Mahidol University

Mahidol University
Wisdom of the Land

Logic and Computer Design Fundamentals

Chapter 3 – Combinational Logic Design

Part 2 – Combinational Logic

Asst.Prof.Dr. Preecha Tangworakitthaworn
Semester 2/2023

Overview

- Part 2 – Combinational Logic
  - Decoding using Decoders
    - Implementing Combinational Functions with Decoders
  - Encoding using Encoders
  - Selecting using Multiplexers
    - Implementing Combinational Functions with Multiplexers

Decoding

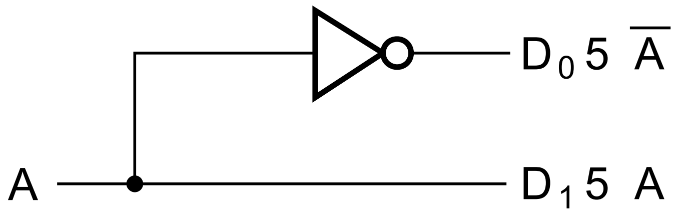
- Decoding - the conversion of an *n*-bit input code to an *m*-bit output code with $n \leq m \leq 2^n$ such that each valid code word produces a unique output code
- Circuits that perform decoding are called ***decoders***
- Here, functional blocks for decoding are
  - called *n*-to-*m* line decoders, where $m \leq 2^n$, and
  - generate 2^n (or fewer) minterms for the *n* input variables

Decoder Examples

- **1-to-2-Line Decoder**

A	D_0	D_1
0	1	0
1	0	1

(a)

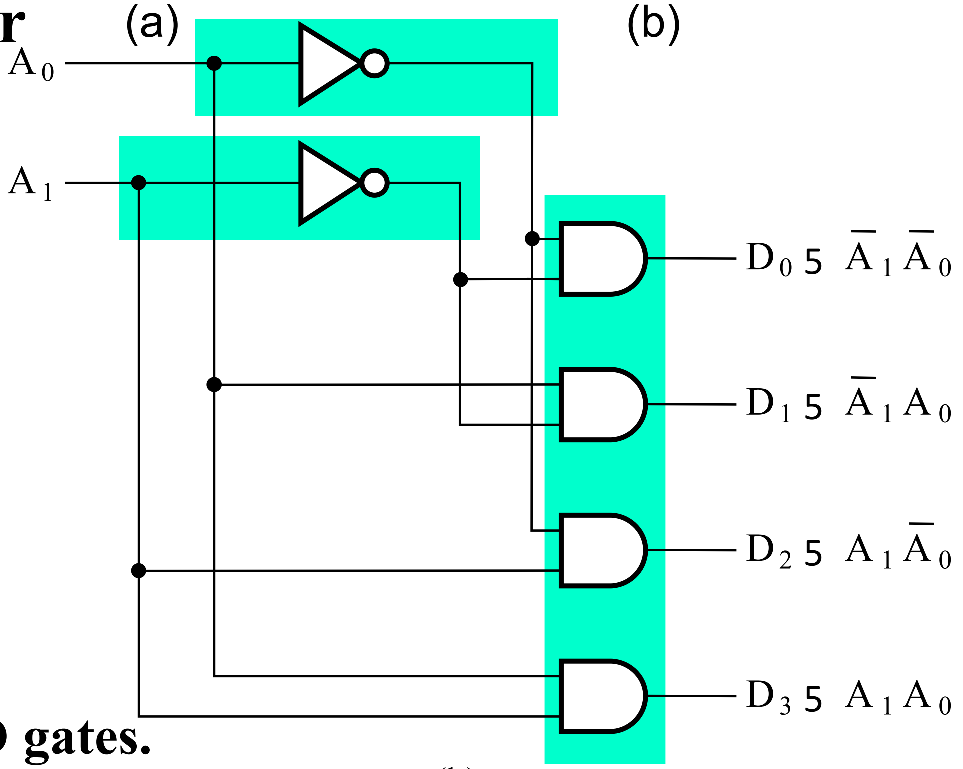


(b)

- **2-to-4-Line Decoder**

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

(a)


A_0
A_1
$D_0 5 \overline{A}_1 \overline{A}_0$
$D_1 5 \overline{A}_1 A_0$
$D_2 5 A_1 \overline{A}_0$
$D_3 5 A_1 A_0$

