

# CSCI 210: Computer Architecture

## Lecture 17: Combinational Logic

Stephen Checkoway

Oberlin College

Mar. 30, 2022

Slides from Cynthia Taylor

# Announcements

- Problem Set 5 due Friday!
- Lab 4 due Sunday
- Join [circuitverse.org](https://circuitverse.org) for Lab 5

# Digital Logic

- Previously: Established rules of Boolean algebra and digital logic
- Today: Building stuff!

# Sum of Products

- Developed from Truth Table form
  - Each product term contains each input exactly once, complemented or not.
  - Need to OR together set of AND terms to satisfy table
  - One product for each 1 in F column

X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

# Programmable Logic Array

- Simple way to create a logical circuit from a truth table, using sum of products
- Either programmed during manufacture, or can be reprogrammed
- Used in CPUs to generate “control signals”

# Programmable Logic Array

- Set of inputs and their complements
- Array of AND gates
  - Form set of product terms
- Array of OR gates
  - Logical sum of product terms
- Multiple outputs

# Creating a PLA

- Prepare the truth table
- Write the Boolean expression in sum of products form.
- Decide the input connection of the AND matrix for generating the required product term.
- Then decide the input connections of OR matrix to generate the sum terms.
- Program the PLA.

# Size

- Only truth table entries that have a True (1) output are represented
- Each different product term will have only one entry in the PLA, even if the product term is used in multiple outputs



# Multiple outputs

Inputs			Outputs		
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

Output functions:  $D(A, B, C)$ ,  $E(A, B, C)$ ,  $F(A, B, C)$

Inputs			Outputs		
A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

### Sum of Products for output D

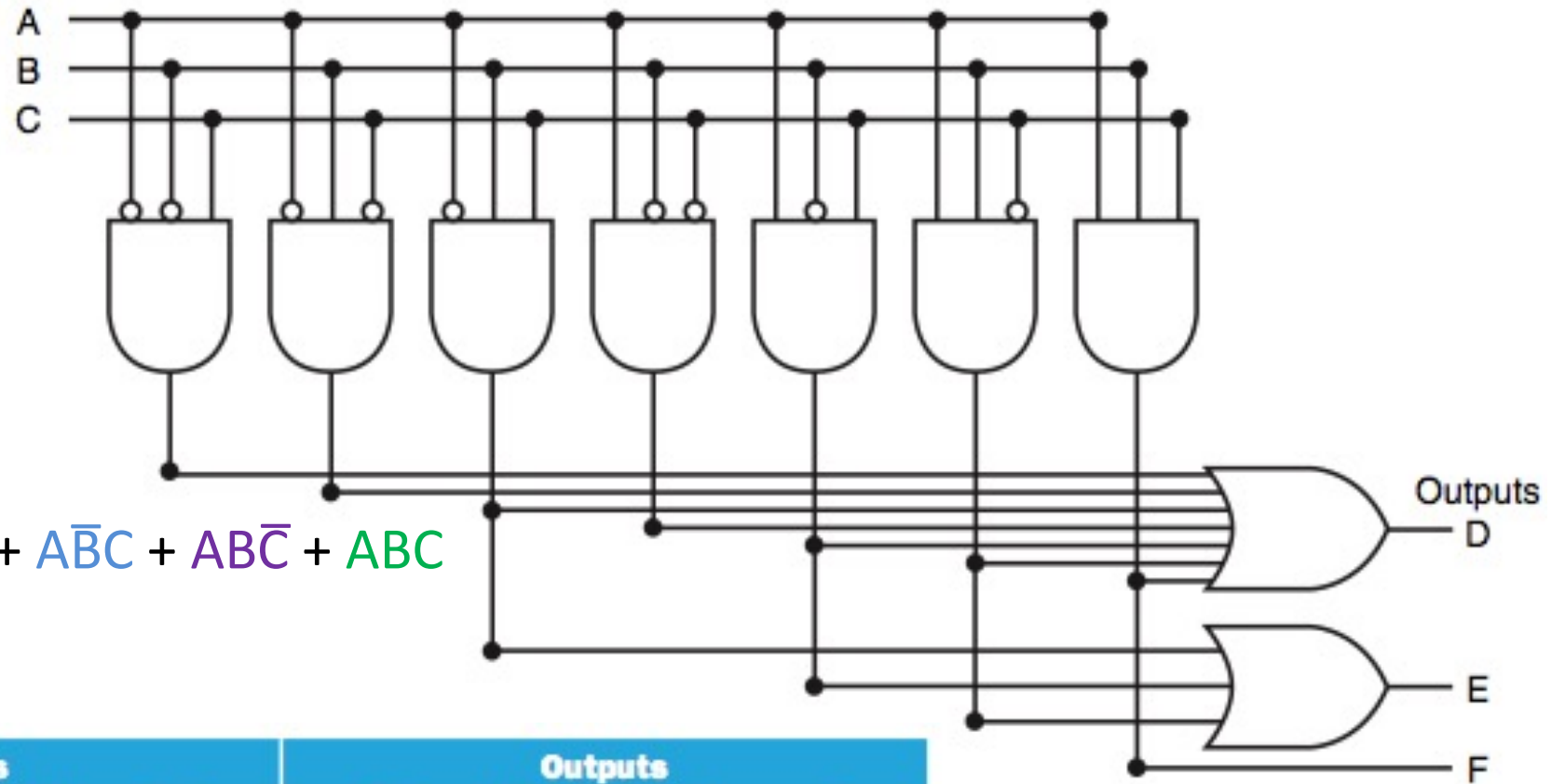
A  $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

B  $\bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

C  $(A+B+\bar{C})(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+B+C)(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$

