LOGIC DESIGN EET-1021

CHAPTER 04

Lecture 23

Combinational Logic

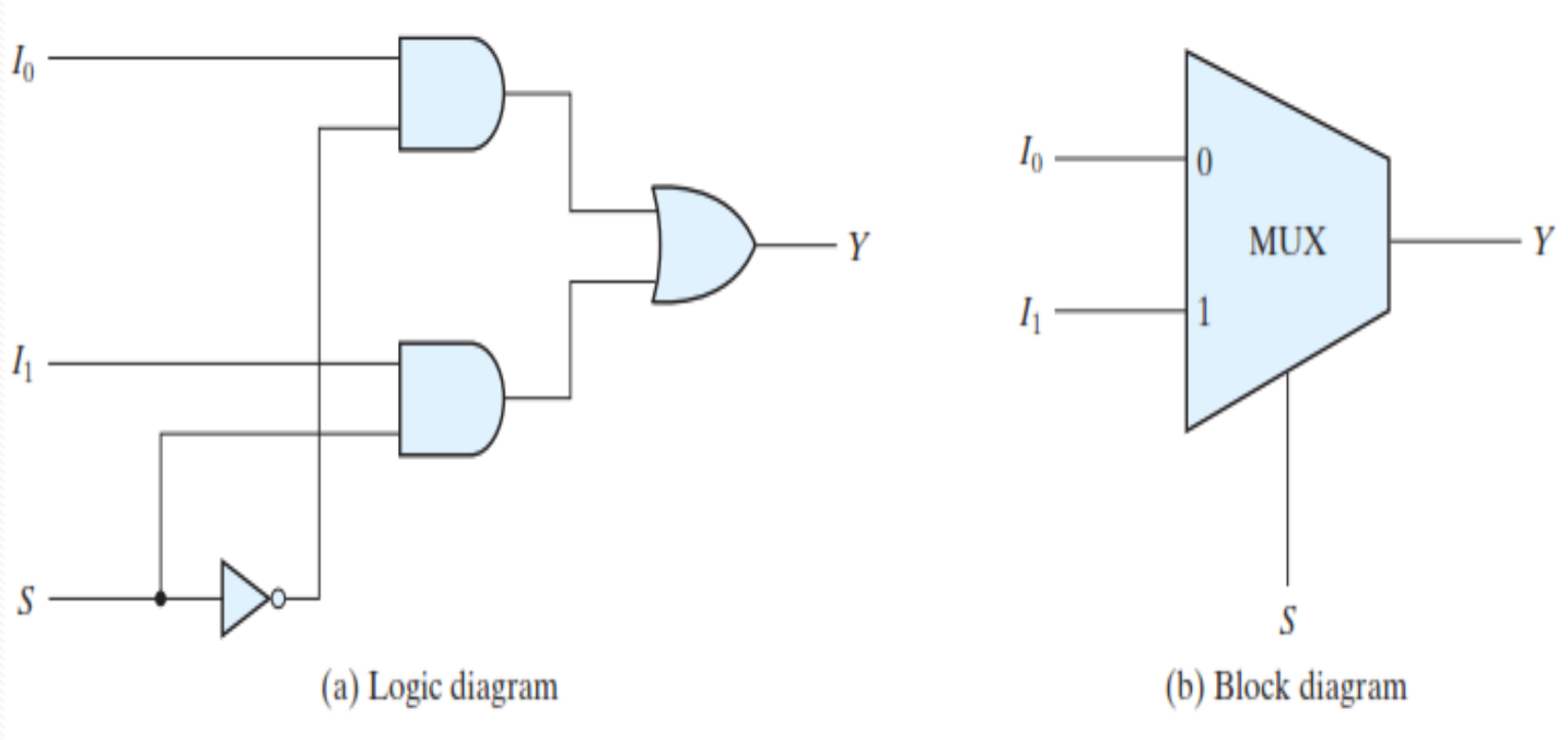
Overview of previous lecture

- **What is Decoder**
- **What is Encoder**
- **Limitation of Encoder**
- **Priority Encoder**

