Chapter4: Combinational Logic

Lecture8- Study Design and Applications of Multiplexers, Function Implementation using Multiplexers

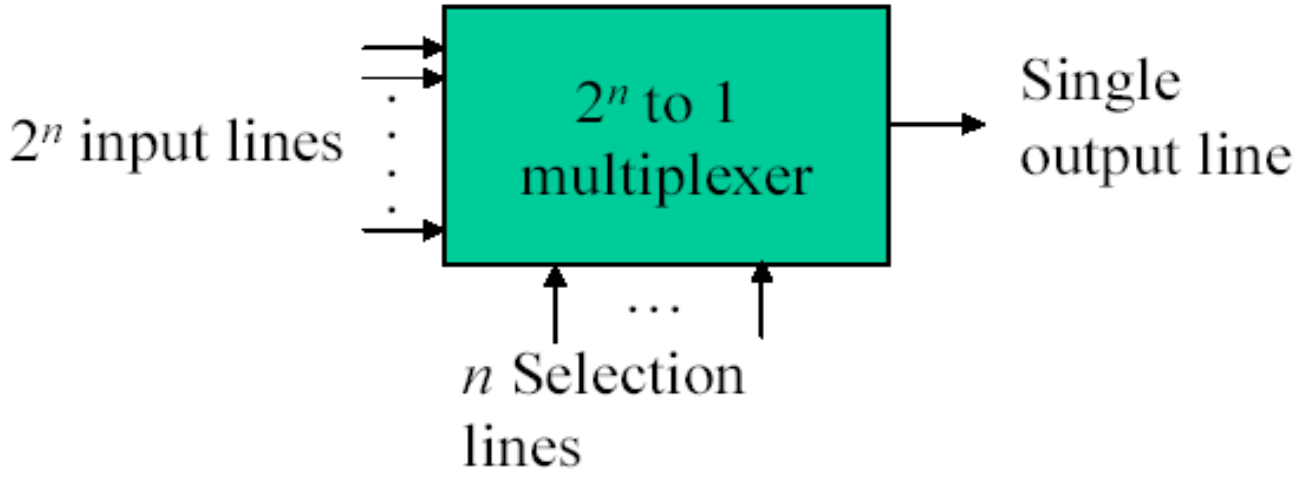
Engr. Arshad Nazir, Asst Prof
Dept of Electrical Engineering
SEECS

Objectives

- Study design and applications of Multiplexers
- Function implementation using Multiplexers

Multiplexers

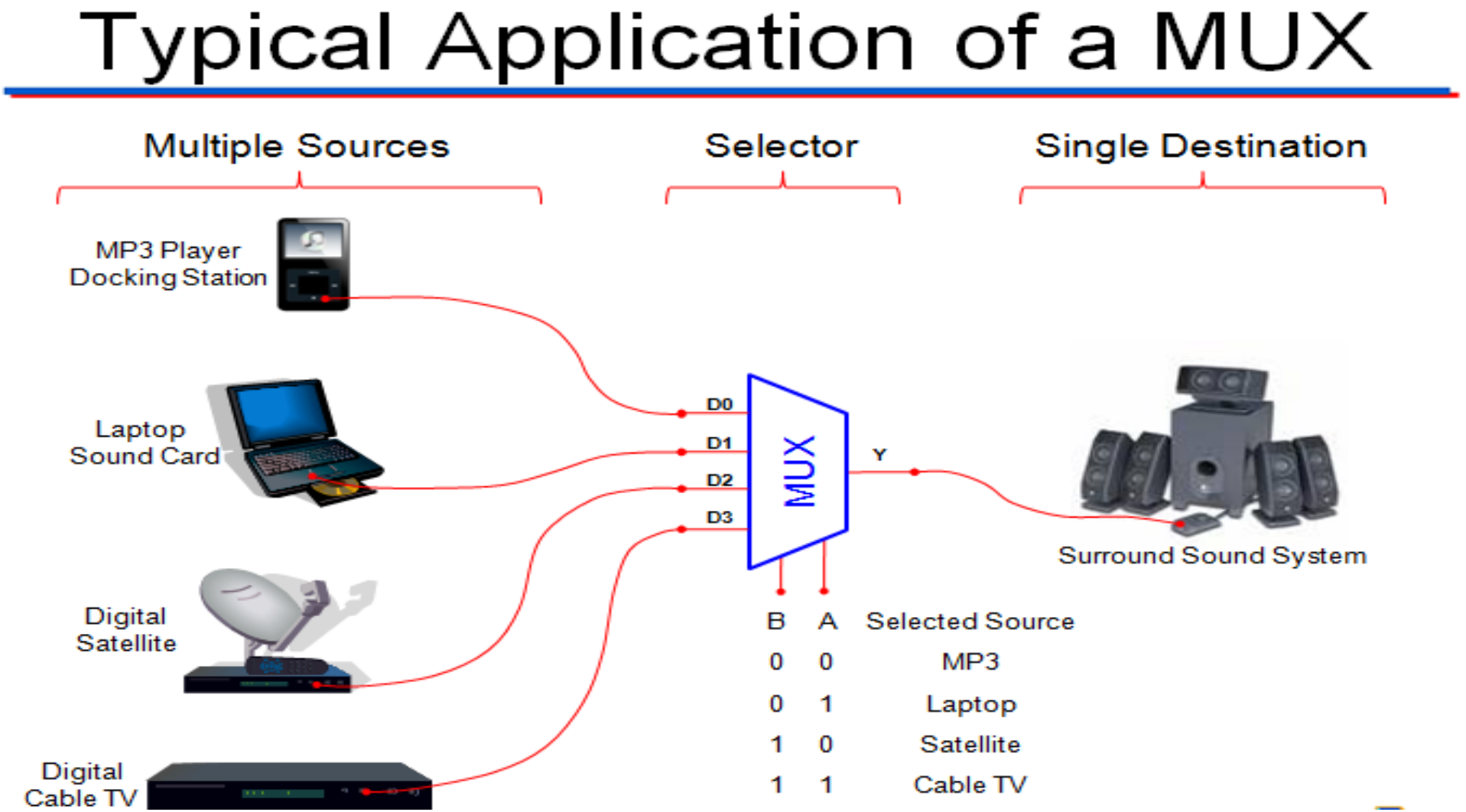
- A multiplexer is a combinational circuit that selects binary information from one of 2^n input lines and directs it to a single line. There are n selection lines.



