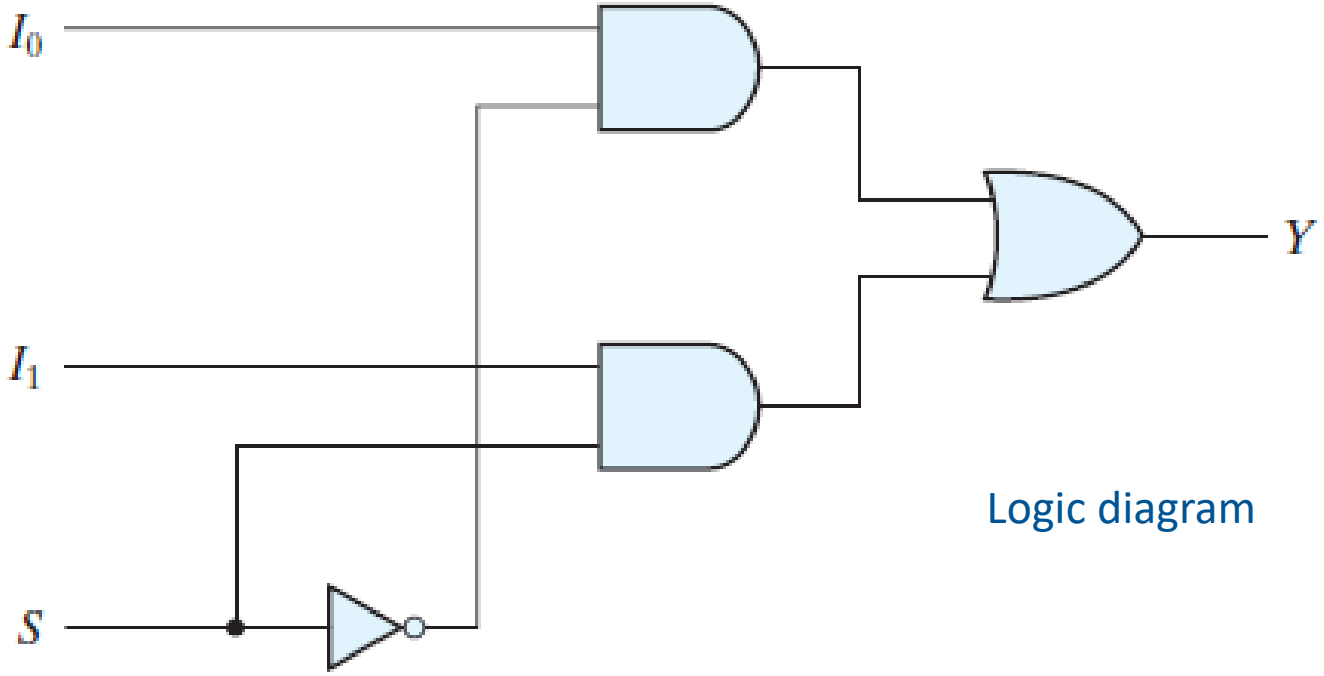
Multiplexer

Sparsh Mittal

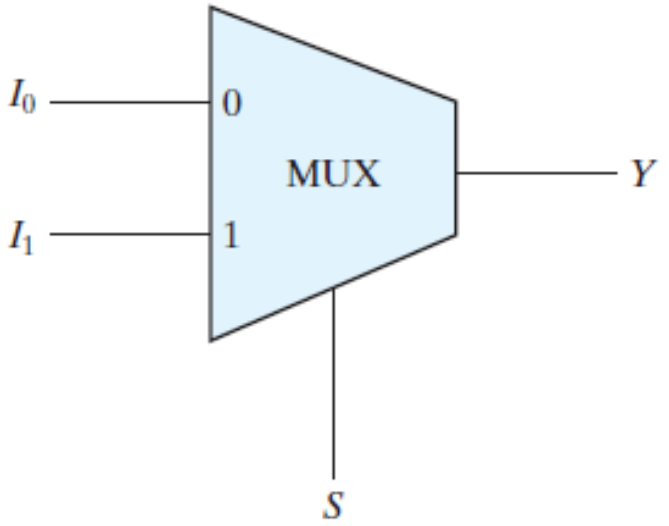
- A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line.
- The selection of a particular input line is controlled by a set of selection lines.
- Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

2-to-1 mux

- A two-to-one-line multiplexer connects one of two 1-bit sources to a common destination.
- The circuit has two data input lines, one output line, and one selection line S.
- When $S = 0$, upper AND gate is enabled and $I0$ has a path to output.
- When $S = 1$, lower AND gate is enabled and $I1$ has a path to output.