(b)

- **Note that the 2-4-line made up of 2 1-to-2-line decoders and 4 AND gates.**

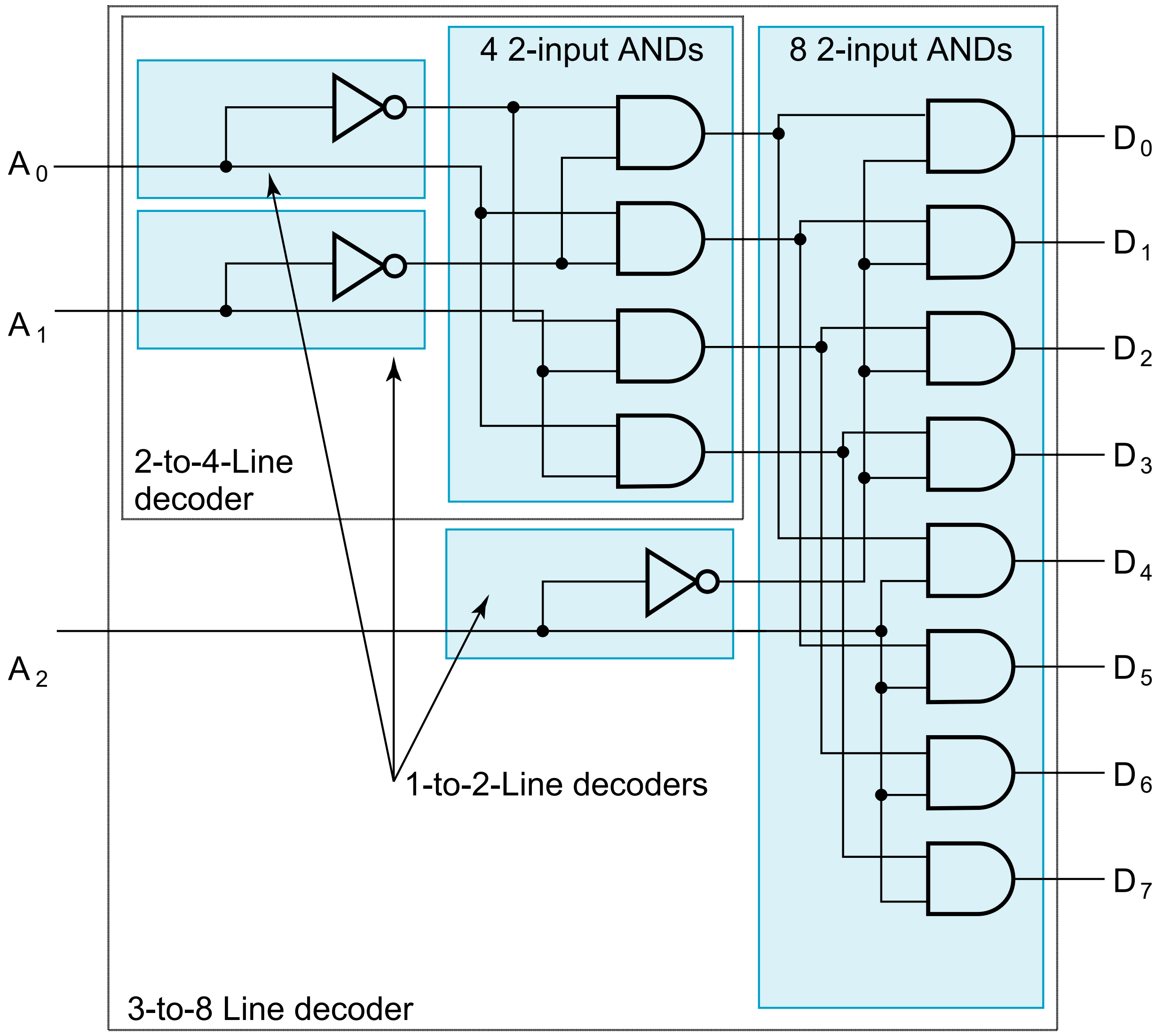
Decoder Expansion - Example 1

- 3-to-8-line decoder
  - Number of output ANDs = 8
  - Number of inputs to decoders driving output ANDs = 3
  - Closest possible split to equal
    - 2-to-4-line decoder
    - 1-to-2-line decoder
  - 2-to-4-line decoder
    - Number of output ANDs = 4
    - Number of inputs to decoders driving output ANDs = 2
    - Closest possible split to equal
      - Two 1-to-2-line decoders
- See next slide for result