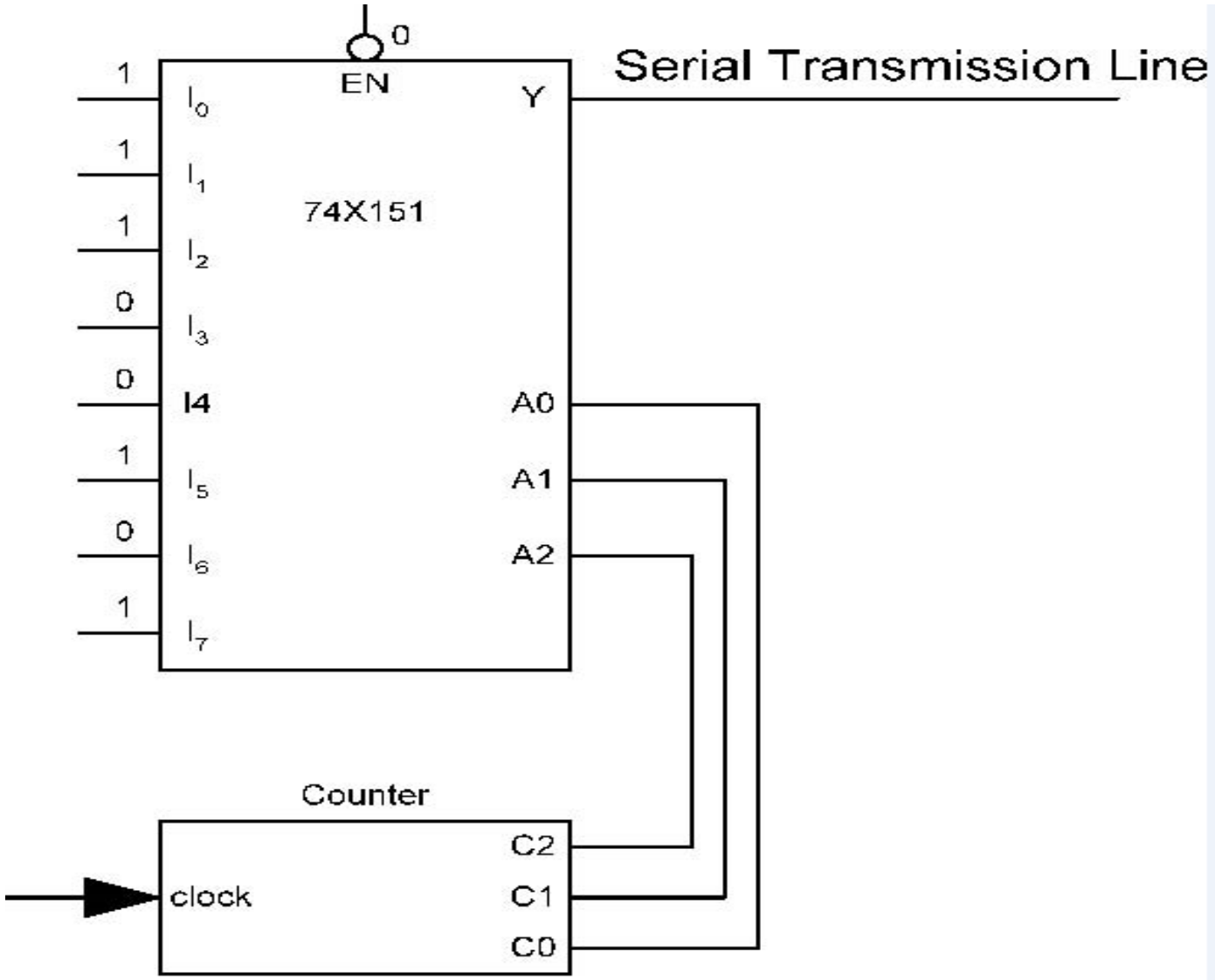
Applications

- Data Selection and Routing: Select data from one of many sources and route it to a single destination.
- Parallel-to-Serial Conversion of binary data.
- Operation Sequencing: When used with control sequencers in physical systems.
- Logic Function Generation: Muxes can be used to implement the logic functions directly from the truth tables without the need for simplification.
- Multiple-bit Selection Logic: Used in arithmetic circuits.
- Synchronous data transmission system to serially transmit four-bit data words from a XMTR to a remote RCVR. These data are said to time-division-multiplexed because four different sets of data are appearing on the same output line at different times.
- Signal routing, data communications, and multiplexed bus control.

Data Selection



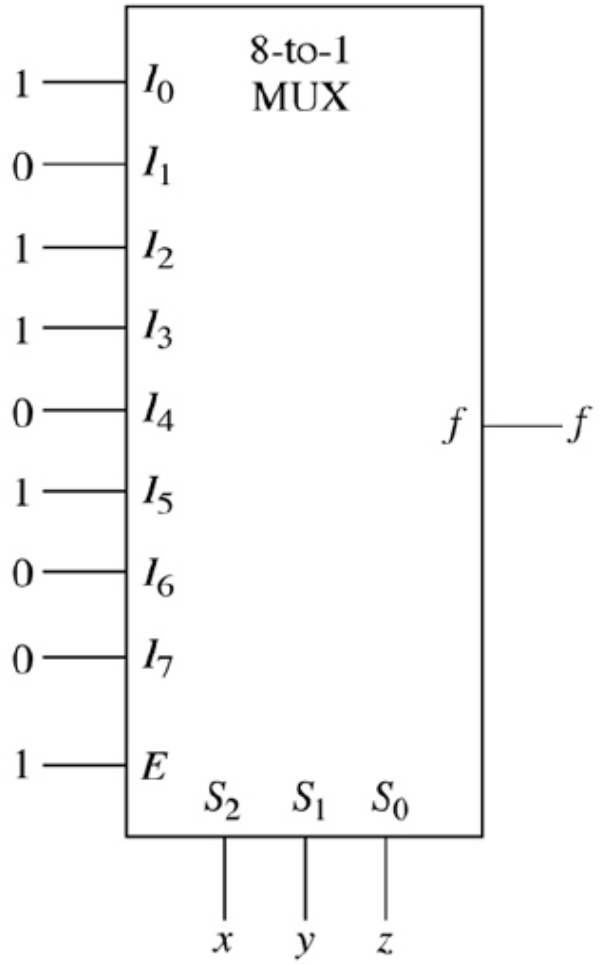
Parallel to Serial Conversion



Functions Implementation using Multiplexers

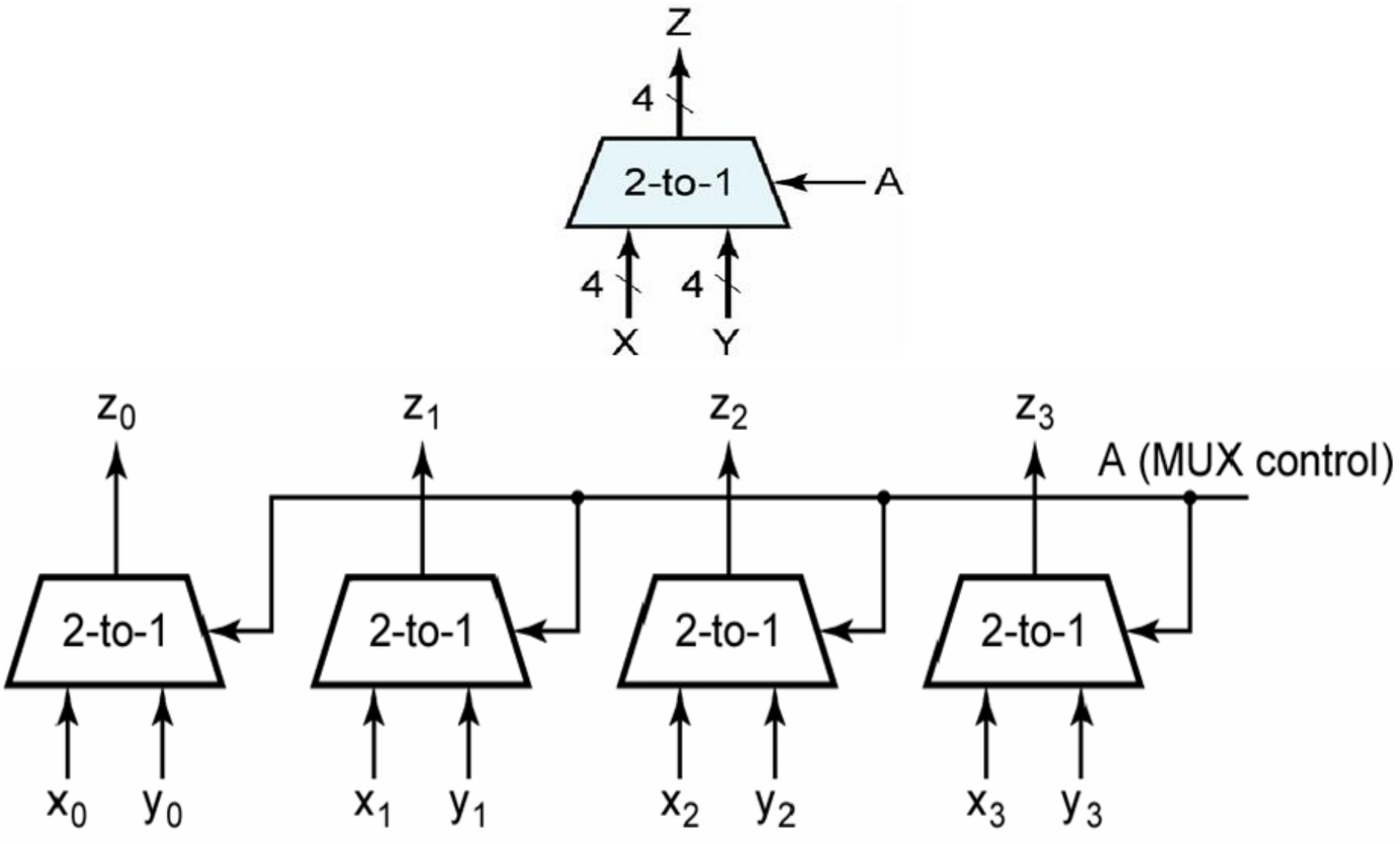
x	y	z	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

