THE UNIVERSITY OF TEXAS AT DALLAS

CS4341 Digital Logic & Computer Design

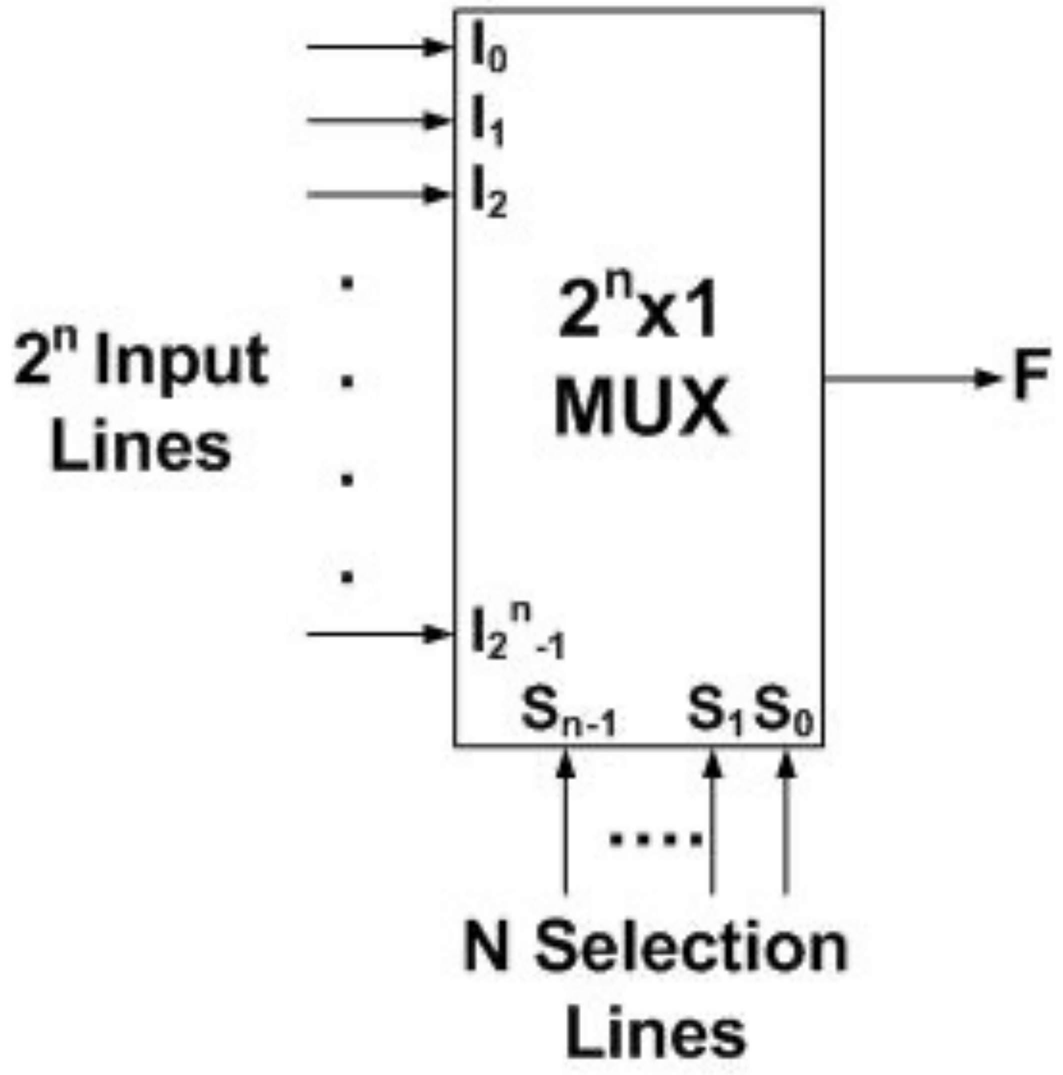
Lecture Notes 10

Omar Hamdy
Assistant Professor
Department of Computer Science

Multiplexers

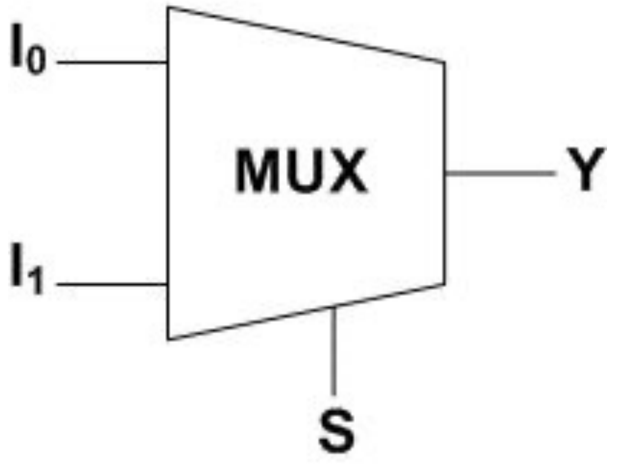
- ➢ Combinational logic is often grouped into larger building blocks to build more complex systems such as the priority circuit and the 7-segment display decoder.
- ➢ Two famous combinational circuits are multiplexers and decoders.
- ➢ Multiplexers (MUX) choose an output from among several possible inputs (usually 2^n) based on the value of a select signal.