Decoder Expansion - Example 1

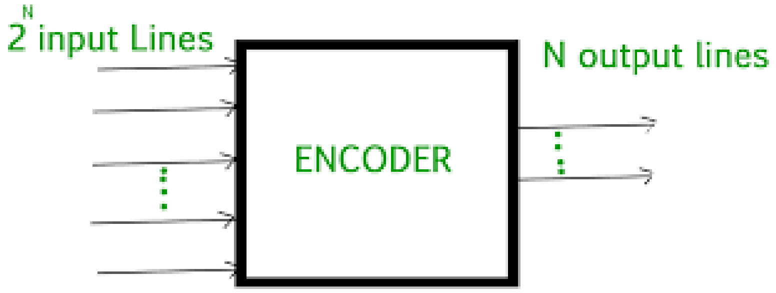
Output

Solution



Encoding

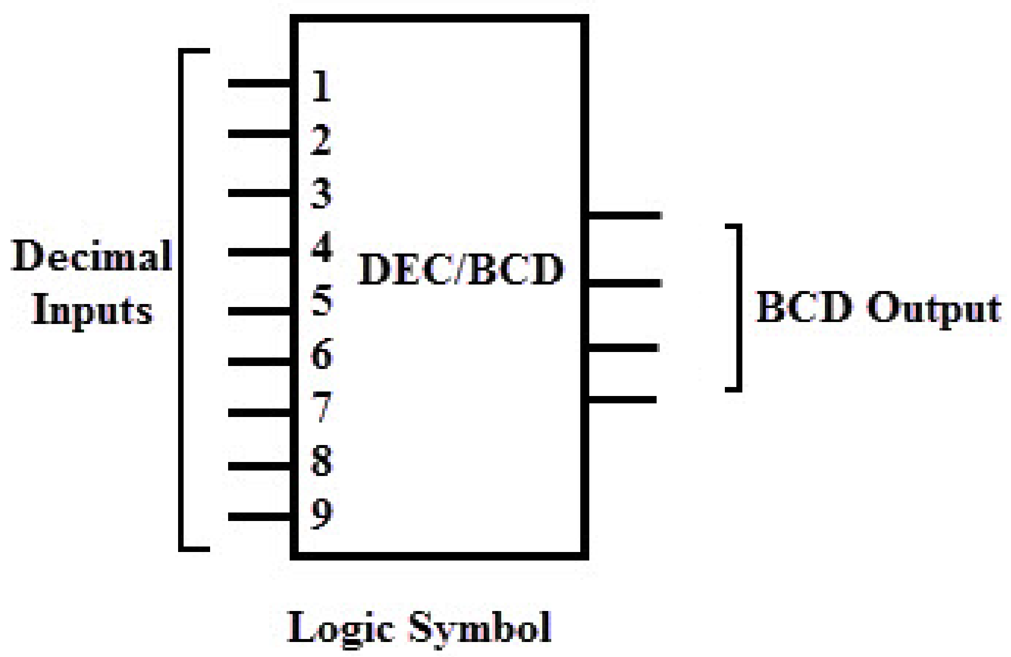
- Encoding - the opposite of decoding - the conversion of an *m*-bit input code to a *n*-bit output code with $n \leq m \leq 2^n$ such that each valid code word produces a unique output code
- Circuits that perform encoding are called ***encoders***
- An encoder has 2^n (or fewer) input lines and *n* output lines which generate the binary code corresponding to the input values



Encoder Example

- **A decimal-to-BCD encoder**
  - **Inputs: 9 bits corresponding to decimal digits 1 through 9, (D_1, …, D_9)**
  - **Outputs: 4 bits with BCD codes**
  - **Function: If input bit D_i is a 1, then the output (A_3, A_2, A_1, A_0) is the BCD code for i,**
- **The truth table could be formed, but alternatively, the equations for each of the four outputs can be obtained directly.**

Decimal-to-BCD encoder



Input *m*: 9 bits
(0 is not applicable for the function)
Why??

