CALIFORNIA STATE UNIVERSITY FULLERTON™

EGEC 281: Designing with VHDL

Fall 2024

Lecture 9: Multiplexer

Rakesh Mahto, Ph.D.
Office: *E 314, California State University, Fullerton*
Office Hours: *Monday and Wednesday 2:00 - 3:30 pm*
Or by appointment
Office Hour Zoom Meeting ID: *894 4126 5483*
Email: *ramahto@fullerton.edu*
Phone No: *657-278-7274*

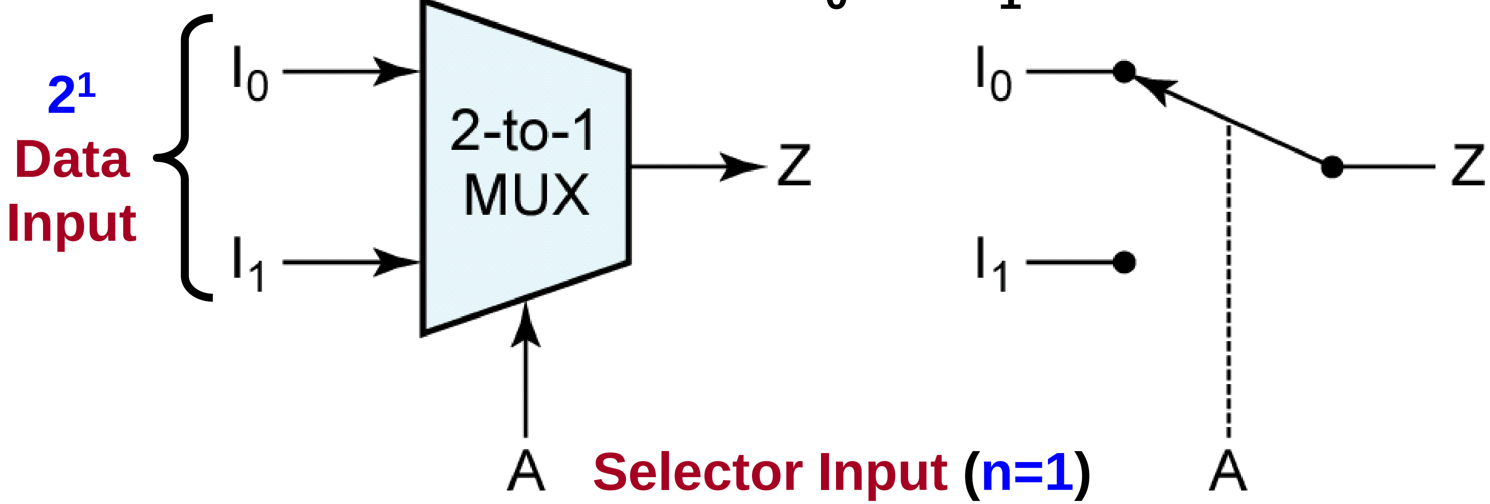
Multiplexers

Multiplexers (MUXs)

Used to direct one of 2^n data inputs to a single output.

- **n select lines needed to select one of 2^n outputs**
- **MUX also known as data selector**

$$Z = A'\,I_0 + A\,I_1$$



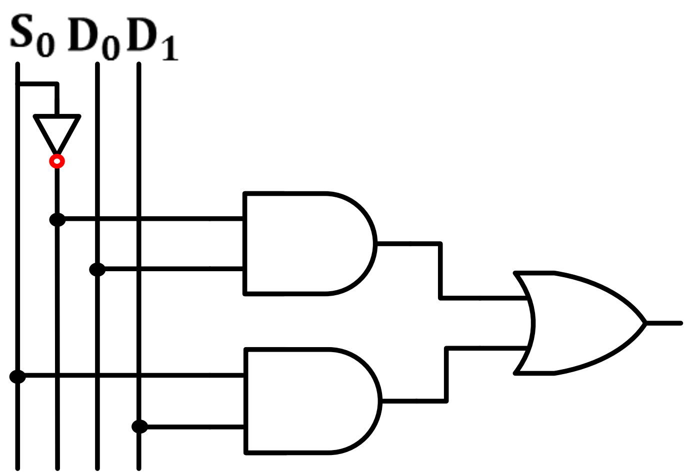
2-to-1 MUX

2 to 1 Multiplexers are use to choose between 2 inputs (D_0, D_1) based on the selector input (S_0). This can be thought of like the following truth function

