CS 221
Logic Design

2023

DR.SOHA AHMED EHSSAN

Combinational circuits

LECTURE 6

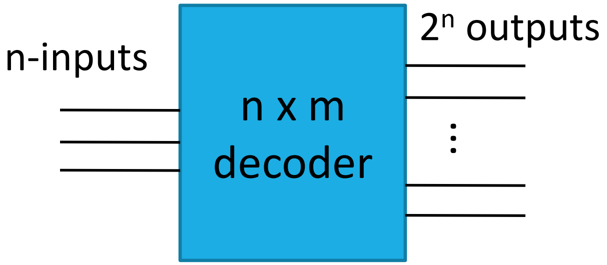
Remember

This chapter includes the most important standard combinational circuits:

- Adders, Subtractors, Comparators, Decoders, Encoders, and Multiplexers

We will know their internal design and the functionality of each.

But, remember **our aim is to know how to think to design a circuit**



Decoders

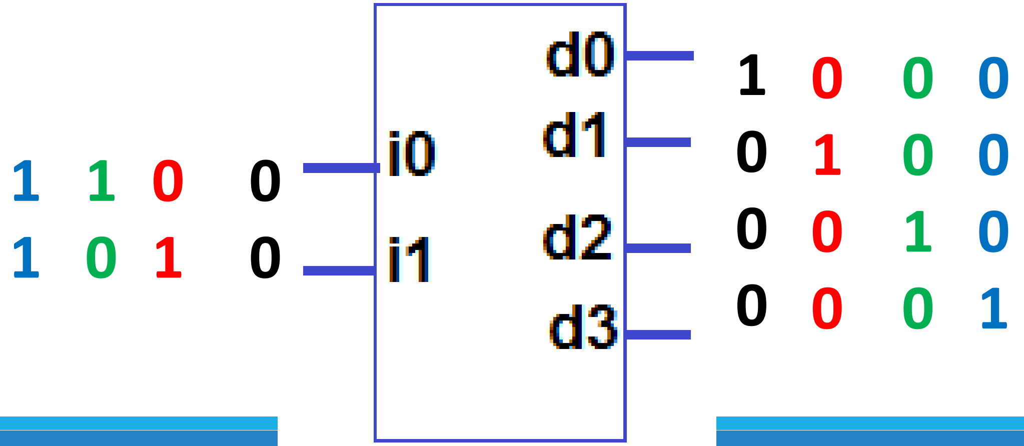
A *Decoder*:

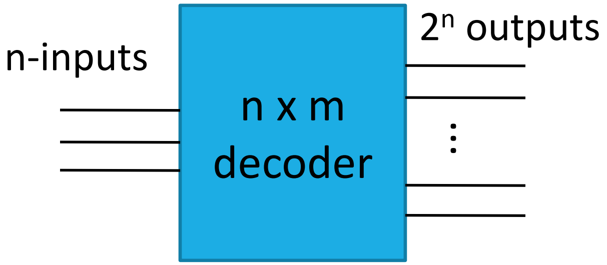
- Is a popular combinational logic building block
- It converts input binary number to one high output

2-input decoder

➔ Has four possible input binary numbers

➔ So, it has four outputs, one for each possible input binary number





Decoders

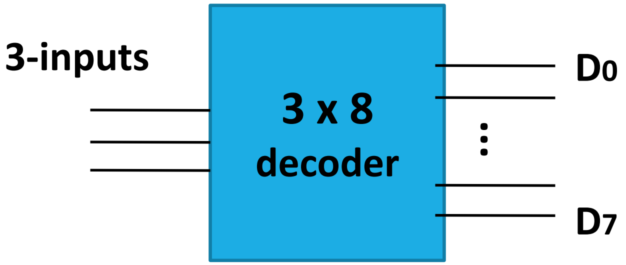
Decoders are called *n-to-m* line decoders, where $m<=2^n$.

A particular application of this decoder is:

Binary-to-octal conversion;

- inputs : binary representation and
- outputs : its correspondence in octal representation.

