جامعة الملك سعود
King Saud University

CSC 220: Computer Organization

Unit 6
Combinational Circuits-2

Prepared by:
Md Saiful Islam, PhD

Updated by:
Isra Al-Turaiki, PhD

Department of Computer Science

College of Computer and Information Sciences

Overview

- Enabling
- Decoders
- Encoders
- Multiplexers
- DeMultiplexers

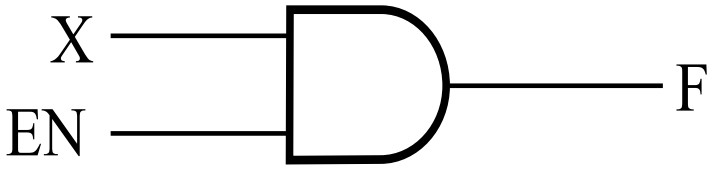
Chapter-3

M. Morris Mano, Charles R. Kime and Tom Martin, **Logic and Computer Design Fundamentals**, Global (5th) Edition, Pearson Education Limited, 2016. ISBN: 9781292096124

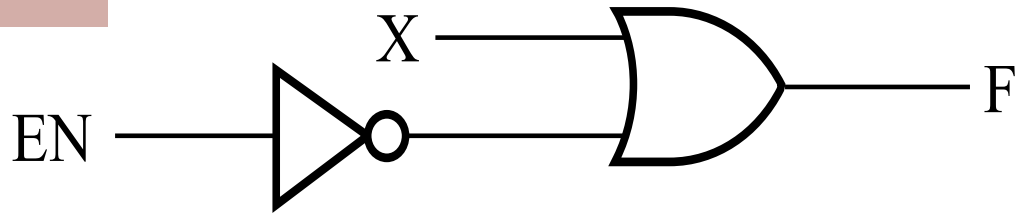
Enabling

- Enabling permits an input signal to pass through to an output
- Disabling blocks an input signal from passing through to an output, replacing it with a *fixed value*
- The value on the output when it is disable can be 0, 1 or Hi-Z

Note: it is possible to modify:
- EN = 0 enables X to reach the output
- EN = 1 blocks X.



(a) When disabled, 0 output



(b) When disabled, 1 output

Decoding

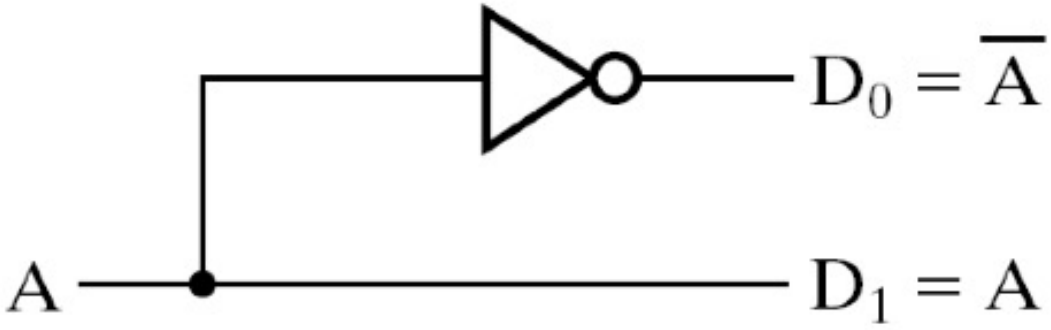
- A *decoder* is a combinational circuit that converts binary information from n input lines to (max. of) 2^n output lines.

- Generate 2^n (or fewer) minterms of n input variables.
  - it activates one and only one of its 2^n outputs based on the input
  - with all other outputs deactivated.

1-to-2-Line Decoder

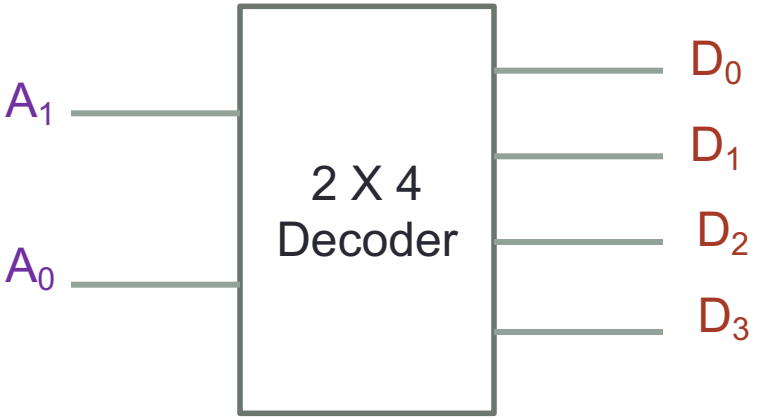
- For n=1 input line and m= 2^n = 2 output lines

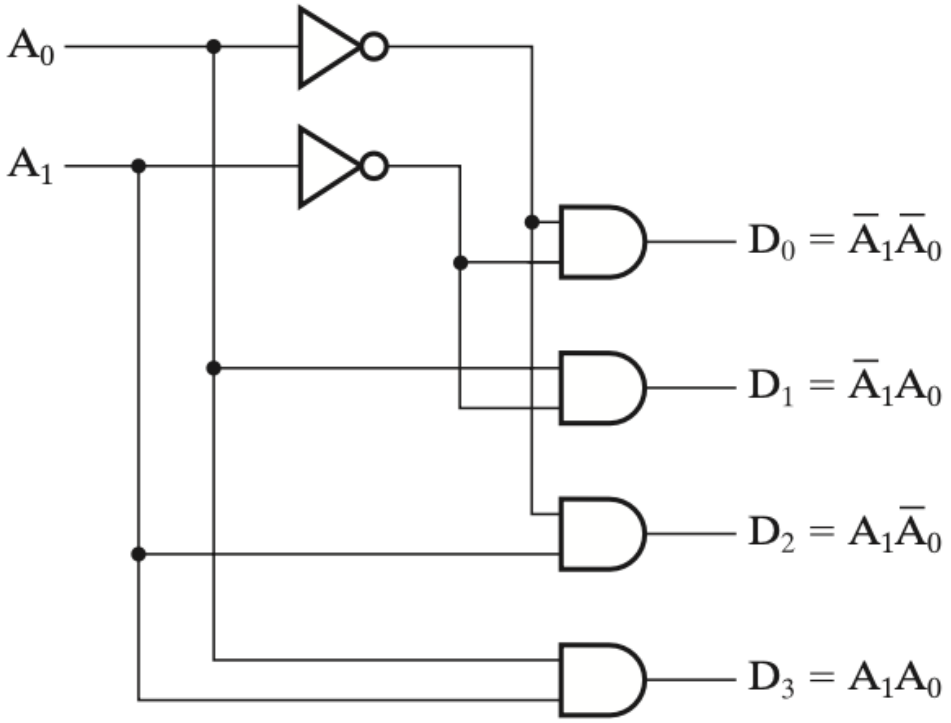
A	D_0	D_1
0	1	0
1	0	1



2-to-4-Line Decoder

- For n inputs =2 and m= 2^n = 4 output lines





A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Decoder and Enabling Combinations