(a)

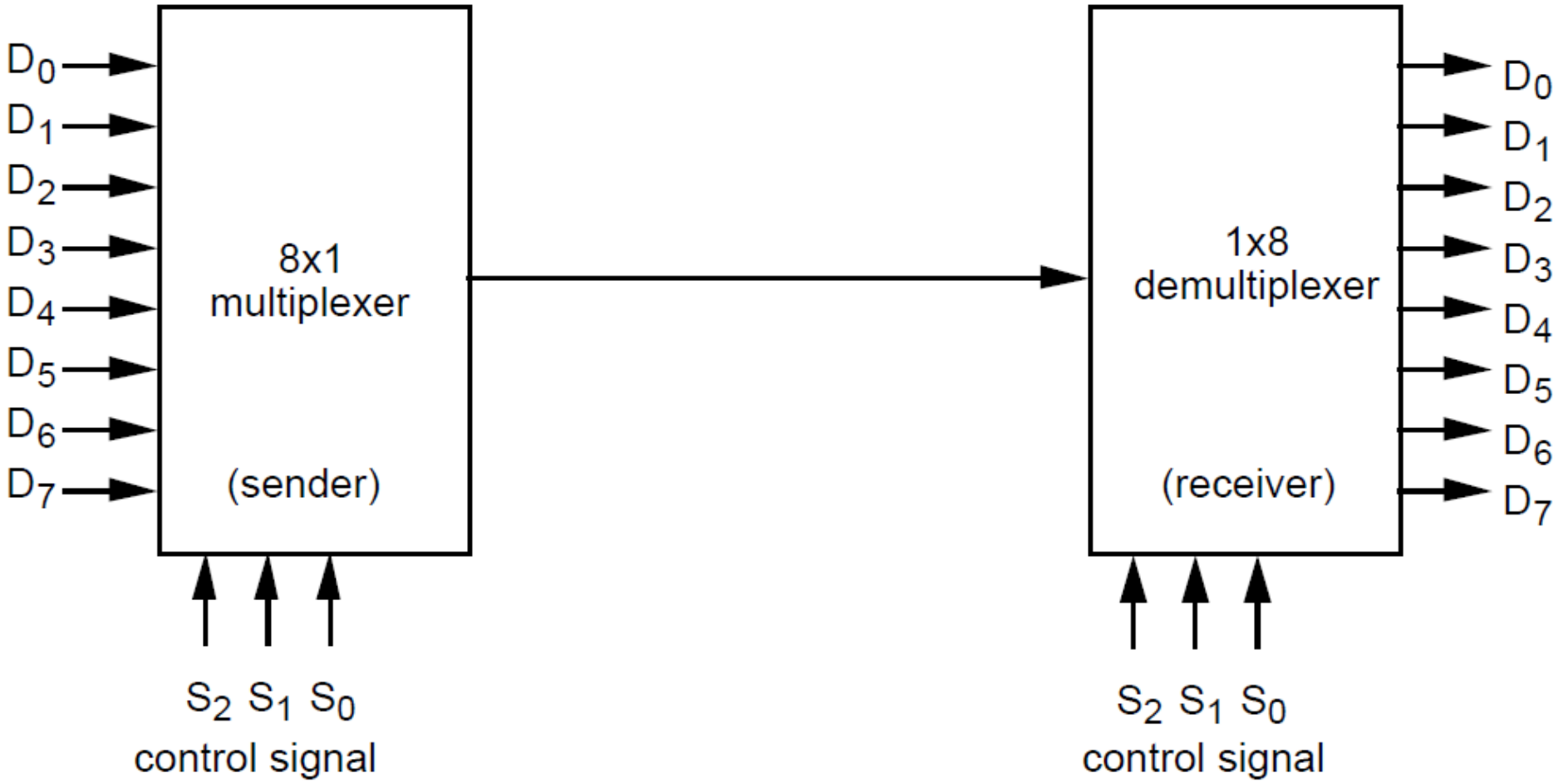


(b)