- ➢ In other words, a 2^n x 1 **multiplexer (MUX)** is a device that selects binary information from one of 2^n input terminals and routes these data to a single output line

Multiplexers



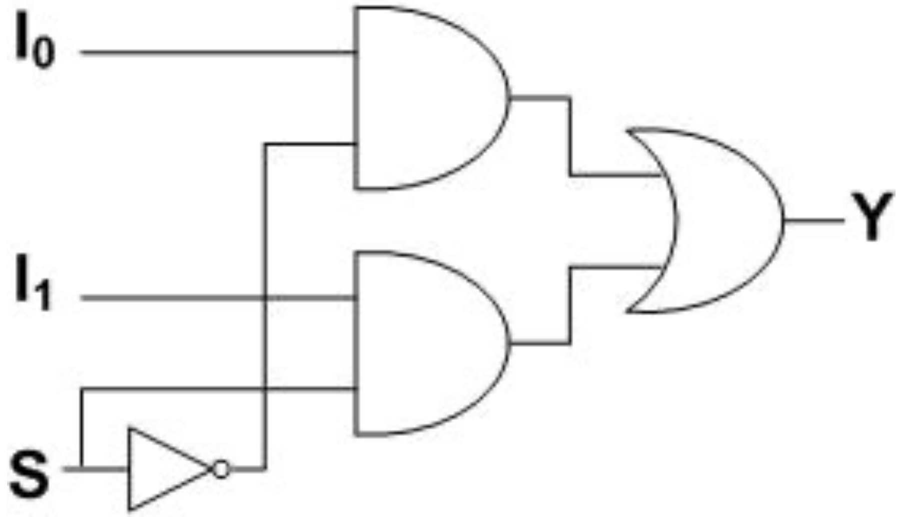
- A 2^n x 1 multiplexer requires n selection lines labeled as S_{n-1}, S_{n-2}, ..., S_1, S_0 to select any one of the 2^n inputs, designated by I_0, I_1, I_2, ..., I_{2^n-1}.
- The bit combination of the selection lines determines an n-bit binary number whose decimal equivalent corresponds to the subscript (**address**) of the selected input terminal.
- S_{n-1}, S_0 represent the msb and lsb respectively