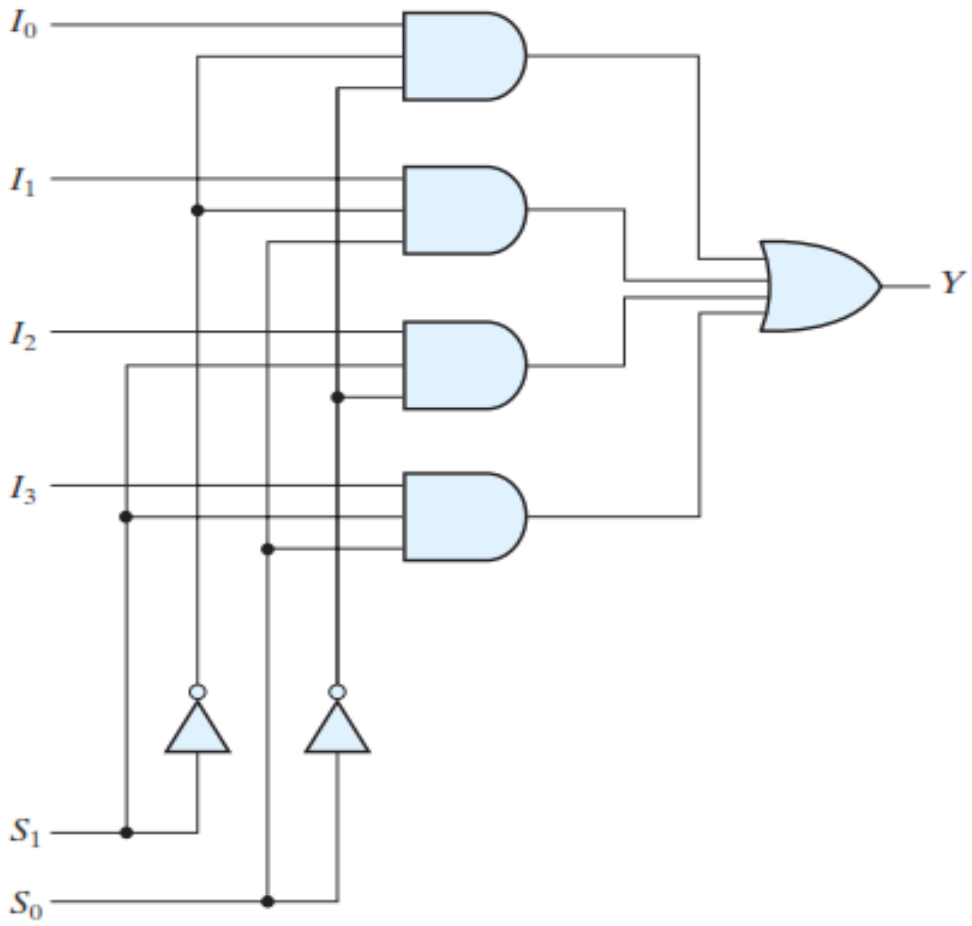
Multiplexers

- 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.

- In general, 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.