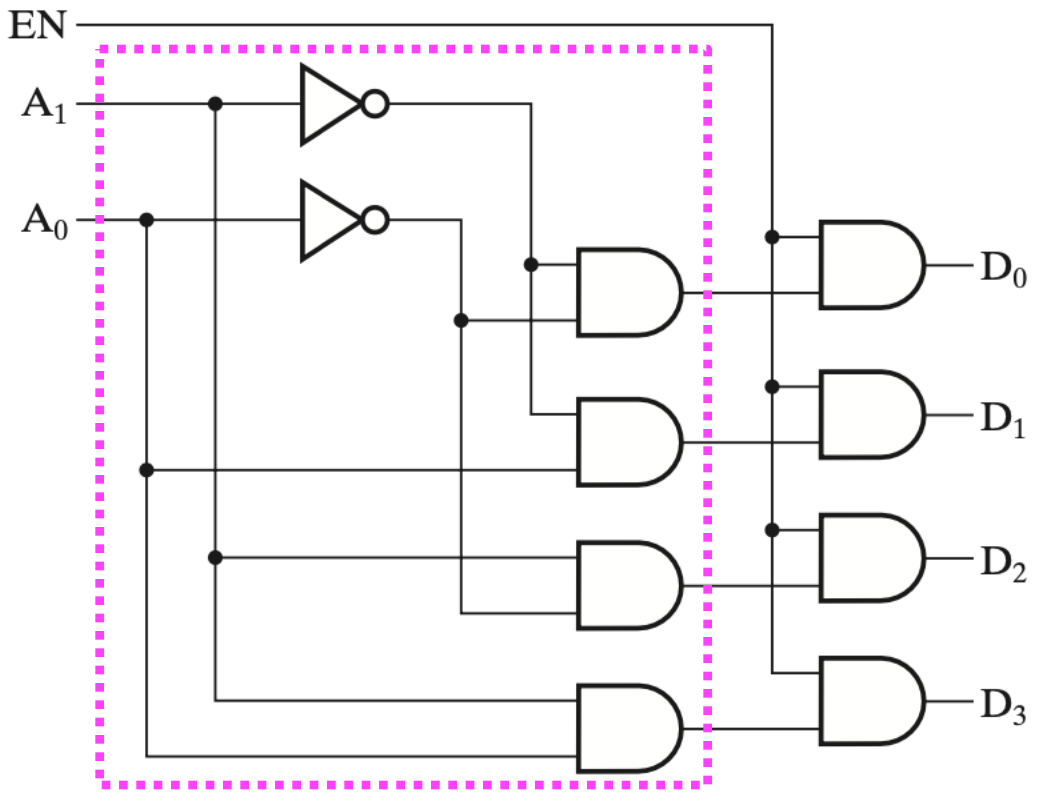
In general, attach m-enabling circuits to the outputs

See truth table below for function

- Note use of X's to denote both 0 and 1
- Combination containing two X's represent four binary combinations

EN	A_1	A_0	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

(a)



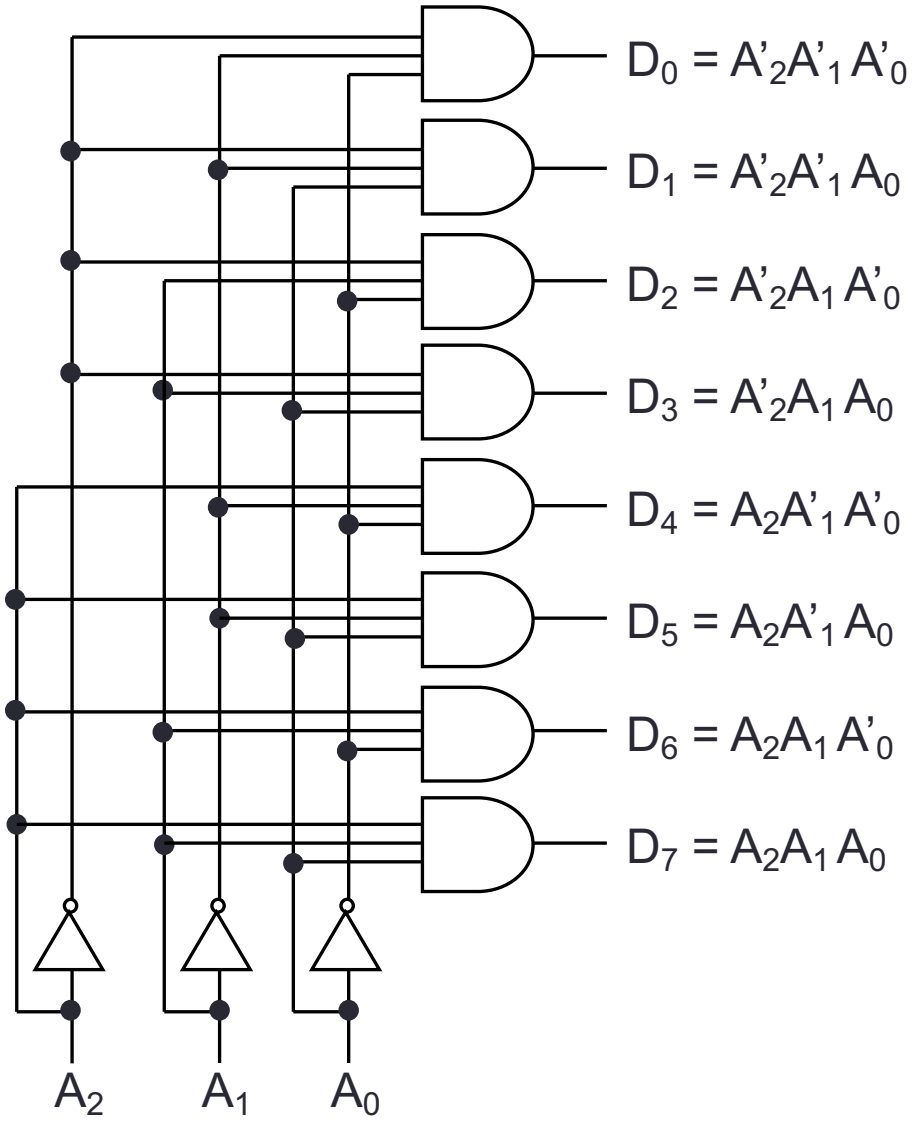
(b)

