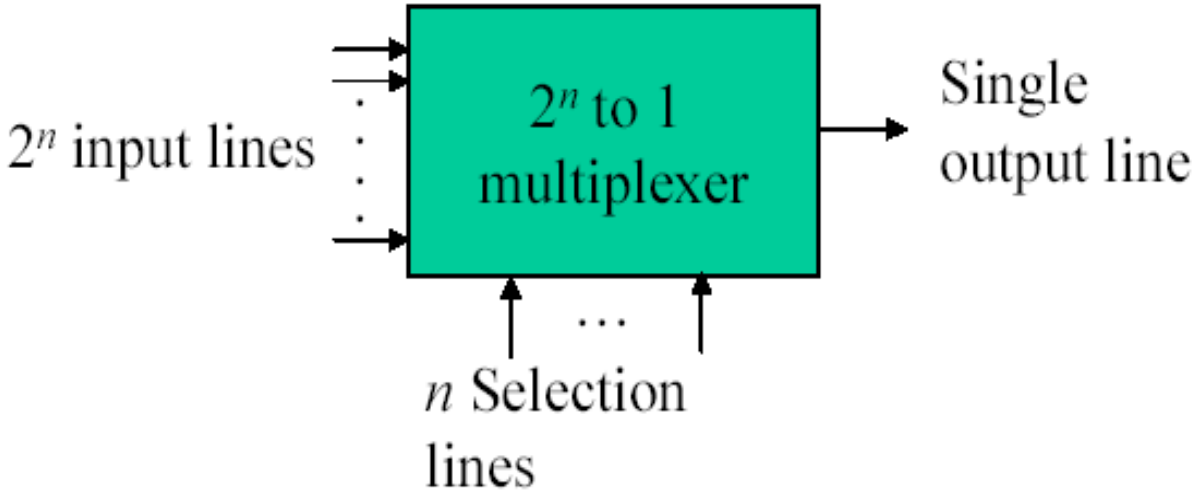
Multiplexer, Demultiplexer, Application of Multiplexer

By Engr. Rimsha

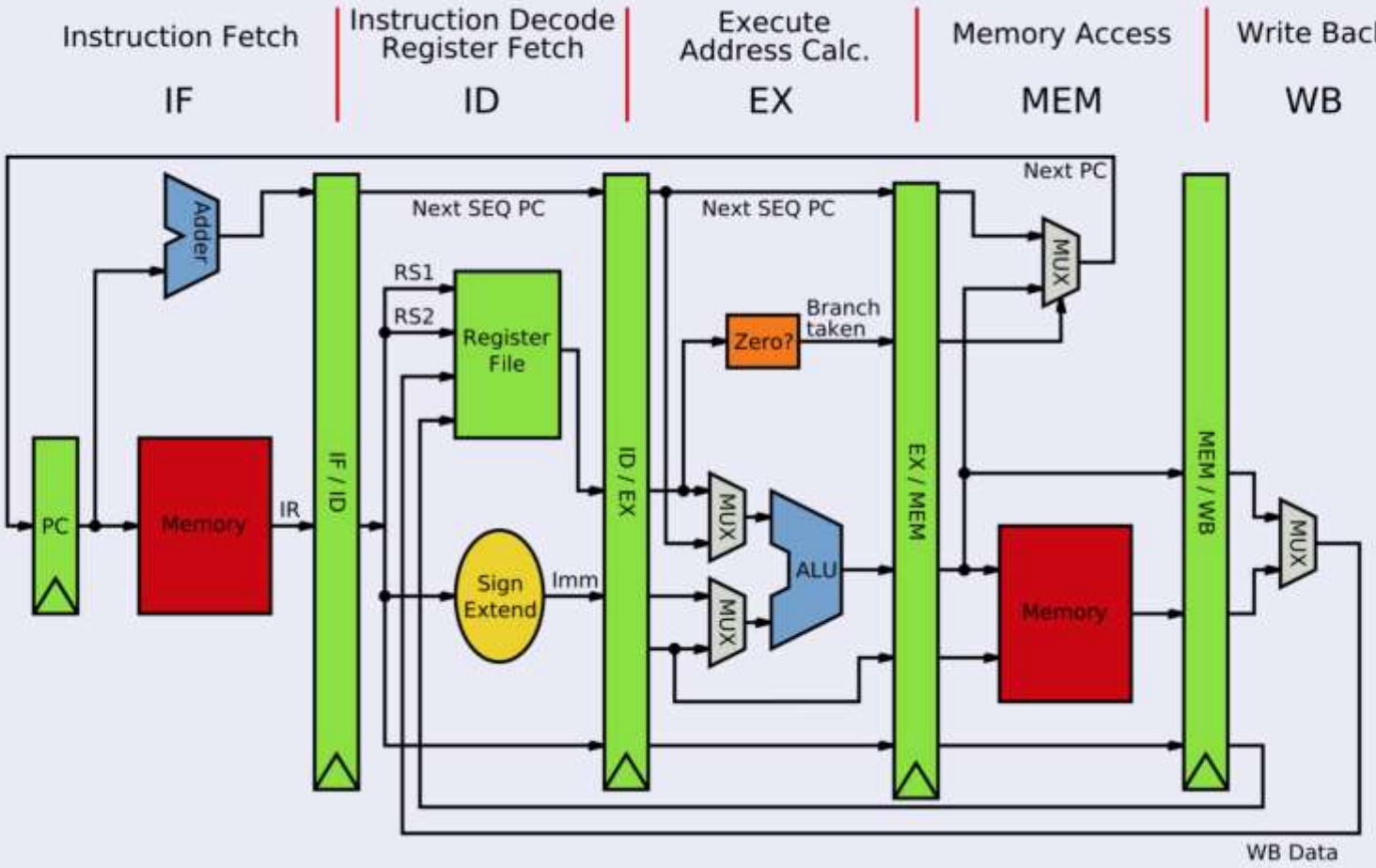
- Basic Concept of Multiplexer
  - Applications
    - Communication of digital voice over fiber using multiplexers and Demultiplexers
- Multiplexers
- Demultiplexers

