CSE 201: Digital Logic Design
Encoder, Priority Encoder, Multiplexer

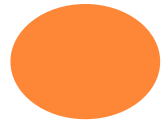
Prepared By

Lec Sumaiya Afroz Mila

CSE, MIST

ENCODER

- Does reverse operation to decoder
- An encoder has 2^n **(or fewer)** input lines and *n* output lines
- Constraint – only one input is active at a time
- An ***encoder*** is a digital circuit that has **2^n or fewer input lines** and **n output lines**. The output lines generate the binary code corresponding to the input value
- Example – Octal to binary encoder
  - Has 8 inputs, one for each octal digit
  - 3 outputs, that generates the corresponding binary number
  - It is assumed that, **only one input has a value of 1 at a given time**



ENCODER

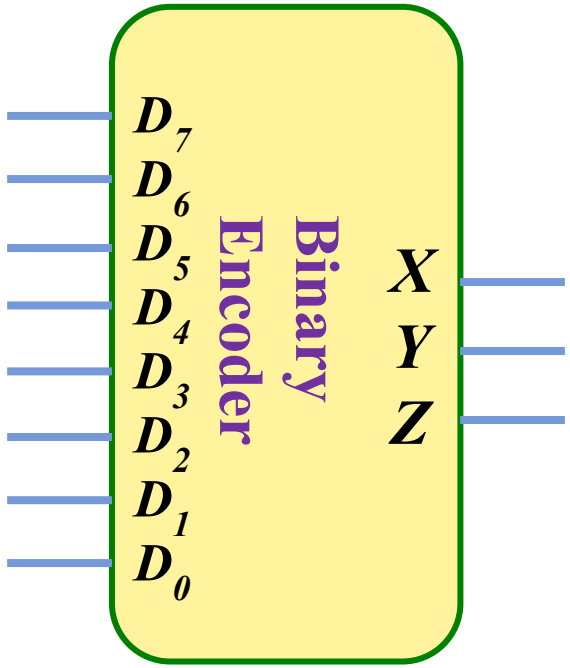
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	X	Y	Z
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Octal to Binary encoder truth table

$$X = D_7 + D_6 + D_5 + D_4$$

$$Y = D_7 + D_6 + D_3 + D_2$$

$$Z = D_7 + D_5 + D_3 + D_1$$



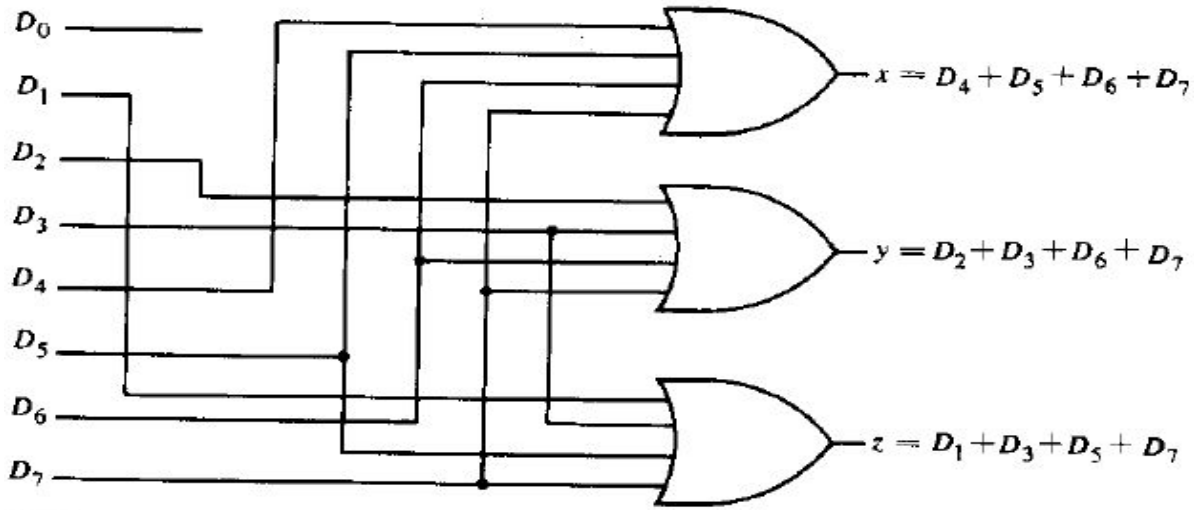
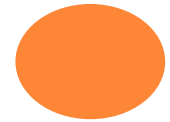