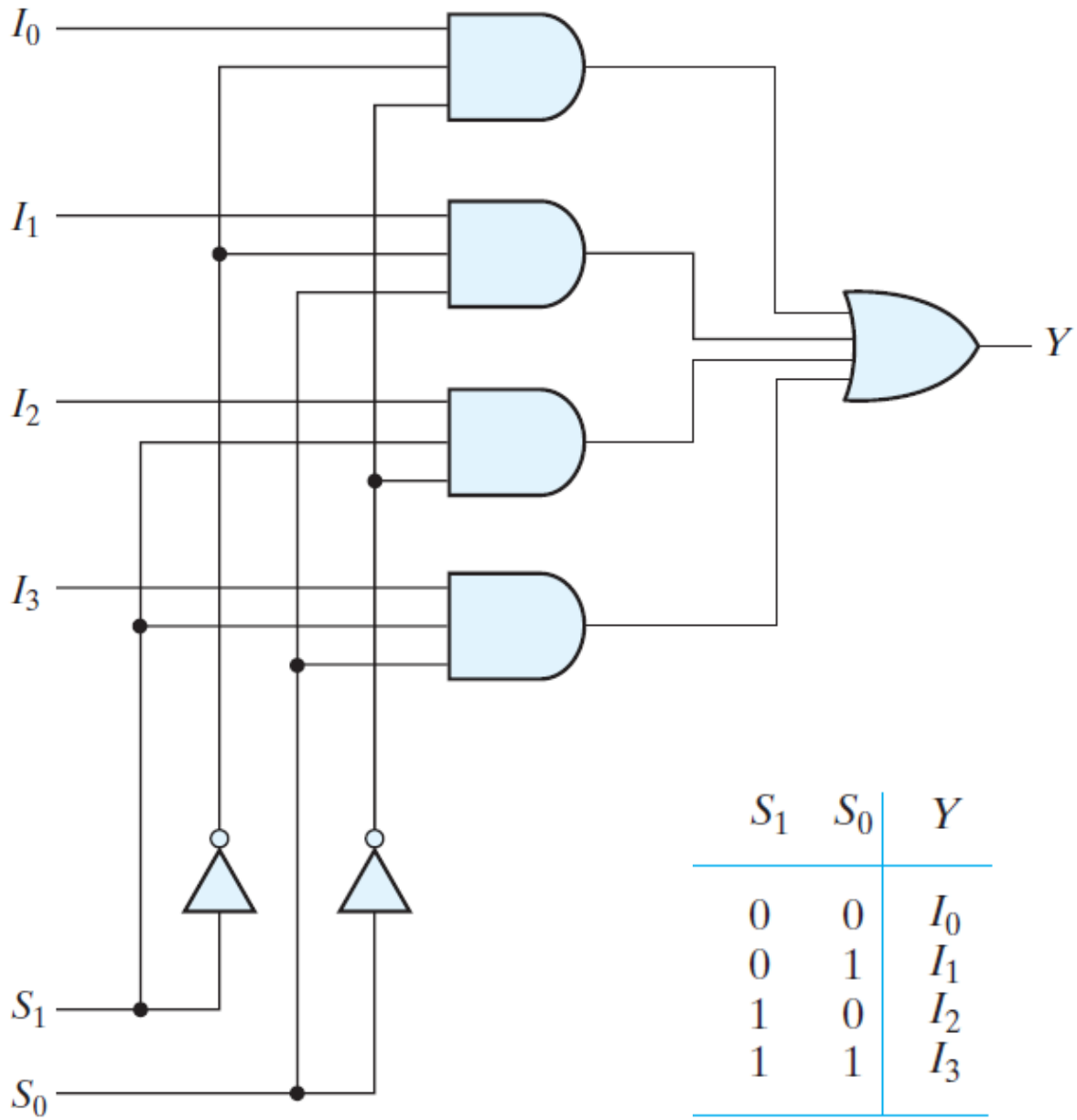
Logic diagram



Block diagram

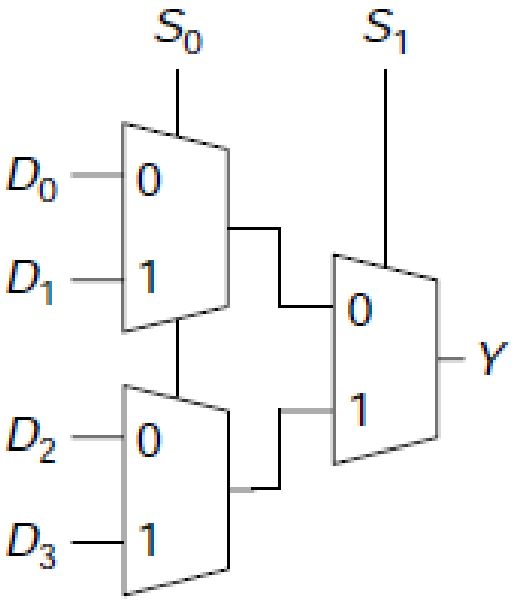
4-to-1 mux

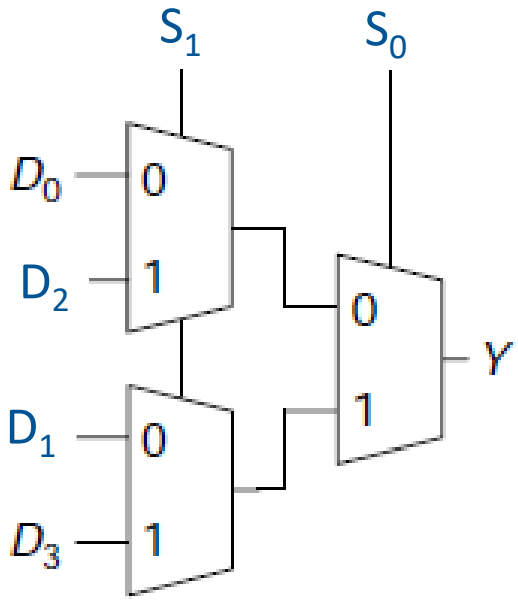
- Each of the four inputs, I0 through I3, is applied to one input of an AND gate.
- Selection lines S1 and S0 are decoded to select a particular AND gate.
- The outputs of the AND gates are applied to a single OR gate that provides the one-line output.
- MUX is also called a data selector.