Two-to-one-line multiplexer



(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

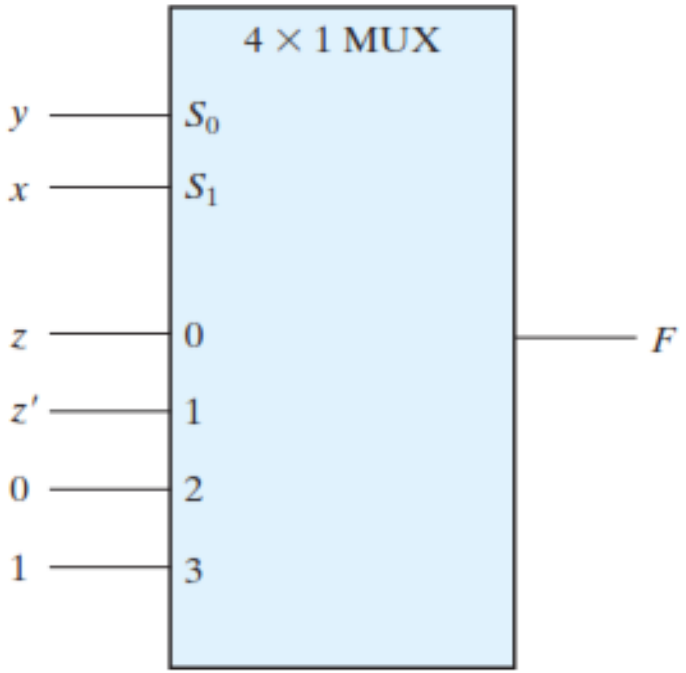
Four-to-one-line multiplexer

Boolean Function Implementation

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

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

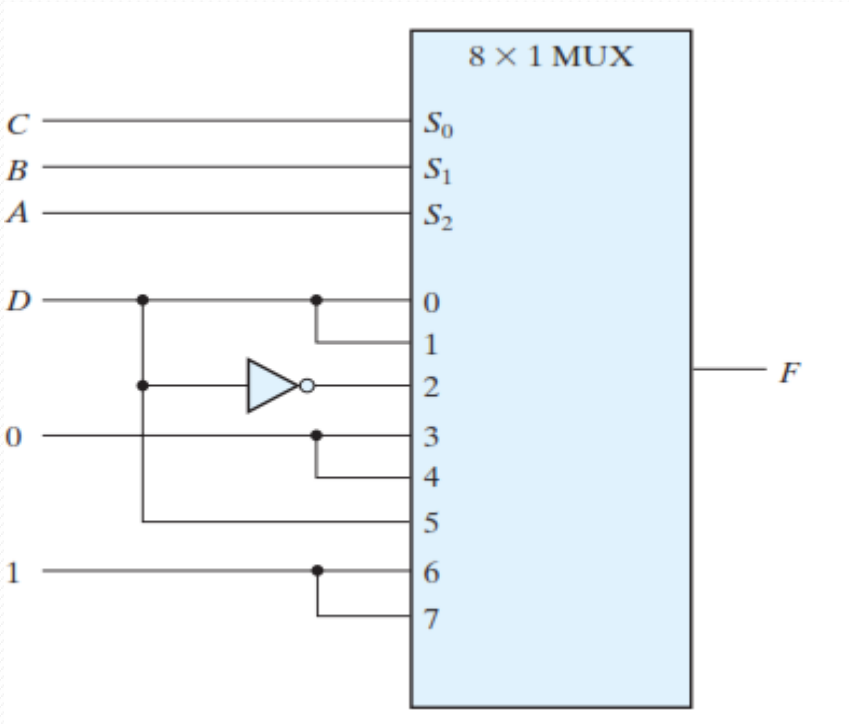


(b) Multiplexer implementation

As a second example, consider the implementation of the Boolean function

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

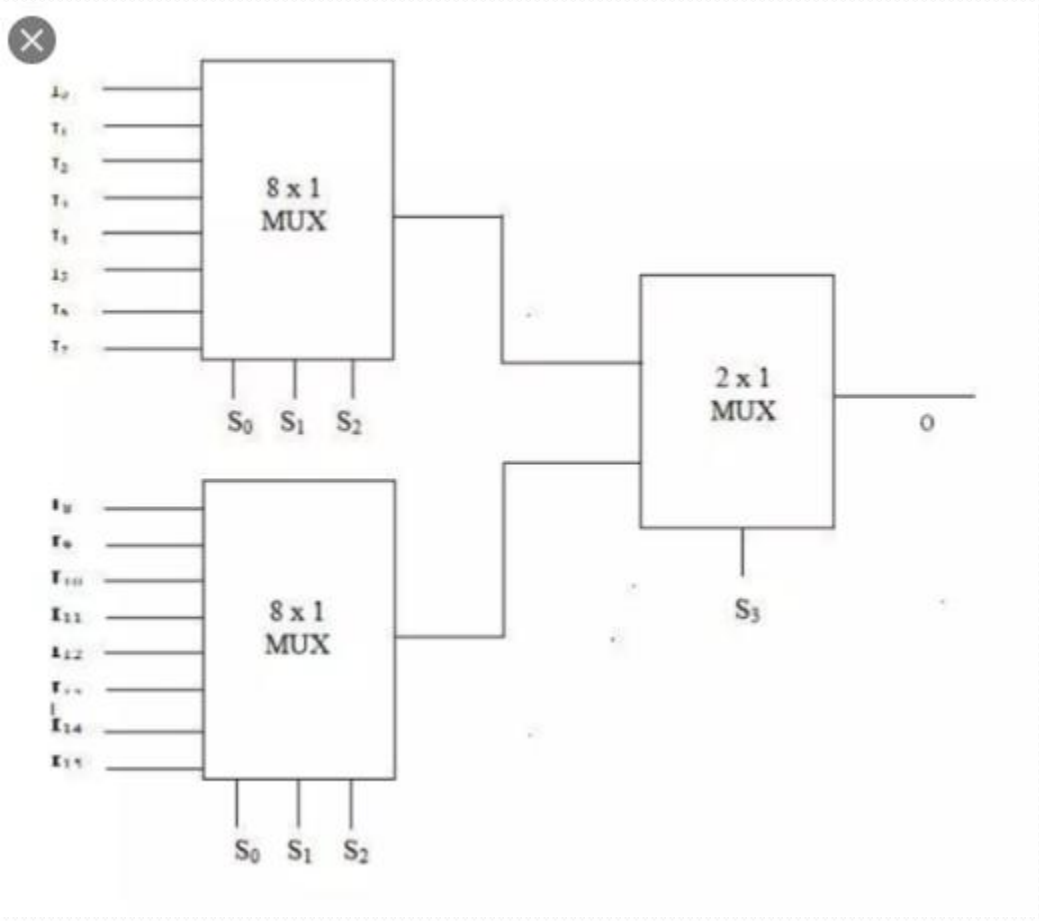
A	B	C	D	F	
0	0	0	0	0	$F = D$
0	0	0	1	1	
0	0	1	0	0	$F = D$
0	0	1	1	1	
0	1	0	0	1	$F = D'$
0	1	0	1	0	
0	1	1	0	0	$F = 0$
0	1	1	1	0	
1	0	0	0	0	$F = 0$
1	0	0	1	0	
1	0	1	0	0	$F = D$
1	0	1	1	1	
1	1	0	0	1	$F = 1$
1	1	0	1	1	
1	1	1	0	1	$F = 1$
1	1	1	1	1	