FIGURE 5-13
Octal-to-binary encoder

ENCODER

- Limitations of the previous encoder –
  - **Only one input can be active at any given time:** If two inputs are active simultaneously, output produces undefined combination. I.e. If D_3 and D_6 are active at the same time, output of the encoder will be 111 which is neither binary 3 nor binary 6. **To solve this limitation**, priority encoder is introduced.
  - When all the inputs are 0, output of the encoder will be 000. Again, output of the encoder will be 000 if D_0 is equal to 1. This can be resolved by providing an additional output indicating that none of the inputs are active.



PRIORITY ENCODER

- A priority encoder is a special type of encoder circuit that includes priority function
- Multiple inputs maybe active at the same time
- Highest priority input gets the precedence
- D_3 has the highest priority, then D_2 has the second highest priority and so on.
- When input D_3 is equal to 1, **regardless of the values of the other inputs**, output for XY will be 11(binary 3)
- As D_2 has the second highest priority level, when D_2 = 1, regardless of the values of the other two lower priority inputs(D_1 and D_0), the output for XY will be 10, given that **D_3 must be equal to 0**
- A ***Valid-output*** indicator, set to 1 when one or more of the inputs are active
- When all inputs are inactive, V=0, other two outputs of the circuit are not used

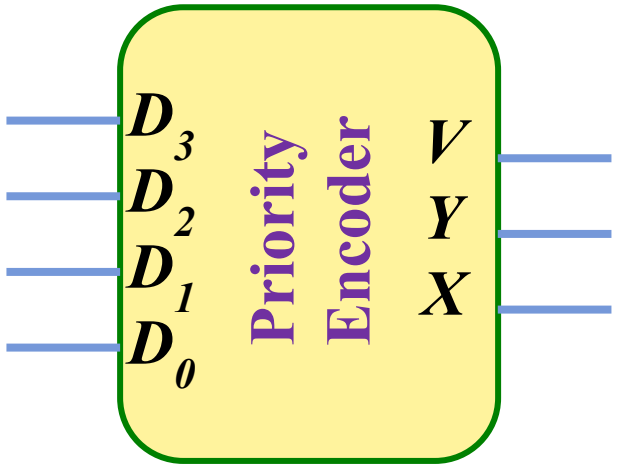
Truth Table of a Priority Encoder

Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
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