S_0	D_1	D_0	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

S_0 \ D_1D_0	00	01	11	10
0	0	1	1	0
1	0	0	1	1

$$F = D_0S_0' + D_1S_0$$

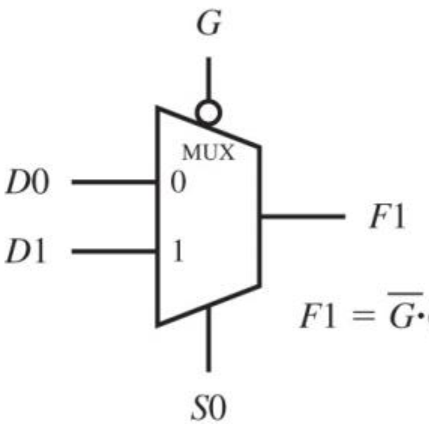


S_0	F
0	D_0
1	D_1

VHDL Design for 2-to-1 Multiplexer

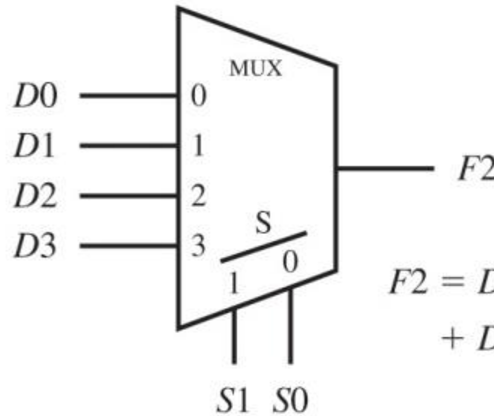

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity comb10 is port (
     D1, D0, S0 : in std_logic;
     F : out std_logic
     );
end comb10;