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

Hierarchical design of MUX





Connection between mux and decoder

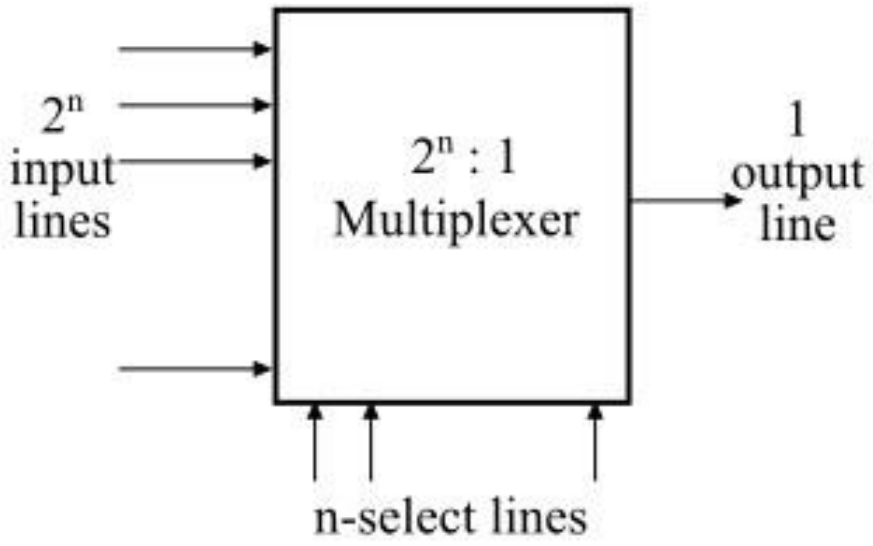


Figure 1 - Multiplexer

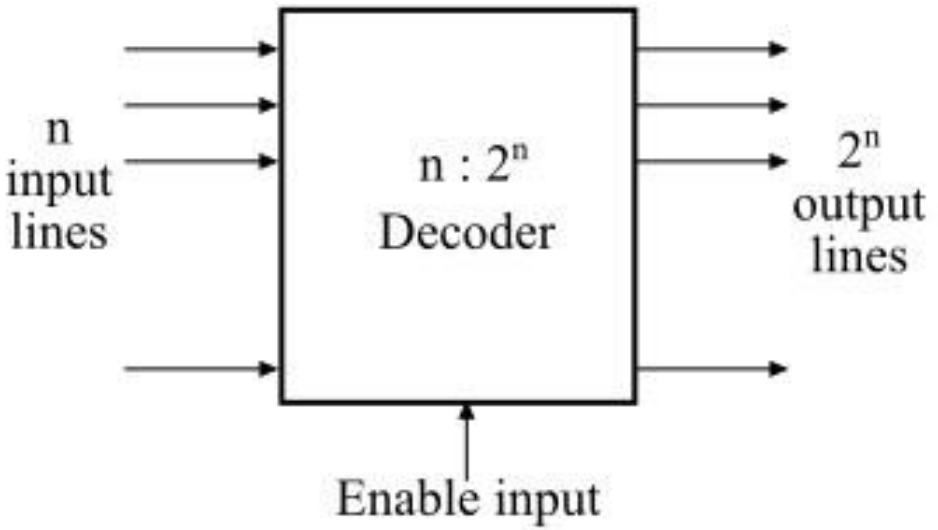
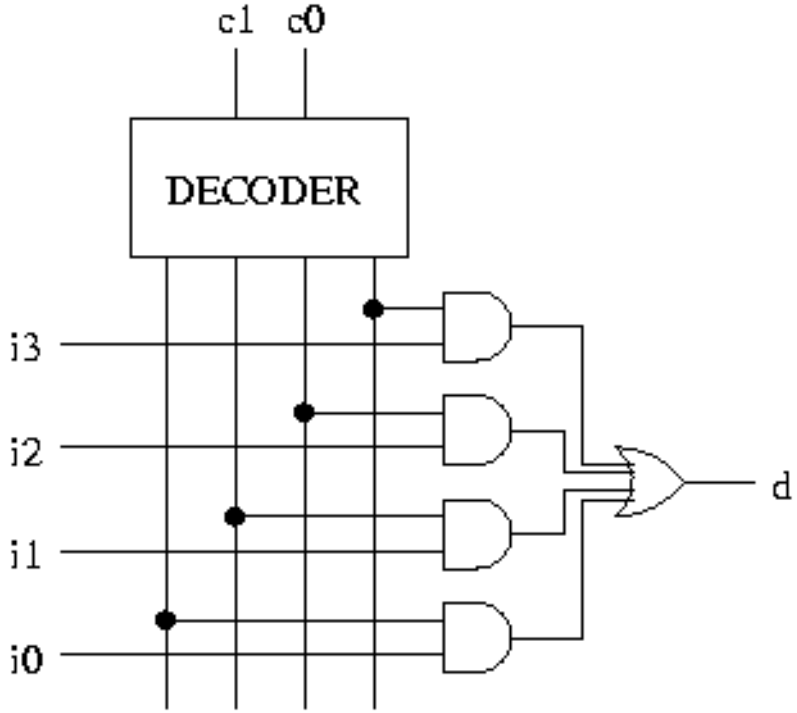