D  $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

Inputs



$$D = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + ABC\bar{C} + ABC$$

$$E = \bar{A}BC + A\bar{B}C + ABC\bar{C}$$

$$F = ABC$$

Inputs			Outputs		
A	B	C	D	E	F
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

# Programmable PLAs

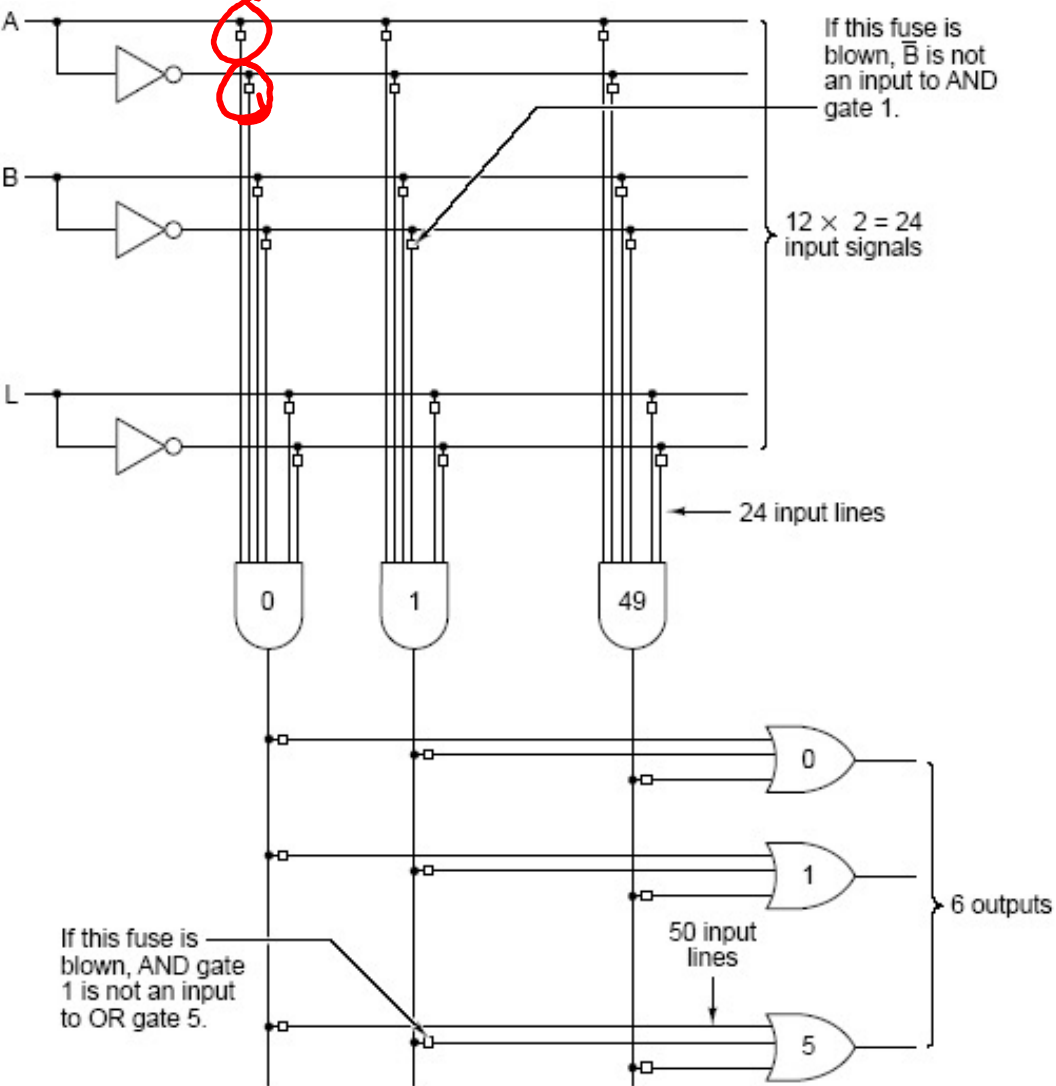
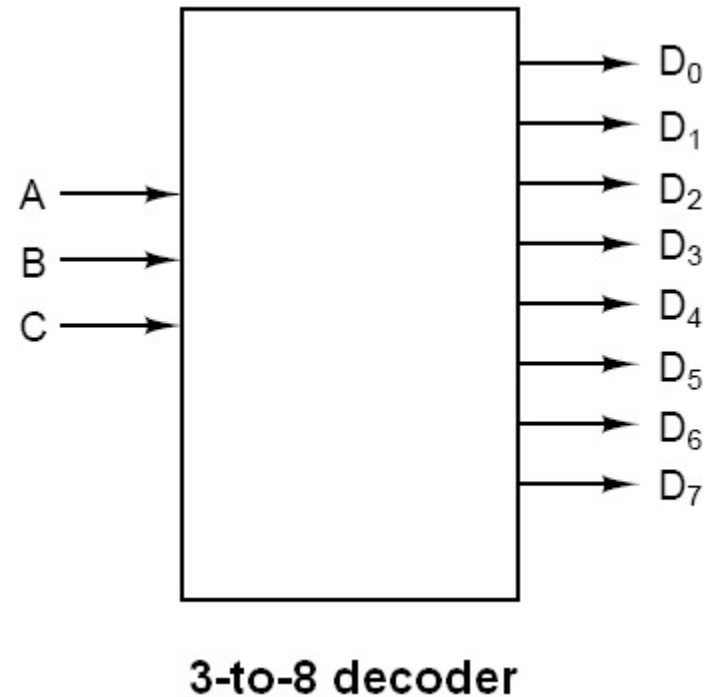


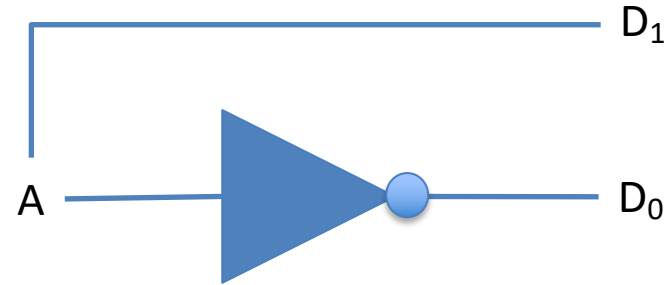
Figure 3-15. A 12-input, 6-output programmable logic array.