3-to-8-Line Decoder

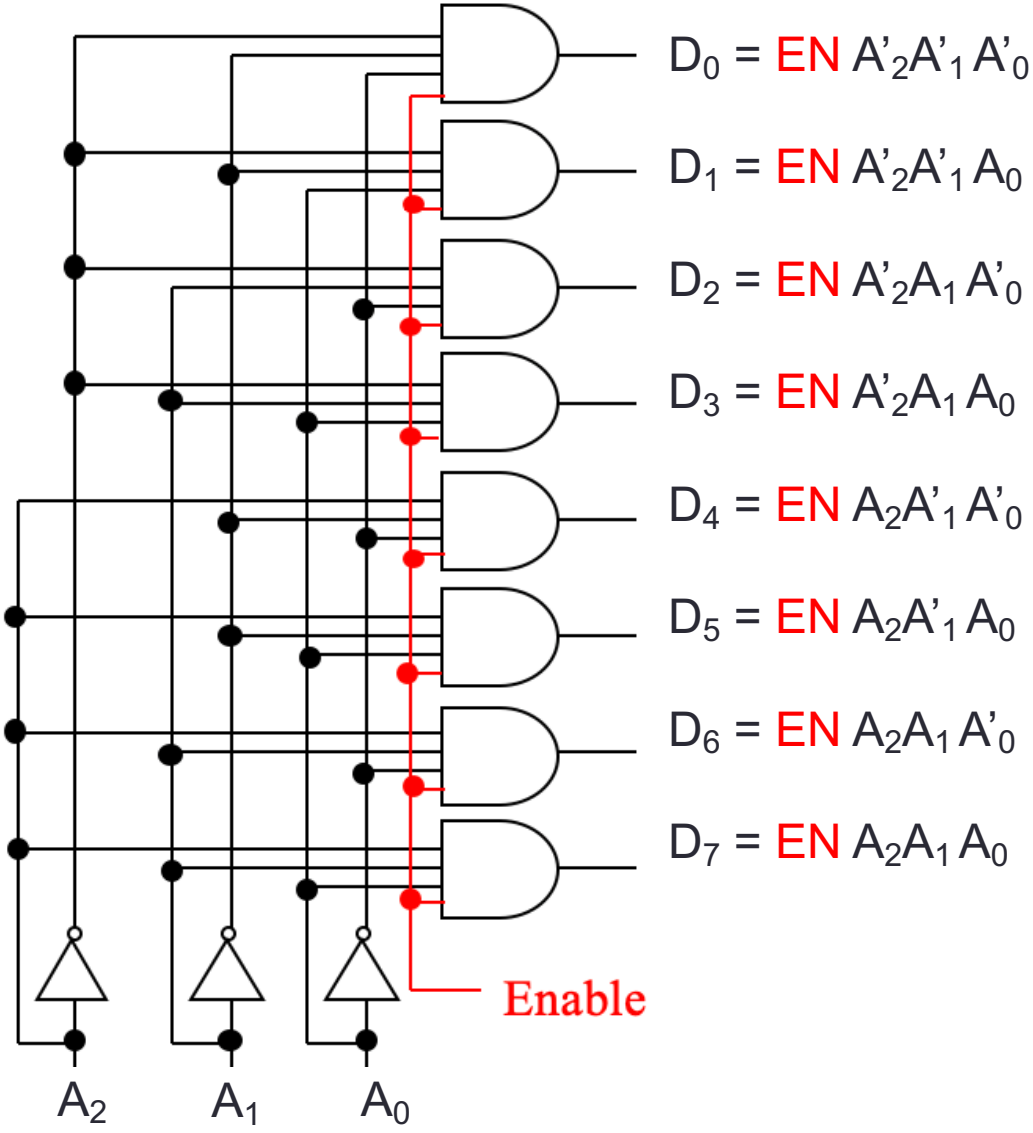
- For n=3 and m=8

A_2	A_1	A_0	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

3-to-8-Line Decoder



3-to-8-Line Decoder with Enable

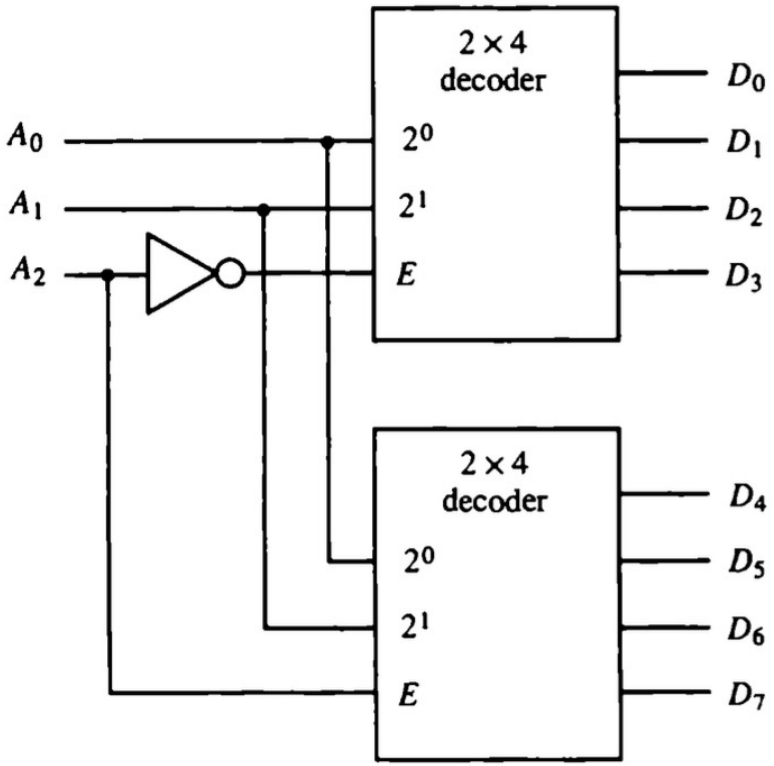


Decoder Expansion

- There are occasions when a certain-size decoder is needed but only smaller sizes are available.
- it is possible to combine two or more decoders with enable inputs to form a larger decoder

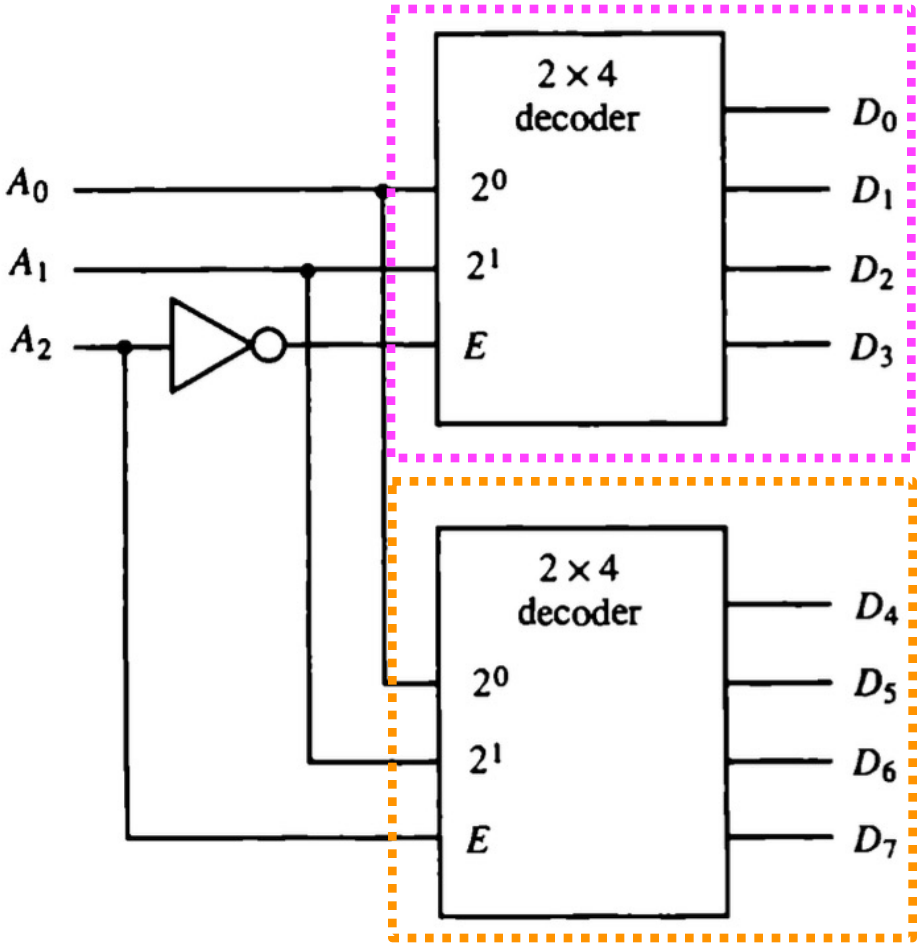
Two 2-to-4-line decoders are combined to achieve a 3-to-8-line decoder.

- When $A_2 = 0$, the upper decoder is enabled and the lower is disabled.
- When $A_2 = 1$, the lower decoder is enabled and the upper is disabled.