Multiple-bit Selection Logic



Synchronous Data Transmission

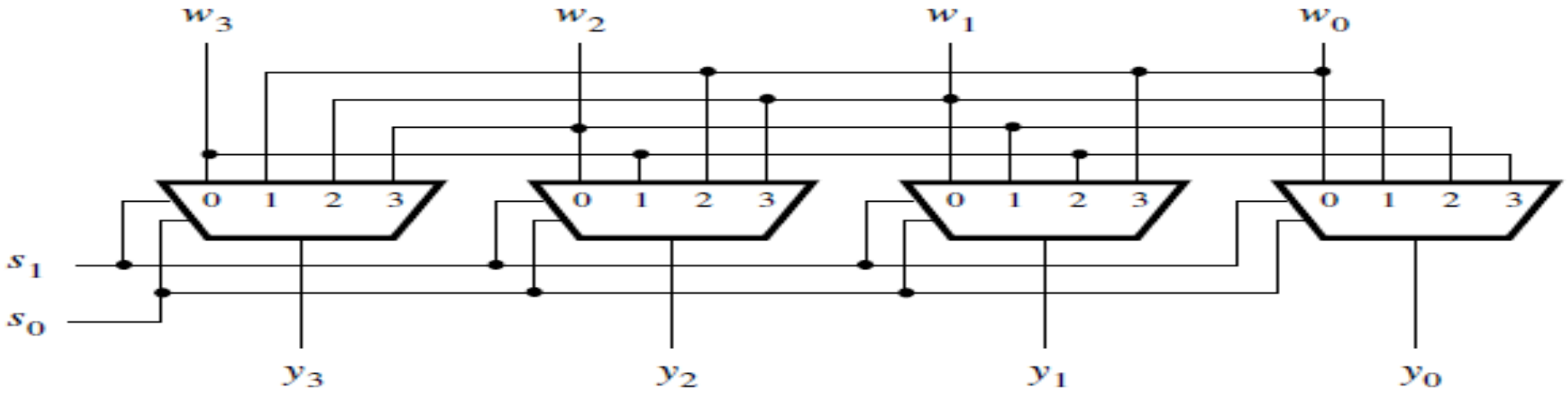


Main function of multiplexer and demultiplexer

Multiplexers use in the design of Barrel Shifters

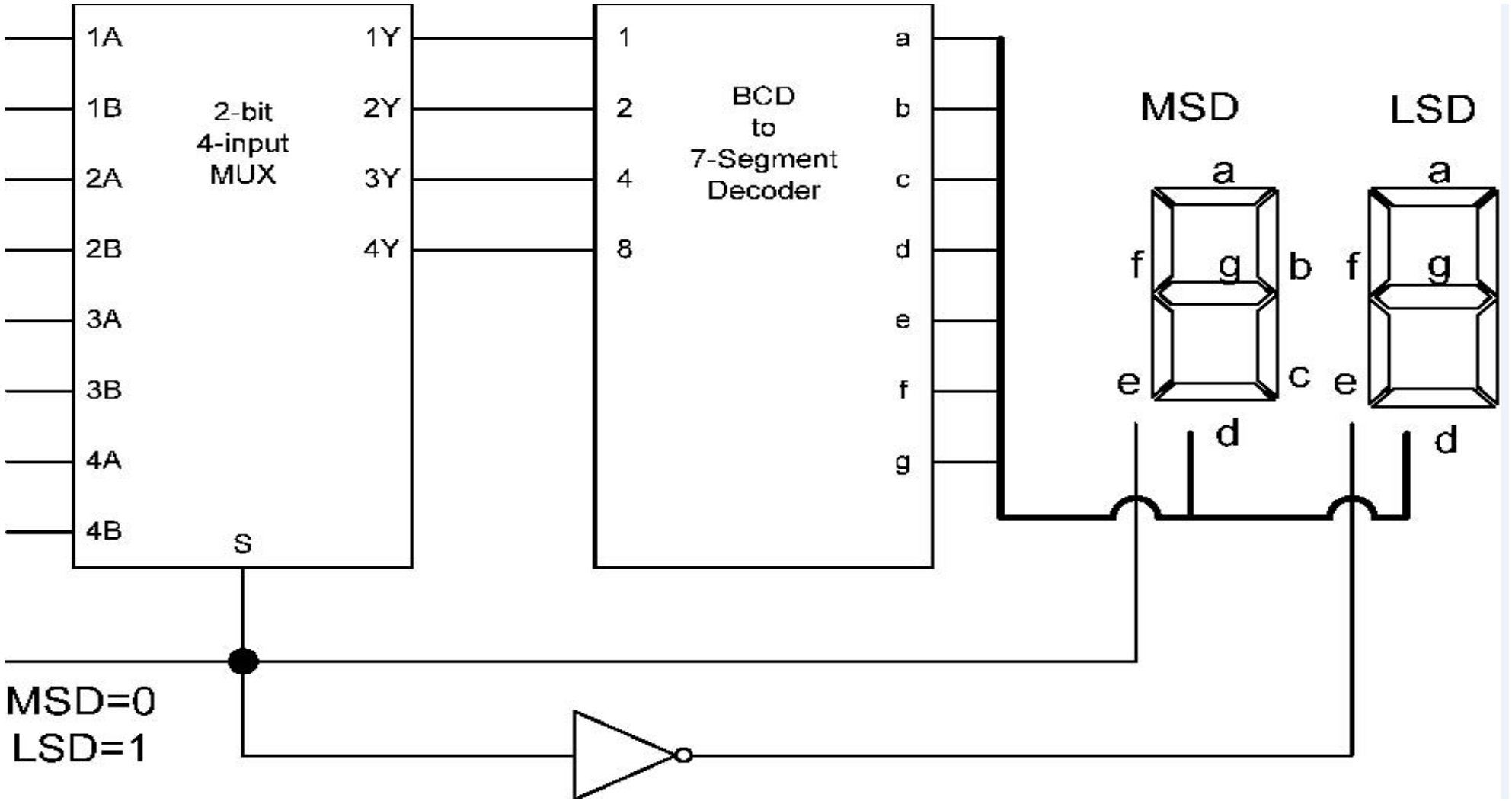
s_1	s_0	y_3	y_2	y_1	y_0
0	0	w_3	w_2	w_1	w_0
0	1	w_0	w_3	w_2	w_1
1	0	w_1	w_0	w_3	w_2
1	1	w_2	w_1	w_0	w_3

(a) Truth table



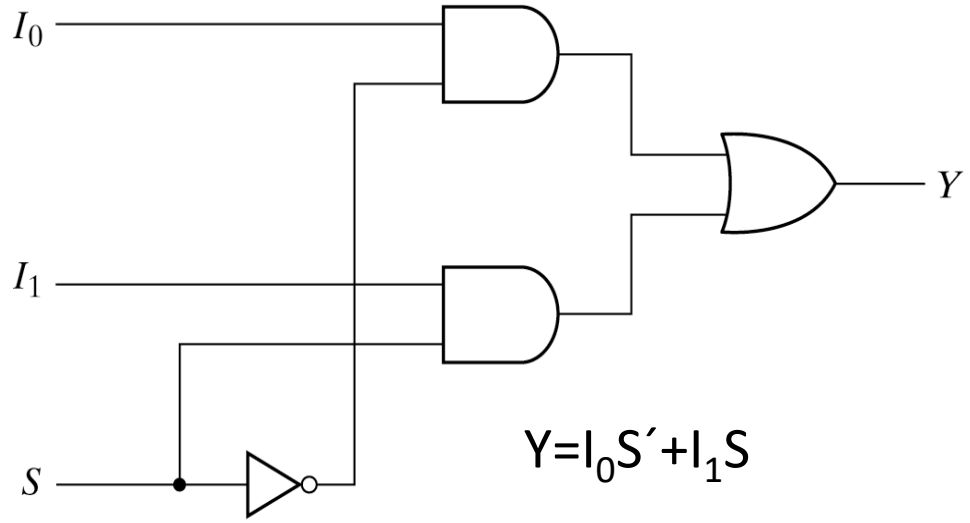
(b) Circuit

Multiplexed Display

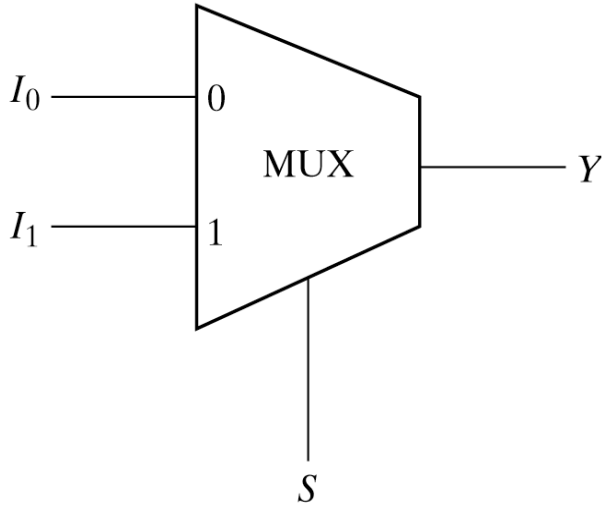


2-to-1-line Multiplexer

- 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 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