# Decoder

- Interprets  $n$  inputs (e.g., A, B, C) as an  $n$ -bit binary number
- Sets output  $D_n$  to 1, all other outputs to 0
- The output is “one hot”

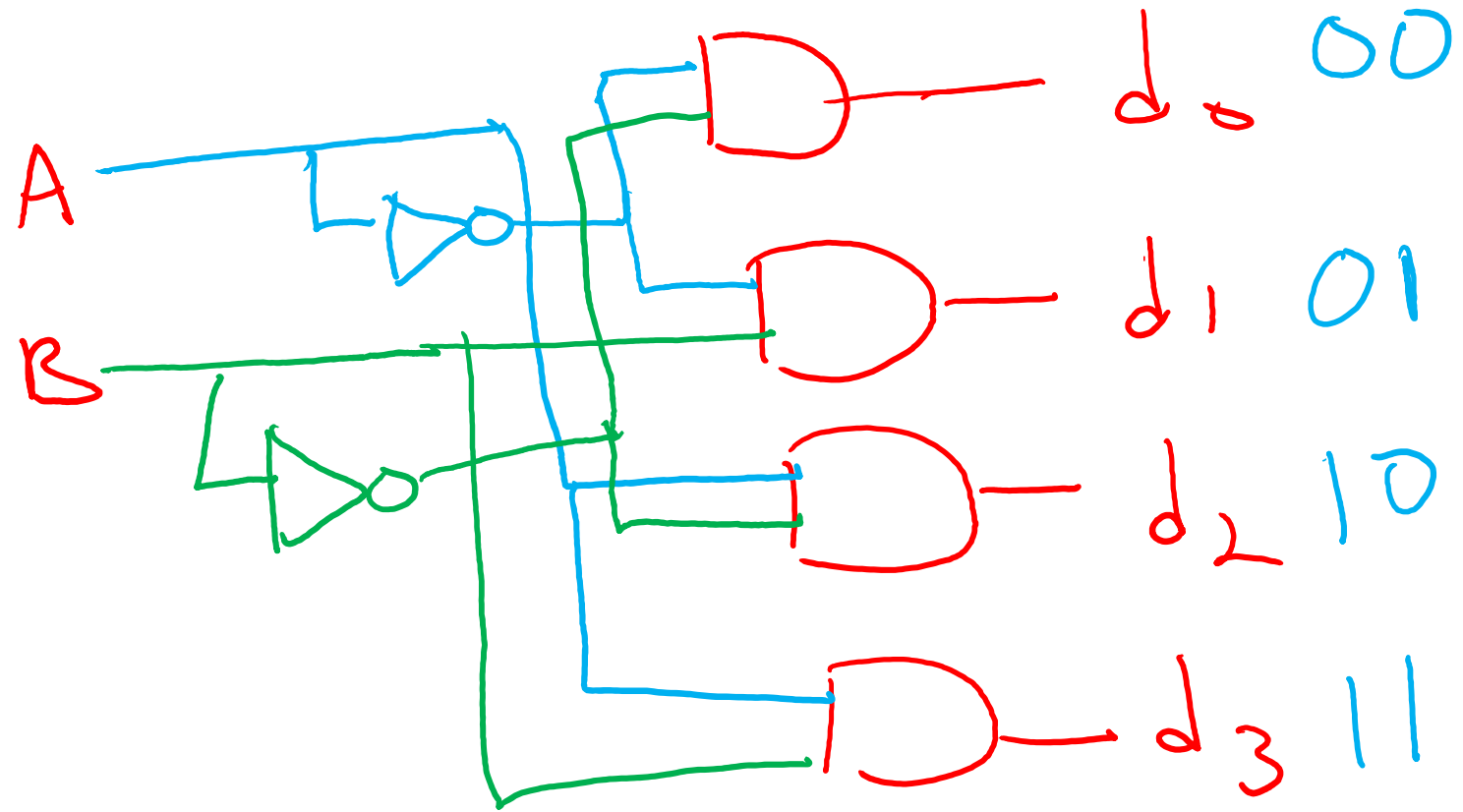


# Creating a Decoder

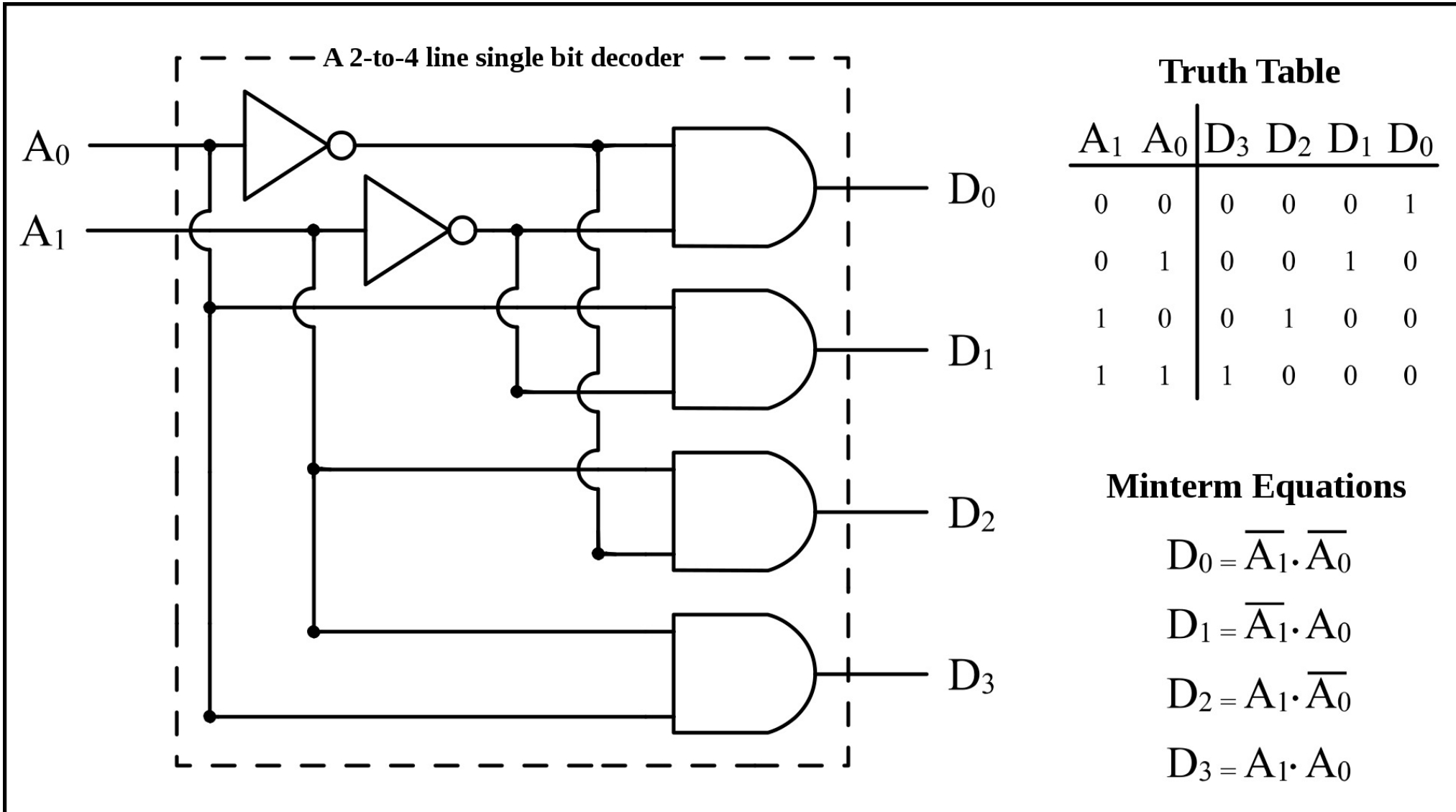


- For each input signal, we need the signal and its complement
- Signals are connected to outputs via AND gates so that inputs turn on output that the digit they represent is present in