Multiplexers

- A multiplexer is a combinational circuit that selects binary information from one of 2^n input lines and directs it to a single line. There are n selection lines.

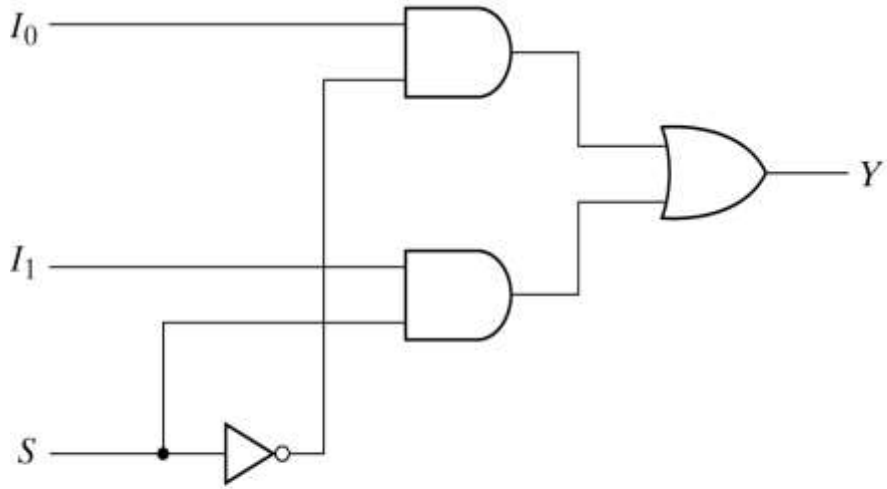


Microarchitecture

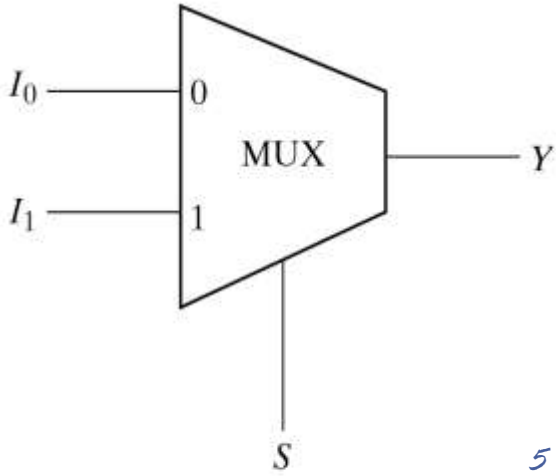


2-to-1-line Multiplexer

- 2-to-1-line multiplexer connects one of two 1-bit sources to a common destination.
- There are two data input lines, one output line and one selection line s. when s=0 the upper AND gate is enabled and and I_0 has path to the output. when s=1 the lower AND gate is enabled and and I_1 has path to the output