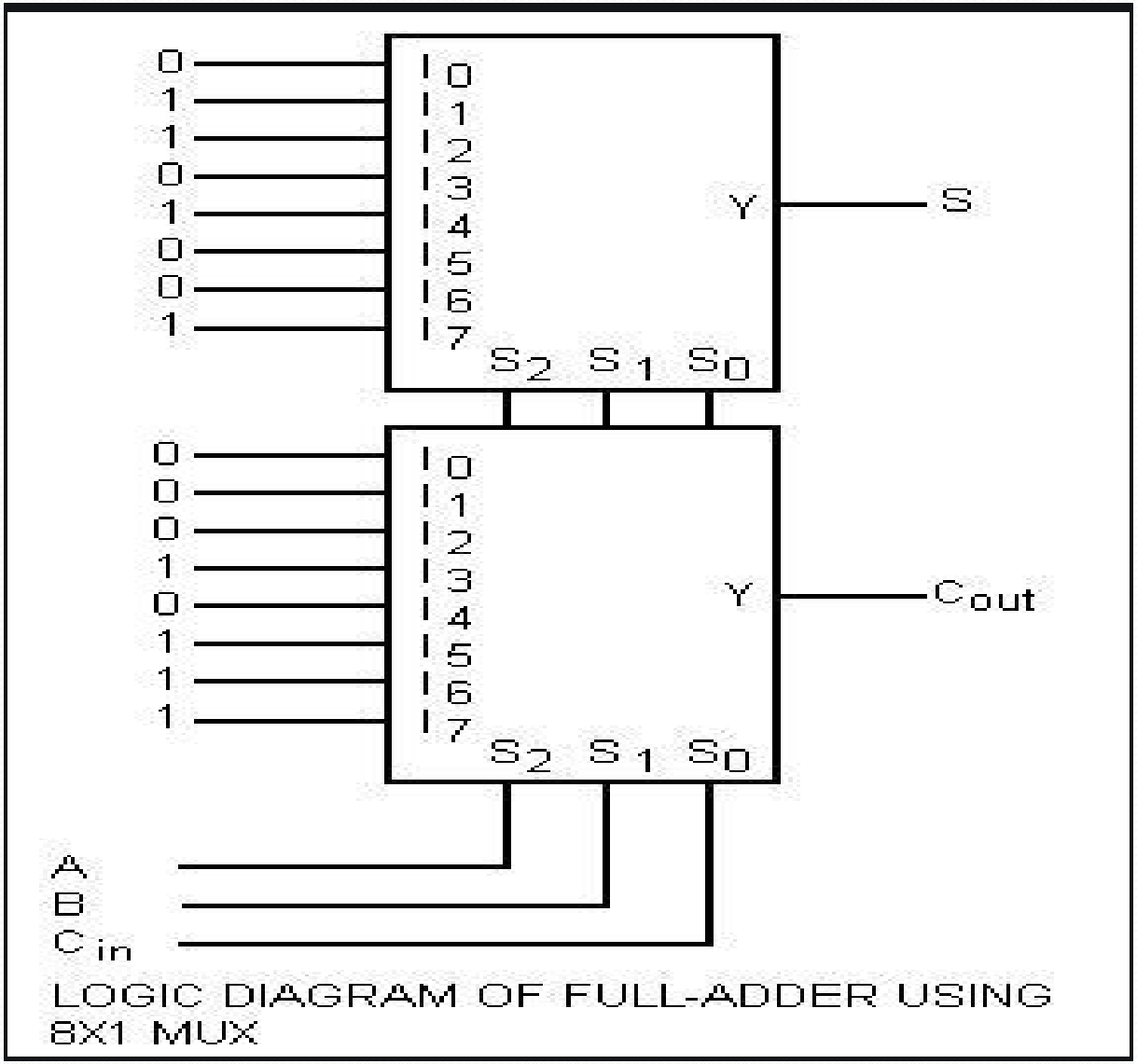
Implementing a four-input function with a multiplexer