Figure 2 - Decoder

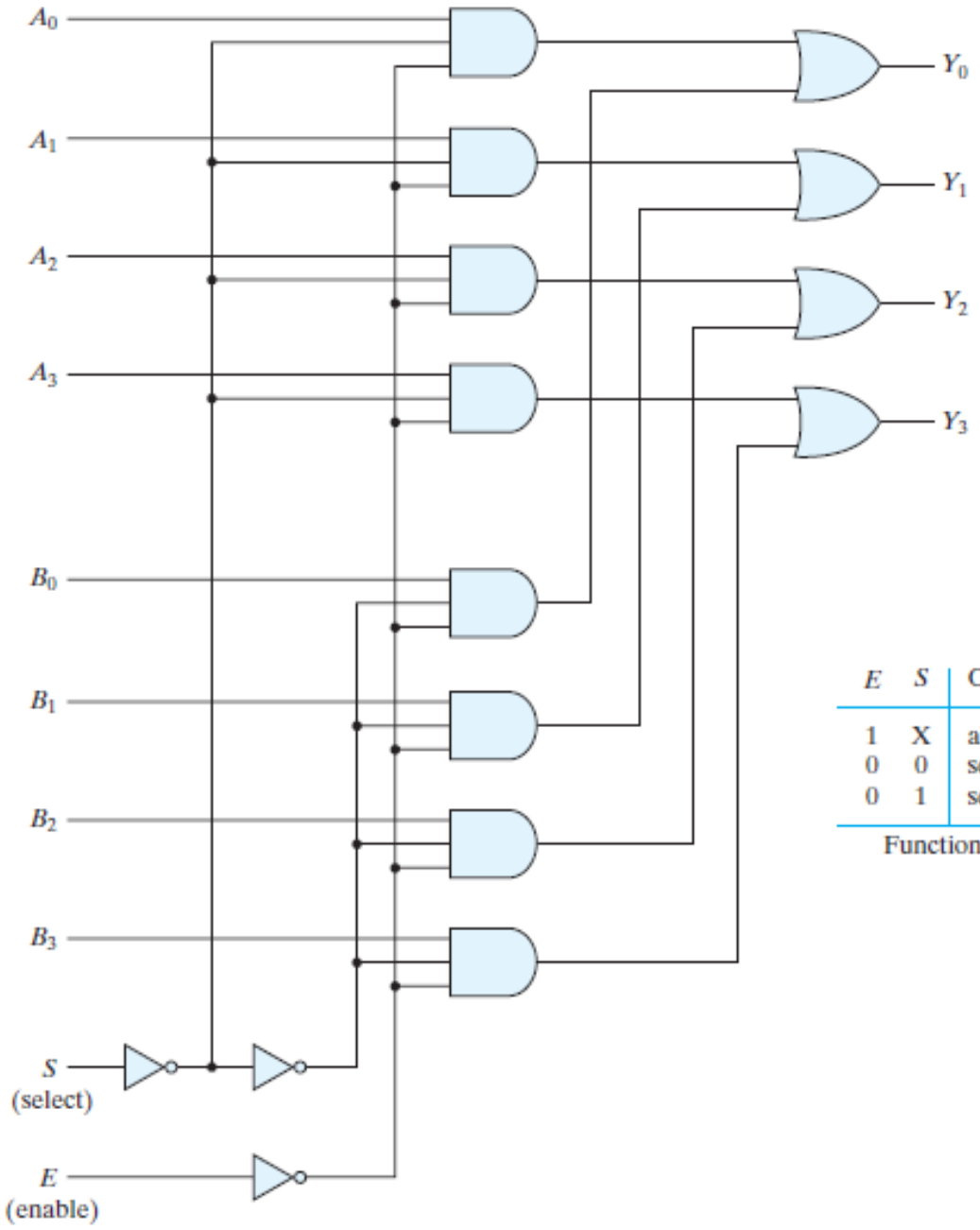
Designing mux using a decoder

- A 2^n -to-1-line multiplexer is constructed from an n -to-2^n decoder by adding 2^n input lines to it, one to each AND gate.
- The outputs of the AND gates are applied to a single OR gate.



Multiple-bit selection

- Common selection inputs can be used to provide multiple-bit selection logic.
- This circuit has four MUXes, each capable of selecting one of two input lines.
- Output Y0 can be selected to come from either input A0 or input B0 and so on.
- Input selection line S selects one of the lines in each of the four MUXes.
- Circuit is enabled using Enable signal