(a) Logic diagram

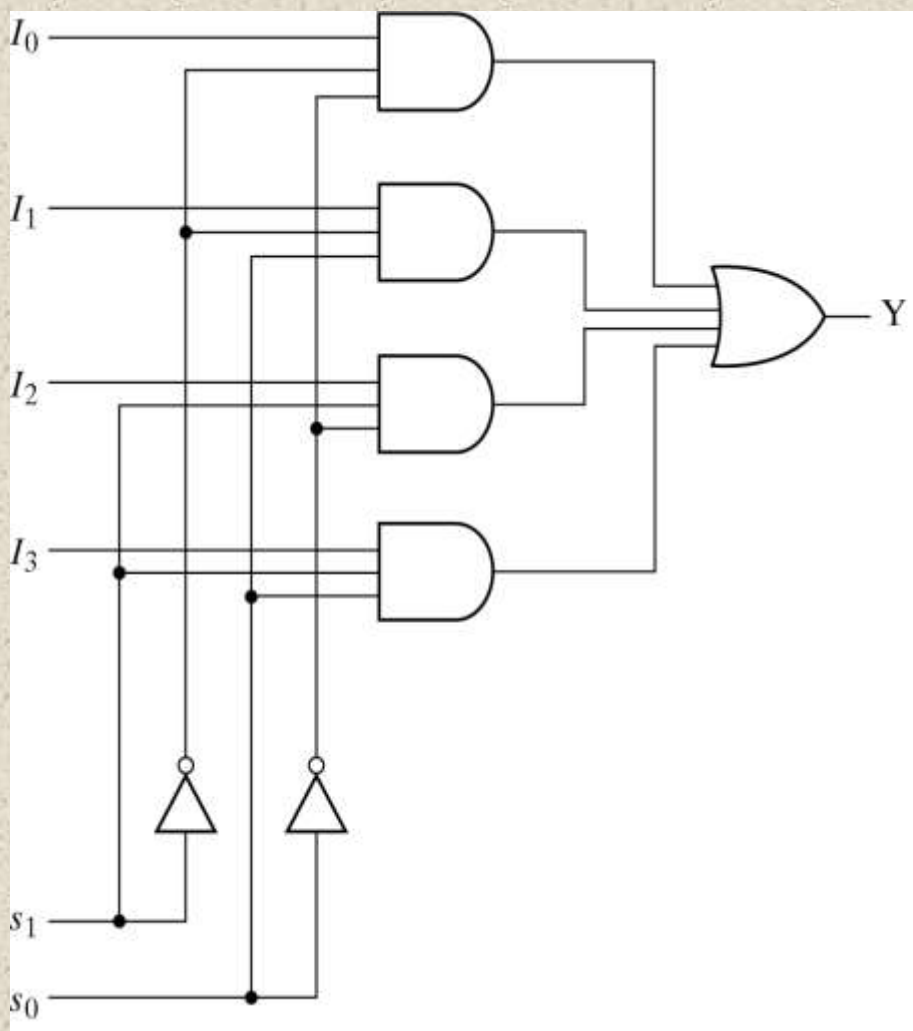


(b) Block diagram

4-to-1-line Multiplexer

- 4-to-1-line multiplexer connects one of four 1-bit sources to a common destination.
- There are four data input lines, one output line and two selection line s_1 and s_0. Selection lines s_1 and s_0 are decoded to select a particular AND gate
- The outputs of the AND gates are applied to a single OR gate that provides the 1-line output
- When s_1s_0=10. The AND gate associated with input I_2 has two of inputs equal to 1 and the third input I_2 connected to output of AND gate. The other three AND gates have at least one input equal to 0, which makes their output equal to 0. The OR gate output is now equal to value of I_2, providing a path from the selected input to the output