2x1 Multiplexers Internal Design

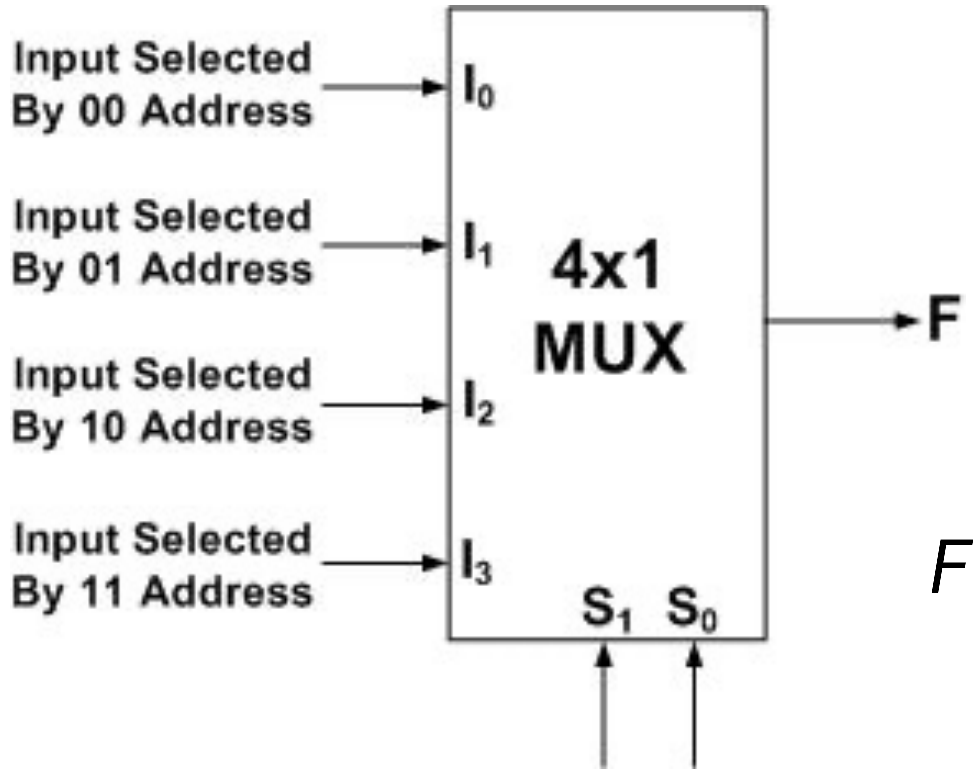


S	F
0	I_0
1	I_1

$F = s'I_0 + sI_1$



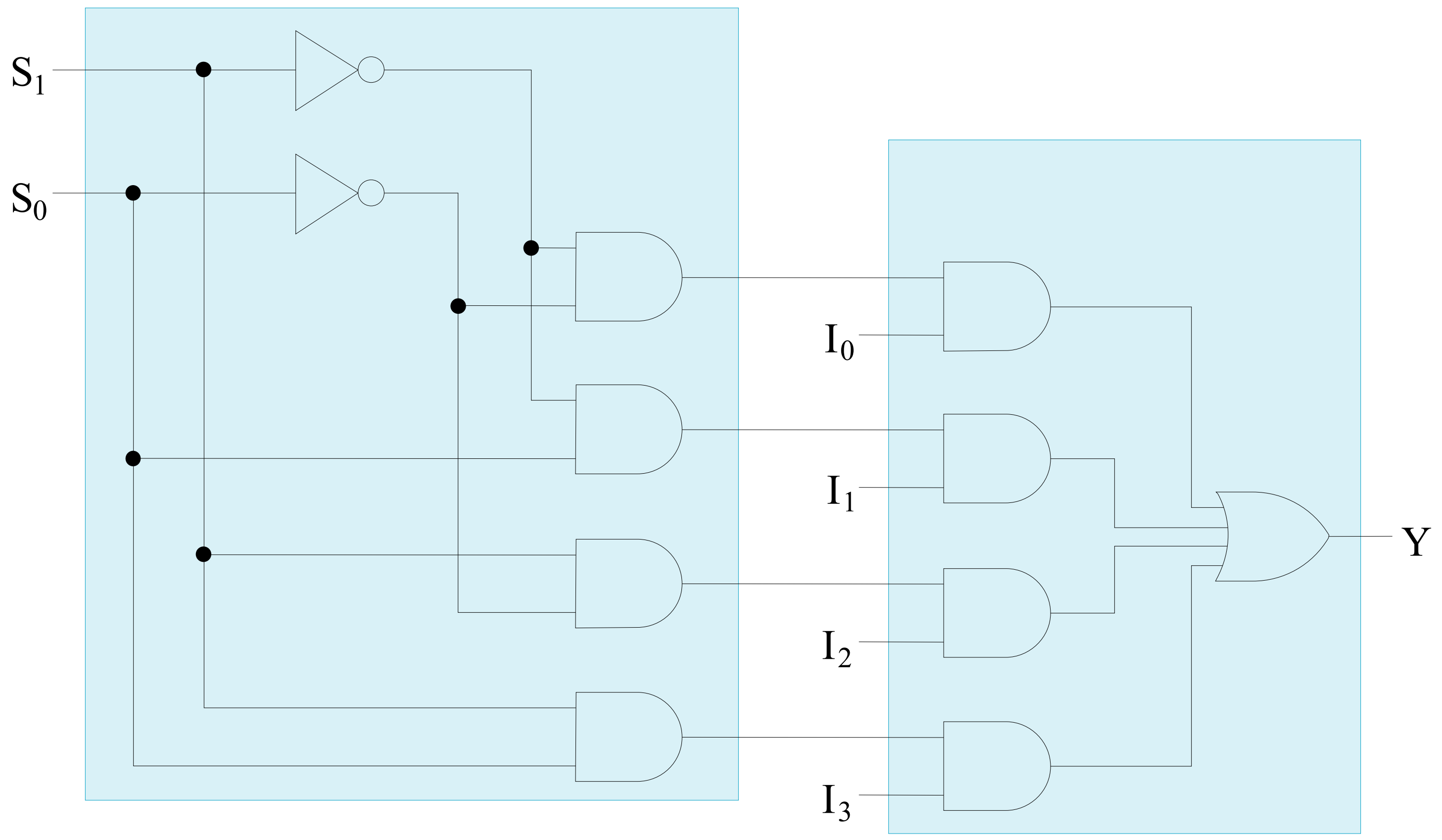
4x1 Multiplexers


Input Selected By 00 Address
Input Selected By 01 Address
Input Selected By 10 Address
Input Selected By 11 Address
I0
I1
I2
I3
4x1 MUX
F
S1
S0