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

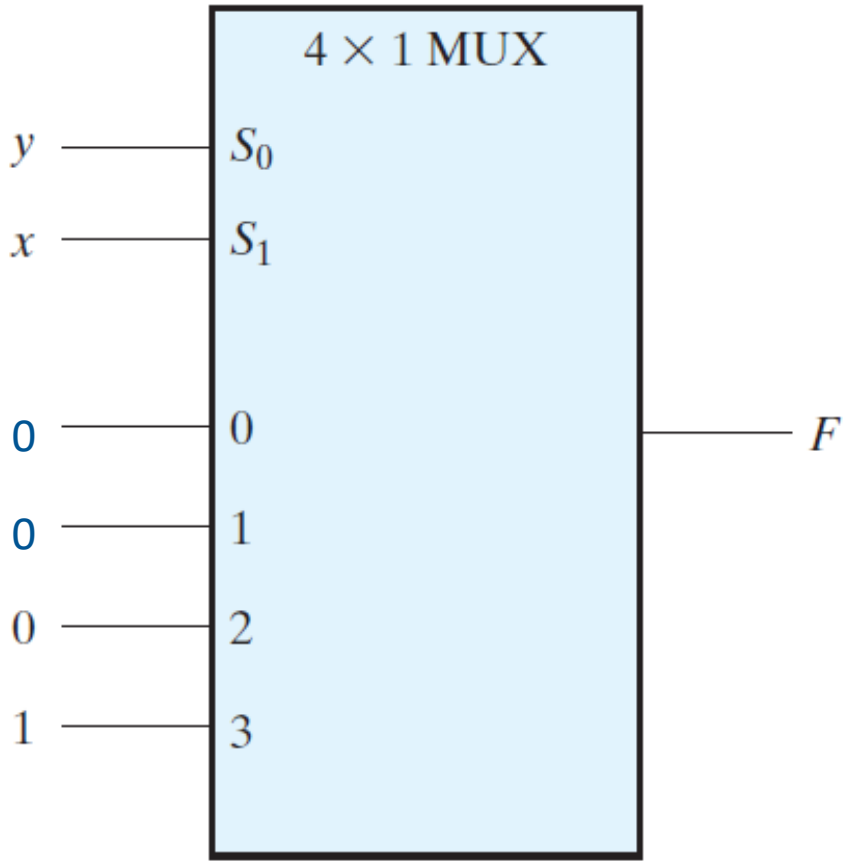
Function table

Quadruple two-to-one-line multiplexer

MUX for implementing a Boolean function

- A MUX is essentially a decoder that includes the OR gate within the unit.
- Minterms of a function are generated in a MUX by the circuit associated with the selection inputs.
- The individual minterms can be selected by the data inputs
- Thus, we can implement a Boolean function of n variables using a MUX with n selection inputs and 2^n data inputs, one for each minterm.

Implement a 2-input AND function using a 4:1 MUX.



(b) Multiplexer implementation

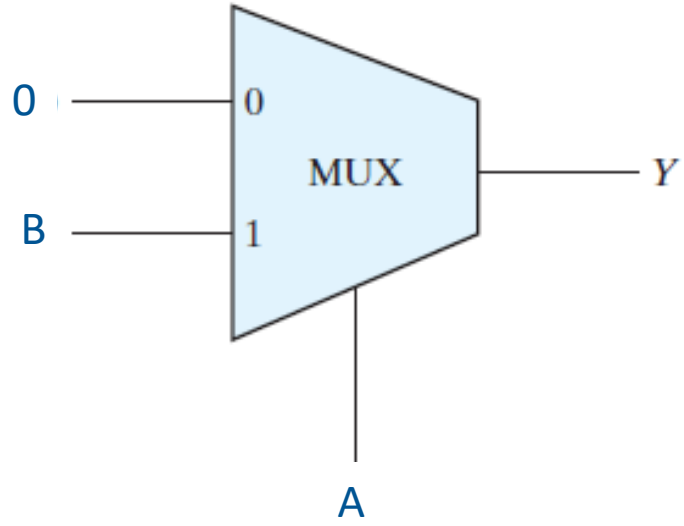
More efficient method

- We will implement a Boolean function of n variables with a MUX that has n - 1 selection inputs.
- The first n - 1 variables of the function are connected to the selection inputs of the multiplexer.
- The remaining single variable of the function is used for the data inputs.
- If the single variable is denoted by z , each data input of the multiplexer will be z , z', 1, or 0.

Implement a 2-input AND function using a 2:1 MUX.

- Inputs are A and B.