4-to-1-line Multiplexer: Logic Circuit



s_1	s_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

(b) Function table

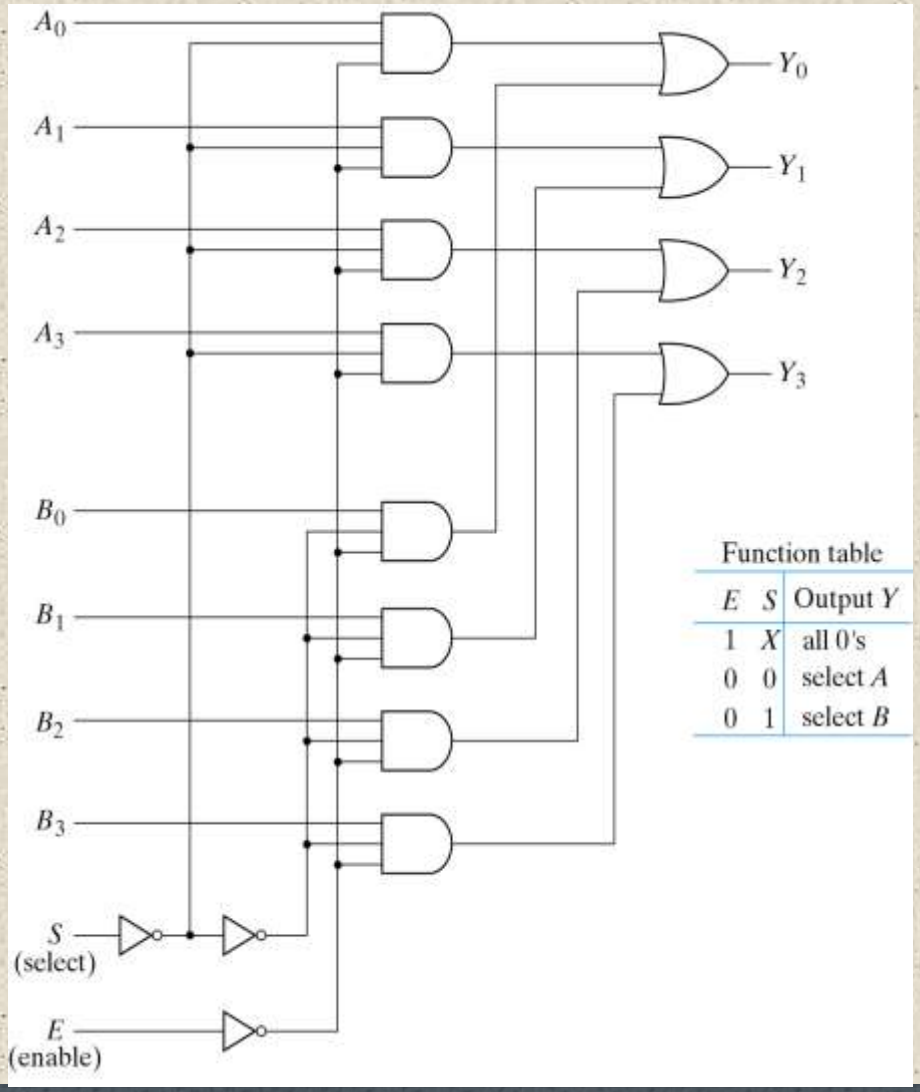
Multiplexer from Decoder

- The AND gates and Inverters in the multiplexer resemble a decoder circuit and they decode the selection input lines
- A 2^n-to-1 line multiplexer is constructed from an n-to-2^n decoder by adding to it 2^n input lines, one to each AND gate. The output of the AND gates are applied to a single OR gate
- The size of a multiplexer is specified by the number 2^n of its data input lines and the single output line
- The n selection lines are implied from the 2^n data lines
- As in decoders, multiplexers may have an enable input to control the operation of the unit

Quadruple 2-to-1-Line Multiplexer

- Multiplexer circuits can be combined with common selection inputs to provide multiple-bit selection logic
- Quadruple 2-to-1-line multiplexer has four multiplexers, each capable of selecting one of two input lines
- Output Y_0 can be selected to come from either input A_0 or B_0. Output Y_1 may have the value A_1 or B_1 and so on
- Input selection line S selects one of the lines in each of the four multiplexers. The enable input E must be active for normal operation
- The circuit contains four 2-to-1 line multiplexers and it selects one of two 4-bit sets of data lines
- The unit is enabled when E=0. Then if s=0, the four A inputs have a path to the four outputs, if s=1 the four B inputs are applied to the outputs.

Quadruple 2-to-1-Line Multiplexer

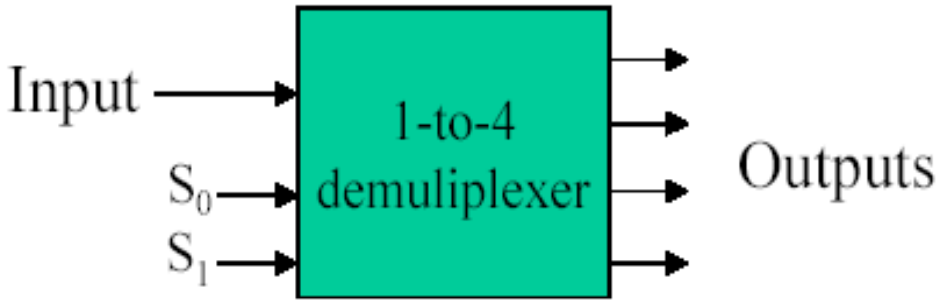


Function table

E	S	Output Y
1	X	all 0's
0	0	select A
0	1	select B

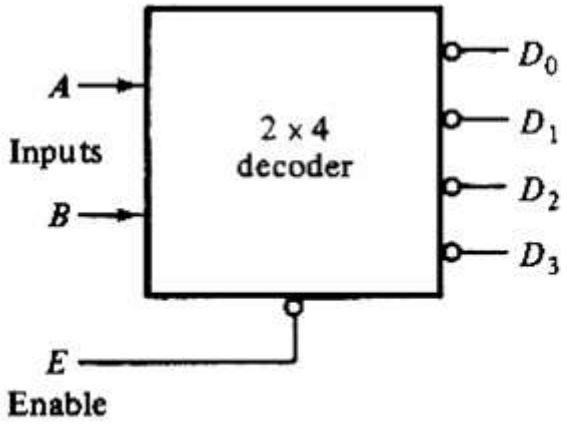
Demultiplexer

- A demultiplexer is a circuit that receives information from a single line and directs it to one of 2^n possible output lines.
- The selection of a specific output is controlled by the bit combination of n selection lines.