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

$$F = s'_1 s'_0 I_0 + s'_1 s_0 I_1 + s_1 s'_0 I_2 + s_1 s_0 I_3$$

4x1 Multiplexers Circuit



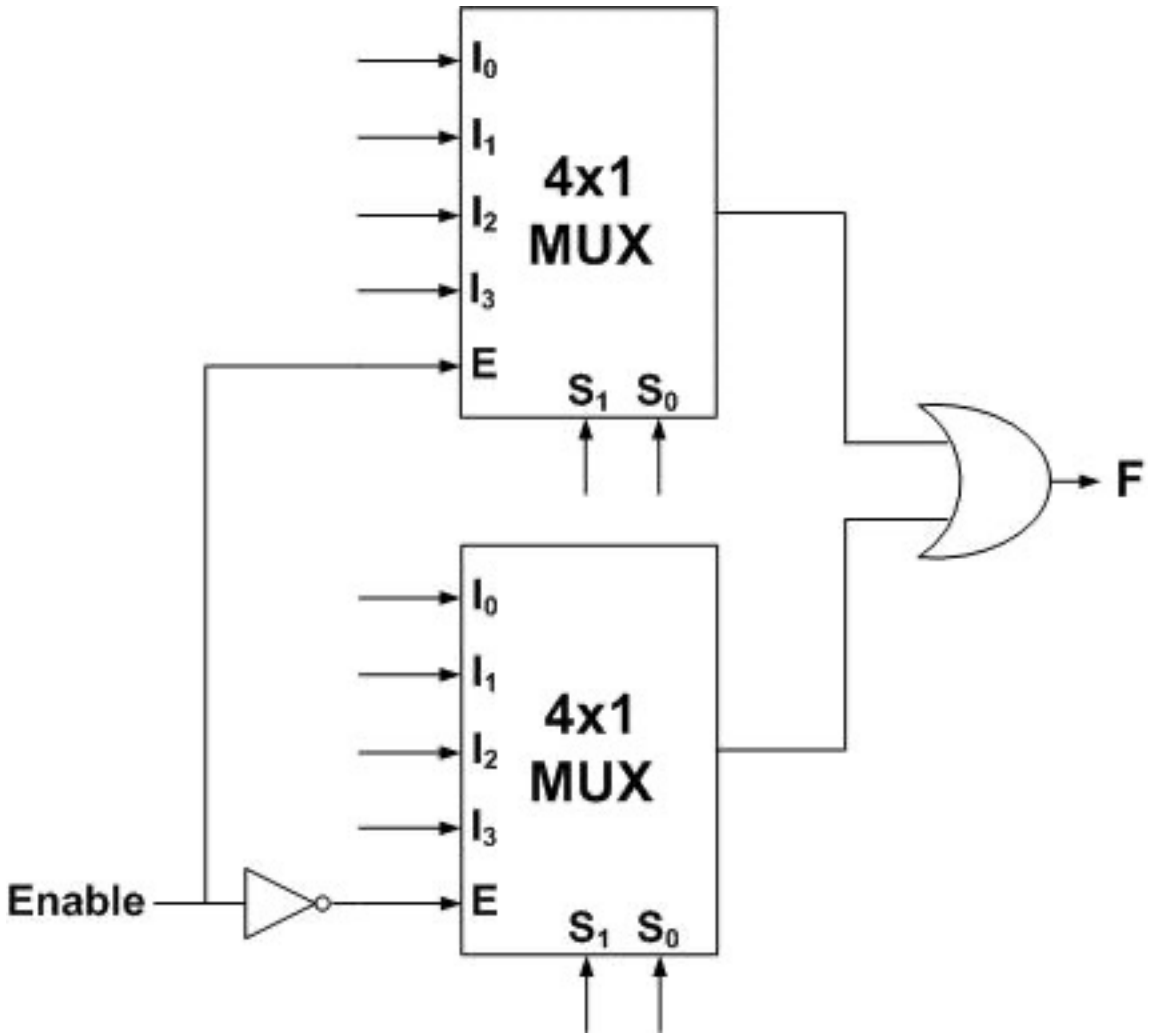
MUX Function

- In general, the output F of a 2^n x 1 multiplexer can be expressed as:

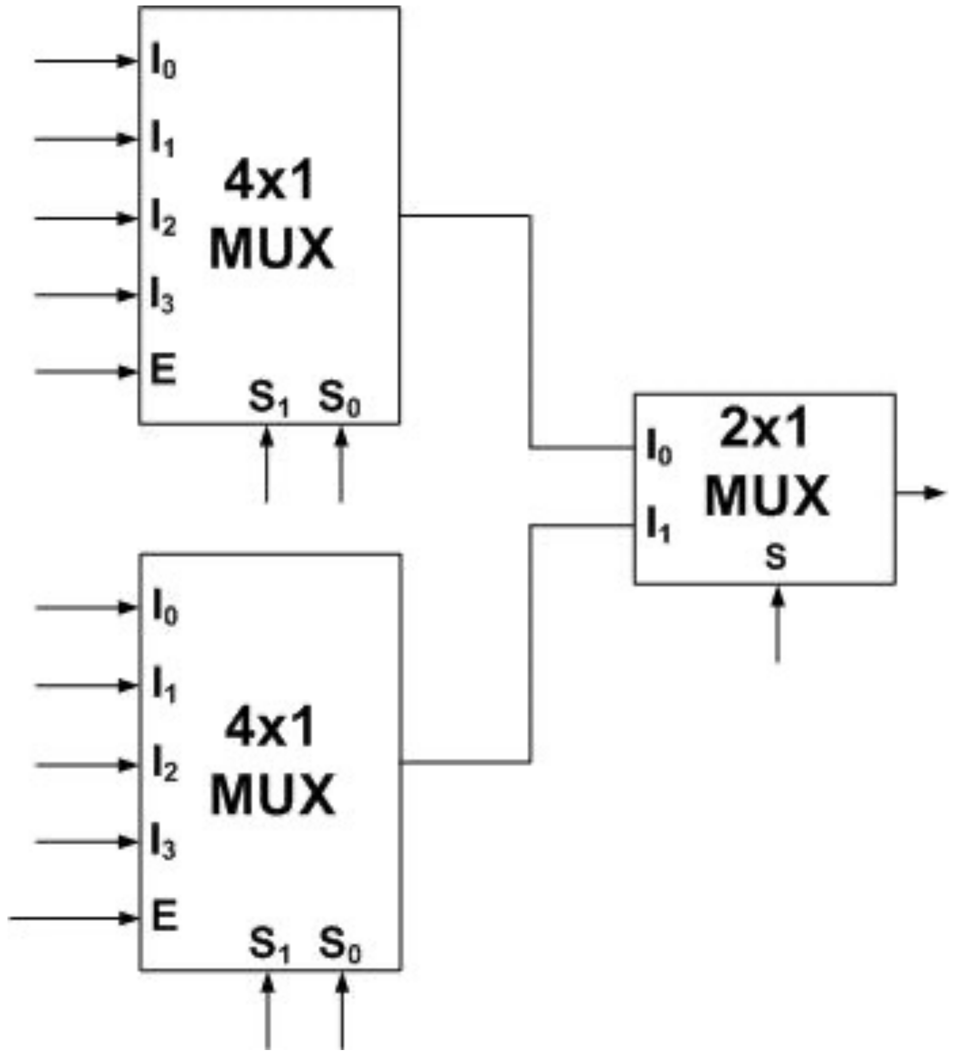
$$\sum_{k=0}^{2^n-1} m_k I_k$$

- Usually, multiplexers have additional input called enable (***E***) to control the operation of the multiplexer where 0 usually means enabled.
- Two of more multiplexers can be combined to form a multiplexer with a larger number of inputs
- Two basic ways to build multiplexer trees:
  - Use the enable line *E* and an OR gate
  - Use an additional multiplexer to select the multiplexer desired

MUX Tree Using Enable *E*



MUX Tree Using Multiplexers

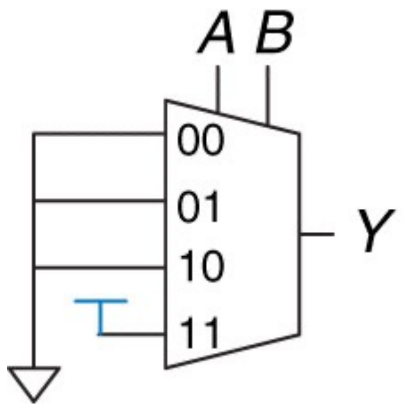


Multiplexer Applications

- Multiplexers can be used as lookup tables to perform logic functions. This can allow MUX to output the same behavior of a desired gate or circuit.
- In general, a 2^N-input multiplexer can be **programmed** to perform any N-input logic function by applying 0's and 1's to the appropriate data inputs
- Example, use 4x1 MUX as an AND gate
- The multiplexer data inputs are connected to 0 or 1 according to the corresponding row of the truth table.

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$Y = AB$



Combining Rows from Truth Tables

- In any truth table, if we combine two rows together, the right most variable can be eliminated and we can still express the function output value of the combined rows as either 0, 1, or the right most variable that was eliminated.