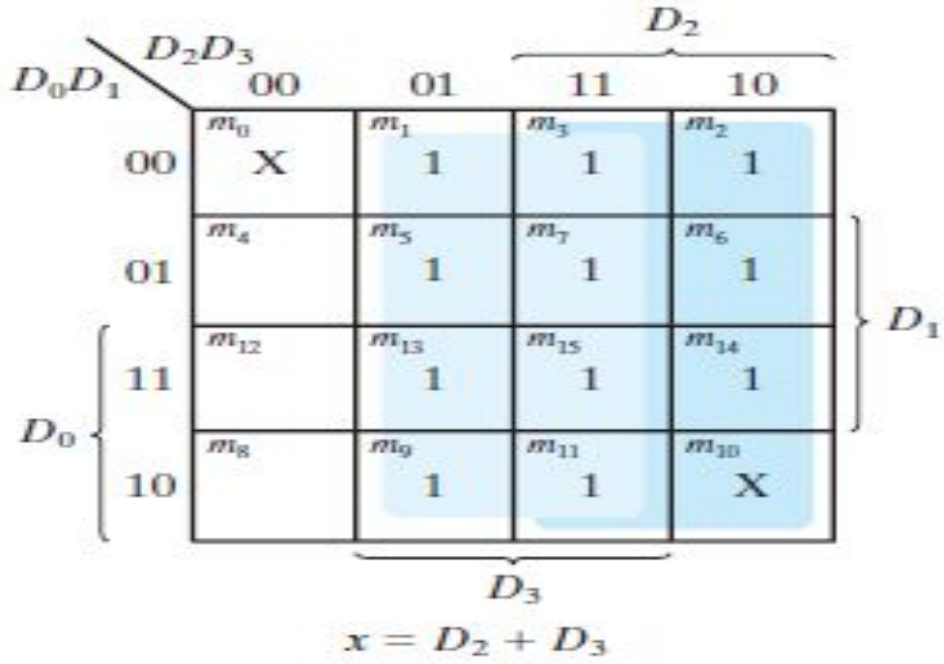
PRIORITY ENCODER

Truth Table of a Priority Encoder

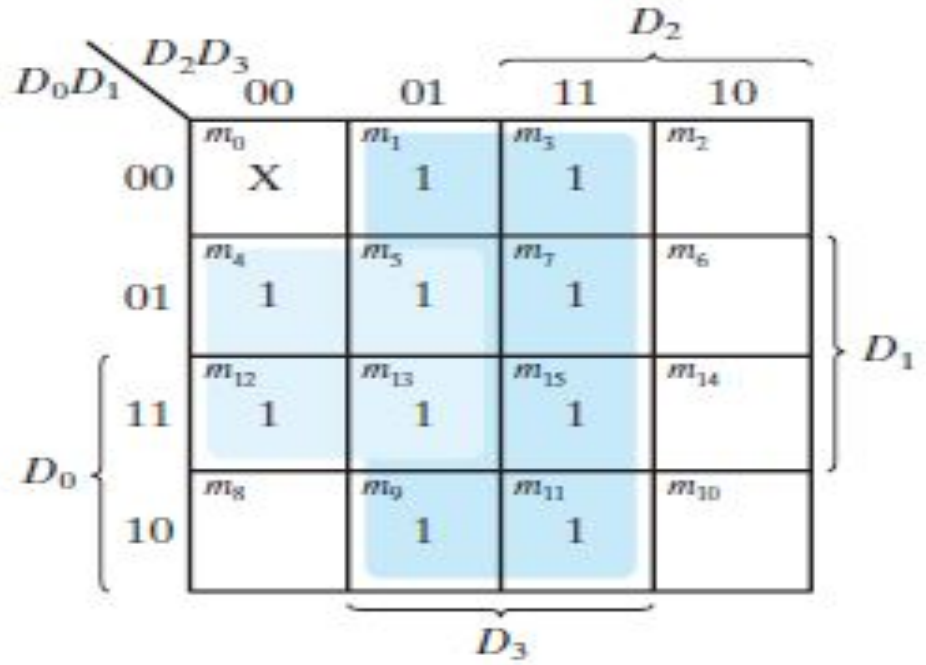
Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1