Decoder Expansion

A_2	A_1	A_0	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



Decoder-Based Combinational Circuits

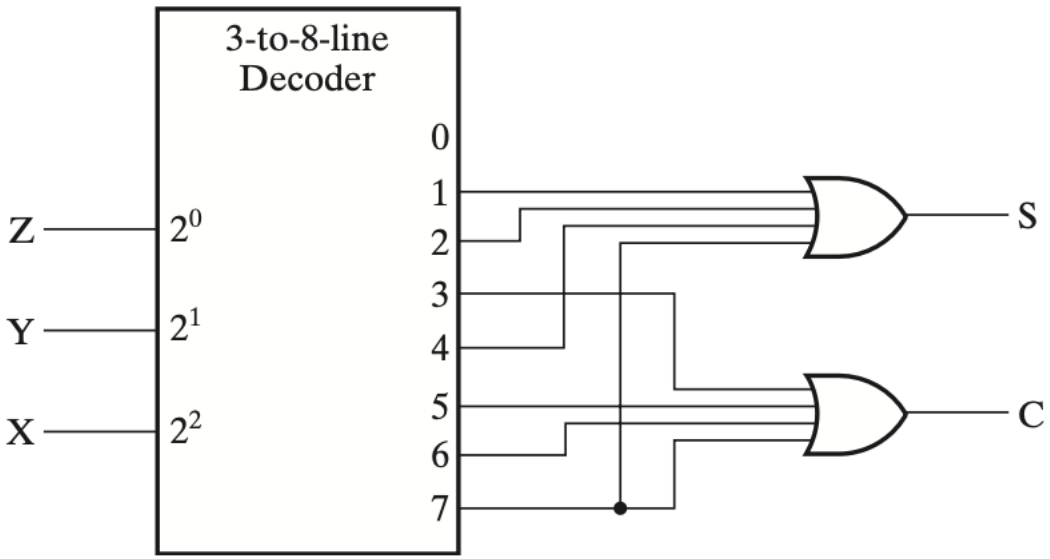
- any Boolean function can be expressed as a sum of minterms:
- use a decoder to generate the minterms and combine them with an external OR gate to form a sum-of-minterms implementation
- any combinational circuit with n inputs and m outputs can be implemented with an n–to–2^n-line decoder and m OR gates.

Example: Full adder

- Number of inputs: 3
- Number of outputs: 2 (S and C)
- Use 3-to-2^3 line decoder and 2 OR gates

Example: Full Adder with a Decoder

X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



S=∑(1,2,4,7)

C=∑(3,5,6,7)

Encoders

- An encoder performs the inverse function of a decoder.
- It has 2^n (or fewer) input lines and n output lines.
- The output lines generate the binary code corresponding to the input value.
- Example: an Octal to binary Encoder

  Inputs: 8 (D_7 … D_0)

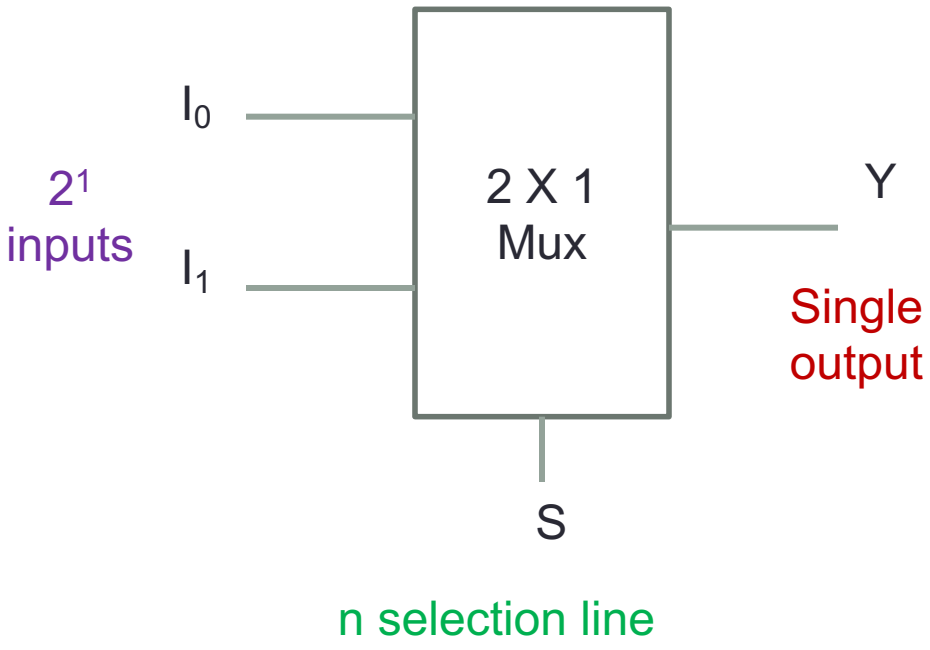
  Outputs: 3 (A_2 … A_0)

$$A_0 = D_1 + D_3 + D_5 + D_7$$
$$A_1 = D_2 + D_3 + D_6 + D_7$$
$$A_2 = D_4 + D_5 + D_6 + D_7$$

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Multiplexer

- Logic circuits that perform selecting are called *multiplexers*
  - A set of inputs (2^n)
  - A single output
  - A set of control lines for making the selection (n)
  - **Example: n=1**



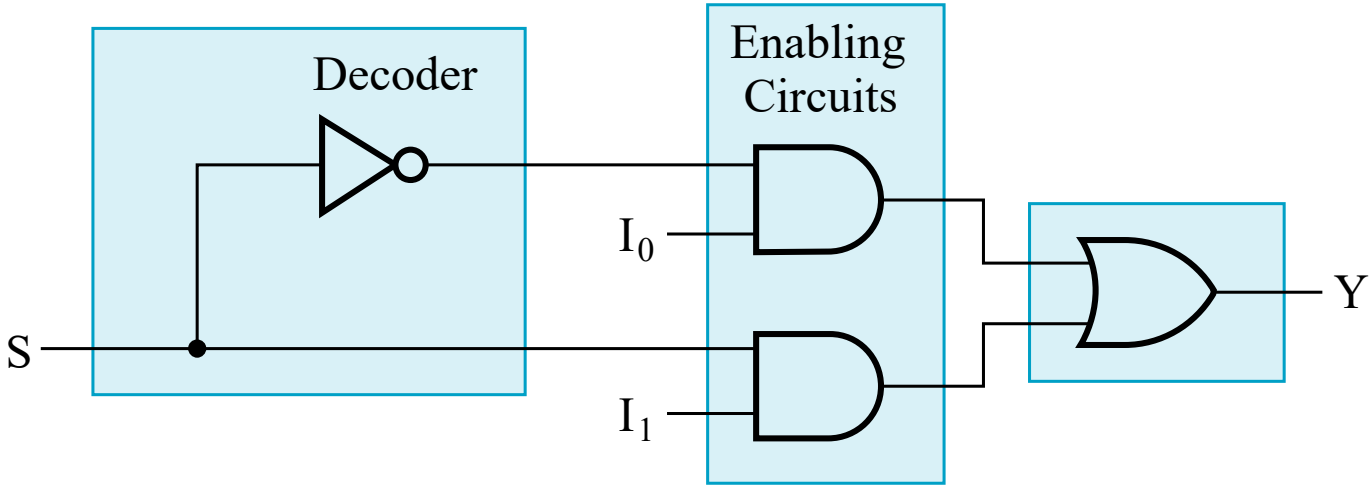
S	I_0	I_1	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$Y = \bar{S}I_0 + SI_1$$

2-to-1-Line Multiplexer

Note the regions of the multiplexer circuit shown:

- 1-to-2-line Decoder
- 2 Enabling circuits
- 2-input OR gate



$$Y = \bar{S}I_0 + SI_1$$

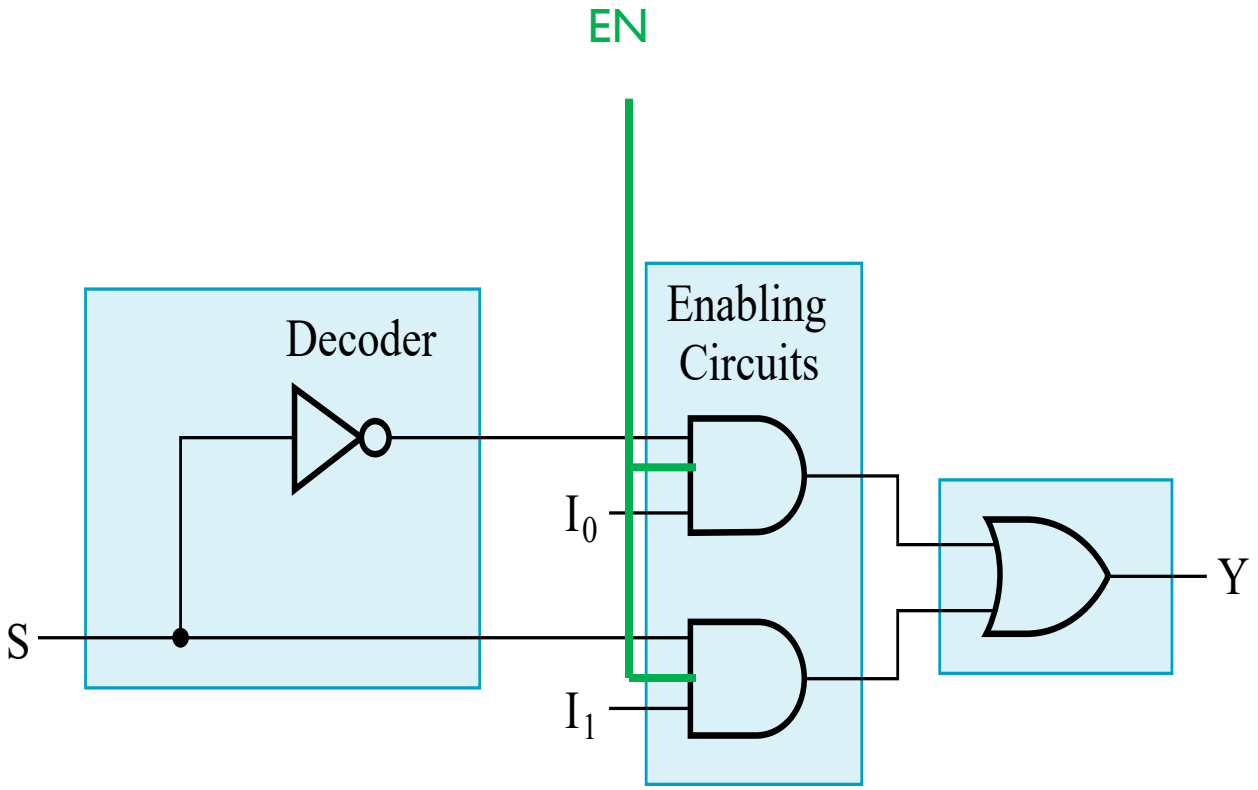
2-to-1-Line Multiplexer with Enable

As in decoders, multiplexers may have an enable input to control the operation of the unit.

When EN= 0, the multiplexer is disabled.

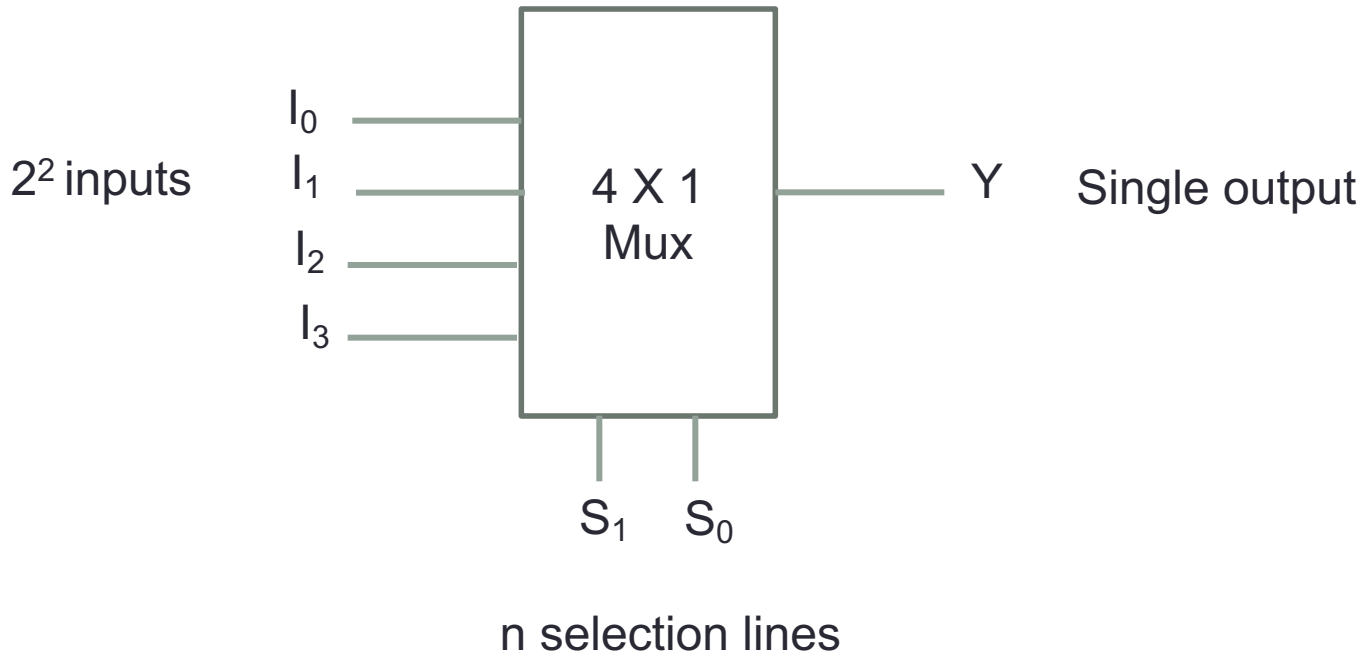
When EN=1, the multiplexer functions normally.