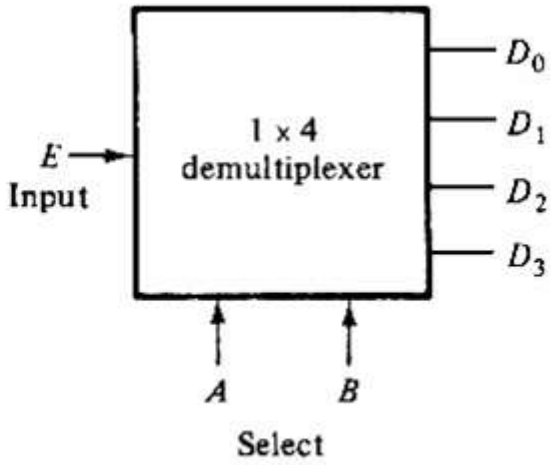


Demultiplexer

- A decoder with an enable input (fig 4-19) can function as demultiplexer (1-to-4-line demultiplexer)
  - E is taken as data input line and A and B are takes as selection inputs



(a) Decoder with enable

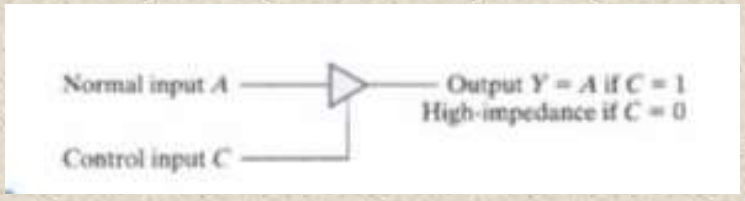


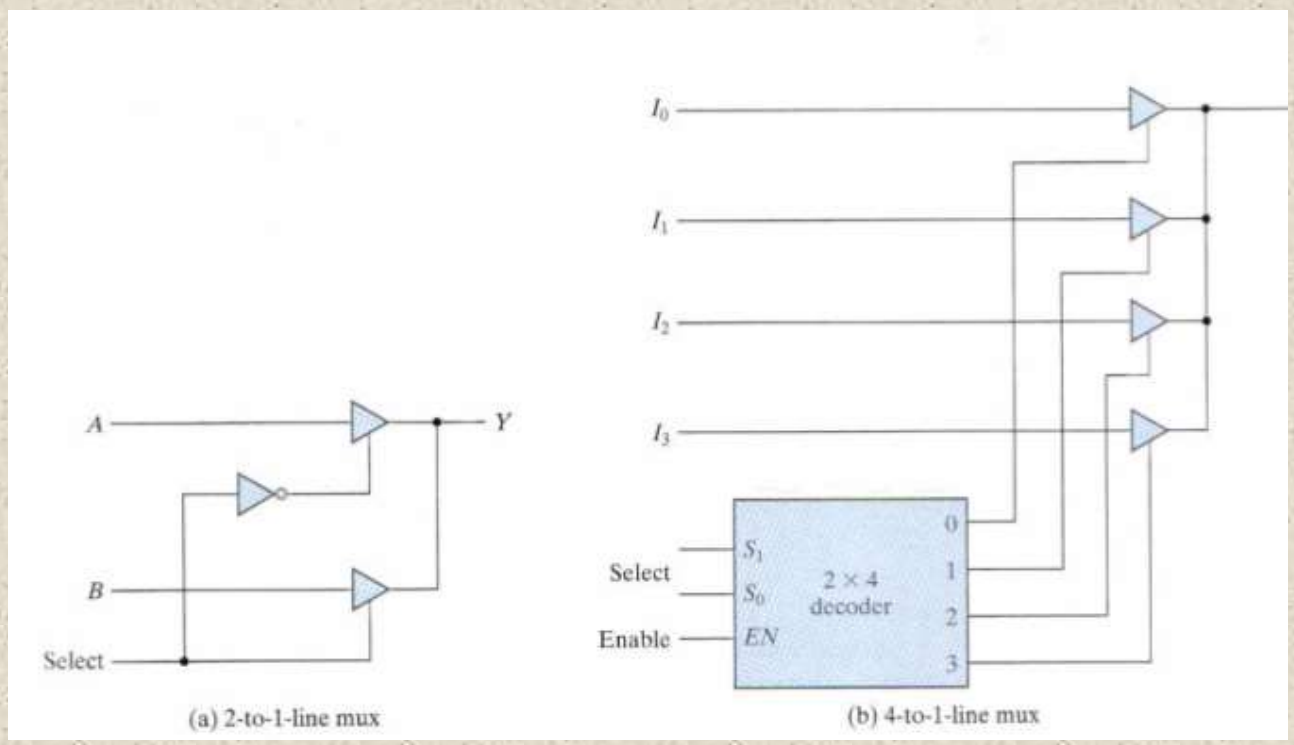
(b) Demultiplexer

Overview of Multiplexer and Demultiplexer

- Multiplexer
  - Function
  - Construction
  - Boolean Function Implementation
- Demultiplexer
  - Function
  - Construction