3-to-8 line Decoder



3 input variables =? Outputs ➔ 2^3=8 outputs

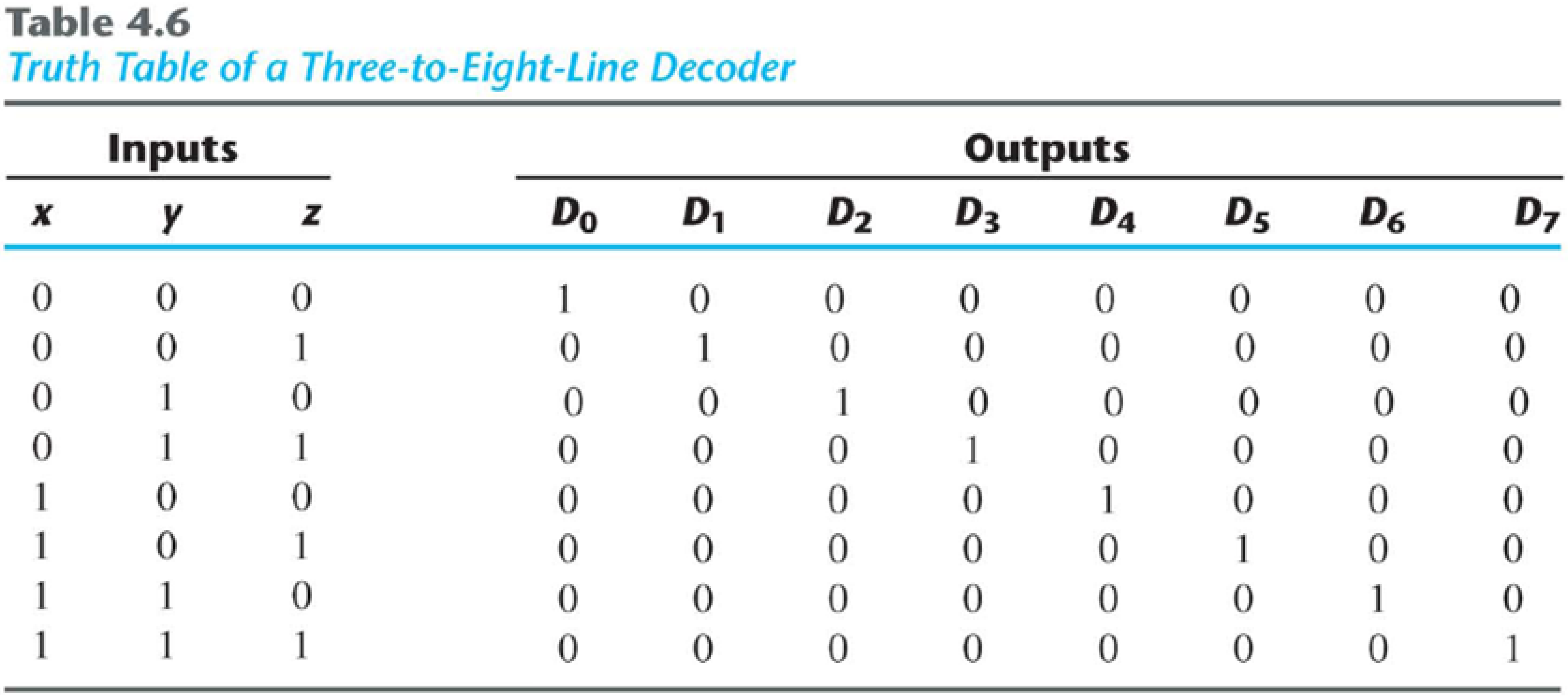
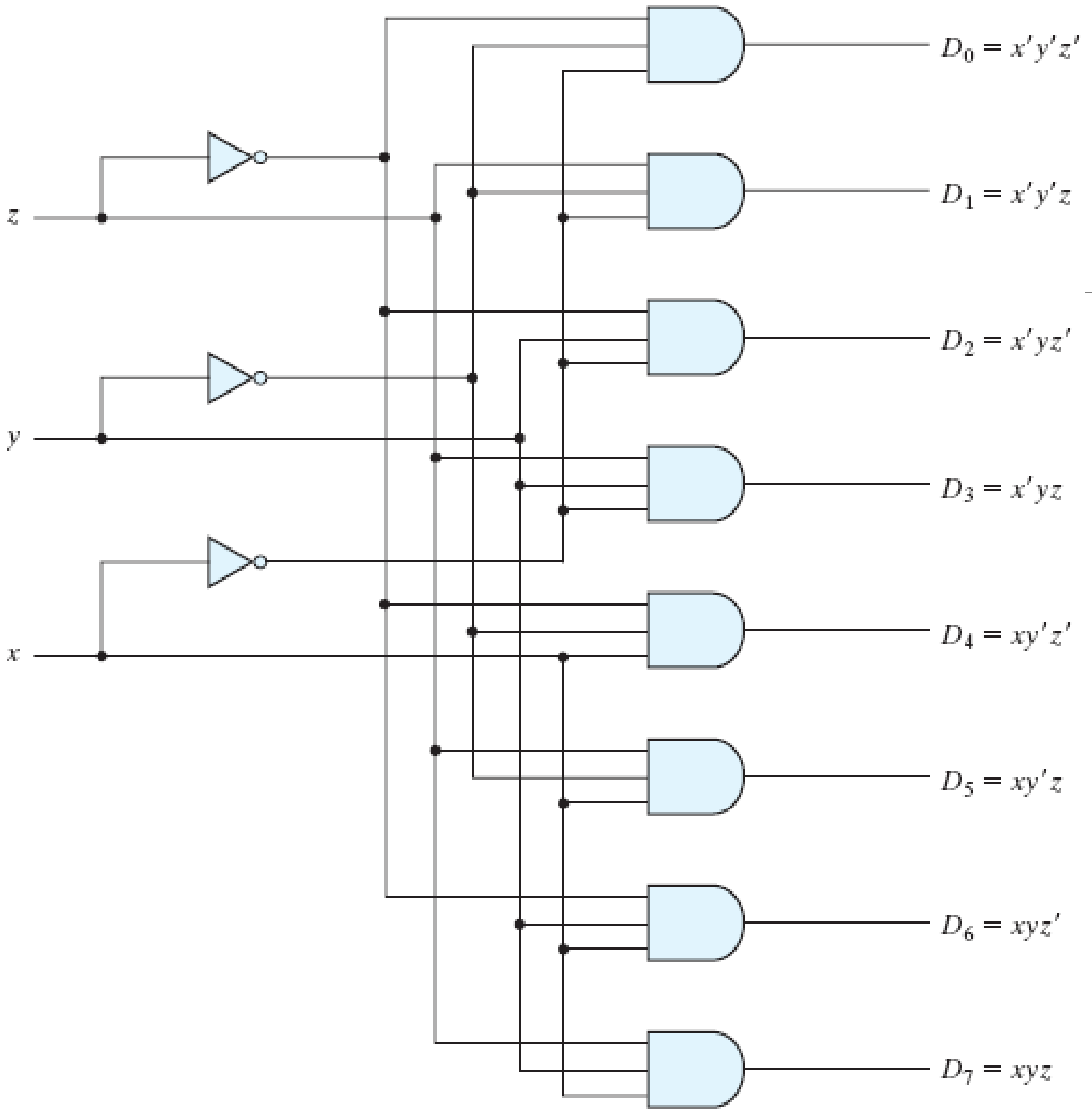


Table 4.6
Truth Table of a Three-to-Eight-Line Decoder

Inputs			Outputs							
x	*y*	*z*	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

D0= x’y’z’, D1=x’y’z, D2=x’yz’, D3=x’yz D4= xy’z’ D5=xy’z D6= xyz’ D7=xyz

Decoders



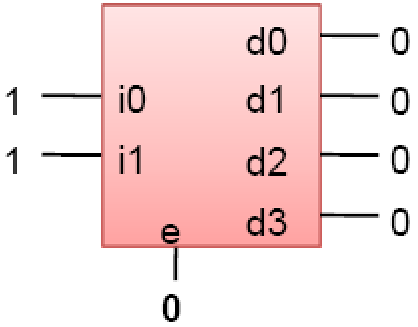
The output whose value is equal to 1 represents ***the minterm*** equivalent of the binary number currently available in the input lines

Decoder with enable input

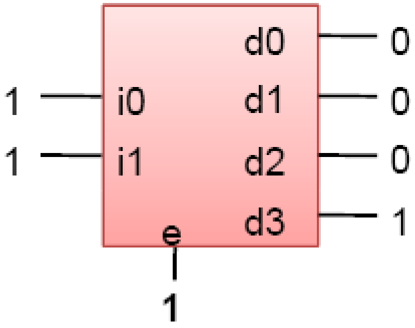
Decoder with enable E

– if E=0, Outputs all 0

– if E=1, Regular behavior



E	i0	i1	d0	d1	d2	d3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1		1		
1	1	0			1	
1	1	1				1