architecture Boolean_function of comb10 is
begin
     F <= (D0 and not S0) or (D1 and S0);
end Boolean_function;
```

Active Low 2-to-1 Multiplexer and 4-to-1 MUX



$$F1 = \overline{G}\cdot(D0\cdot\overline{S0} + D1\cdot S0)$$

(a)



$$F2 = D0\cdot\overline{S1}\cdot\overline{S0} + D1\cdot\overline{S1}\cdot S0 + D2\cdot S1\cdot\overline{S0} + D3\cdot S1\cdot S0$$

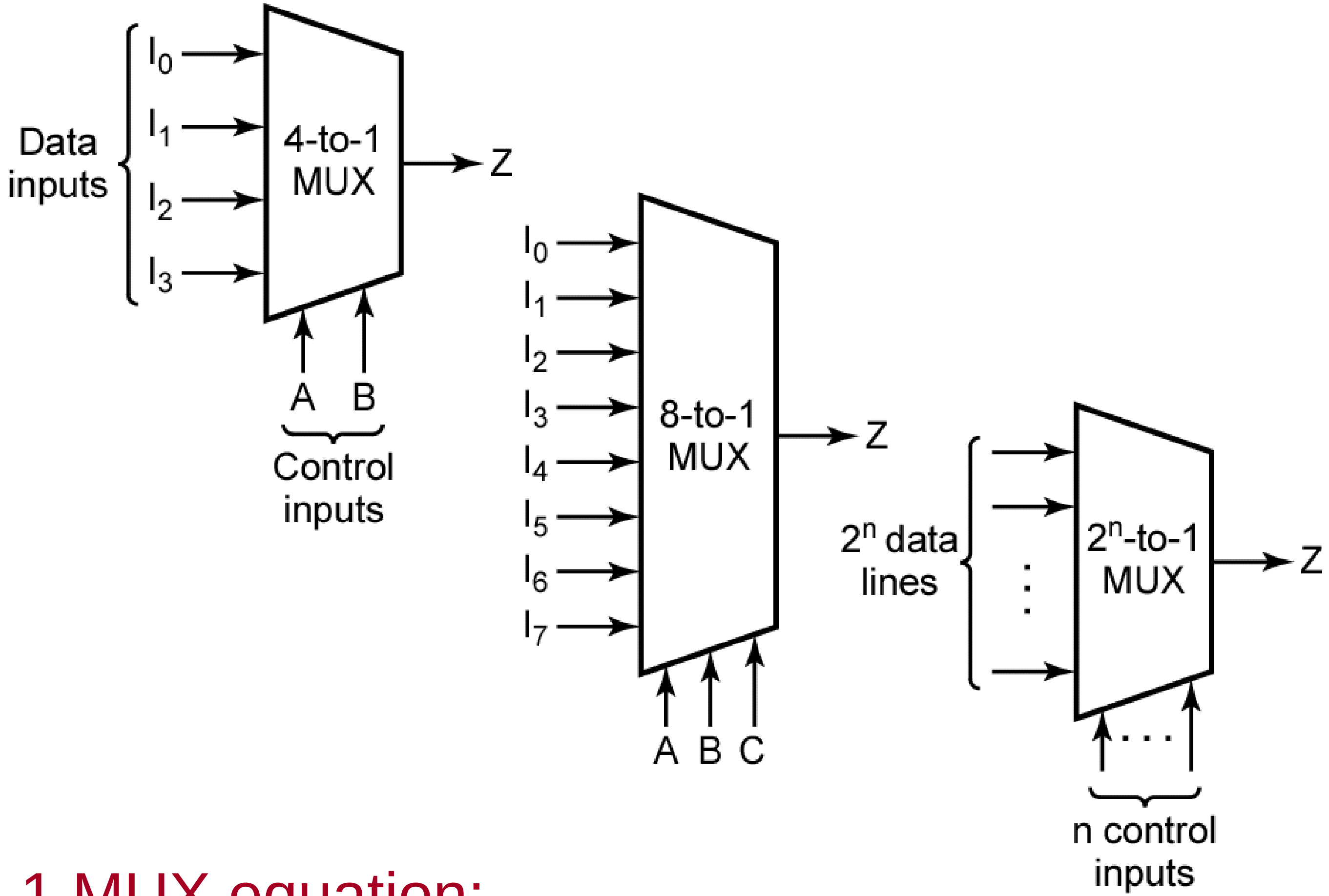
(b)

$G0$	$S0$	F
1	×	0
0	0	$D0$
0	1	$D1$

VHDL Design for 4-to-1 Multiplexer


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity comb11 is port (
     D3, D2, D1, D0, S1, S0, G : in std_logic;
     F1, F2 : out std_logic
     );
end comb11;

architecture Boolean_functions of comb11 is
begin
     F1 <=  not G and ((D0 and not S0) or (D1 and S0));
     F2 <=  (D0 and not S1 and not S0) or (D1 and not S1 and S0) or
            (D2 and S1 and not S0) or (D3 and S1 and S0);
end Boolean_functions;
```