# 2-to-4 Decoder

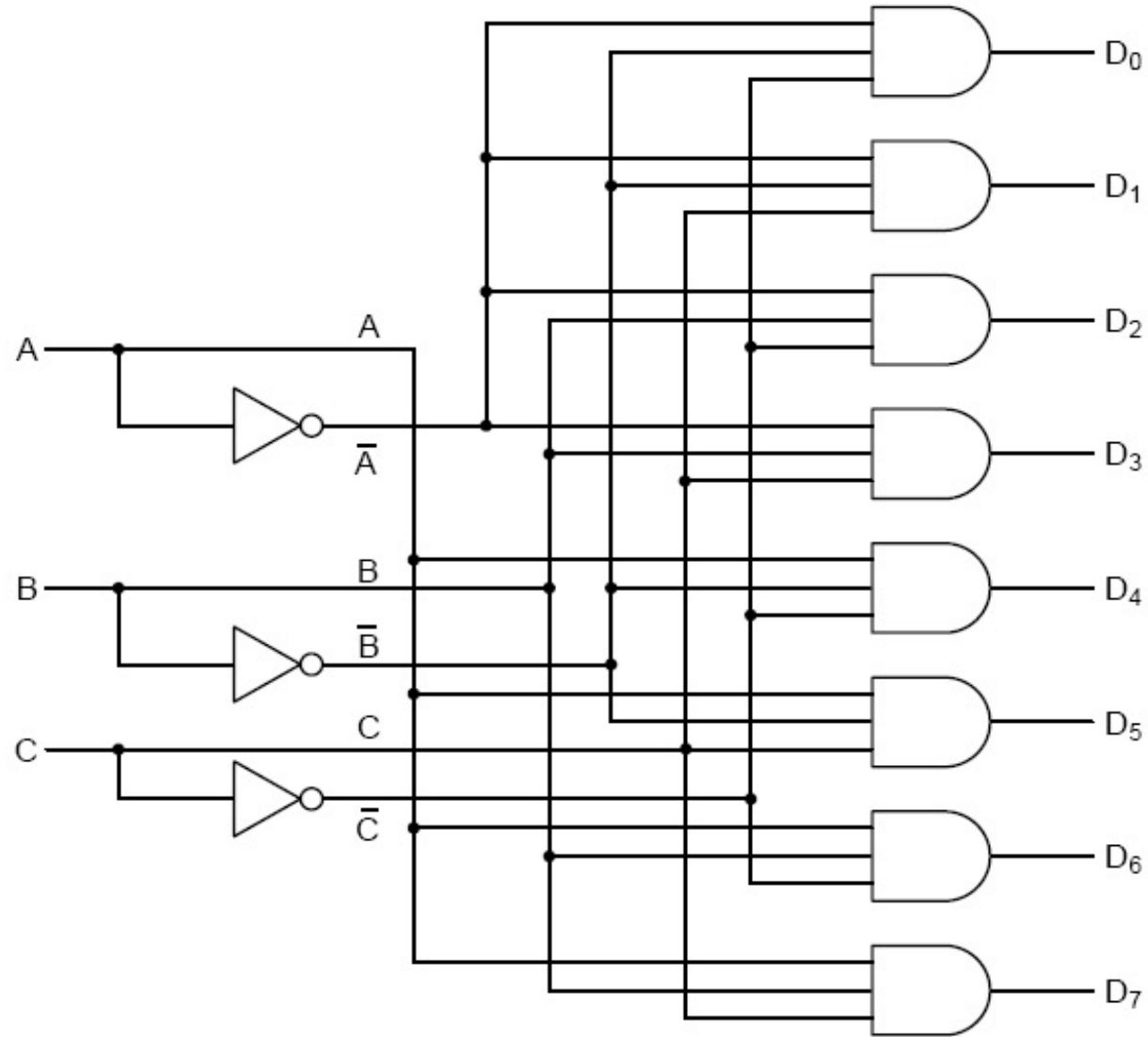


# 2-to-4 Decoder

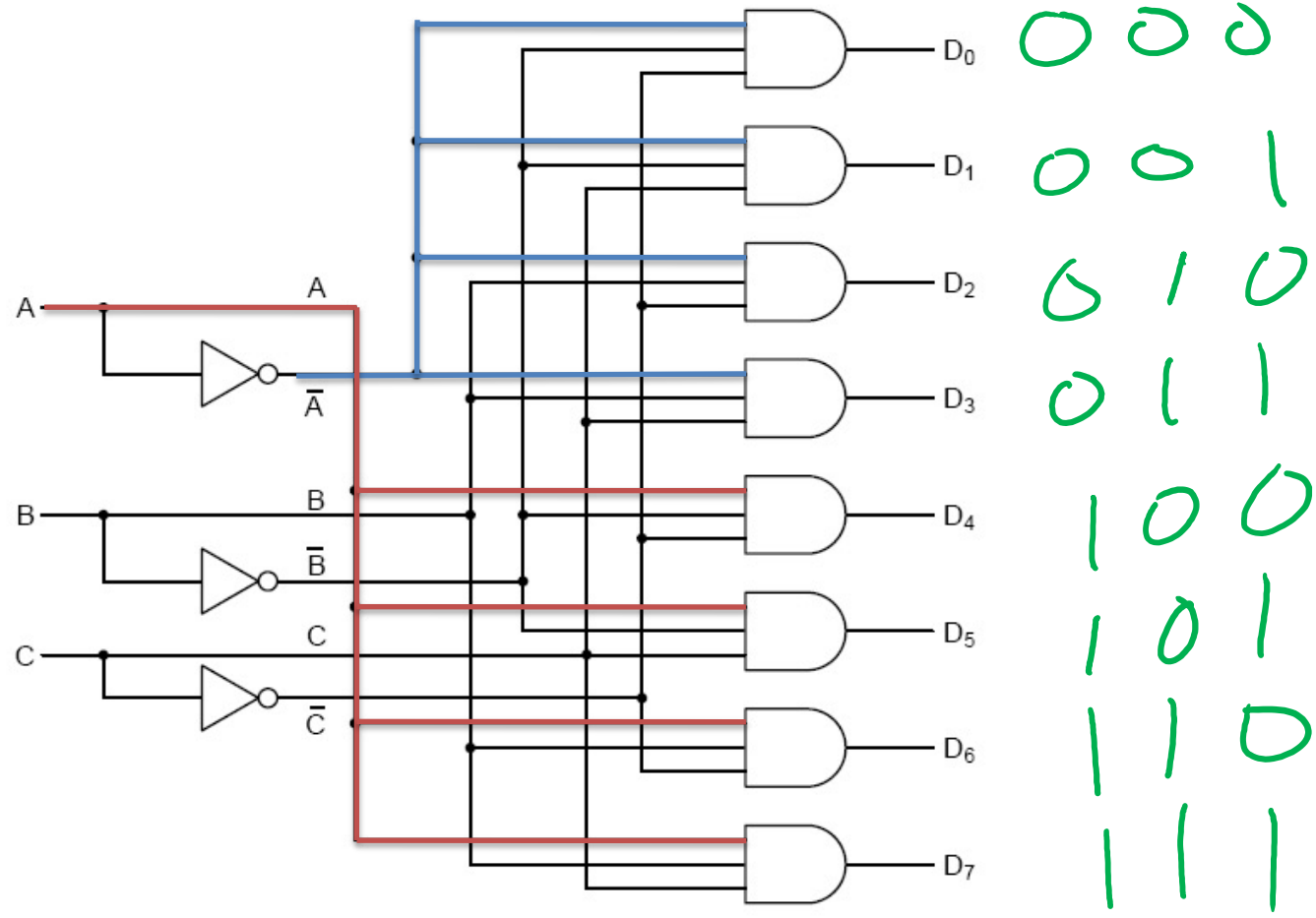




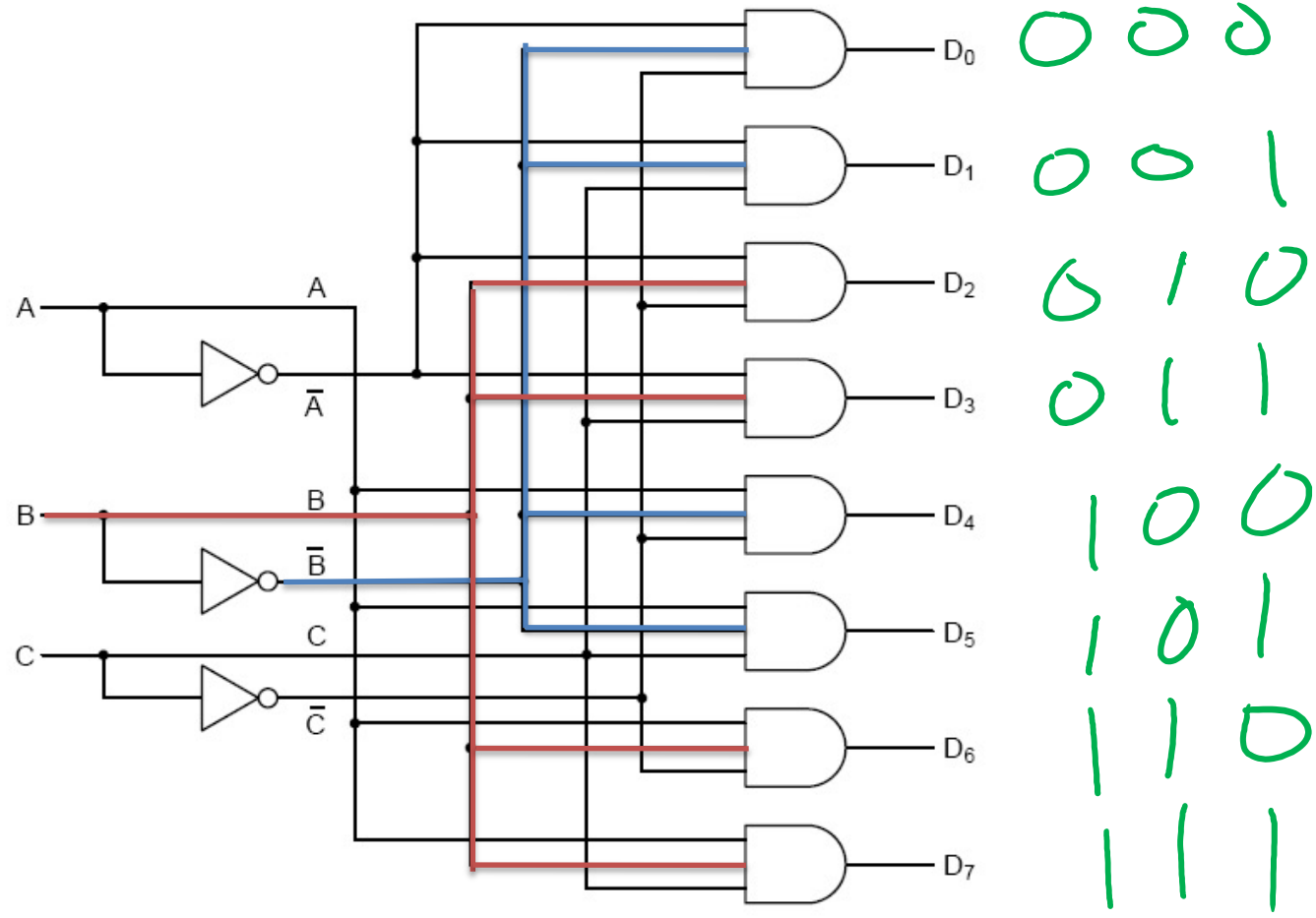
# 3-to-8 Decoder



# 3-to-8 Decoder, A is our