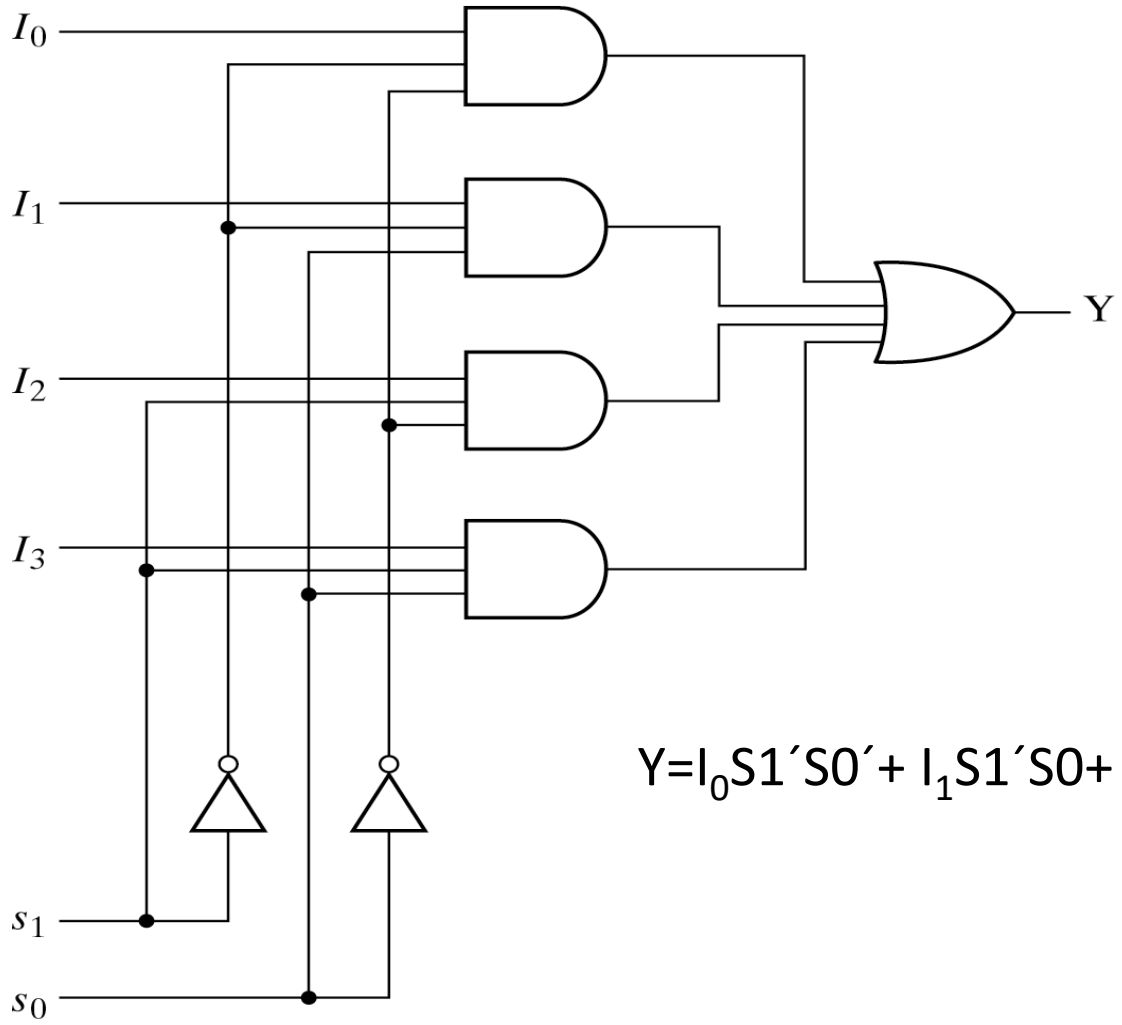


(b) Block diagram

Fig. 4-24 2-to-1-Line Multiplexer

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



(a) Logic diagram

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

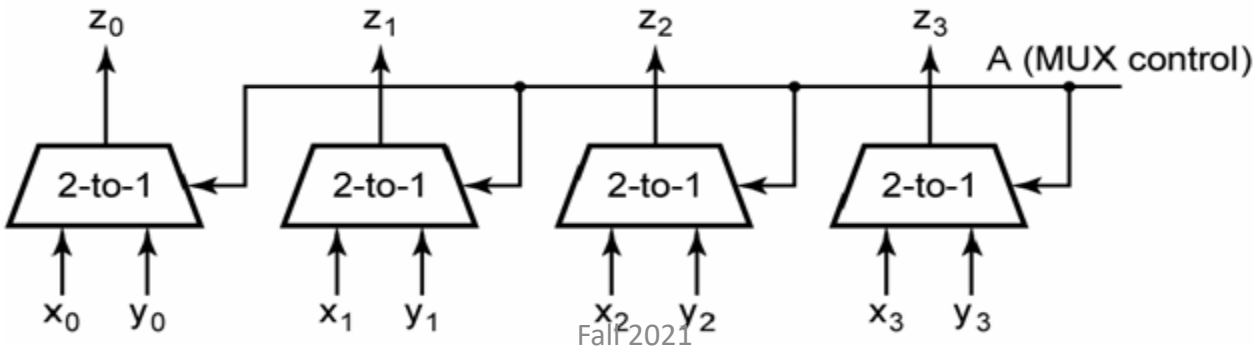
(b) Function table

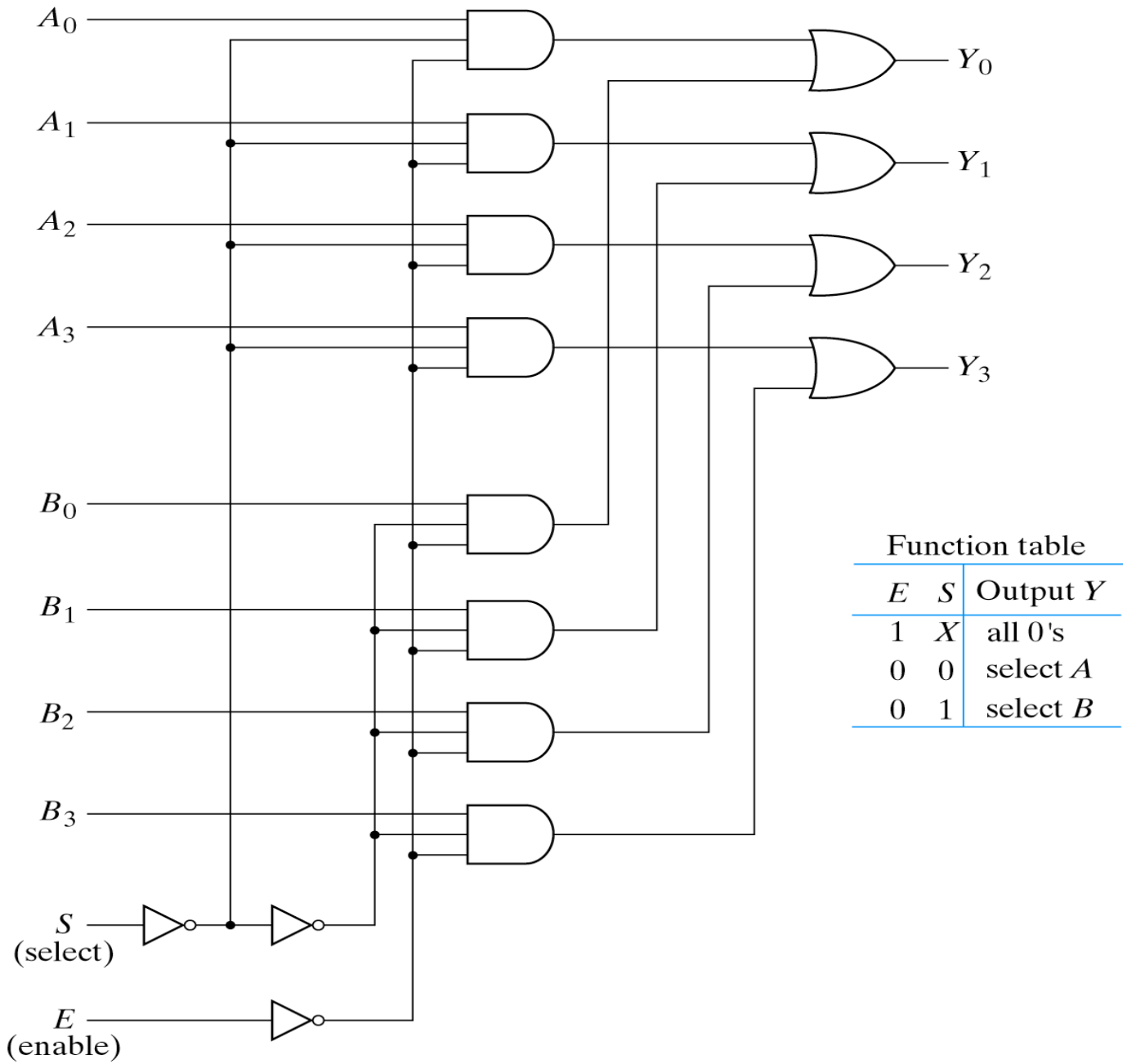
$$Y=I_0S1'S0'+ I_1S1'S0+ I_2S1S0'+ I_3S1S0$$