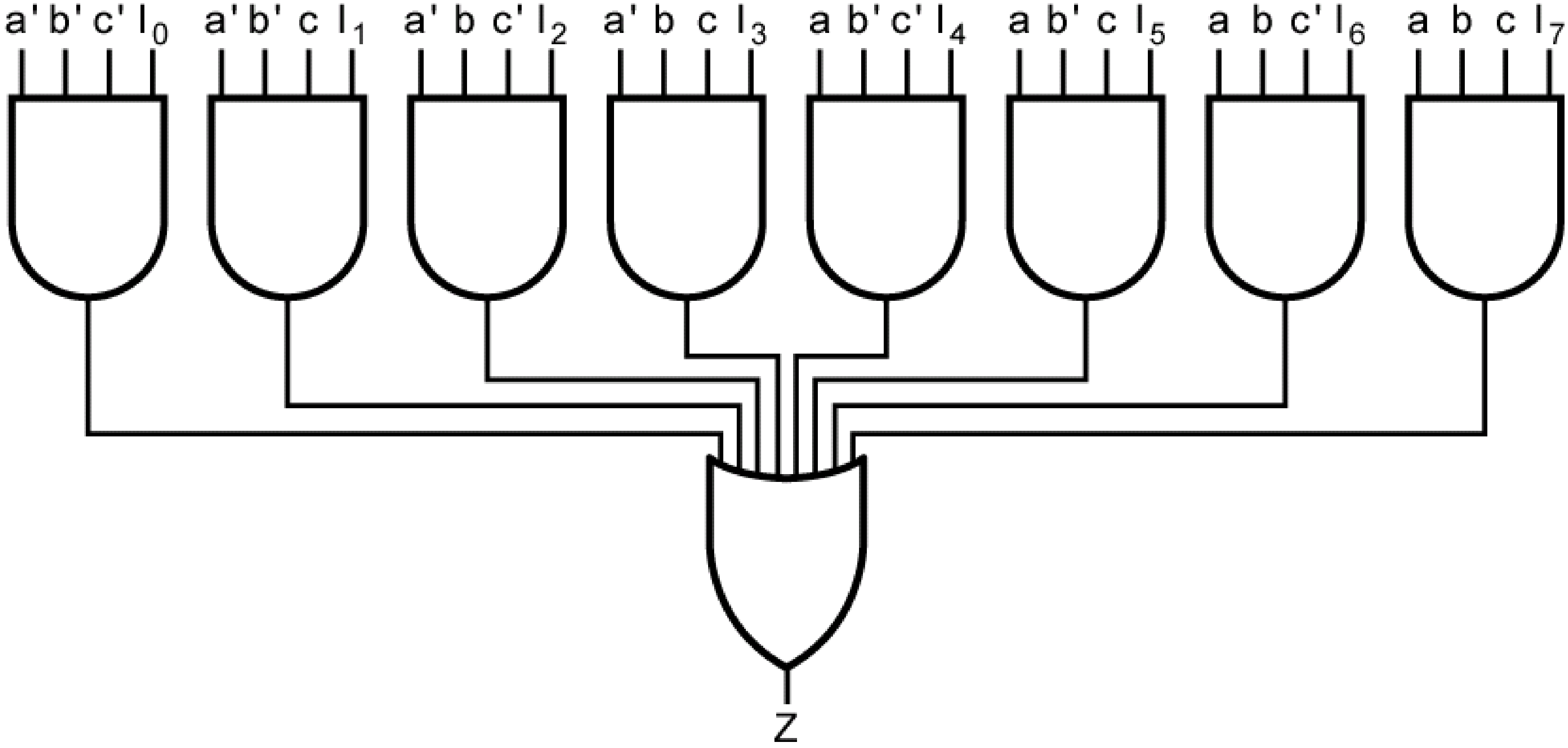
Multiplexers



8-to-1 MUX equation:

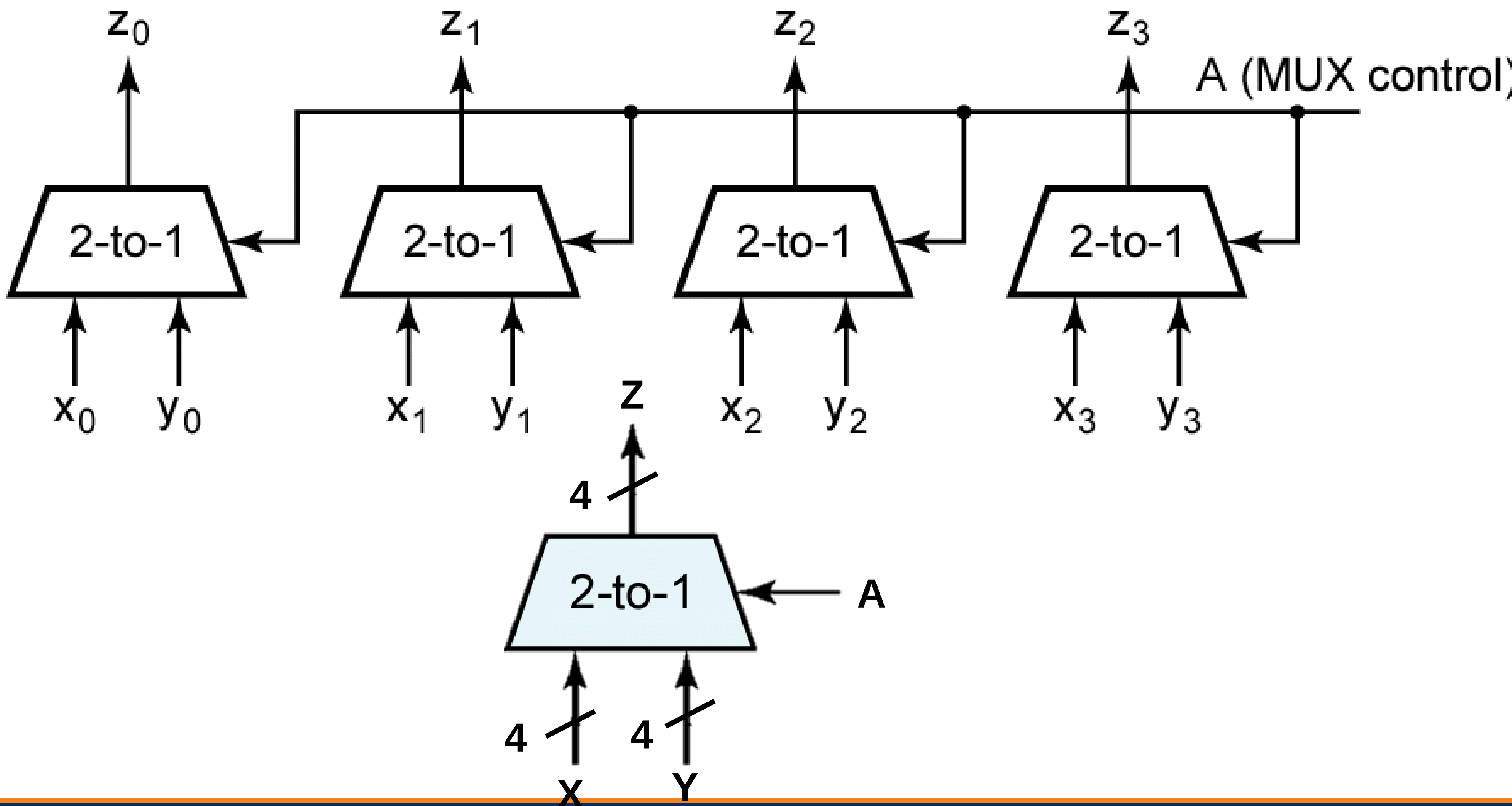
$$Z = A'B'C'I_0 + A'B'CI_1 + A'BC'I_2 + A'BCI_3 + AB'C'I_4 + AB'CI_5 + ABC'I_6 + ABCI_7$$