$$x = D_2 + D_3$$
$$y = D_3 + D_1 D_2'$$
$$V = D_0 + D_1 + D_2 + D_3$$



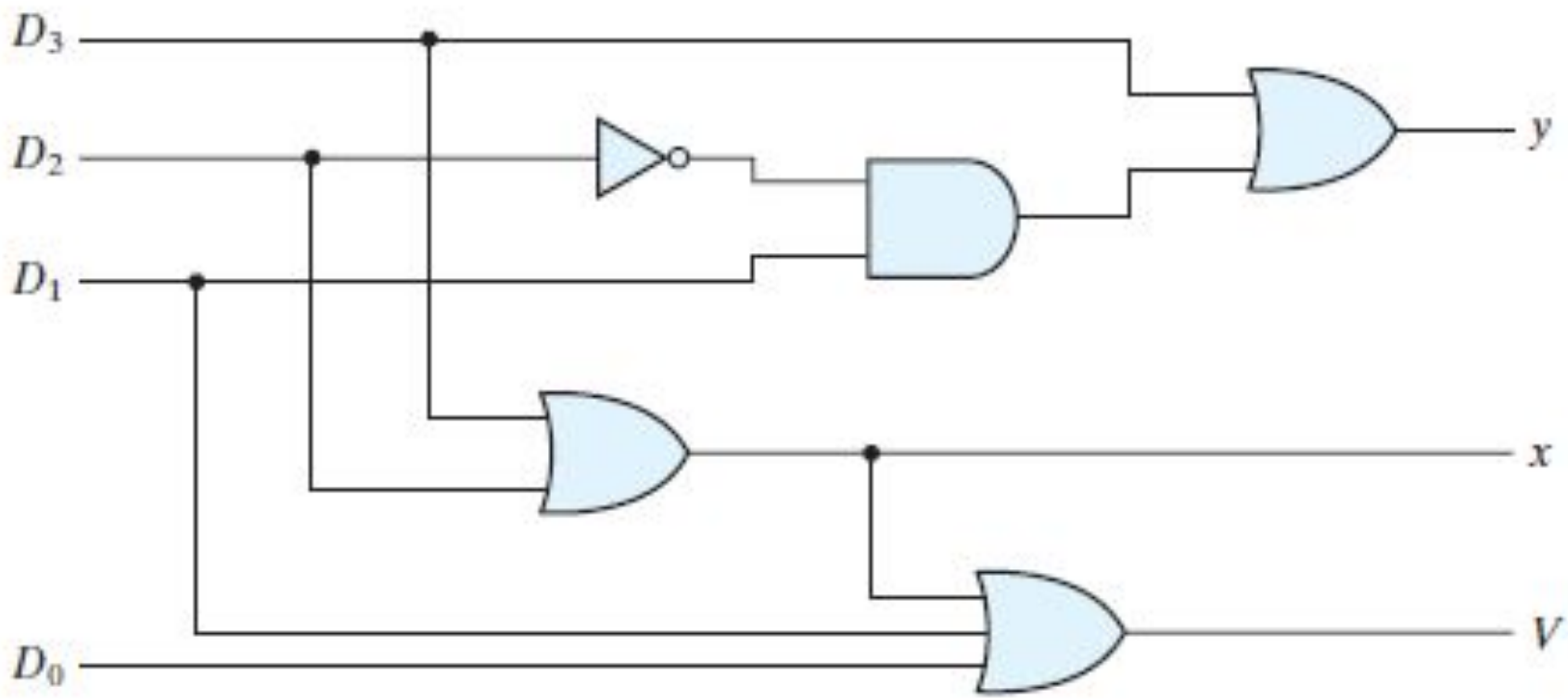
$$x = D_2 + D_3$$



$$y = D_3 + D_1 D'_2$$

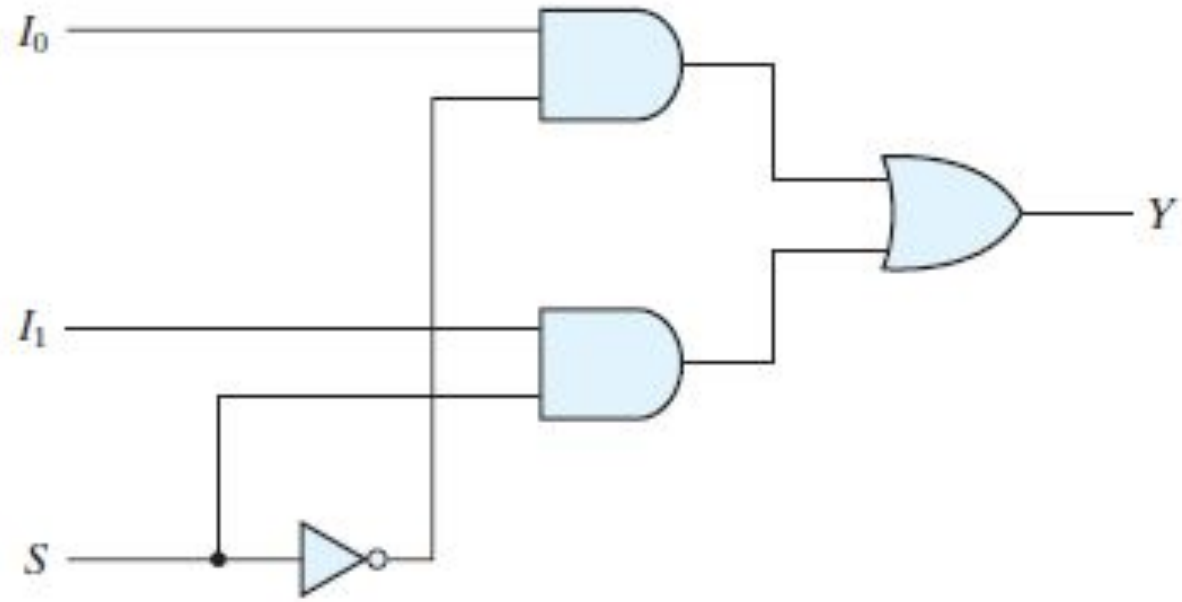
PRIORITY ENCODER

$$x = D_2 + D_3$$
$$y = D_3 + D_1 D_2'$$
$$V = D_0 + D_1 + D_2 + D_3$$



MULTIPLEXER

- A ***digital multiplexer*** is a combinational circuit that selects binary information from **one of many input lines** and directs it to a **single output line**
- There are n selection lines whose bit combinations determine which input is selected.