MSB

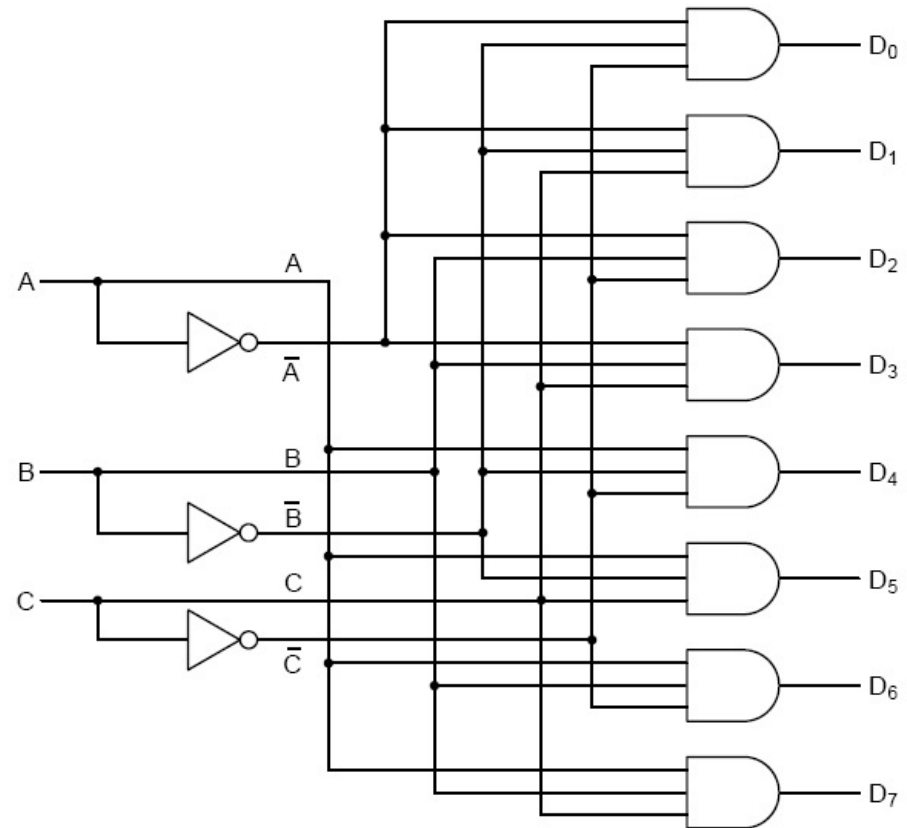


# 3-to-8 Decoder, B

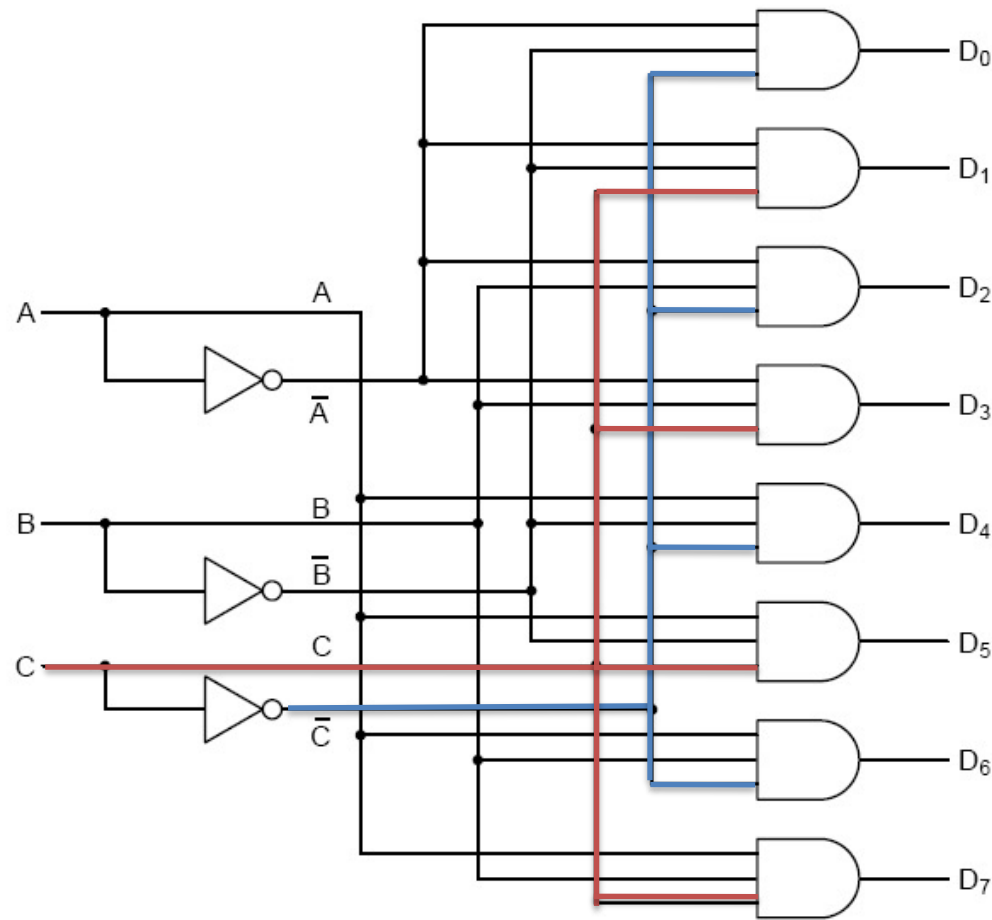


C corresponds to the lowest order bit for the input. In our 3-bit decoder, if C is 1, we should send 0 to \_ and 1 to \_

