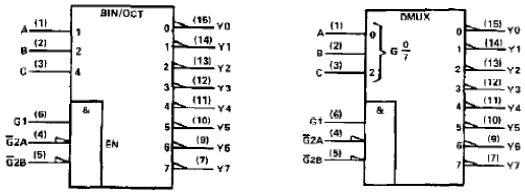

## 3:8 decoder – MSI IC 74138




## Decoder-Demultiplexer



## 3:8 decoder – MSI IC 74138




## Truthtable from Datasheet

| INPUTS | | | | | OUTPUTS | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ENABLE | | SELECT | | | | | | | | | | |
| G1 | G2* | C | B | A | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
| X | H | X | X | X | H | H | H | H | H | H | H | H |
| L | X | X | X | X | H | H | H | H | H | H | H | H |
| H | L | L | L | L | L | H | H | H | H | H | H | H |
| H | L | L | L | H | H | L | H | H | H | H | H | H |
| H | L | L | H | L | H | H | L | H | H | H | H | H |
| H | L | L | H | H | H | H | H | L | H | H | H | H |
| H | L | H | L | L | H | H | H | H | L | H | H | H |
| H | L | H | L | H | H | H | H | H | H | L | H | H |
| H | L | H | H | L | H | H | H | H | H | H | L | H |
| H | L | H | H | H | H | H | H | H | H | H | H | L |

## Encoders

- Inverse operation of that of Decoders
- $2^n$ input lines → n output lines
- Output is binary code of the input

## Octal-to-Binary Encoder

- Assumption – only one i/p has value 1 at any given time.

| Inputs | | | | | | | | Outputs | | |
|---|---|---|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | X | Y | Z |
| 1 | 0 | 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 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Z =D1+D3+D5+D7

Y = D2+D3+D6+D7

X = D4+D5+D6+D7

Issues –

- When 2 i/ps are simultaneously 1.
- When none of the i/p is 1, o/p is 0

## Priority Encoder

- If two or more inputs are equal to 1 at the same time, the input having highest priority will take precedence.

| Inputs | | | |
|---|---|---|---|
| D0 | D1 | D2 | D3 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| X | 1 | 0 | 0 |
| X | X | 1 | 0 |
| X | X | X | 1 |

| Outputs | | |
|---|---|---|
| A | B | V |
| X | X | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## Maps for Priority Encoders

- A = ?

| D0D1 \ D2D3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | 1 | 1 |
| 01 | | 1 | 1 | 1 |
| 11 | | 1 | 1 | 1 |
| 10 | | 1 | 1 | 1 |

## Maps for Priority Encoders

- A = D2+D3

| D0D1 \ D2D3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | 1 | 1 |
| 01 | | 1 | 1 | 1 |
| 11 | | 1 | 1 | 1 |
| 10 | | 1 | 1 | 1 |

## Maps for Priority Encoders

- B = ?

| D0D1 \ D2D3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | 1 | |
| 01 | 1 | 1 | 1 | |
| 11 | 1 | 1 | 1 | |
| 10 | | 1 | 1 | |

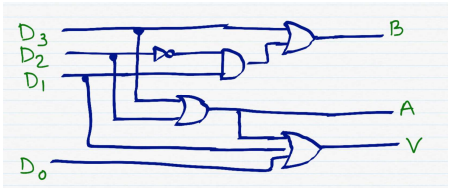
## Maps for Priority Encoders

- B = D3+D1D2'

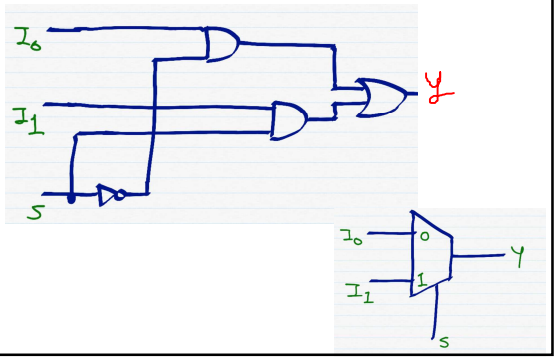
| D0D1 \ D2D3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | 1 | |
| 01 | 1 | 1 | 1 | |
| 11 | 1 | 1 | 1 | |
| 10 | | 1 | 1 | |

## Four input Priority Encoder

- A = D2+D3
- B = D3+D1D2'
- V = D0+D1+D2+D3



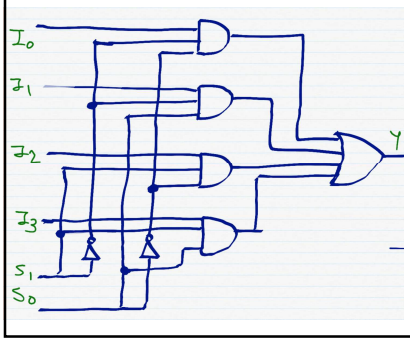

## Multiplexer




## Multiplexer

- Selects 'binary' information from one of the many i/p lines and connects it to a single o/p line.
- $2^n$ i/p lines, n selection lines
- Combination of select lines determines which i/p is selected.
- Also known as "Data selector"

## Four-to-One line multiplexer




| S1 | S0 | Y |
|---|---|---|
| 0 | 0 | $I_0$ |
| 0 | 1 | $I_1$ |
| 1 | 0 | $I_2$ |
| 1 | 1 | $I_3$ |

## Mux MSI IC's

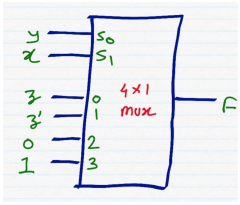
- 74153 – Dual 4:1 Mux
- 74151 – 8:1 Mux
- 74150 – 16:1 Mux

## Boolean function implementation

- f(x,y,z) = Σ (1,2,6,7)

| X | Y | Z | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

F
Z
Z'
0
1



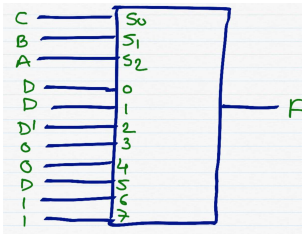

## f(A,B,C,D) = Σ (1,3,4,11,12,13,14,15)

| A | B | C | D | O/p |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

## f(A,B,C,D) = Σ (1,3,4,11,12,13,14,15)

| A | B | C | D | O/p |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

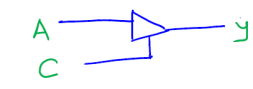
D D D' 0 0 1 1



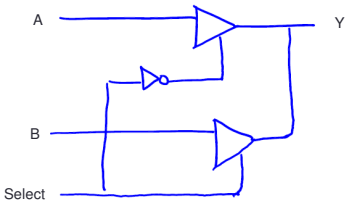

## Three-State Gates

- Third-state is high-impedance state –
- A) logic behaves like an open ckt, i.e. o/p is disconnected
- B) Ckt has no logic significance
- C) Ckt connected to the output of the three-state gate is not affected by the inputs.

## Three-State Gate -Buffer




Output y = A if C=1
High Impedance if c=0




2-to-1 line Mux

## 4-to-1 line Mux