Tri-State Buffer





MUX Application

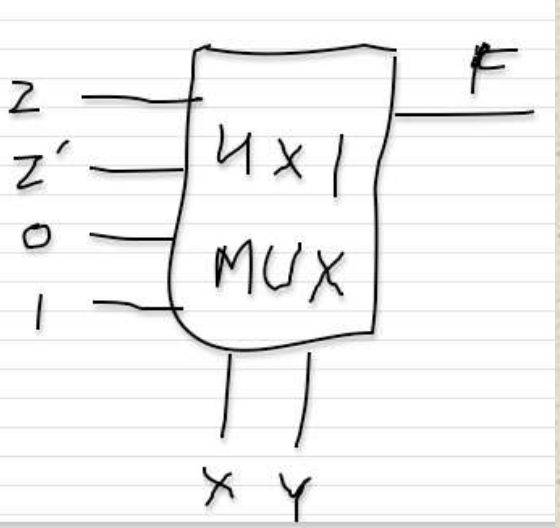
- Boolean Function Implementation using MUX

Implementation of Boolean Function with Multiplexers

F (X, Y, Z) = Σ (1, 2, 6, 7)

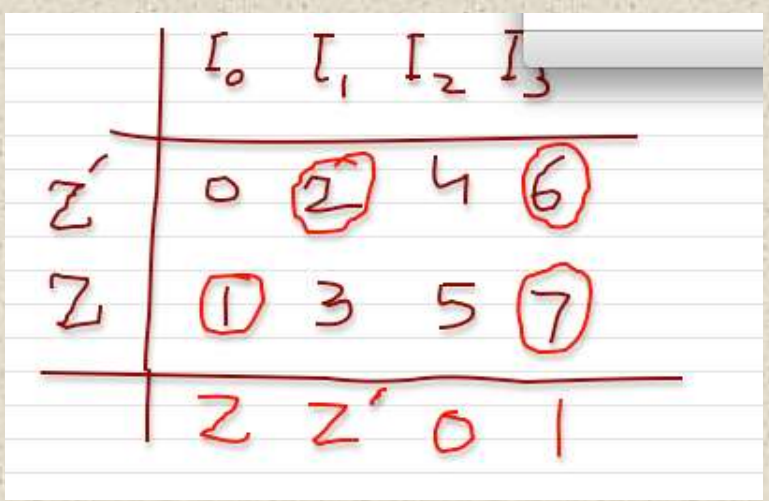
x	y	z	F	
0	0	0	0	$F = z$
0	0	1	1	
0	1	0	1	$F = z'$
0	1	1	0	
1	0	0	0	$F = 0$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	