Output *n*: 4 bits

Encoder Example (continued)

- **Input D_i is a term in equation A_j if bit A_j is 1 in the binary value for i.**
- **Equations:**

$$A_3 = D_8 + D_9$$

$$A_2 = D_4 + D_5 + D_6 + D_7$$

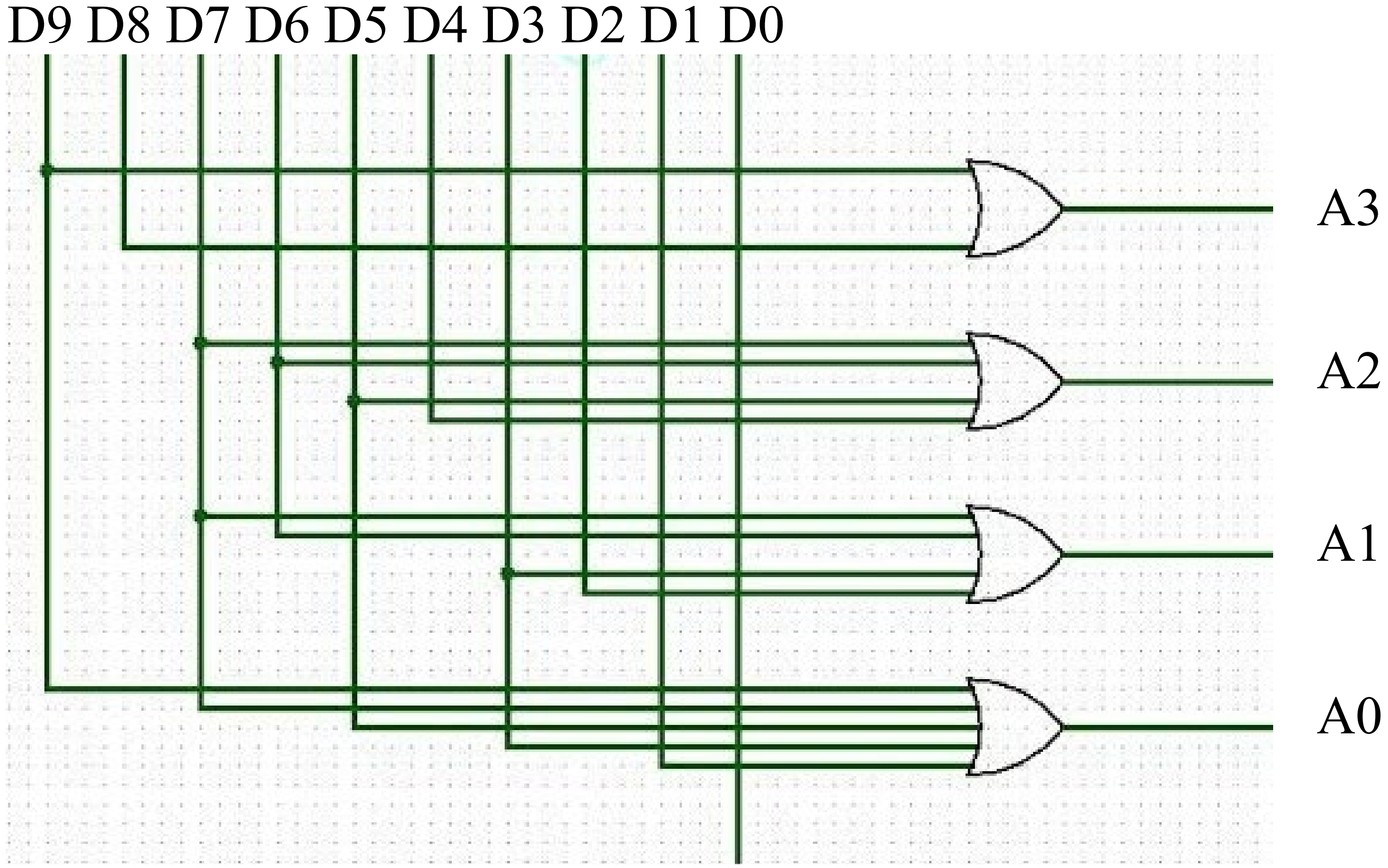
$$A_1 = D_2 + D_3 + D_6 + D_7$$

$$A_0 = D_1 + D_3 + D_5 + D_7 + D_9$$

How to draw Truth table??