EN	S	I_0	I_1	Y
0	X	X	X	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1



4-to-1-Line Multiplexer

Example: n=2

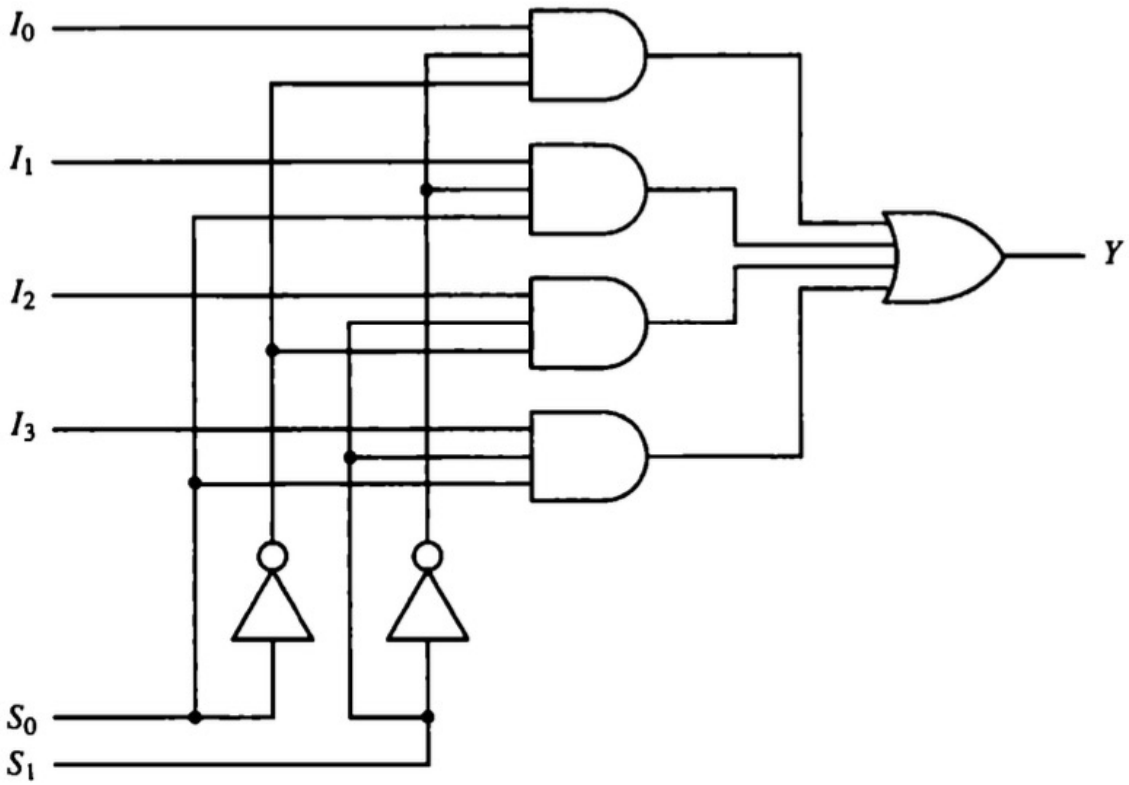


4-to-1-Line Multiplexer

Condensed Truth Table for 4-to-1-Line Multiplexer

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

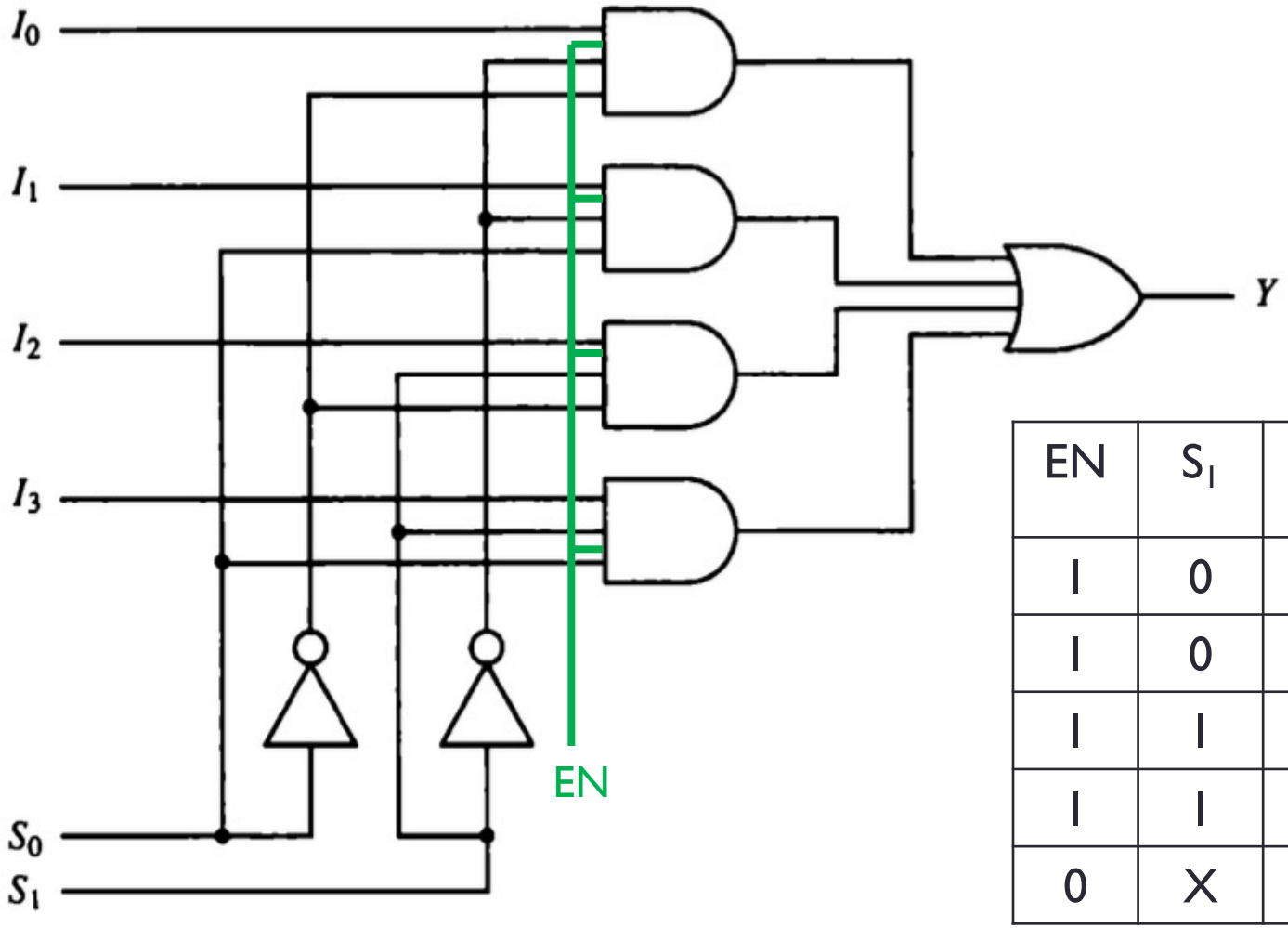
Figure 2-4 4-to-1-line multiplexer.



$$Y = (\overline{S}_1\overline{S}_0)I_0 + (\overline{S}_1S_0)I_1 + (S_1\overline{S}_0)I_2 + (S_1S_0)I_3$$

4-to-1-Line Multiplexer with Enable

Figure 2-4 4-to-1-line multiplexer.



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

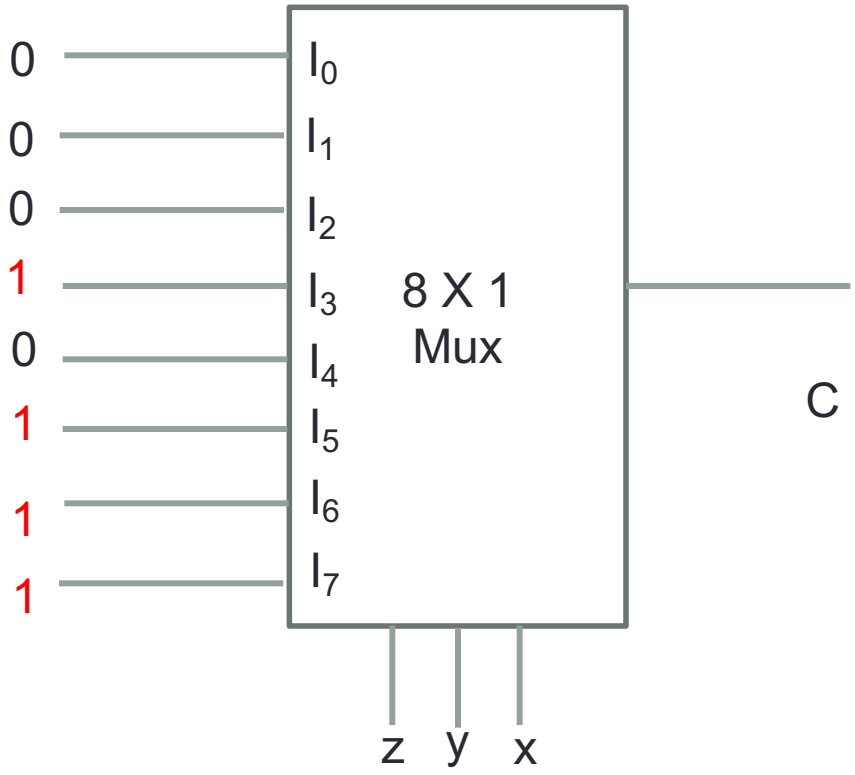
Y = EN $(S_1'S_0')I_0$ + EN $(S1'S0)I_1$ + EN $(S_1S_0')\ I_2$ + EN $(S_1S_0)\ I_3$