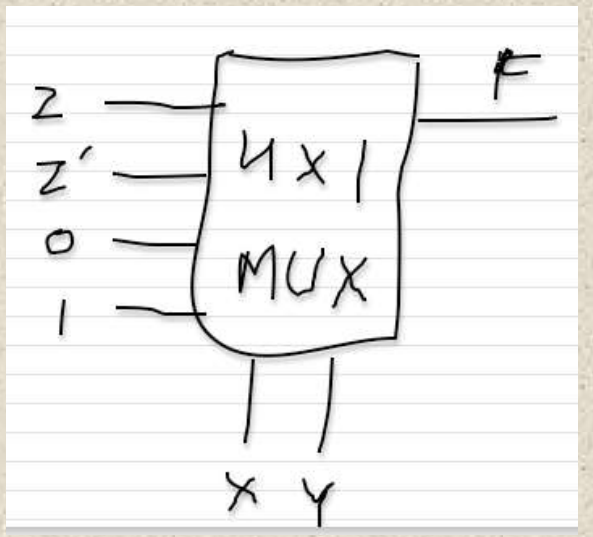
(a) Truth table



Implementation of Boolean Function with Multiplexers Alternative Method

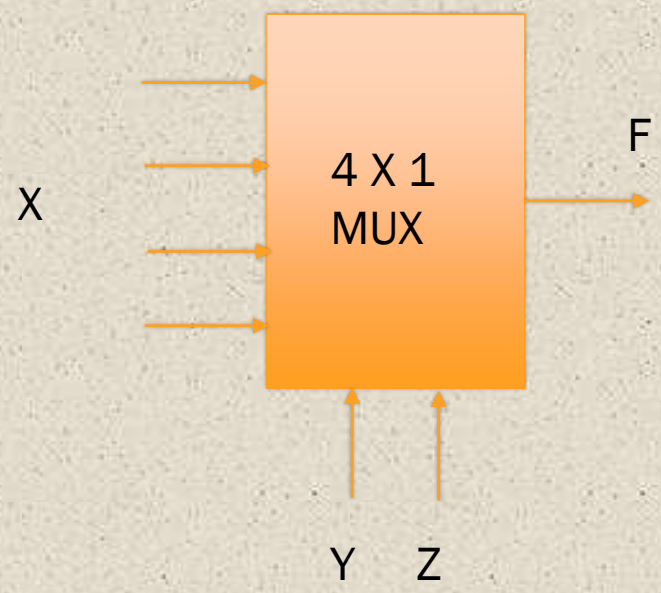
$F(X, Y, Z) = \Sigma(1, 2, 6, 7)$





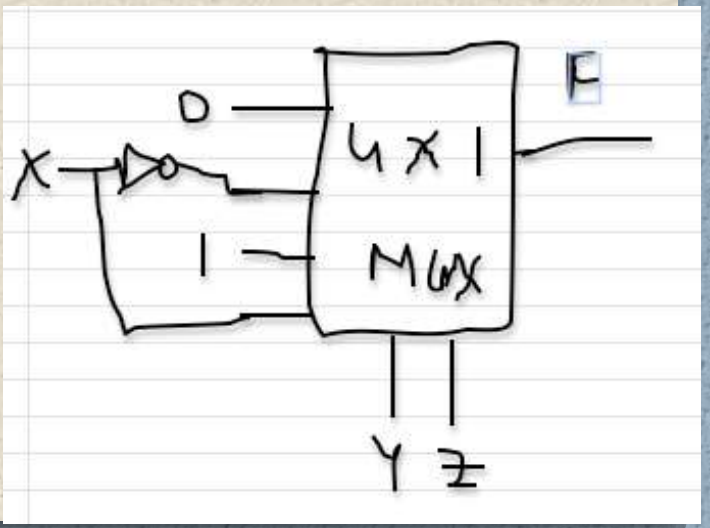
Changing the variables at Select Lines

$F(X, Y, Z) = \Sigma(1, 2, 6, 7)$

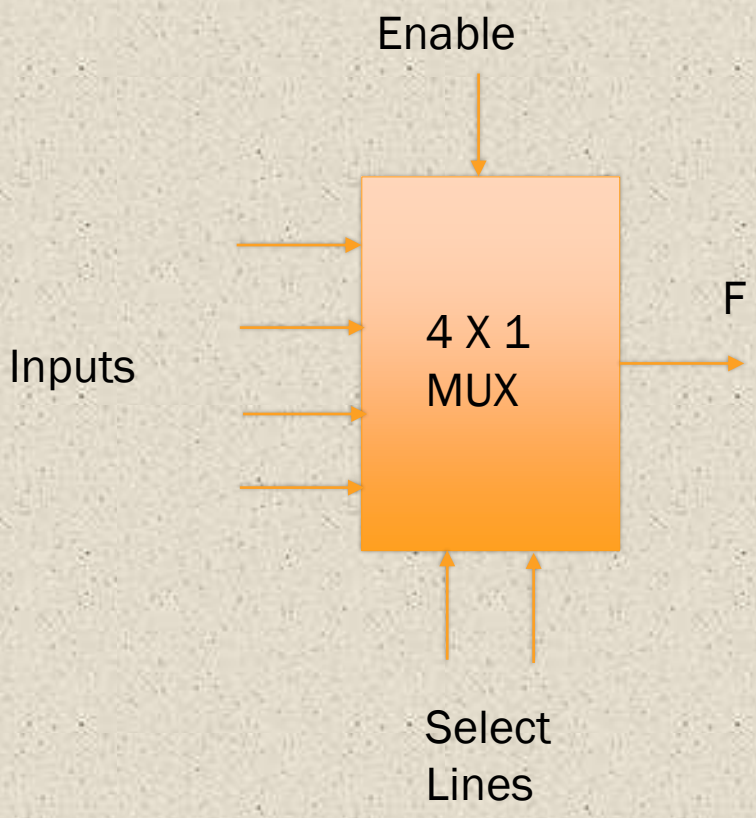


Implementation using MUX

	I_0	I_1	I_2	I_3
x'	0	1	2	3
x	4	5	6	7
	0	x'	1	x



Multiplexer with Enable Input



Circuit Implementation using MUX

$$F(A,B,C,D) = \Sigma(0,1,5,6,14,15)$$

If Three variables are Chosen on Select lines.
size of Multiplexer is 8 x 1

option 1

B, C, D on select lines

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\bar{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	A'	A'	0	0	0	A'	1	A

Circuit Implementation using MUX

$F(A,B,C,D) = \Sigma(0,1,5,6,14,15)$

If Three variables are chosen on select lines.
size of multiplexer is 8 x 1

option 2

A, B, C on select lines

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
D'	0	2	4	6	8	10	12	14
D	1	3	5	7	9	11	13	15
	1	0	D	D'	0	0	0	1