Example

Construct a 16×1 multiplexer with two 8×1 and one 2×1 multiplexers.



Implement a full adder with two 8×1 multiplexers.

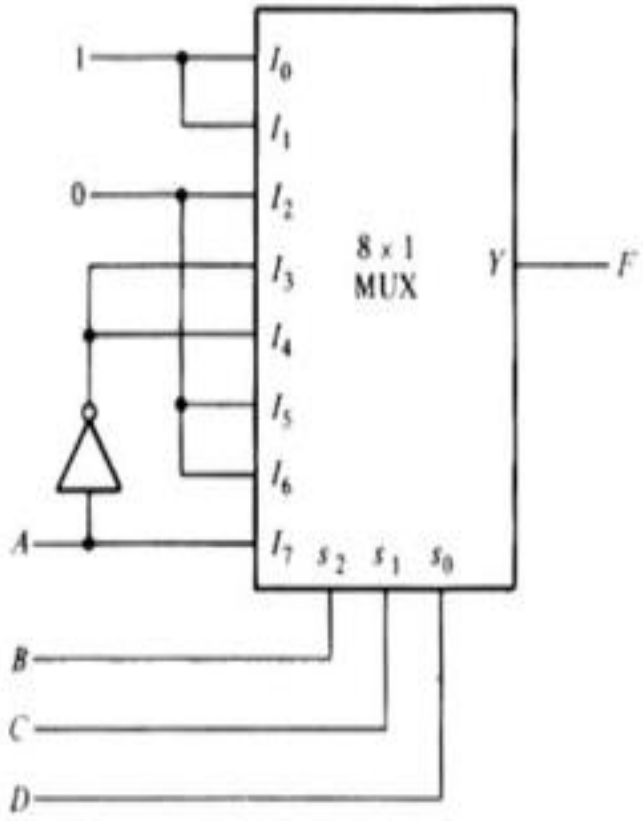


LOGIC DIAGRAM OF FULL-ADDER USING 8X1 MUX

Implement the given Boolean function by using 8× 1 multiplexer.

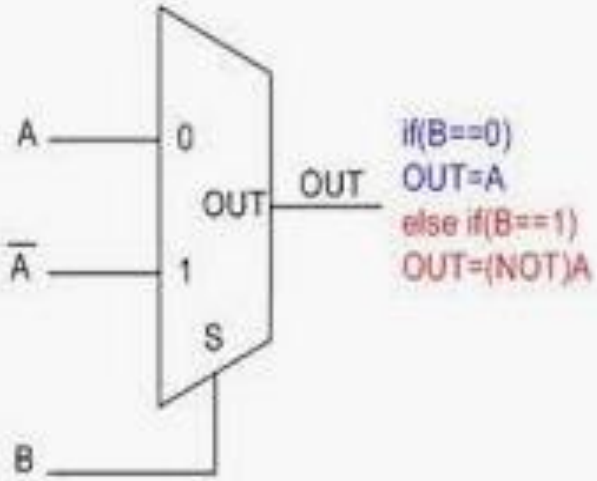
$F(A,B,C,D) = \sum(0,1,3,4,8,9,15)$

	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



Implement XOR gate by using 2 × 1 Multipler

Inputs		Output
A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0



THANK YOU