Mux-Based Combinational Circuits

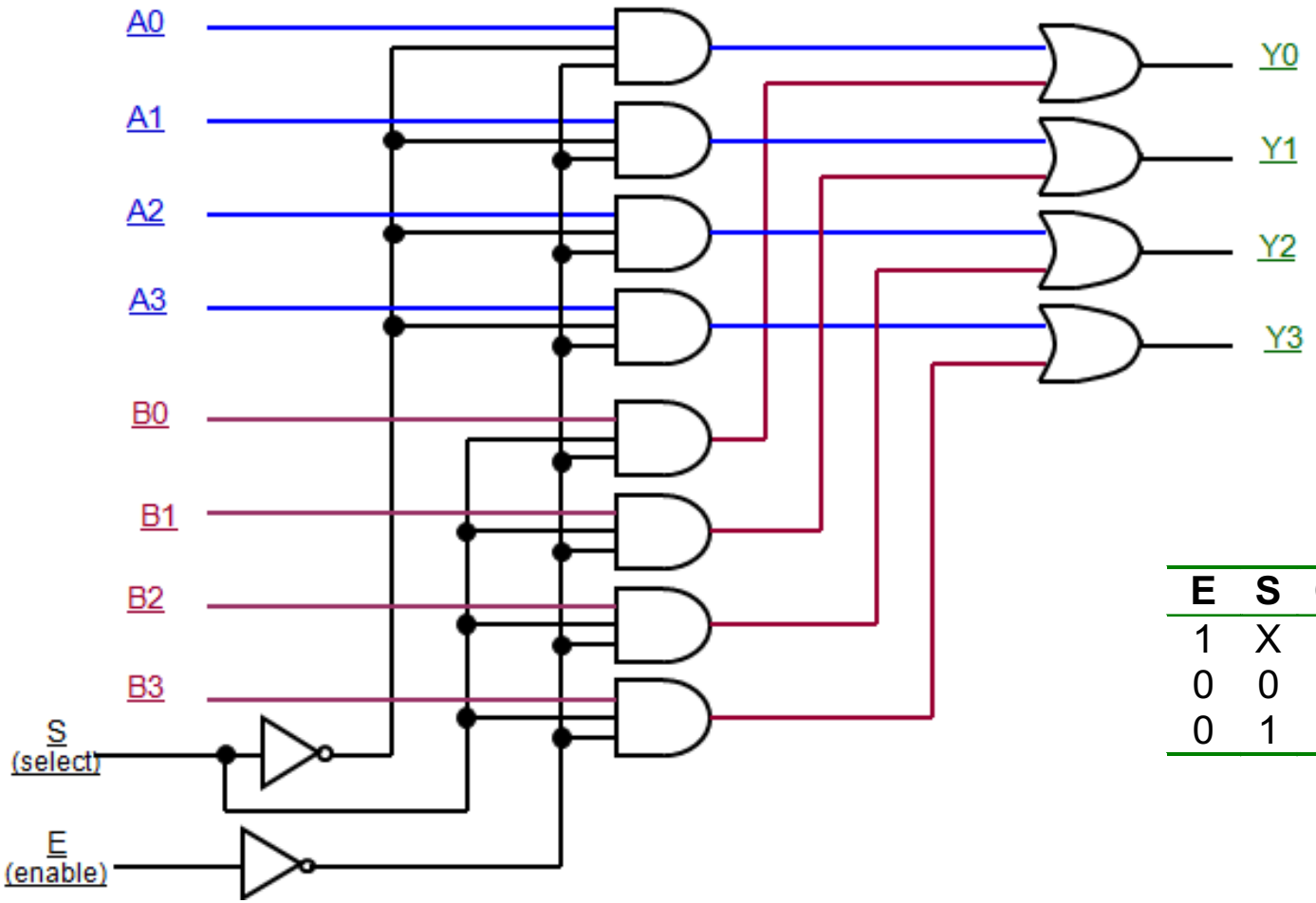
Example: Full adder with a multiplexer

X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$C=\sum(3,5,6,7)$



Multiplexer Width Expansion

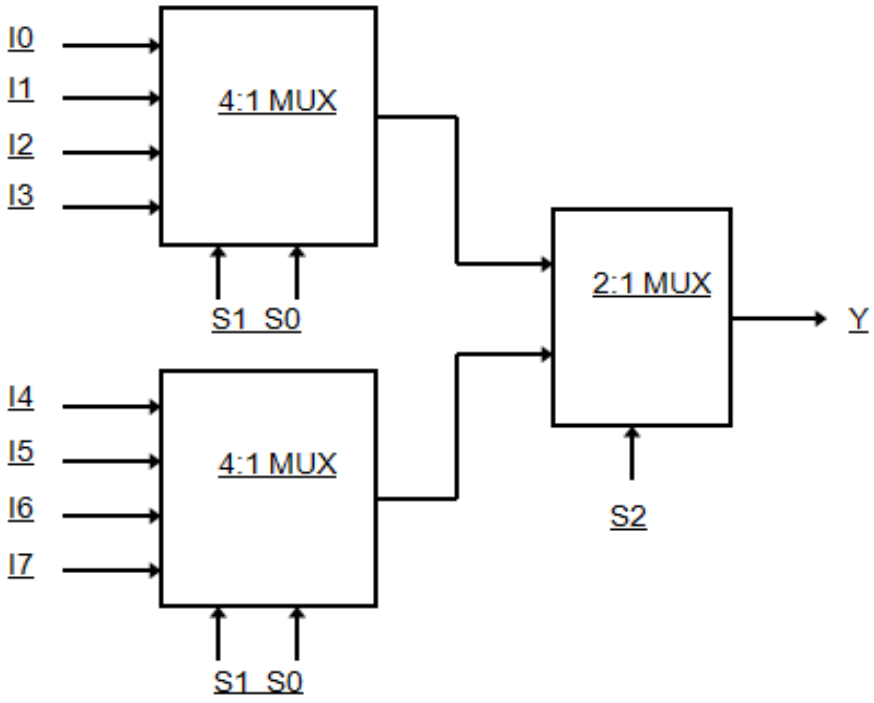


E	S	Output Y
1	X	all 0's
0	0	select A
0	1	select B

Quadruple 2:1 multiplexer

Larger Multiplexers

- Larger multiplexers can be constructed from smaller ones.
- An 8-to-1 multiplexer can be constructed from smaller multiplexers like this (note placement of selector lines):