Clicker	0	1
A	D0, D1, D2, D3	D4, D5, D6, D7
B	D4, D5, D6, D7	D0, D1, D2, D3
C	D0, D2, D4, D6	D1, D3, D5, D7
D	D1, D3, D5, D7	D0, D2, D4, D6



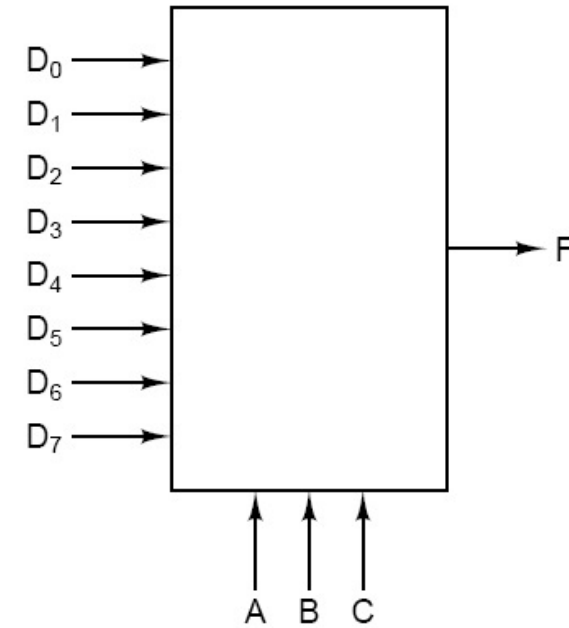