- Generally there are 2^n input lines, n selection lines
- Each of the two input lines **I_0 and I_1** applied to one input of an AND gate
- Selection line **S** decoded to select a particular AND gate



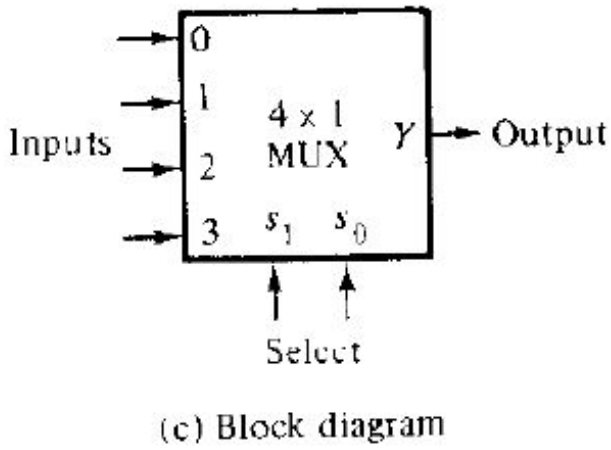
2 to 1 line multiplexer

4 TO 1 LINE MULTIPLEXER

- In this circuit, there are **4 input lines**, **2 selection lines**
- Each of the four input lines **I_0 to I_3** applied to one input of an AND gate
- Selection lines **S_1 and S_0** decoded to select a particular AND gate