Fig. 4-25 4-to-1-Line Multiplexer

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





Function table

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

Fig. 4-26 Quadruple 2-to-1-Line Multiplexer

Boolean function implementation

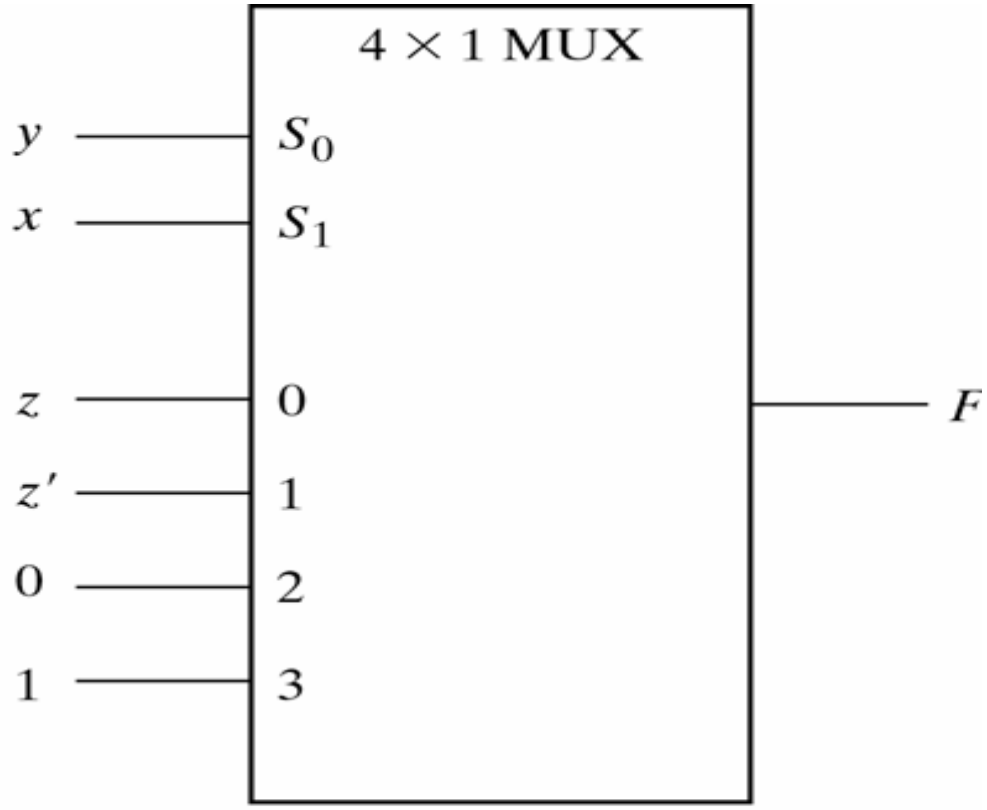
- MUX: A decoder + an OR gate
- 2^n-to-1 MUX can implement any Boolean function of n input variables with n selection inputs.
- We can also use smaller size decoders like 2^{n-1}-to-1 MUX to implement n variable function. In this case n-1 will be selection inputs and a single data input.
- Any of the variables can be taken as data and select inputs and implementation will vary.
- Smaller MUXes can be combined to design a larger Mux. Under such designs the most significant select variables should be connected to select inputs of last MUX in the hierarchy.
- Three-state buffers can also be used to construct larger MUX.

Function implementation using MUX

An Example: F(A,B,C)=S(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

Fig. 4-27 Implementing a Boolean Function with a Multiplexer

Function Implementation using MUX

Procedure: The following steps should be followed to implement given function:

- assign an ordering sequence of the input variable
- the rightmost variable (D) will be used for the input lines
- assign the remaining n-1 variables to the selection lines w.r.t. their corresponding sequence
- construct the truth table
- consider a pair of consecutive minterms starting from m_0
- determine the input lines

Note: Alternately we can also use implementation Table method to realize a given function using MUXes of any size.