A	B	Y=A.B	
0	0	0	Y=0
0	1	0	
1	0	0	Y=B
1	1	1	

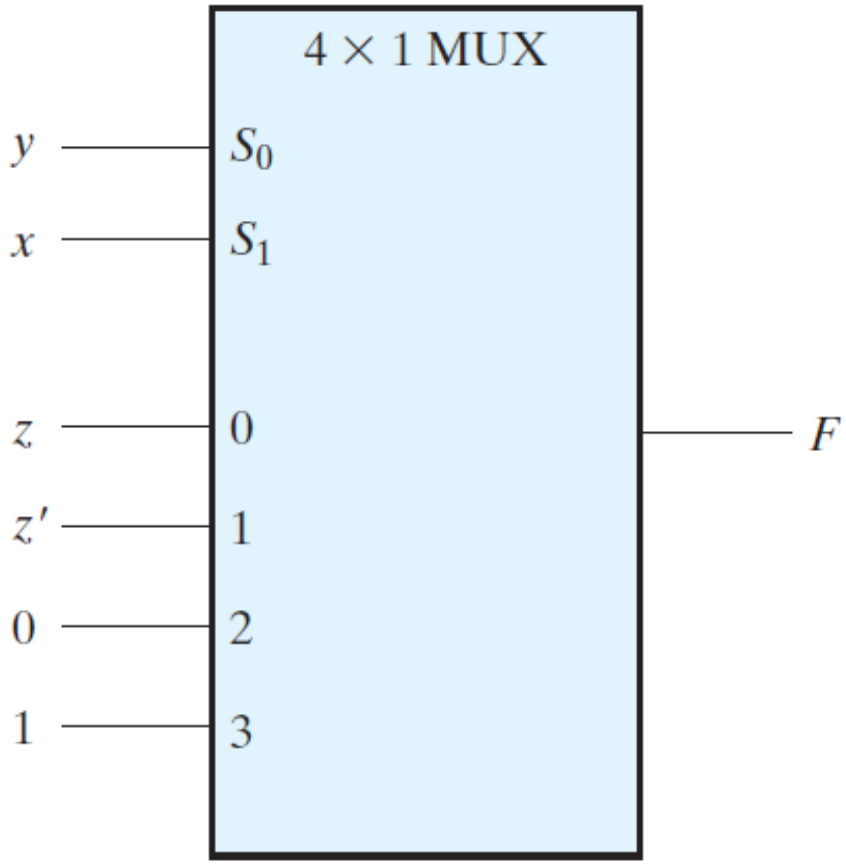


Example 1

- $F(x, y, z) = \sum(1, 2, 6, 7)$

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



(b) Multiplexer implementation

Procedure

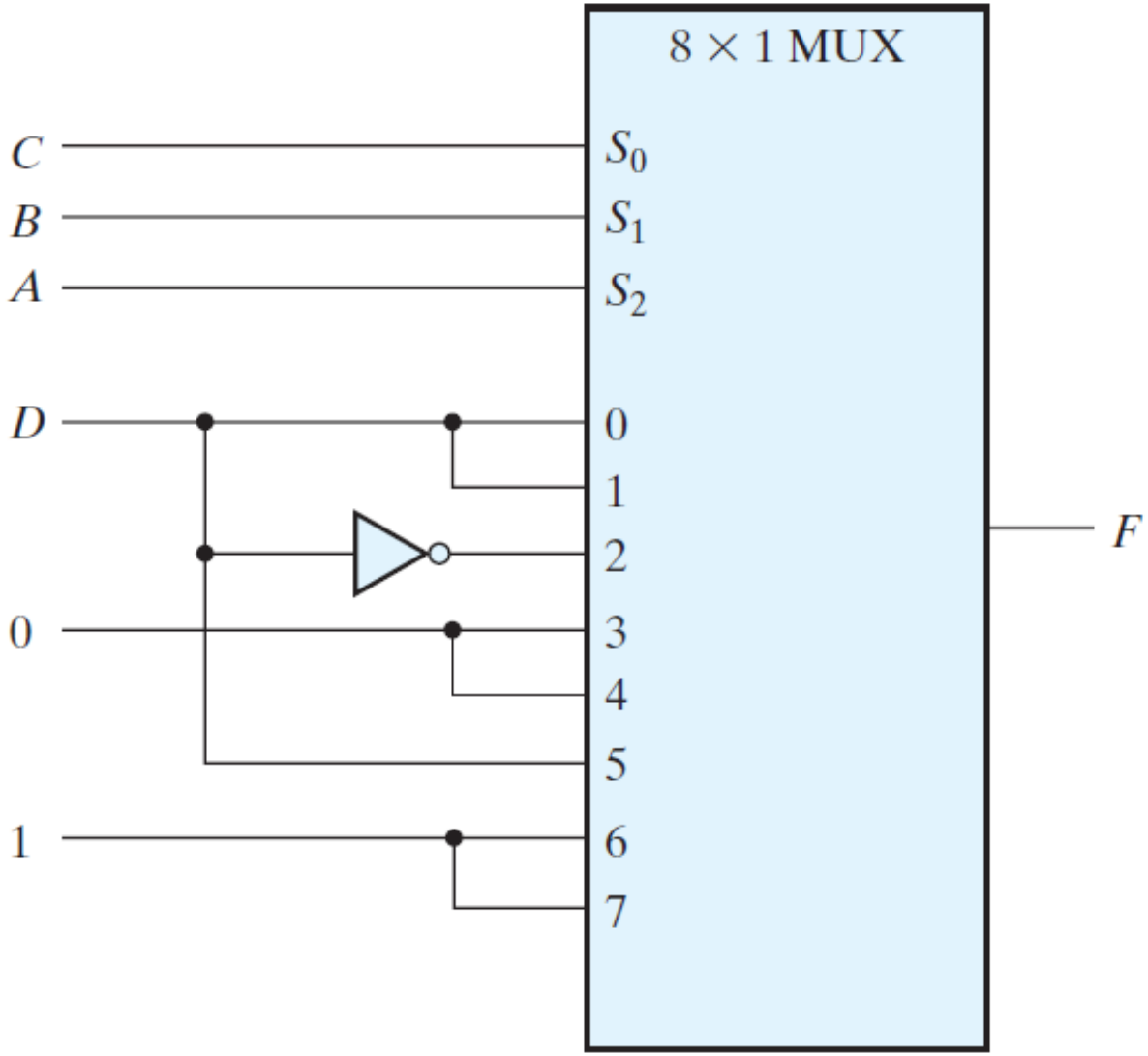
- Boolean function is listed in a truth table.
- Then first n - 1 variables in the table are applied to the selection inputs of the multiplexer.
- For each combination of the selection variables, we evaluate the output as a function of the last variable.
- This function can be 0, 1, the variable, or the complement of the variable.
- These values are then applied to the data inputs in the proper order.

Example 2

- $F(A, B, C, D) = \sum(1, 3, 4, 11, 12, 13, 14, 15)$