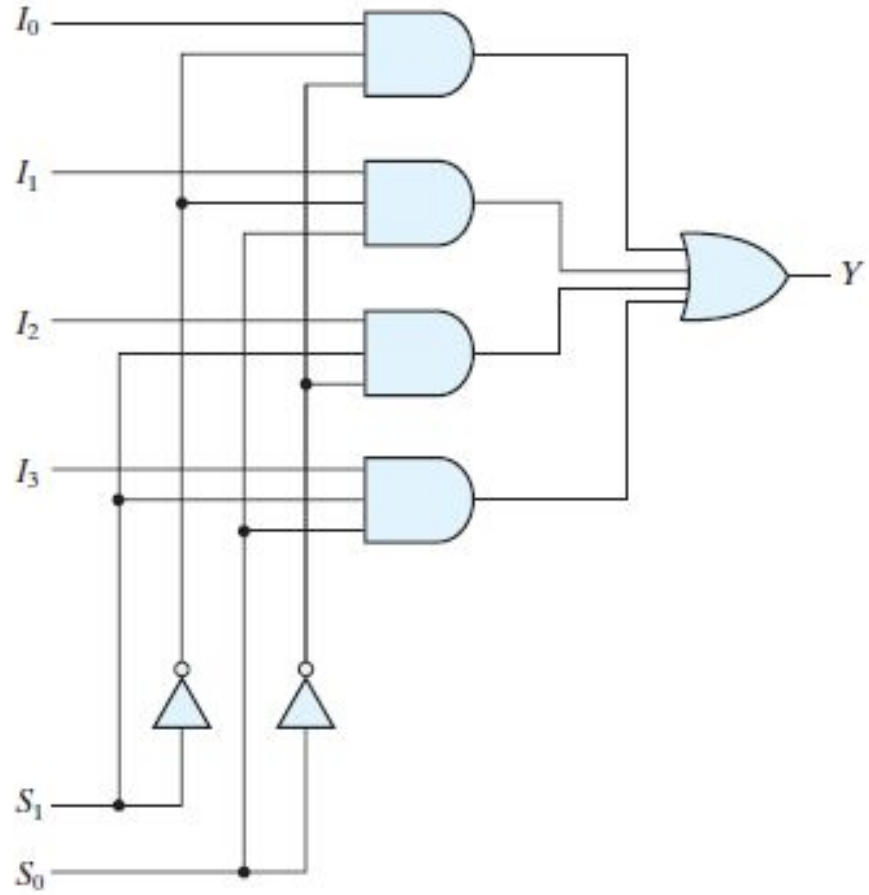
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Function table of a 4 to 1 line multiplexer



(c) Block diagram

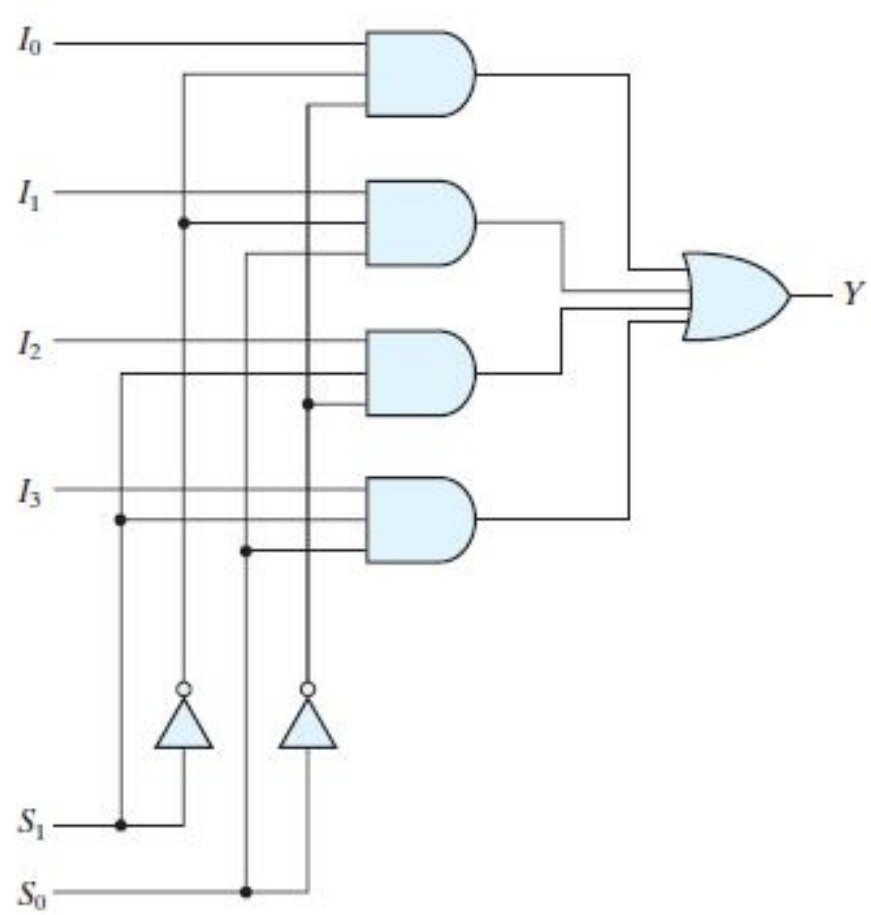
Block diagram of a 4 to 1 line multiplexer