Logic Diagram for 8-to-1 MUX



Quad Multiplexer Used to Select Data For 4 Bit Bus

Control Variable *A* selects one of two 4-bit data words.

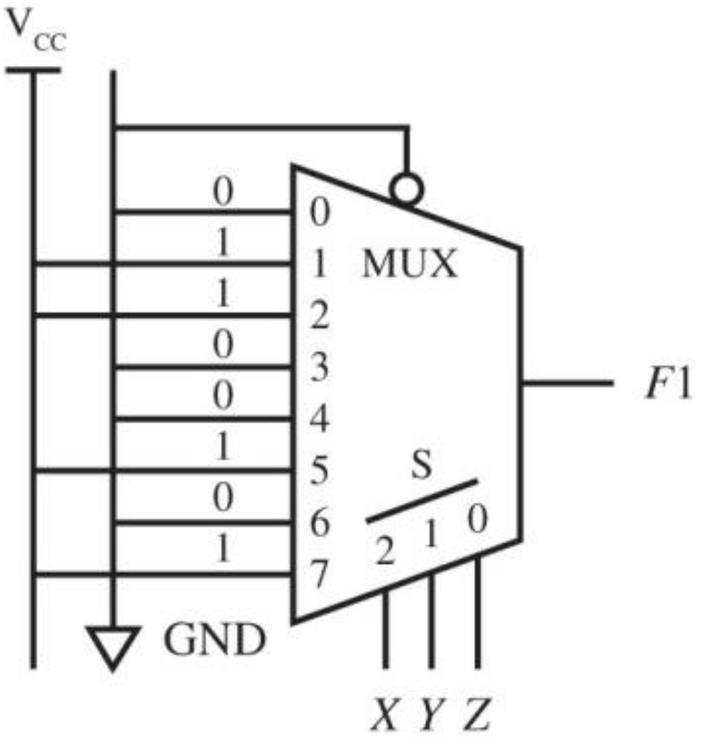


Designing Logic Circuits with MUXs

Designing Logic Circuits with MUXs

Technique 0:

- Use compact minterm form of the function. Function does not need to be reduced.
- Connect the data inputs of the MUX to the function values Vcc for 1 and GND for 0.
- Connect select lines of MUX to the input variables.
- Need a 2^n-to-1 MUX for any n-variable function.