- Variable A must be connected to selection input $S2$ so that A , B, and C correspond to selection inputs $S2$, $S1$, and $S0$, respectively

A	B	C	D	F	
0	0	0	0	0	$F = D$
0	0	0	1	1	
0	0	1	0	0	$F = D$
0	0	1	1	1	
0	1	0	0	1	$F = D'$
0	1	0	1	0	
0	1	1	0	0	$F = 0$
0	1	1	1	0	
1	0	0	0	0	$F = 0$
1	0	0	1	0	
1	0	1	0	0	$F = D$
1	0	1	1	1	
1	1	0	0	1	$F = 1$
1	1	0	1	1	
1	1	1	0	1	$F = 1$
1	1	1	1	1	

A	B	C	D	F	
0	0	0	0	0	$F = D$
0	0	0	1	1	
0	0	1	0	0	$F = D$
0	0	1	1	1	
0	1	0	0	1	$F = D'$
0	1	0	1	0	
0	1	1	0	0	$F = 0$
0	1	1	1	0	
1	0	0	0	0	$F = 0$
1	0	0	1	0	
1	0	1	0	0	$F = D$
1	0	1	1	1	
1	1	0	0	1	$F = 1$
1	1	0	1	1	
1	1	1	0	1	$F = 1$
1	1	1	1	1	



Application of MUX and DEMUX

- We can design serial data trans-receiver.
- It transmits data bit-by-bit serially.