CSCI 210: Computer Architecture
Lecture 17: Combinational Logic

Stephen Checkoway
Oberlin College
Nov. 10, 2021
Slides from Cynthia Taylor

Announcements

- Problem Set 5 due Friday!
- Lab 4 due Sunday
- Join 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, microprocessors

Programmable Logic Array

- Set of inputs and inverted inputs
- Array of AND gates
  - Form set of product terms
- Array of OR gates
  - Logical sum of product terms

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.

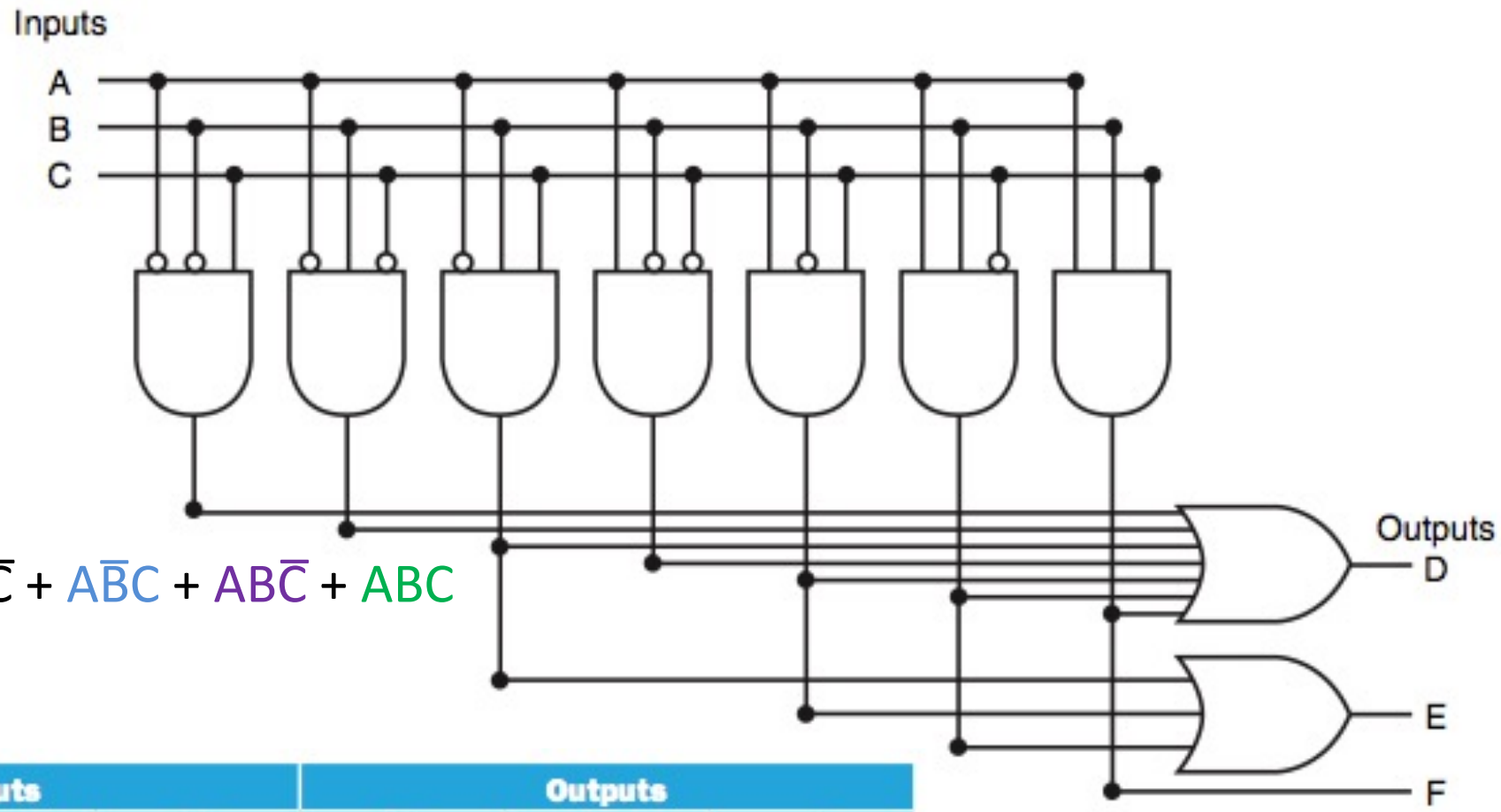
Size

- 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