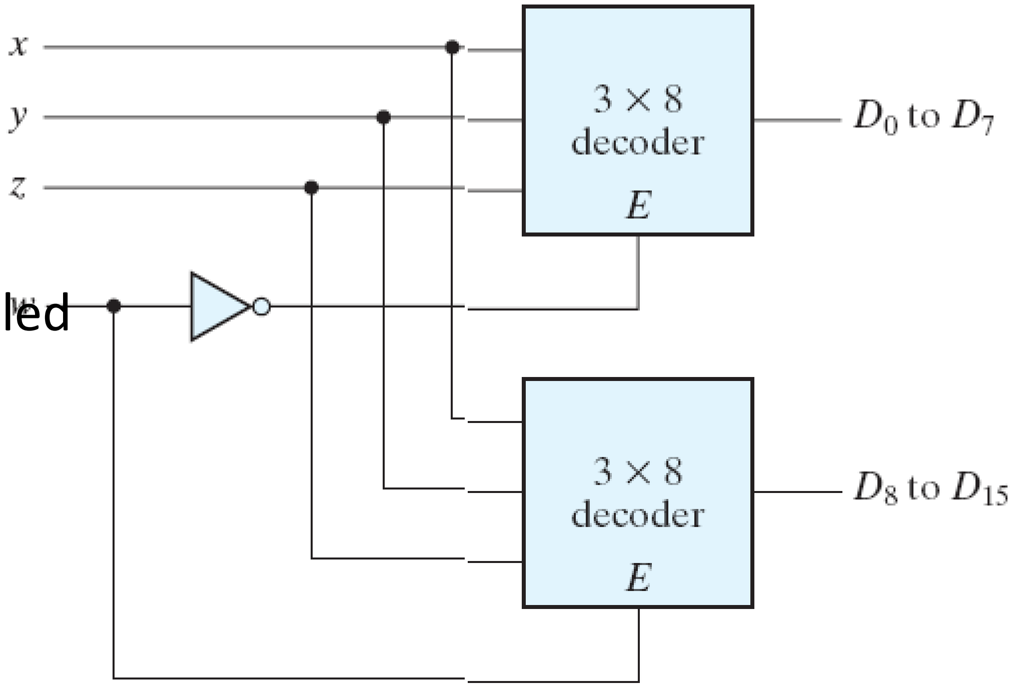


How to use as a block

4-to-16 line Decoder

The 4-to-16 line decoder with two 3-to-8 line decoder with enable inputs

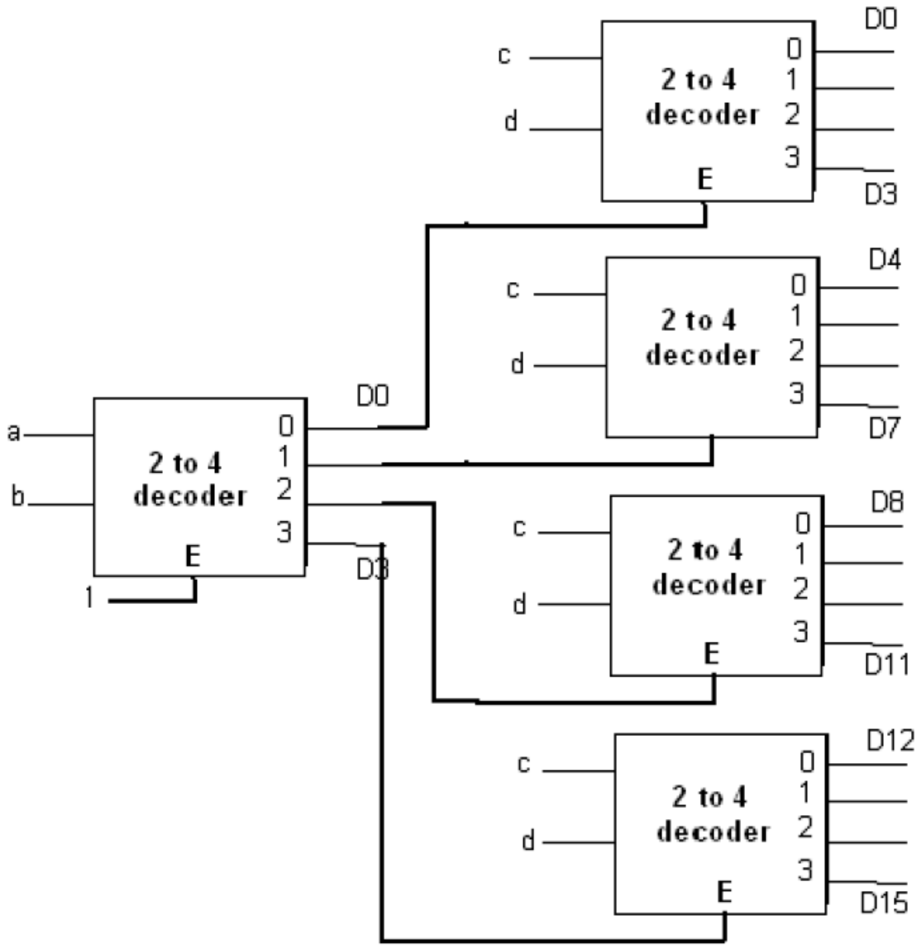
- When w=0,
  - the top decoder is enabled and the other is disabled
  - The bottom decoder outputs all 0’s and the top eight outputs generate minterms 0000 to 0111
- When w=1,
  - the bottom decoder is enabled and the other is disabled
  - The top decoder outputs all 0’s and the bottom eight outputs generate minterms 1000 to 1111



Exercise

Can you sketch a 4×16 decoder using a number of 2×4 decoders?

4×16 decoder using a number of 2×4 decoders



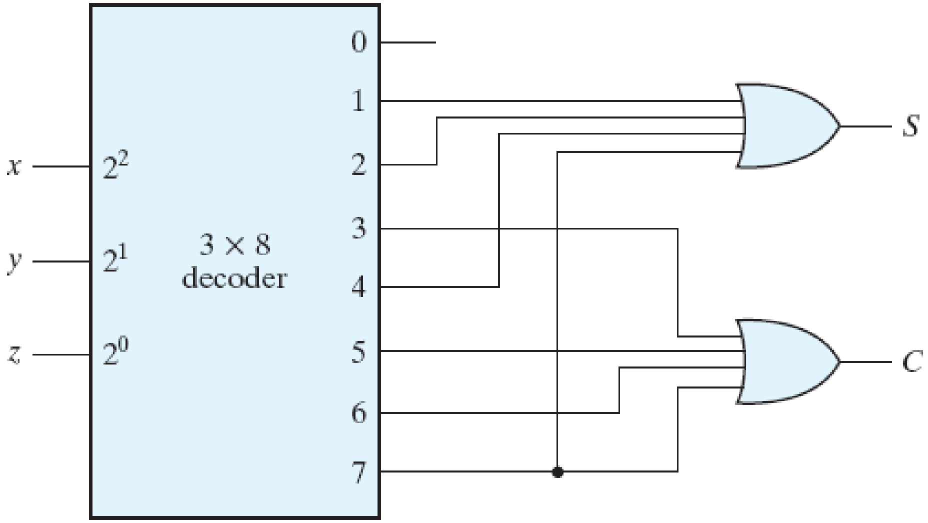
How to use as a block

Boolean function implementation

Since any Boolean function can be expressed in sum of minterms form:

➔ A decoder with an external OR gate provides an implementation of the function Ex: Full-adder functions

S(x,y,z)= Σ(1,2,4,7) C(x,y,z)=Σ(3,5,6,7)