Circuit Implementation using MUX

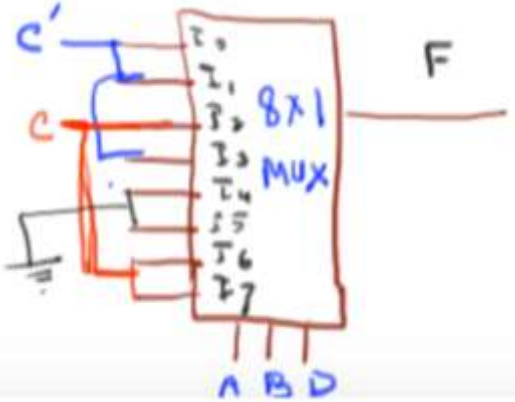
$F(A,B,C,D) = \Sigma(0,1,5,6,14,15)$

If Three variables are chosen on select lines.
size of Multiplexer is 8 x 1

option 3

A, B, D
on select lines

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
C′	0	1	4	5	8	9	12	13
C	2	3	6	7	10	11	14	15
	C′	C′	C	C′	0	0	C	C

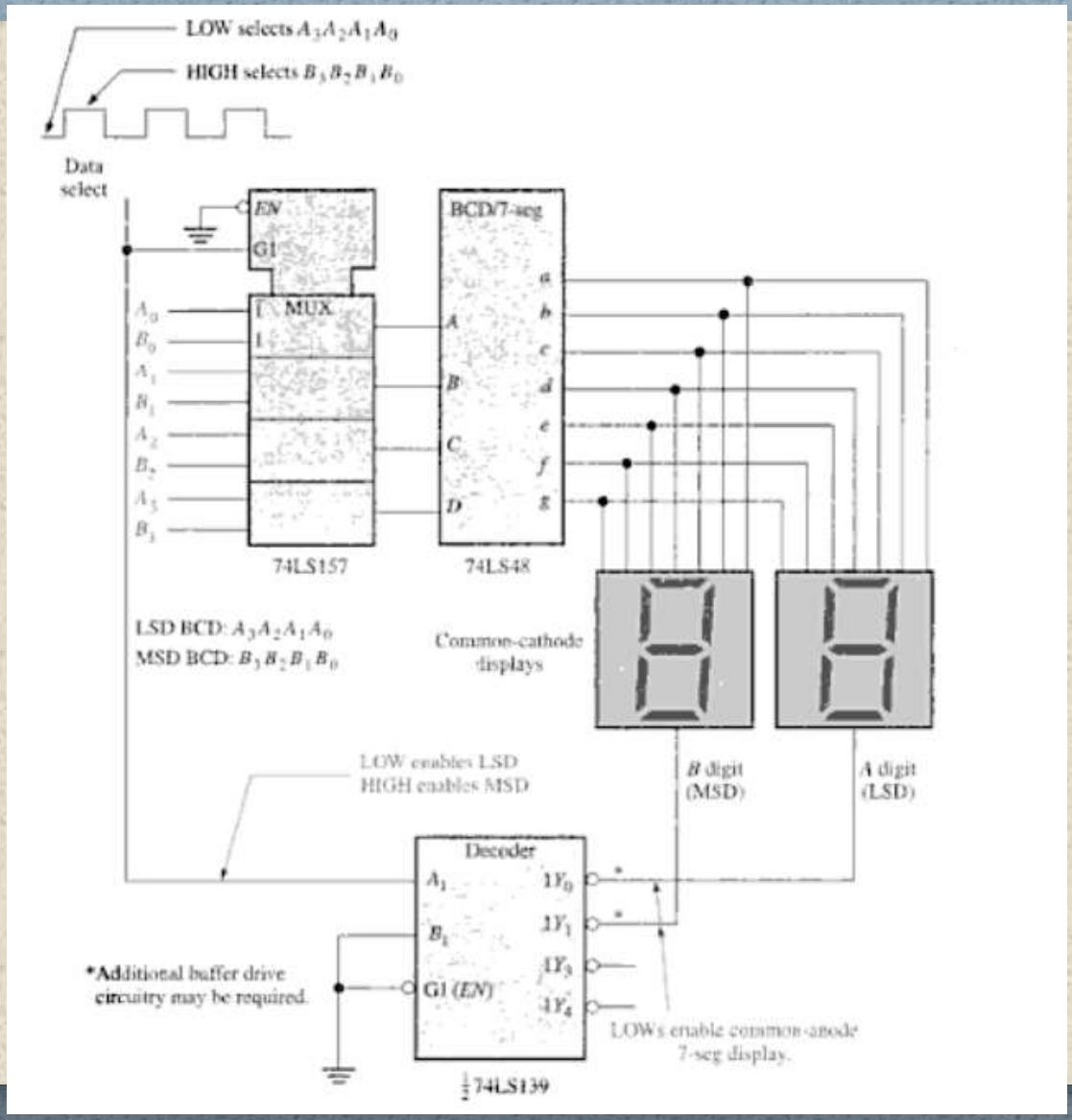


A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Practice Problems

- Writing of Boolean Function if circuit with MUX is given
- Choosing more than one variable at the input of MUX
- Implementation of Multiple outputs using appropriate size of multiplexers

Application of Multiplexer



End of Lecture