A	B	F
0	0	0
0	1	0
0	0	0
0	1	1
1	0	1
1	1	0
1	0	1
1	1	1

A	F
0	0

A	F
0	B

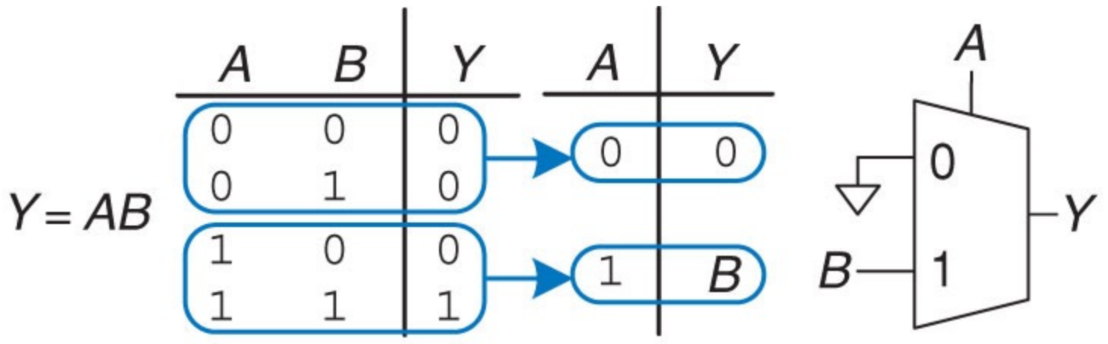
A	F
1	B’

A	F
1	1

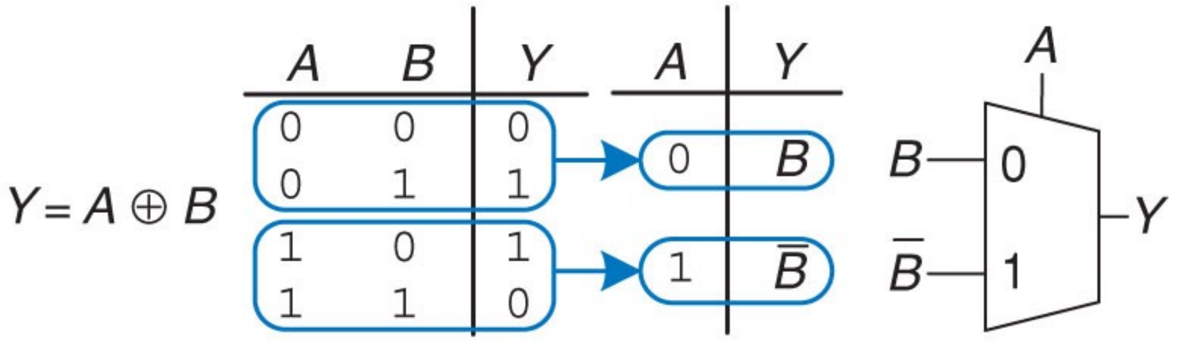
Reducing MUX Size

- Multiplexer size can be cut in half, using only a 2^{N-1} input multiplexer to perform any N-input logic function if one of the control signals is used as an input to the multiplexer.
- MUX data inputs are then going to be one of the signal literals, as well as 0's and 1's
- By combining pairs of rows in the truth table, the rightmost input variable is eliminated.
- Output of the combined rows will be 0, 1 or the eliminated input variable.
- Use the reduced MUX to design the new truth table

Reducing MUX Size - Examples



(a)



(b)