Full-adder with a decoder

Encoders

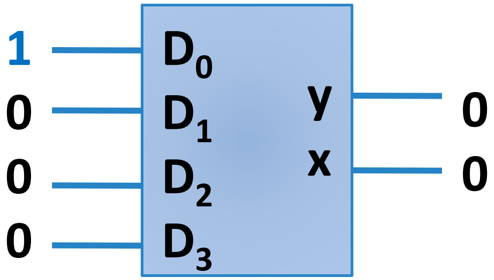
An encoder

- It is a digital circuit that performs the inverse operation of a decoder
- It has 2^n inputs lines and n output lines

An example of an encoder is:

- Octal-to-binary conversion, it has:
  - 8 inputs : the octal code and
  - 3 outputs : the corresponding binary number

It is assumed that only one input has a value of 1 at any given time



Octal to binary Encoder

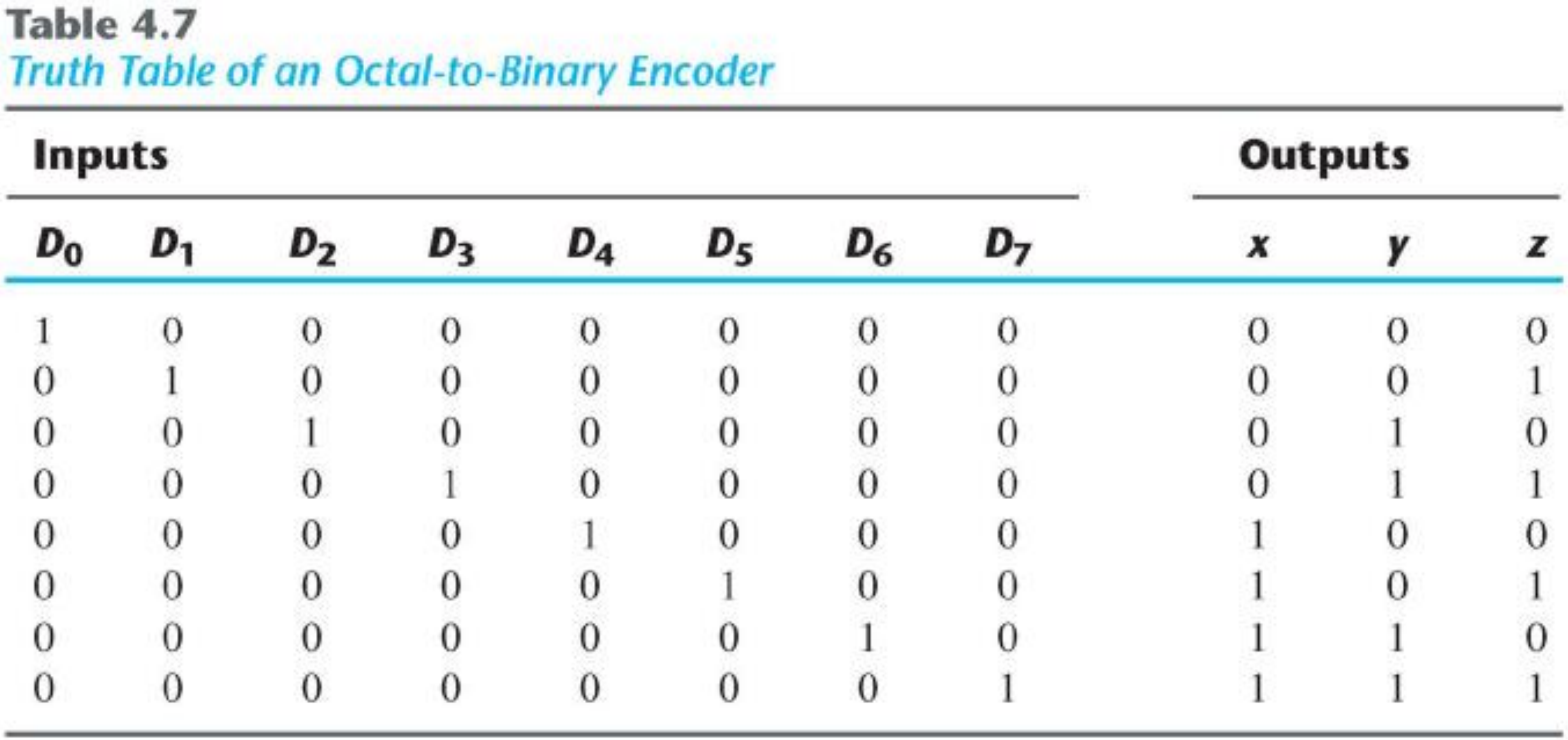


Table 4.7

Truth Table of an Octal-to-Binary Encoder

Inputs								**Outputs**		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

It can be implemented with 3 OR gates

$$z = D_1 + D_3 + D_5 + D_7$$
$$y = D_2 + D_3 + D_6 + D_7$$
$$x = D_4 + D_5 + D_6 + D_7$$

But, it has some limitations

Octal to binary Encoder: limitations

This encoder has some limitations:

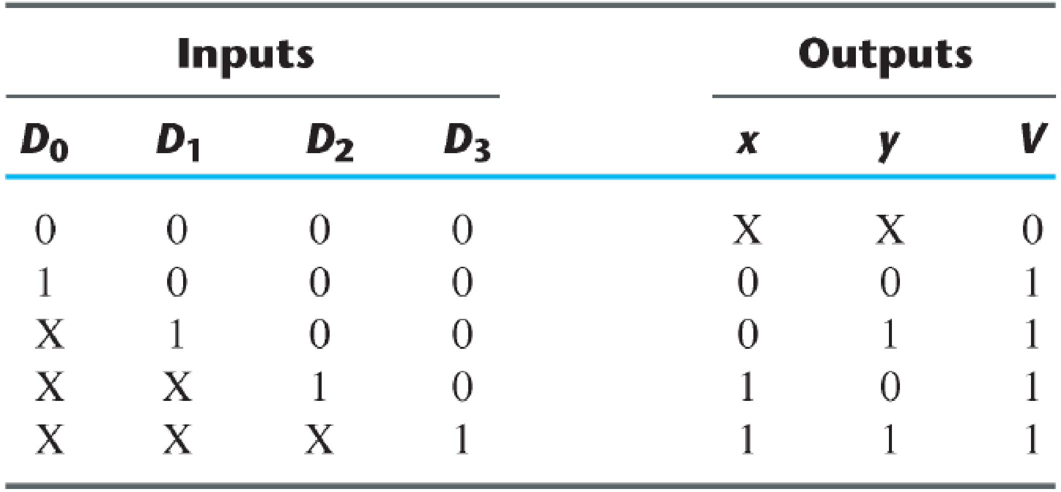
1. It is assumed that only one input can be active at any given time,
   - If 2 inputs are active simultaneously, the output produces an undefined combination
   - To resolve this ➔ higher priority with higher subscript
2. A output with all 0's is generated when all the input are 0;
   - But this output is the same as when D0 is equal to 1
   - To resolve this, ➔ one more output is used to indicate whether at least one input is equal to 1.