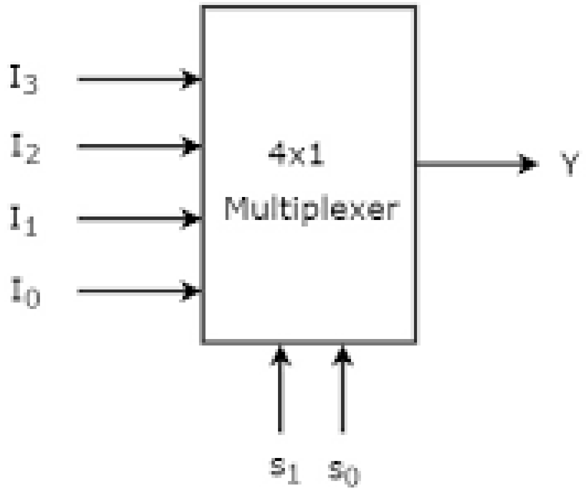
Decimal-to-BCD encoder

Output Solution



Multiplexers

- **A multiplexer selects information from an input line and directs the information to an output line.**
- **A typical multiplexer has *n* control inputs (S_{n-1}, … S_0) called *selection inputs*, 2^n information inputs (I_{2^n-1}, … I_0), and one output Y.**

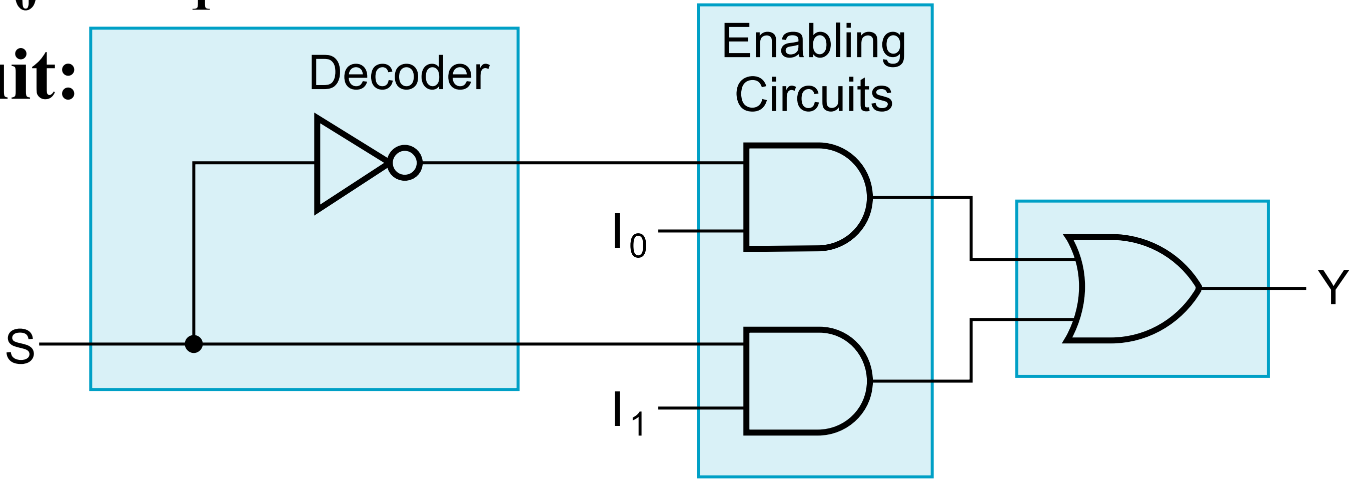


2-to-1-Line Multiplexer

- **Since $2 = 2^1$, $n = 1$**
- **The single selection variable S has two values:**
  - **$S = 0$ selects input I_0**
  - **$S = 1$ selects input I_1**
- **The equation:**