4 to 1 line multiplexer

4 TO 1 LINE MULTIPLEXER

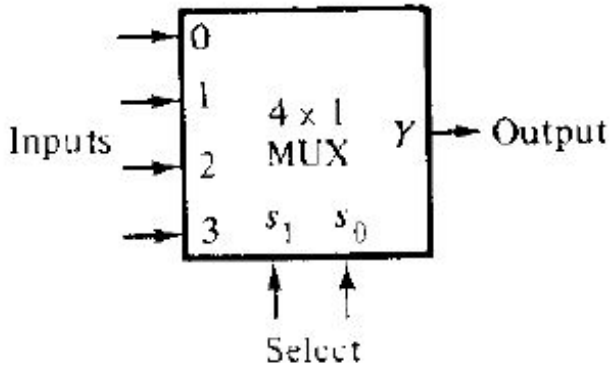
- When S_1S_0=10, AND gate associated with I_2 has **two of its input equal to 1** and the third input is connected to I_2.
- Other 3 AND gates has atleast one input equal to 0, making their output equal to 0
- The OR gate output is now equal to the value of I_2



4 to 1 line multiplexer

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

Function table of a 4 to 1 line multiplexer



(c) Block diagram

Block diagram of a 4 to 1 line multiplexer

FUNCTION IMPLEMENTATION USING MUX

- Any boolean function can be implemented by using a multiplexer
- A boolean function of ***n+1*** variables can be implemented by taking ***n*** variables to the ***selection lines*** of the multiplexer.
- The remaining variable is used as the input line for the multiplexer

Example- $F(A, B, C) = \sum(1, 3, 5, 6)$

Steps:

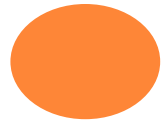
1. Choose the selector variables.
   Lets choose,
   - B, C as selector S_1 and S_0
   - A as input line

FUNCTION IMPLEMENTATION USING MUX

Example- $F(A, B, C) = \sum(1, 3, 5, 6)$

Steps:

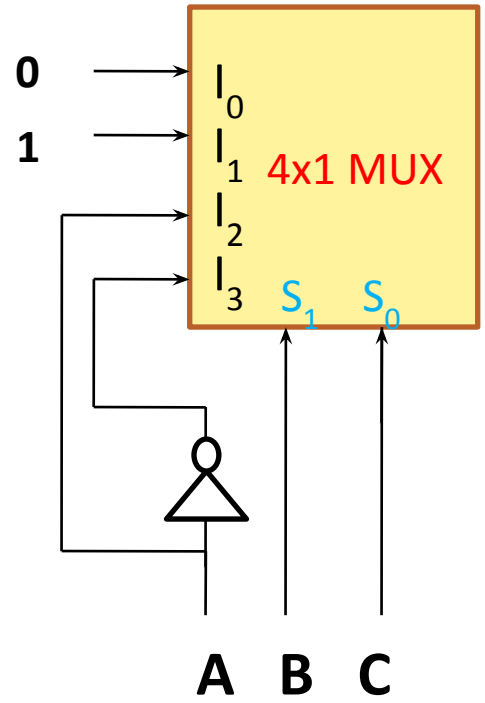
2. In the first row, list the name of the input lines of the multiplexers horizontally
3. In the second row, list the minterms where A is complemented
4. In the third row, list the minterms where A is uncomplemented
5. Circle the minterms for which the function outputs 1
6. Fourth row presents the multiplexer inputs
   - If the two minterms in a column are not circled, apply 0 to the corresponding multiplexer input
   - If the two minterms in a column are circled, apply 1 to the corresponding multiplexer input
   - If the bottom minterm is circled and the top is not circled, apply A to the corresponding multiplexer input
   - If the top minterm is circled and the bottom is not circled, apply A’ to the corresponding multiplexer input