How to design?
Know internal ?

Priority encoder

A *priority encoder* is an encoder circuit that overcomes the limitations of octal-to-binary encoder previously implemented.

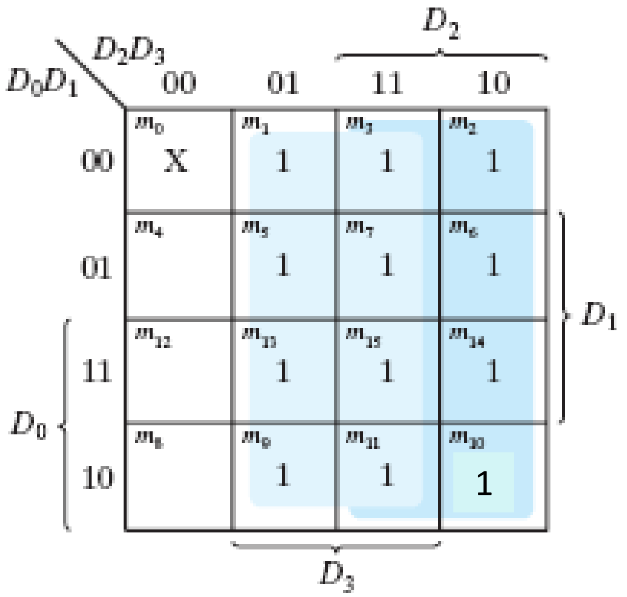
Table 4.8
Truth Table of a Priority Encoder



Inputs				Outputs		
D_0	D_1	D_2	D_3	x	y	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

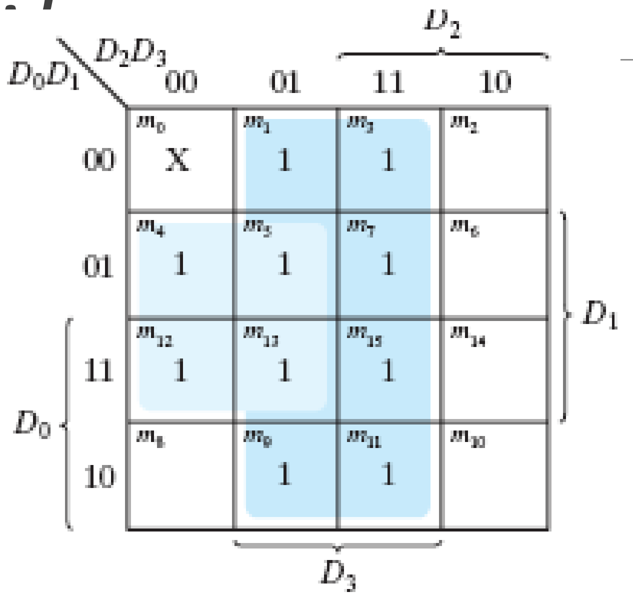
How to design?
Know internal ?

Priority encoder(Cont.)



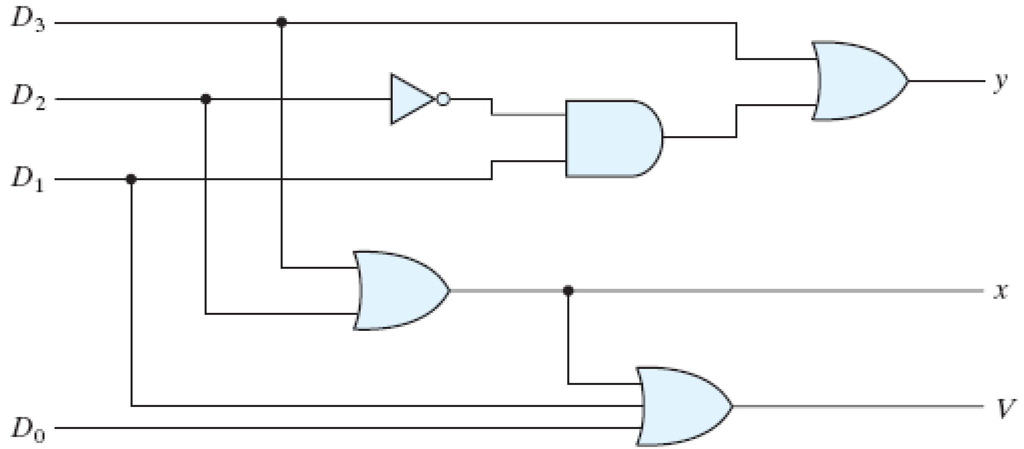
D_0D_1 \ D_2D_3	00	01	11	10
00	m_0 X	m_1 1	m_3 1	m_2 1
01	m_4	m_5 1	m_7 1	m_6 1
11	m_{12}	m_{13} 1	m_{15} 1	m_{14} 1
10	m_8	m_9 1	m_{11} 1	m_{10} 1

$x = D_2 + D_3$



D_0D_1 \ D_2D_3	00	01	11	10
00	m_0 X	m_1 1	m_3 1	m_2
01	m_4 1	m_5 1	m_7 1	m_6
11	m_{12} 1	m_{13} 1	m_{15} 1	m_{14}
10	m_8	m_9 1	m_{11} 1	m_{10}

$y = D_3 + D_1D'_2$



Multiplexers

A multiplexer (MUX) is a combinational circuit that

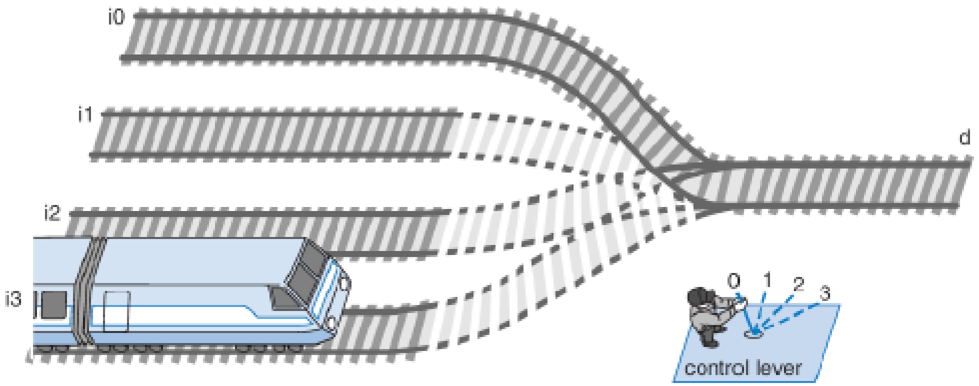
- selects binary information from one of many input lines and
- directs it to a single output line.