# 3-8 Decoder, C



0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

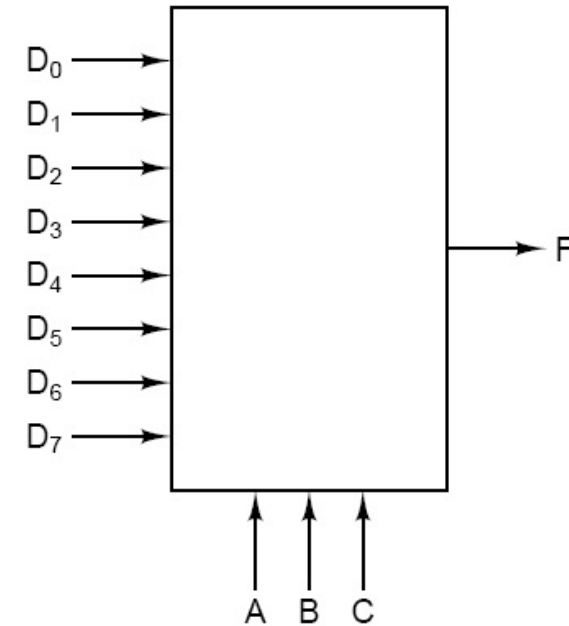
# Multiplexer

- Select one signal from a group of  $2^n$  inputs, to be output on a single output line.