FUNCTION IMPLEMENTATION USING MUX

$F(A, B, C) = \sum(1, 3, 5, 6)$

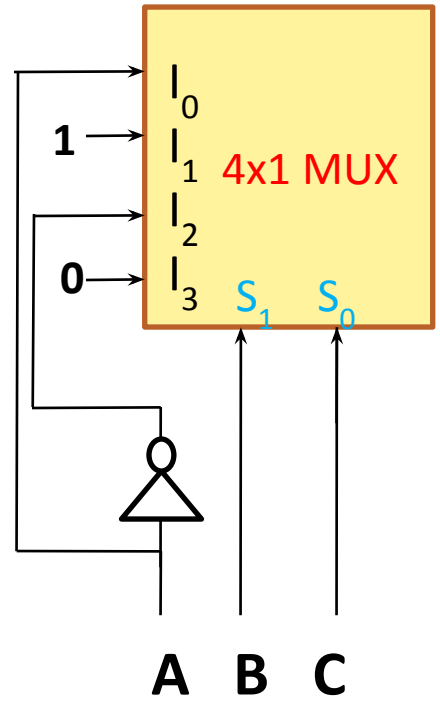
MUX input line	I_0	I_1	I_2	I_3
A’	0	1	2	3
A	4	5	6	7
Input values	0	1	A	A’



FUNCTION IMPLEMENTATION USING MUX

$F(A, B, C) = \sum(1, 2, 4, 5)$

MUX input line	I_0	I_1	I_2	I_3
A’	0	1	2	3
A	4	5	6	7
Input values	A	1	A’	0

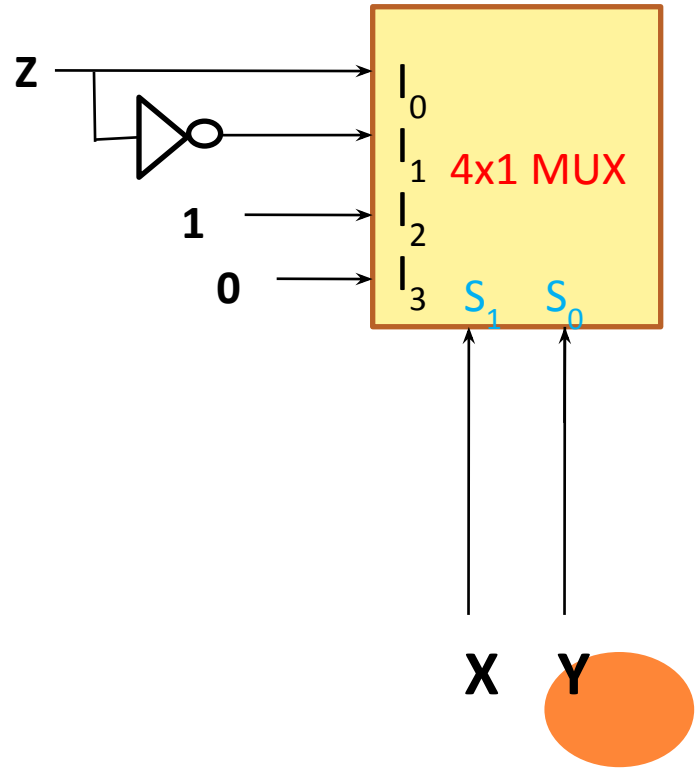


FUNCTION IMPLEMENTATION USING MUX

- It is not necessary to choose the leftmost variable in the ordered sequence of a variable list for the data inputs of the multiplexer. Any of the variable can be chosen to be the input
- F(A, B, C) = ∑(1, 2, 4, 5)

MUX input line	I_0	I_1	I_2	I_3
Z'	0	2	4	6
Z	1	3	5	7
Input values	Z	Z'	1	0

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



Self Study

- Why is a multiplexer called data selector?
- Logic diagram of a *quadruple 2 to 1 line multiplexer*.
- Implement the following function with a multiplexer

F(A, B, C,D) = ∑(0,1, 3, 4, 8,9,15)