Multiplexers used in Functions Implementation

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	

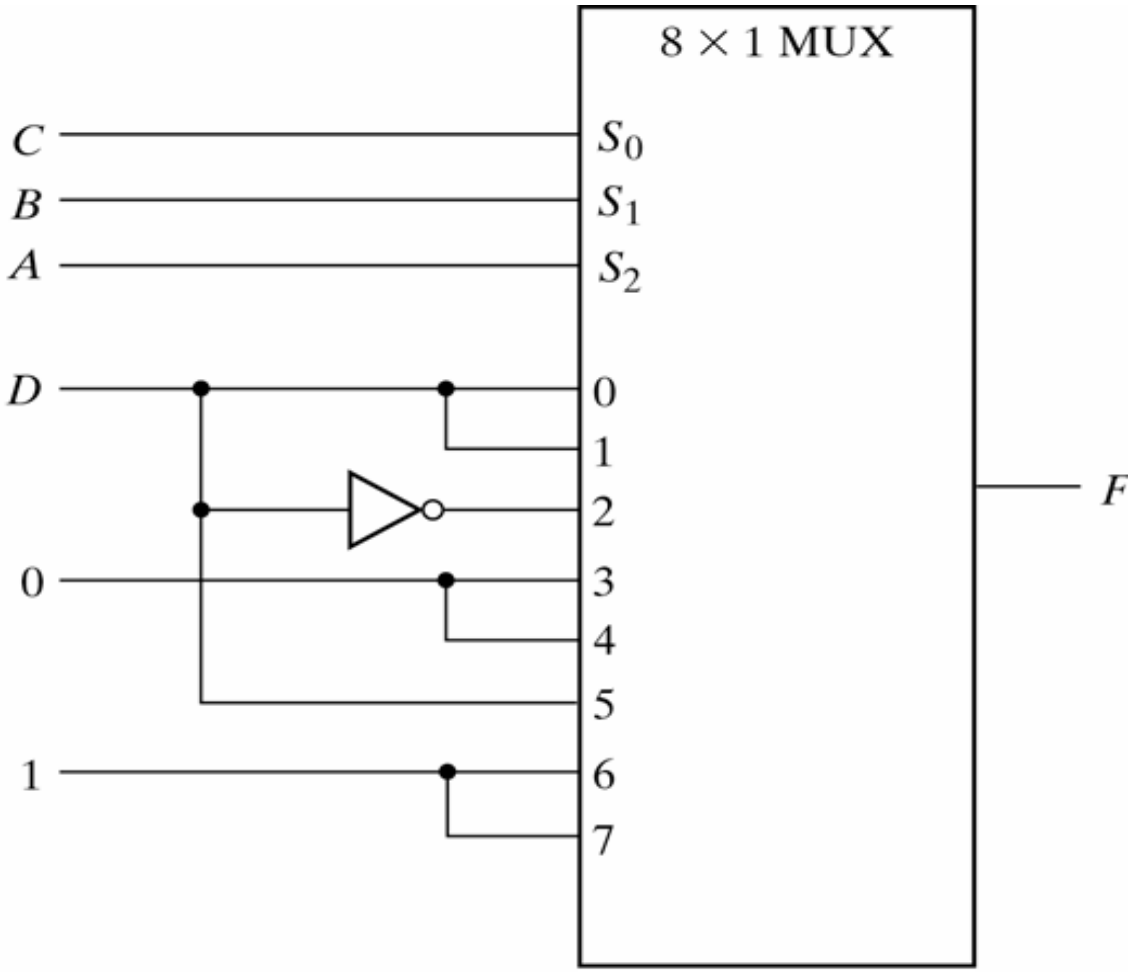
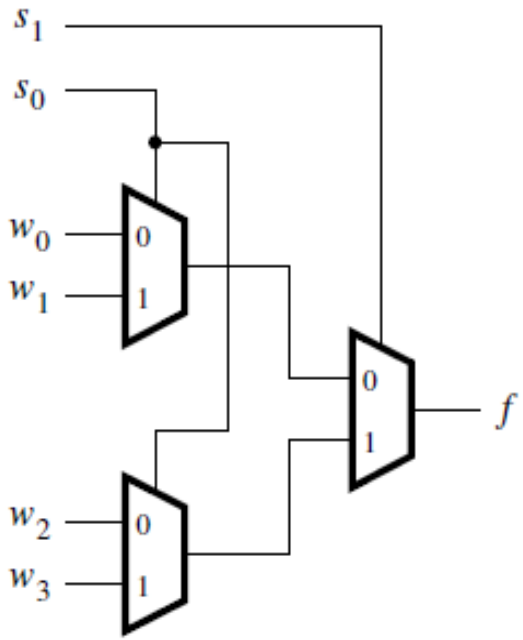
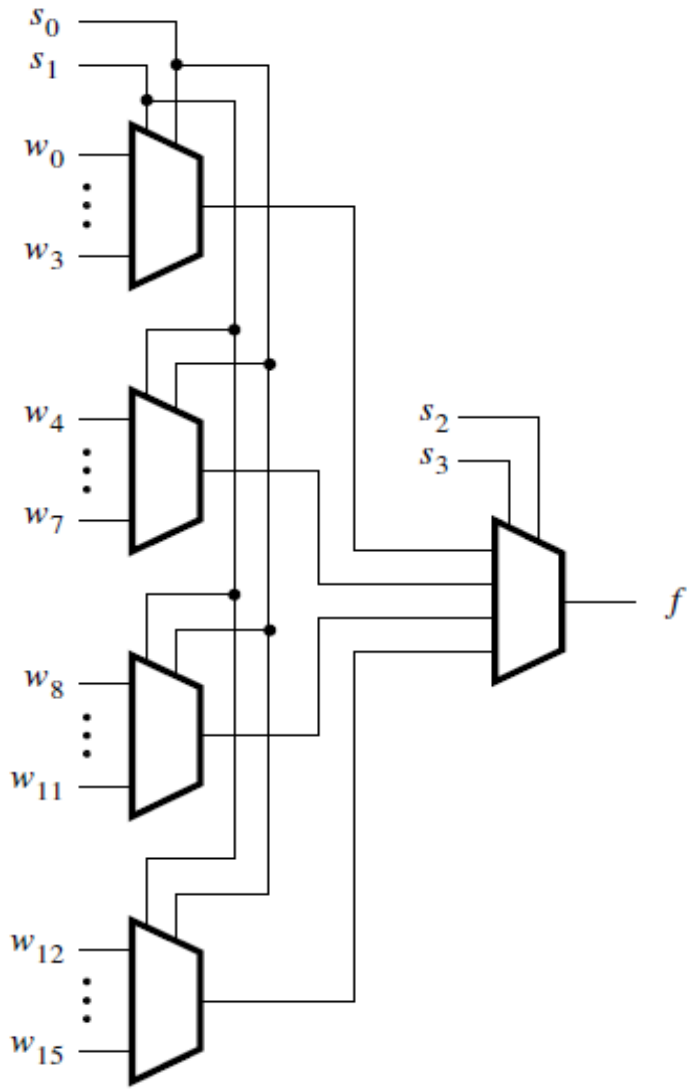


Fig. 4-28 Implementing a 4-Input Function with a Multiplexer

Cascading Multiplexers





Problem Solving Session

Problem: 4-33

Implement a full adder with two 4 x 1 multiplexers. Take X, Y as selection inputs and Z as the data input.

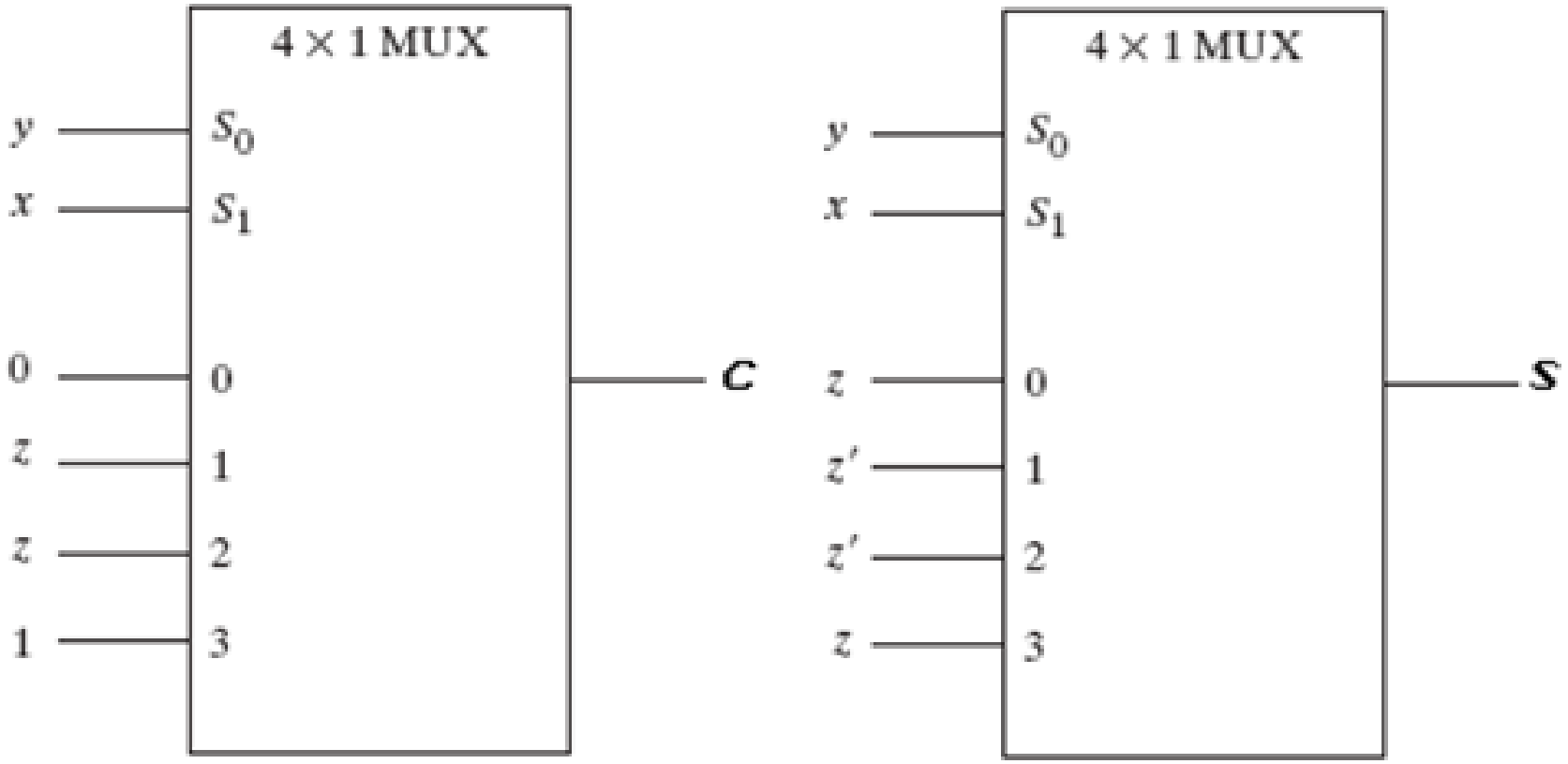
Solution:

Design procedure:

1. Derive the truth table that defines the required relationship between inputs and outputs.

X	Y	Z	C	C	S	S
0	0	0	0	C=0	0	S=Z
0	0	1	0		1	
0	1	0	0	C=Z	1	S=Z′
0	1	1	1		0	
1	0	0	0	C=Z	1	S=Z′
1	0	1	1		0	
1	1	0	1	C=1	0	S=Z
1	1	1	1		1	

2. We connect the first two variables of the functions to the selection inputs of the multiplexer. The remaining single variable of the function is used for the data inputs.