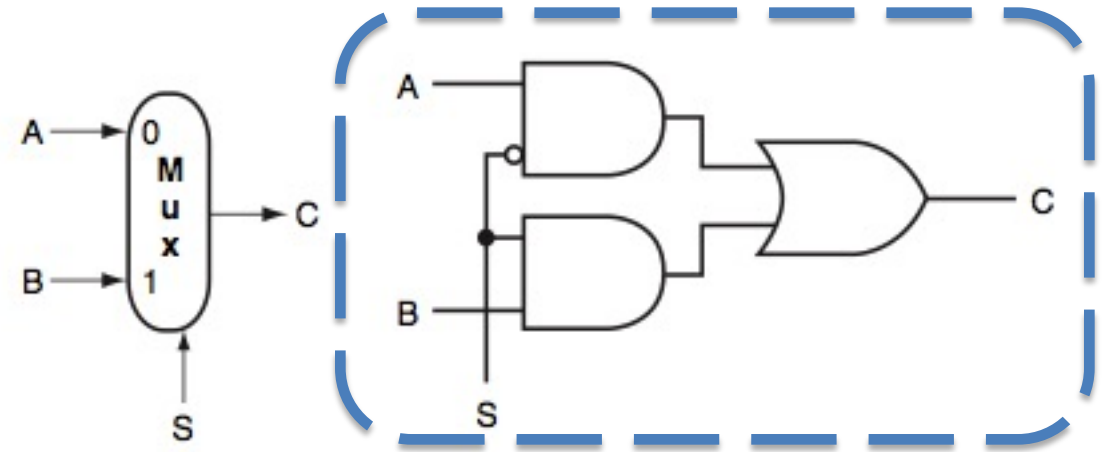
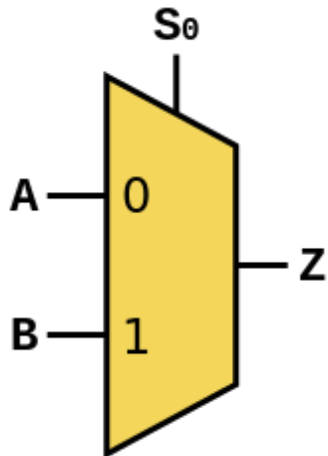
# Multiplexer

- Lines  $D_0, \dots, D_7$  are the data input lines and  $F$  is the output line.
- Lines  $A$ ,  $B$ , and  $C$  are called the select lines. They are interpreted as a three-bit binary number, which is used to choose one of the  $D$  lines to be output on line  $F$ .



# 2-to-1 Multiplexer

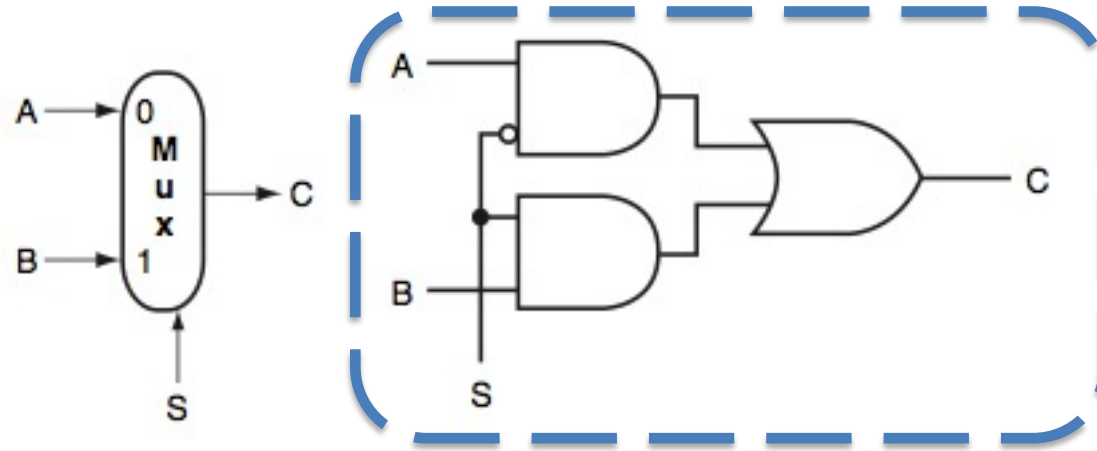
- S is the selector
- 0 selects A; 1 selects B
- Common circuit symbol



Implementation



A = 1, B = 0, and S is 1. C will be



- A. 0
- B. 1
- C. Unclear

# Arbitrarily Large Multiplexer

- If there are  $n$  data inputs, there will need to be  $\lceil \log_2 n \rceil$  selector inputs.
- The multiplexer consists of
  - A decoder that generates  $n$  signals, each indicating a different control value
  - An array of  $n$  AND gates, each combining one of the inputs with a signal from the decoder
  - A single large OR gate that incorporates the outputs of the AND gates

# Reading

- Next lecture: ALU
  - Read Section 3.4
- Problem Set 5
  - Due Friday
- Lab 4
  - Due Sunday