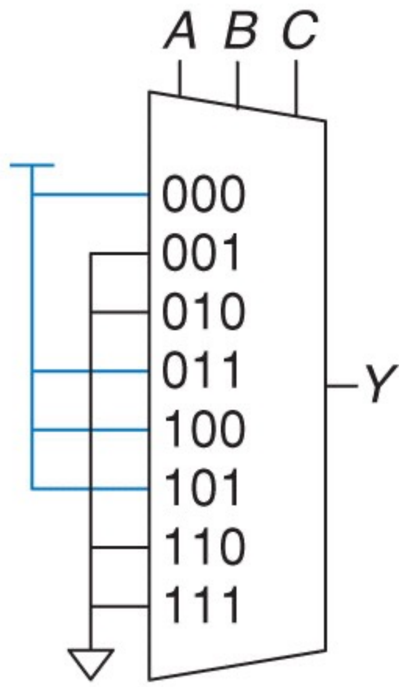
Example: Design Using MUX

- Design the Boolean function F = AB’ + B’C’ + A’BC using 8x1 mux

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

$$Y = A\overline{B} + \overline{B}\,\overline{C} + \overline{A}BC$$

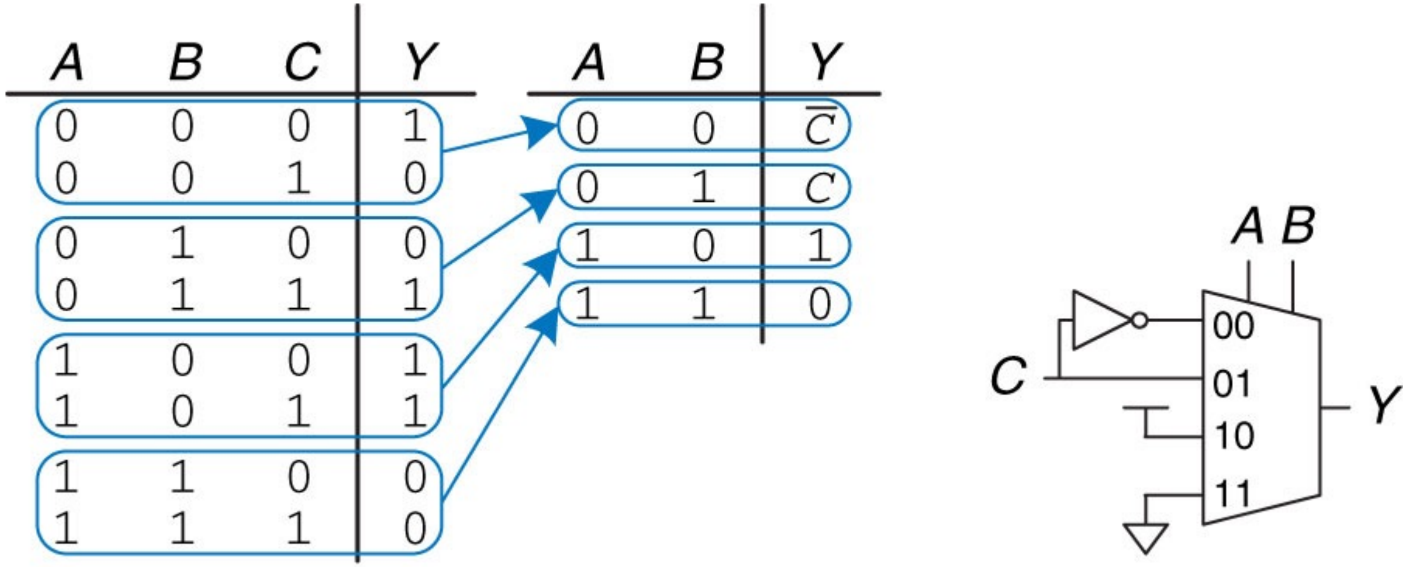
(a)



(b)

Example: Design Using MUX

- Design the Boolean function F = AB’ + B’C’ + A’BC using 4x1 mux



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

(a)

A	B	Y
0	0	$\overline{C}$
0	1	C
1	0	1
1	1	0

(b)

(c)

To Do List

- ➢Review lecture notes
- ➢Study chapter 2
- ➢Start studying for your exam 1