It is also called a

Data selector

There are:

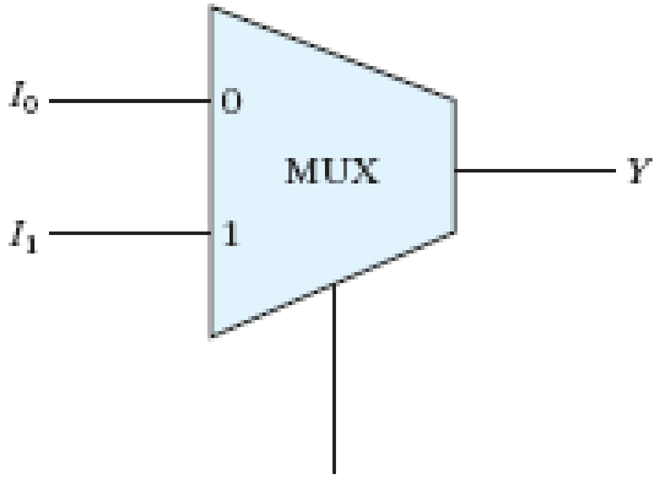
- 2^n input lines
- *n* selection lines whose bit combinations determine which input is selected, and
- Single output



Two-to-one multiplexer

It has:

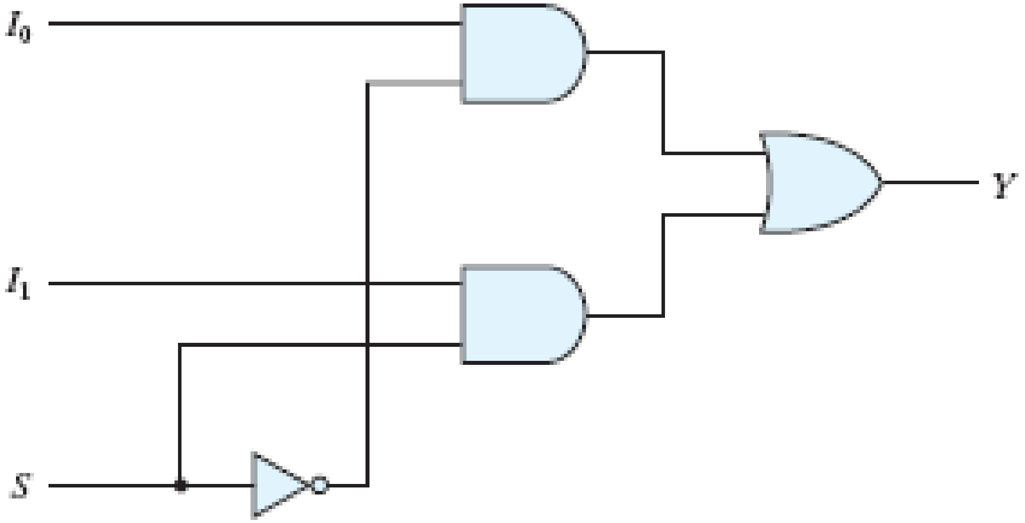
- 2 inputs, 1 selection and 1 output



(b) Block diagram

S	Y
0	I_0
1	I_1

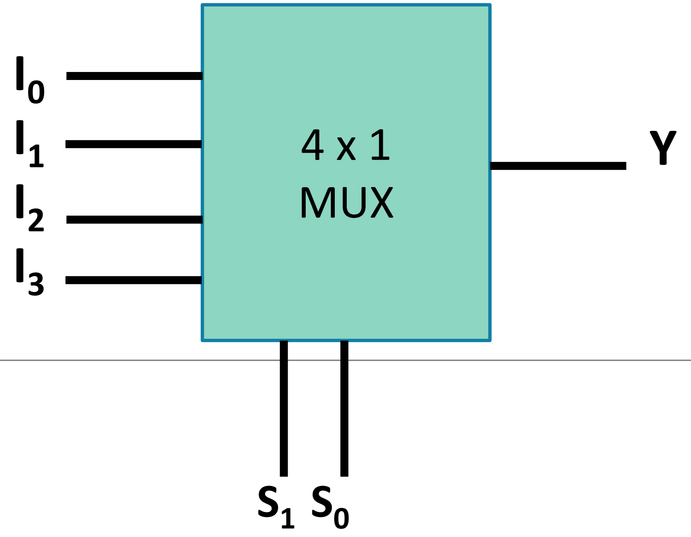
$Y= S'I_0 + SI_1$



(a) Logic diagram

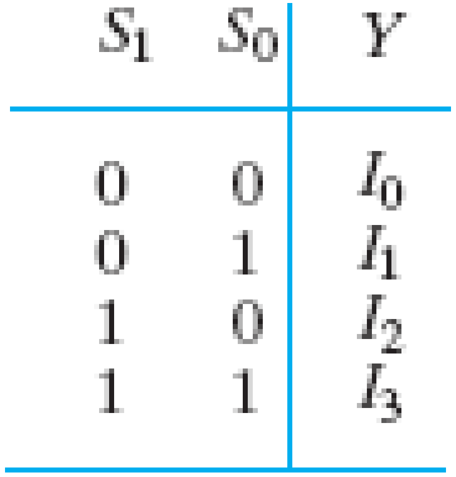
How to design?
Know internal ?

Four-to-one multiplexer



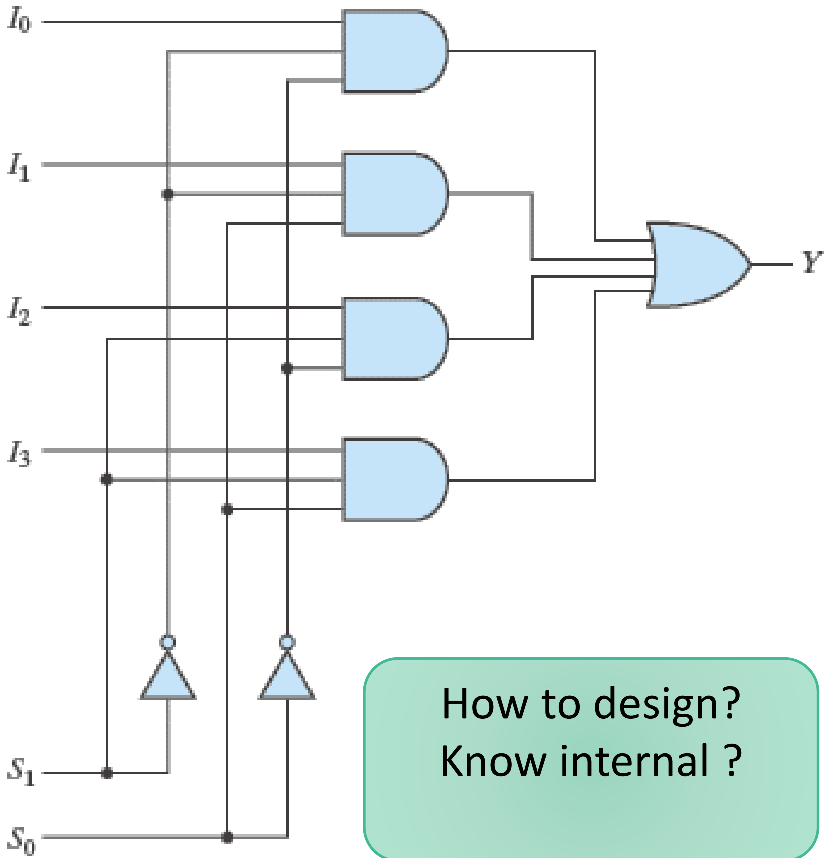
It has:

- 4 inputs, 2 selection and 1 output



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

$$Y= S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$$



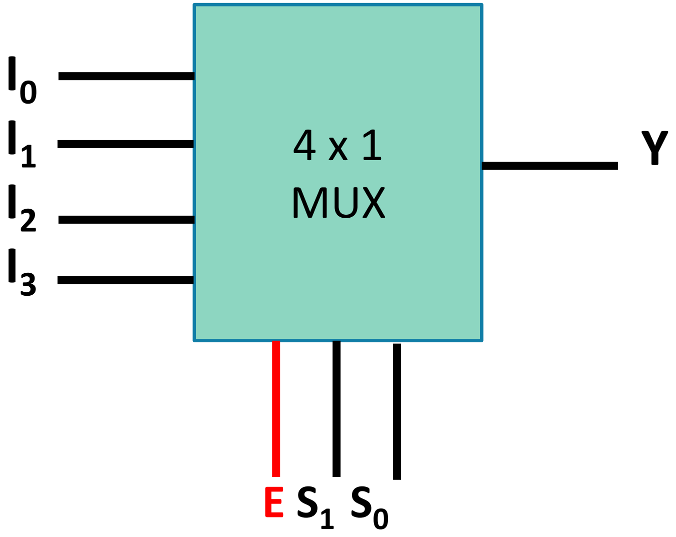
How to design?
Know internal ?

Multiplexer with enable input

As decoder, multiplexer may have an enable input:

- When E=0, the outputs are disabled
- When E=1, normal multiplexer

E	S1	S0	Y
0	x	X	All 0’s
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3



How to use as a block

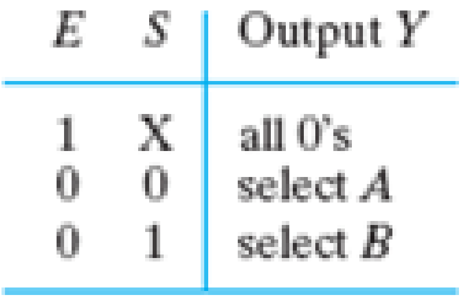
Quadruple 2-to-1 line multiplexer

The circuit has 4 MUXs,

- Each MUX:
  - select one of two input lines: A_0 or B_0
  - Output Y_0
  - Selection line S selects one MUX from the 4 MUXs.

It can be seen as a circuit that selects one of two 4-bit numbers

The enable input E must be active for normal operation.



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

Function table

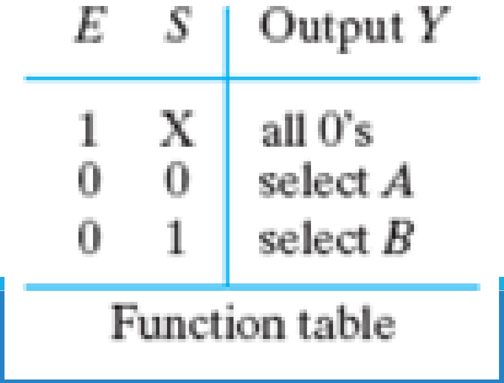
Quadruple 2-to-1 line multiplexer (Cont.)

The unit is enabled when E=0

If S=0, the four A inputs have a path to the four outputs

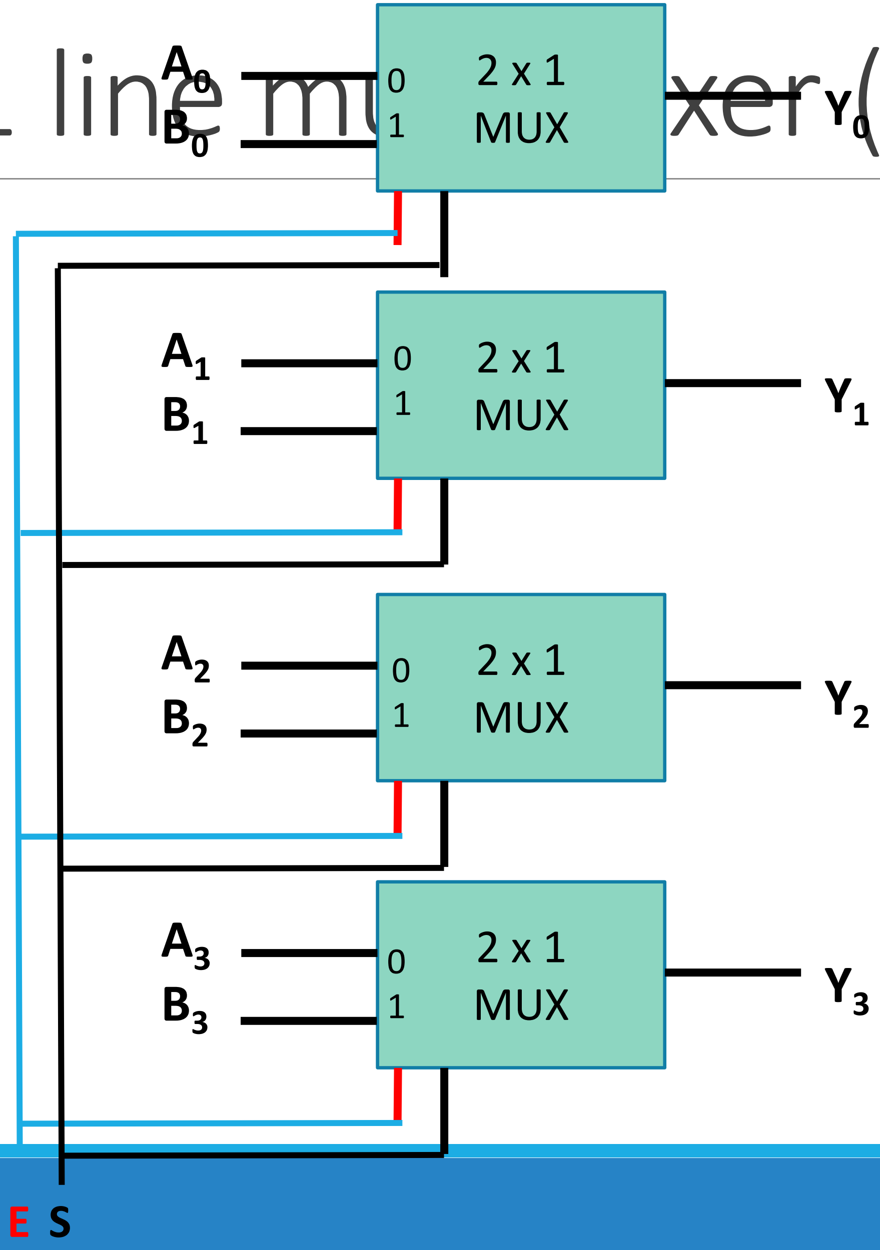
If S=1, the four B inputs have a path to the four outputs