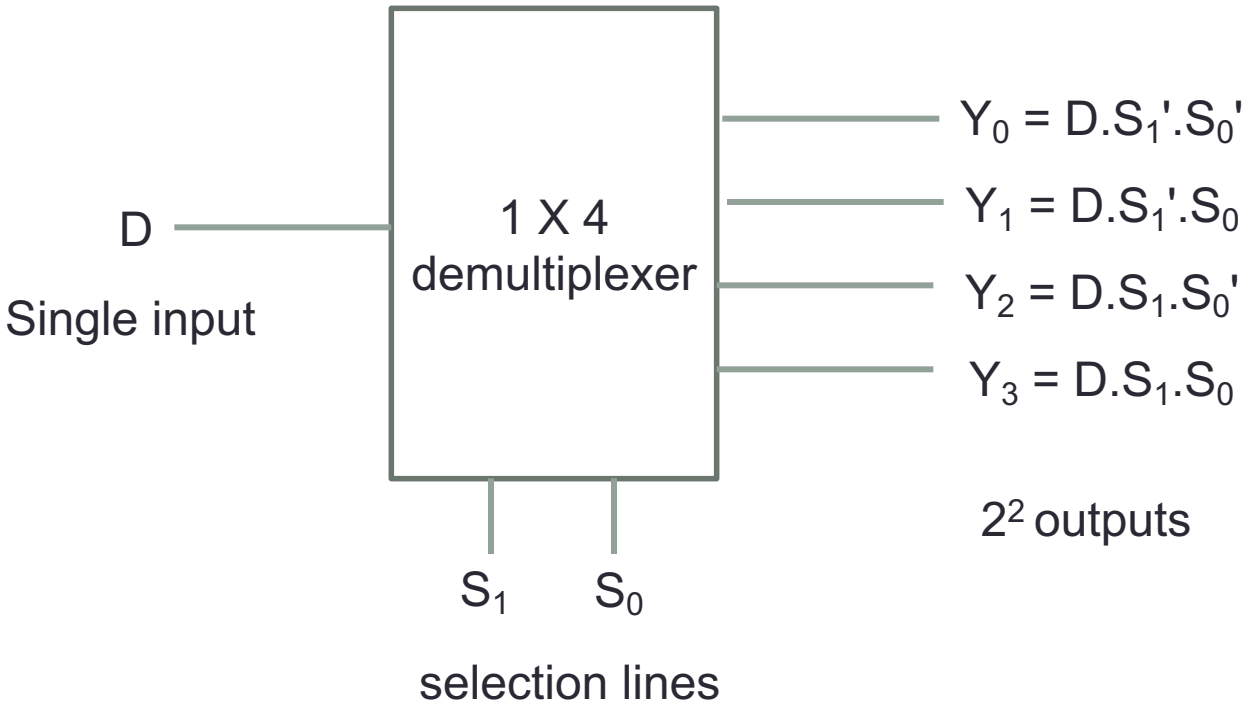


S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

Demultiplexer

The inverse of selection is *distribution*:

- information received from a single line is transmitted to one of 2^n possible output lines.
- The circuit which implements such distribution is called a demultiplexer.



Demultiplexer

S_1	S_0	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

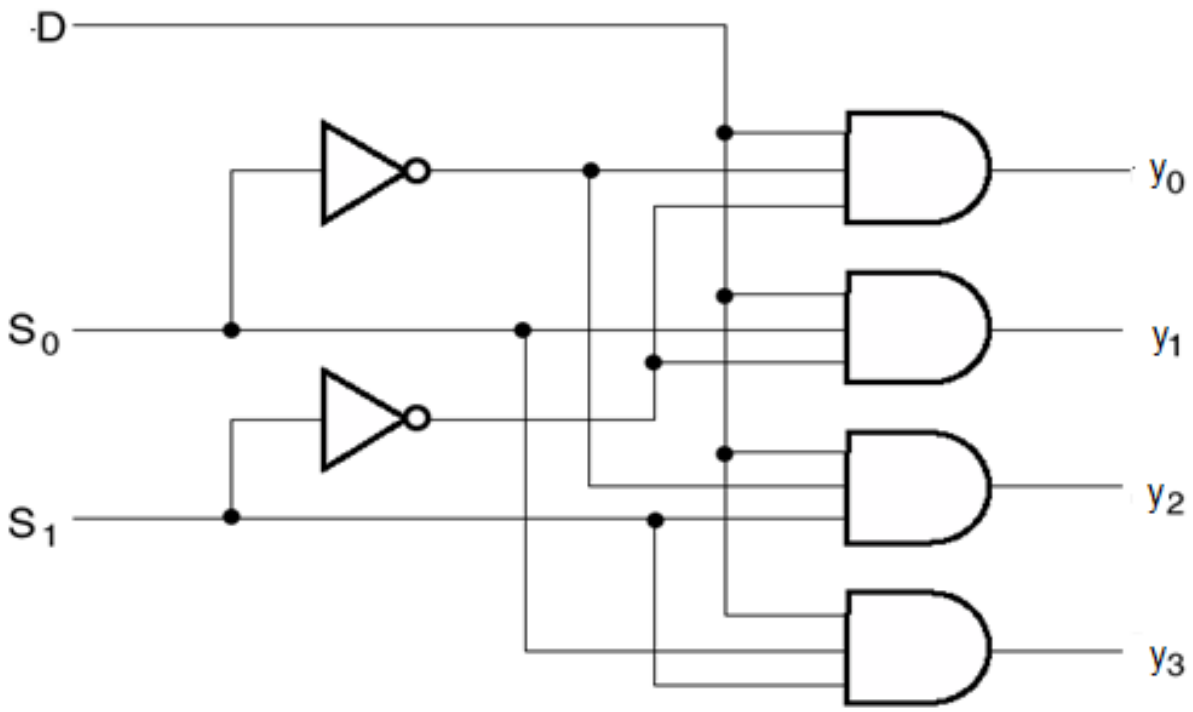
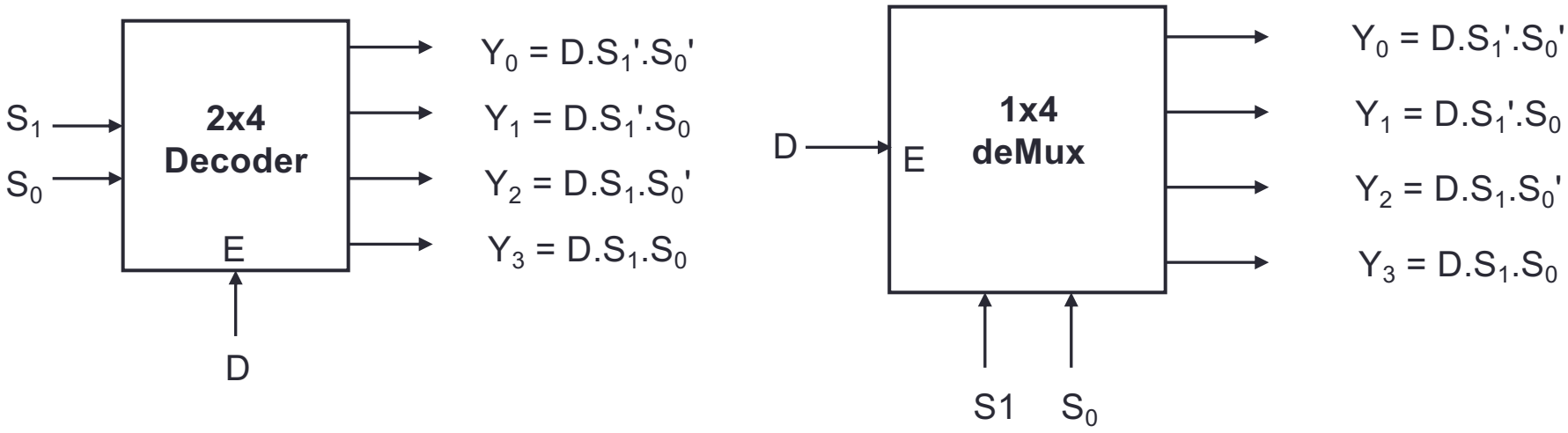


Fig. 3-24 1-to-4-Line Demultiplexer

The demultiplexer is actually identical to a decoder with enable.

Demultiplexer



- For the demultiplexer: input EN provides the data, while the other inputs act as the selection variables.

- Although the two circuits have different applications, their logic diagrams are exactly the same. a decoder with enable input is referred to as a decoder/ demultiplexer.

Summary

Decoder:

- n-to-2^n decoder is simply a minterm generator, with output corresponding to exactly one minterm.
- For each possible input condition, one and only one output signal will be at logic 1

Encoders:

- It has 2^n (or fewer) input lines and n output lines.
- Assign a unique output code (a binary number) for each input signal applied to the device
- Opposite of the decoder

Summary

Multiplexers:

- it has a set of inputs (2^n) and a single output
- the specific input being determined by a selection code (n lines).

Demultiplexer:

- Connects a single input line to one of 2^n output lines, the specific output being determined by a selection code (n lines)