$F_1(X,Y,Z) = \Sigma\ m(1,2,5,7)$

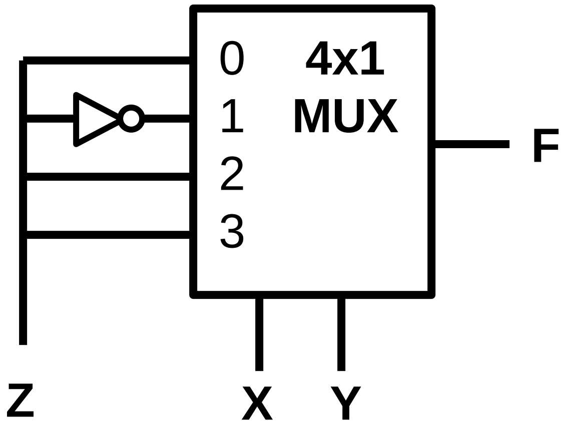
Designing Logic Circuits with MUXs

Technique 1:

- **For n variables, use n-1 variables as inputs to select lines. Hence, we need a 2^{n-1}-to-1 MUX.**

Example: **Given $F_1(X,Y,Z) = \Sigma\, m(1,2,5,7)$, 3 variables so we need a 2^2-to-1 MUX. Lets allocate X and Y to select lines, leaving Z for the Data Lines.**

	X	Y	Z	F		
I_0	0	0	0	0	Z	Notice F and Z are the same
	0	0	1	**1**		
I_1	0	1	0	**1**	Z'	Notice F and Z are Complemented
	0	1	1	0		
I_2	1	0	0	0	Z	Notice F and Z are the same
	1	0	1	**1**		
I_3	1	1	0	0	Z	Notice F and Z are the same
	1	1	1	**1**		



Designing Logic Circuits with MUXs

Technique 2 (MAP technique):

Given: $F_1(X,Y,Z) = \Sigma\, m(1,2,5,7)$

1. **First row:** list the inputs to MUX
   Second row: list the minterms so that the unused variable is set to zero
   Third row: list the minterms so that the unused variable is set to one.

Mux Selector Variables

		X'Y'	X'Y	XY'	XY
row 1		I_0	I_1	I_2	I_3
row 1	Z'	m_0	m_1	m_2	m_3
row 3	Z	m_4	m_5	m_6	m_7

Unused Variable

Designing Logic Circuits with MUXs

Technique 2 (MAP technique):

2.If F = 1, circle that minterm.

Mux Selector Variables

		X'Y'	X'Y	XY'	XY
row 1		I_0	I_1	I_2	I_3
row 1	**Z'**	m_0	m_1	m_2	m_3
row 3	**Z**	m_4	m_5	m_6	m_7

Unused Variable

$$F_1(X,Y,Z) = \Sigma\, m(1,2,5,7)$$

$I_0 = 0,\ I_1 = 1,\ I_2 = Z',\ I_3 = Z$

Designing Logic Circuits with MUXs

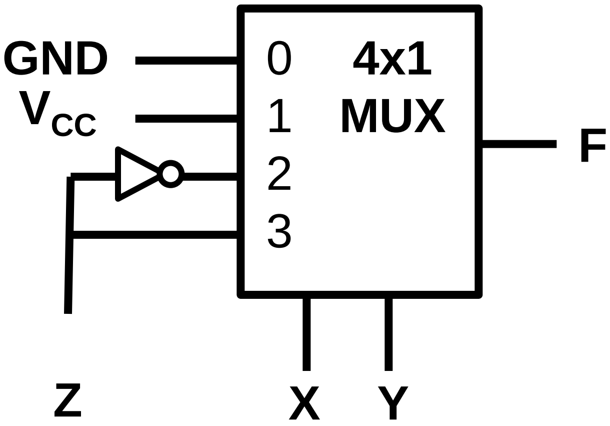
Technique 2 (MAP technique):

3.If upper minterm in a column is circled then MUX line I_i = Z'.
If lower minterm in a column is circled then I_i = Z.
If both minterms in a column are circled then I_i = 1.
If neither minterm in a column is circled then I_i = 0.

Mux Selector Variables

		X'Y'	X'Y	XY'	XY
row 1		I_0	I_1	I_2	I_3
row 1	**Z'**	m_0	m_1	m_2	m_3
row 3	**Z**	m_4	m_5	m_6	m_7

Unused Variable



$$I_0 = 0,\ I_1 = 1,\ I_2 = Z',\ I_3 = Z$$

Q&A