Problem 4-35

Implement the following Boolean functions with a 4:1 multiplexer and external gates.

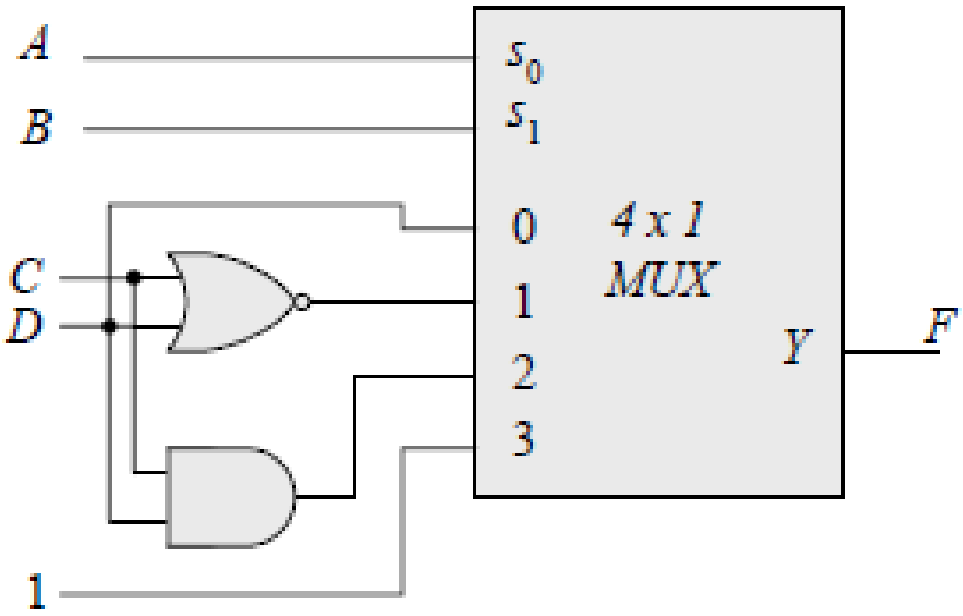
a) $F(A,B,C,D)=\sum(1,3,4,11,12,13,14,15)$

b) $F(A,B,C,D)=\sum(1,2,4,7,8,9,10,11,13,15)$

Connect inputs A and B to the selection lines. The input requirements for the data lines will be a function of variables C and D. These values are obtained by expressing F as a function of C and D for each of the four cases when AB=00, 01, 10, and 11. The function may have to be implemented with external gates and with connections to power and ground.

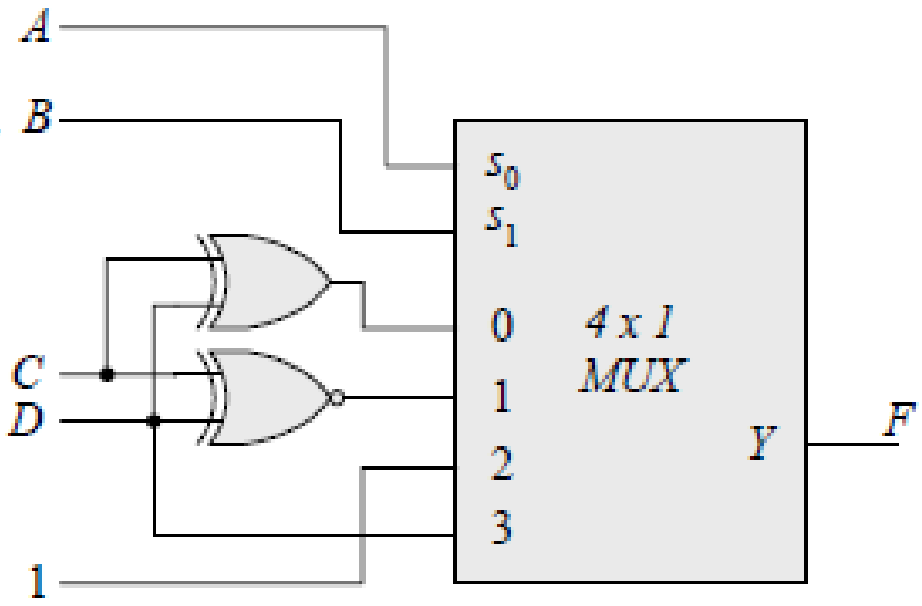
(a)

Inputs ABCD	*F*	
0000	0	
0001	1	$AB = 00$
0010	0	$F = D$
0011	1	
0100	1	
0101	0	$AB = 01$
0110	0	$F = C'D'$
0111	0	$= (C + D)'$
1000	0	
1001	0	$AB = 10$
1010	0	$F = CD$
1011	1	
1100	1	
1101	1	$AB = 11$
1110	1	$F = 1$
1111	1	



(b)

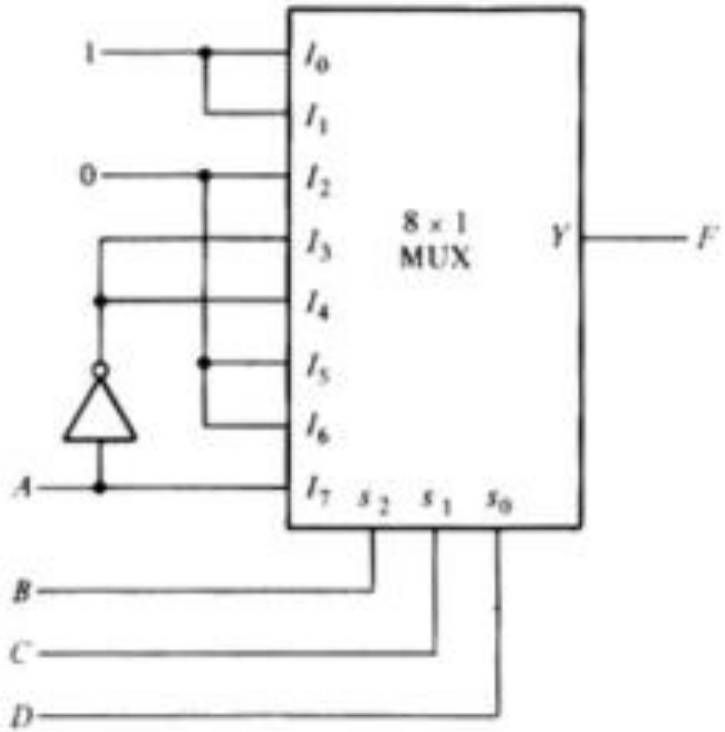
Inputs ABCD	F	
0000	0	
0001	1	$AB = 00$
0010	1	$F = C'D + CD'$
0011	0	
0100	1	
0101	0	$AB = 01$
0110	0	$F = C'D' + CD$
0111	1	
1000	1	
1001	1	$AB = 10$
1010	1	$F = 1$
1011	1	
1100	0	
1101	1	$AB = 11$
1110	0	$F = D$
1111	1	



Boolean Function Implementation

- Implement the following function with multiplexer: F(A,B,C,D)= ∑(0,1,3,4,8,9,15).

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



The End