When E=1, the outputs are all 0’s regardless the value of S



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

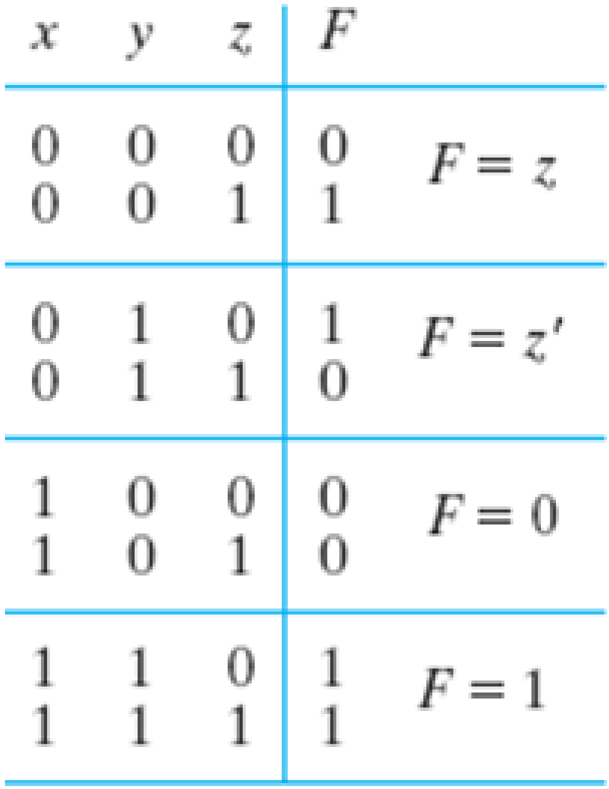
Function table



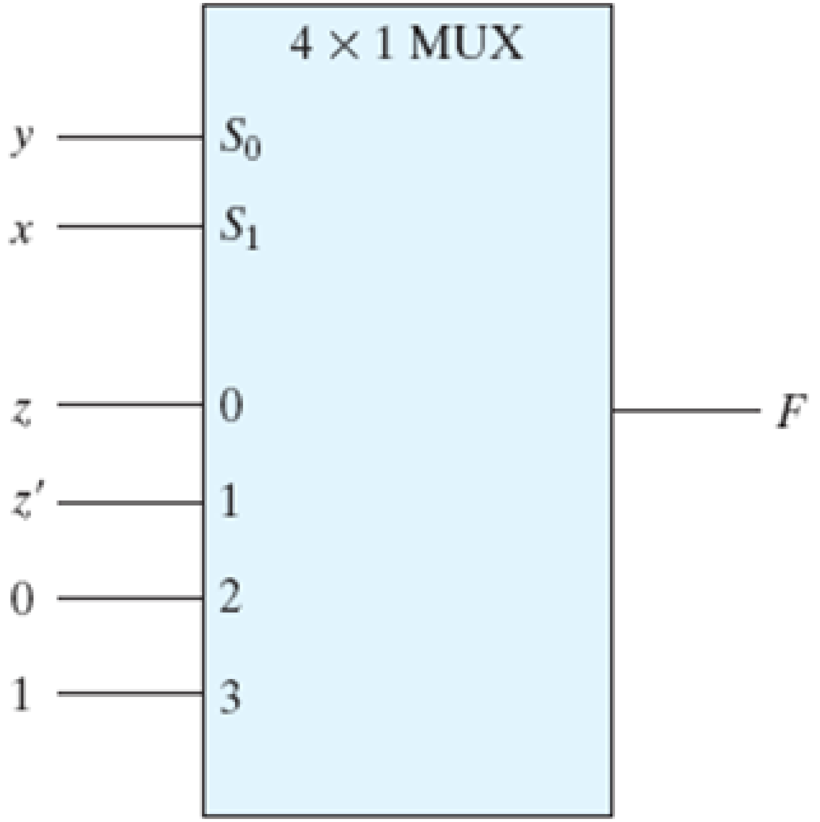
How to use as a block

Boolean function implementation

Example1: *F(x,y,z)=Σ(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	



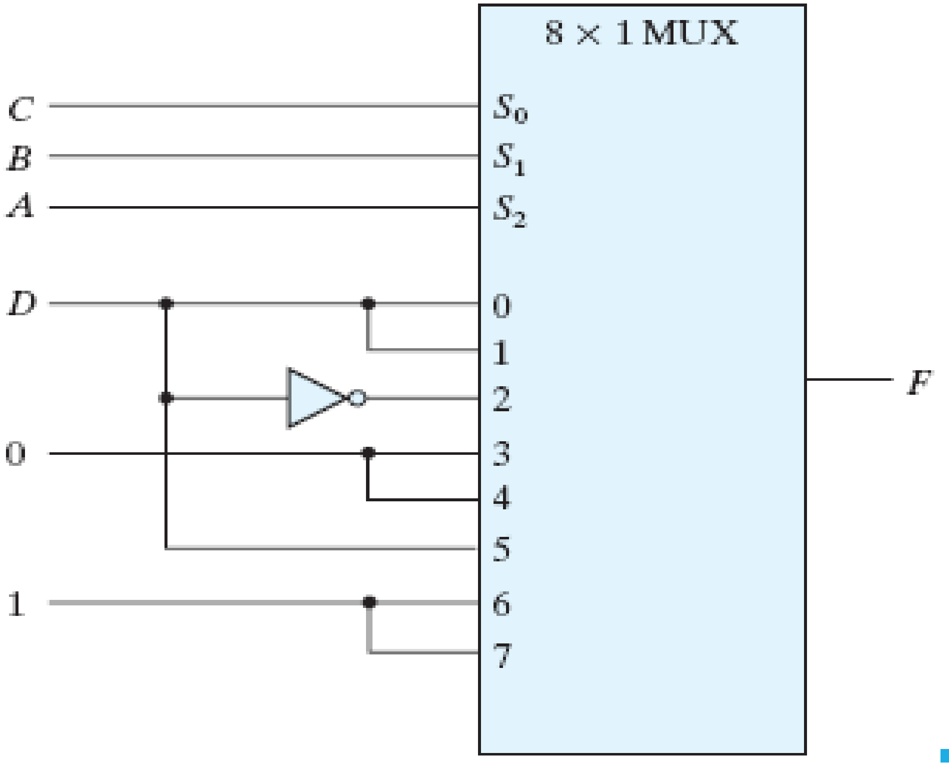
The first n-1 variables are applied to selection inputs and we evaluate the output as a function of the last variable

How to use as a block

Boolean function implementation (Cont.)

Example2: *F(A, B, C, D)=Σ(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	



THANK YOU