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

- **The circuit:**

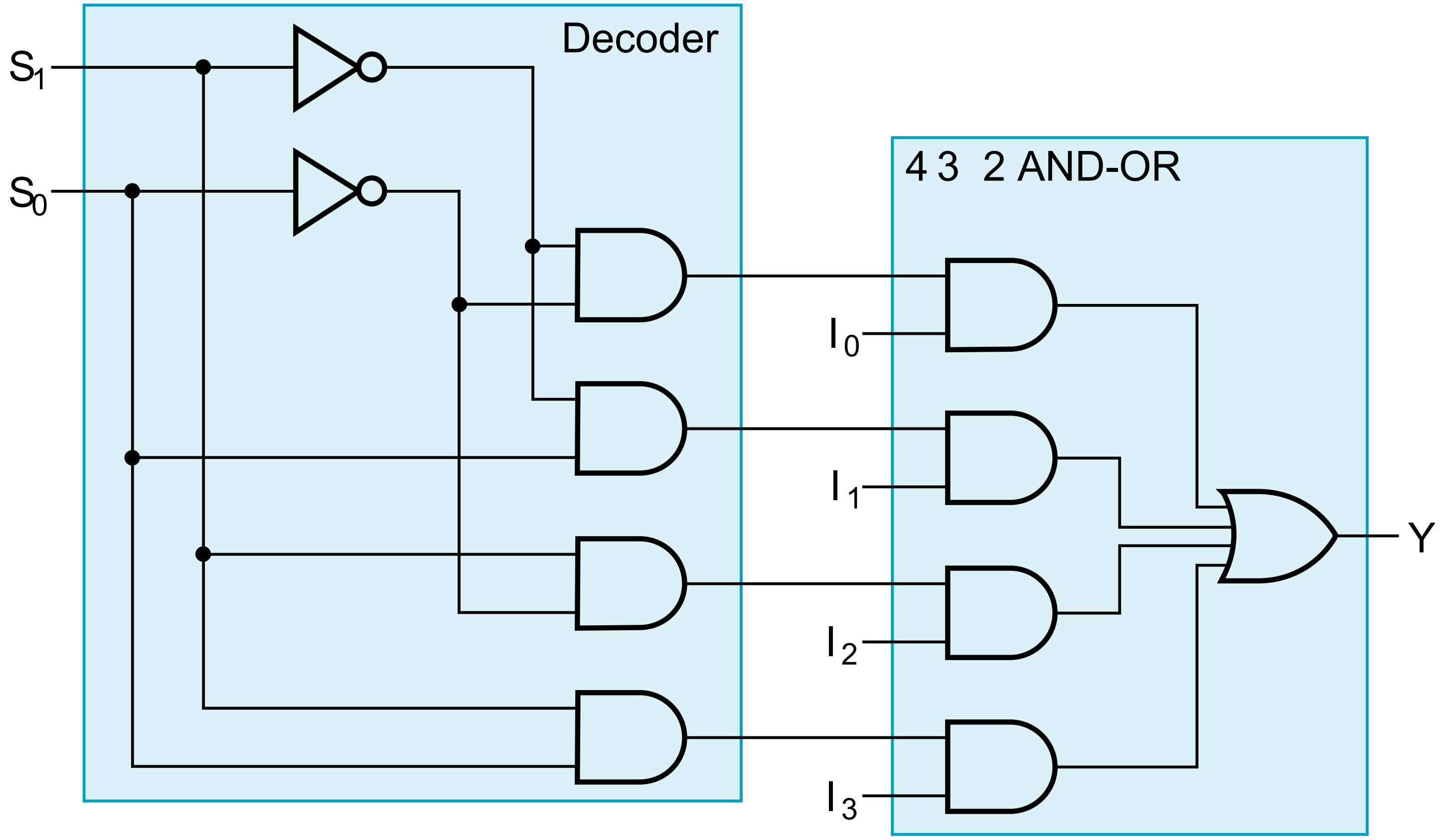


2-to-1-Line Multiplexer (continued)

- **Note the regions of the multiplexer circuit shown:**
  - **1-to-2-line Decoder**
  - **2 Enabling circuits**
  - **2-input OR gate**
- **To obtain a basis for multiplexer expansion, we combine the Enabling circuits and OR gate into a 2 × 2 AND-OR circuit:**
  - **1-to-2-line decoder**
  - **2 × 2 AND-OR**
- **In general, for an 2^n-to-1-line multiplexer:**
  - ***n*-to-2^n-line decoder**
  - **2^n × 2 AND-OR**

Example: 4-to-1-line Multiplexer

- 2-to-2^2-line decoder
- $2^2 \times 2$ AND-OR


Decoder
4 3 2 AND-OR
S_1
S_0
I_0
I_1
I_2
I_3
Y