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



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

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

$F = ABC$

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

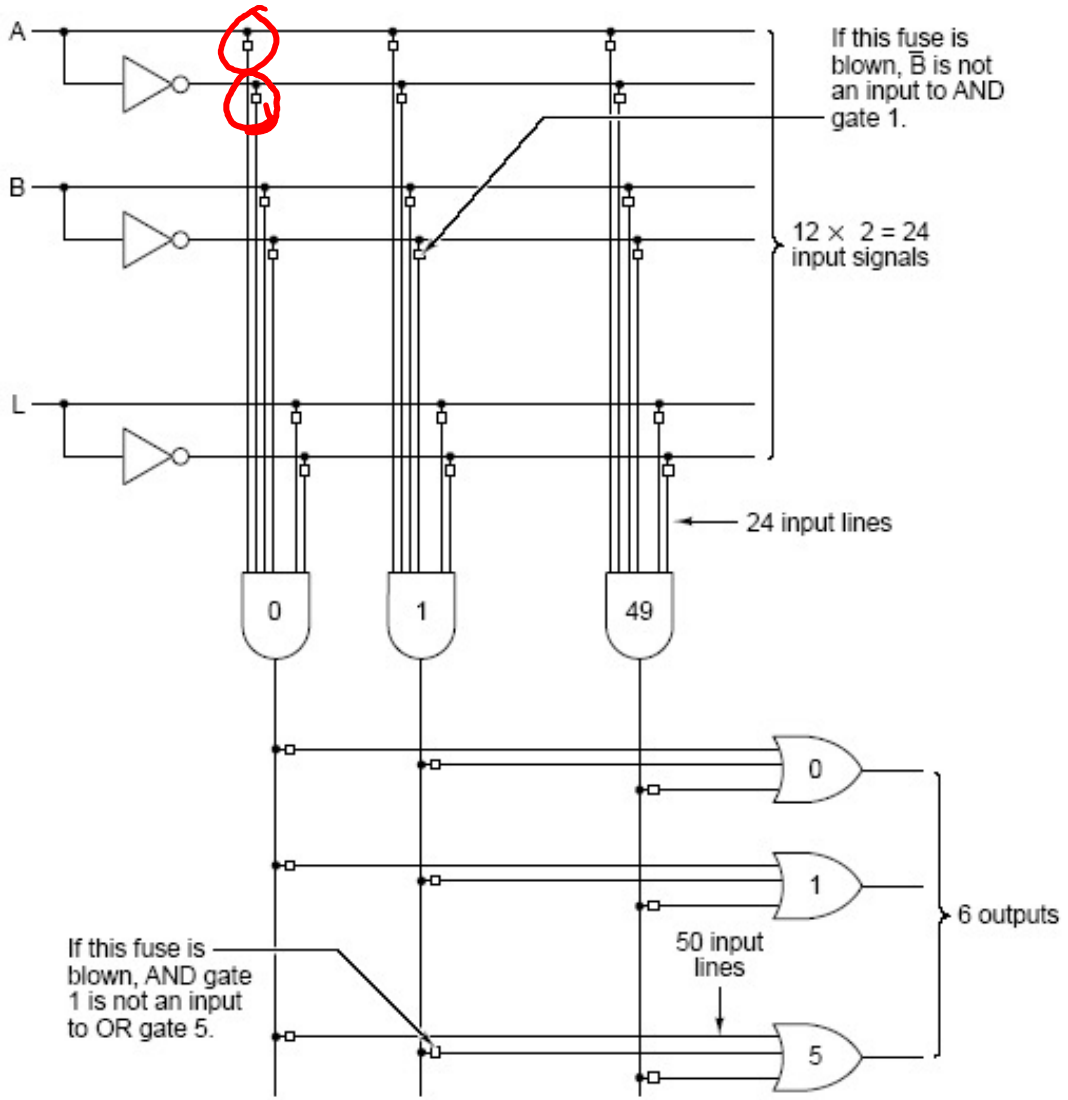
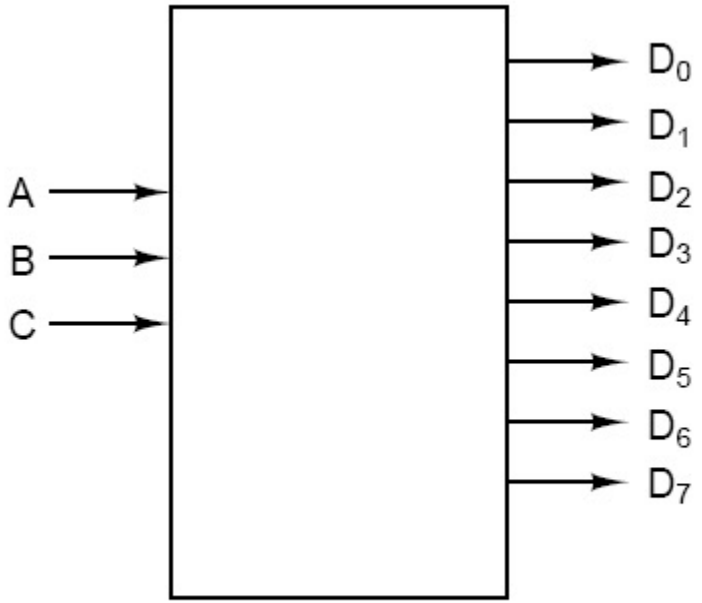


Figure 3-15. A 12-input, 6-output programmable logic array.

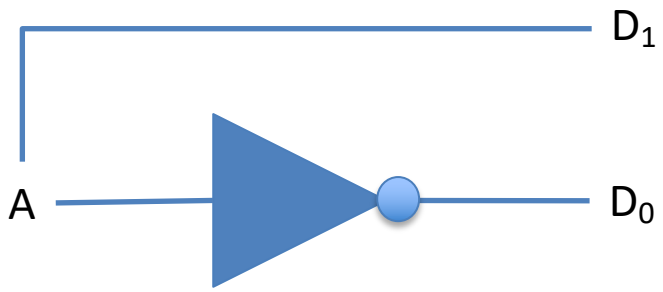
Decoder

- Interprets n inputs (ABC) as an n-bit binary number
- Sets output Dn to 1, all other outputs to 0
- The output is “one hot”



3-to-8 decoder

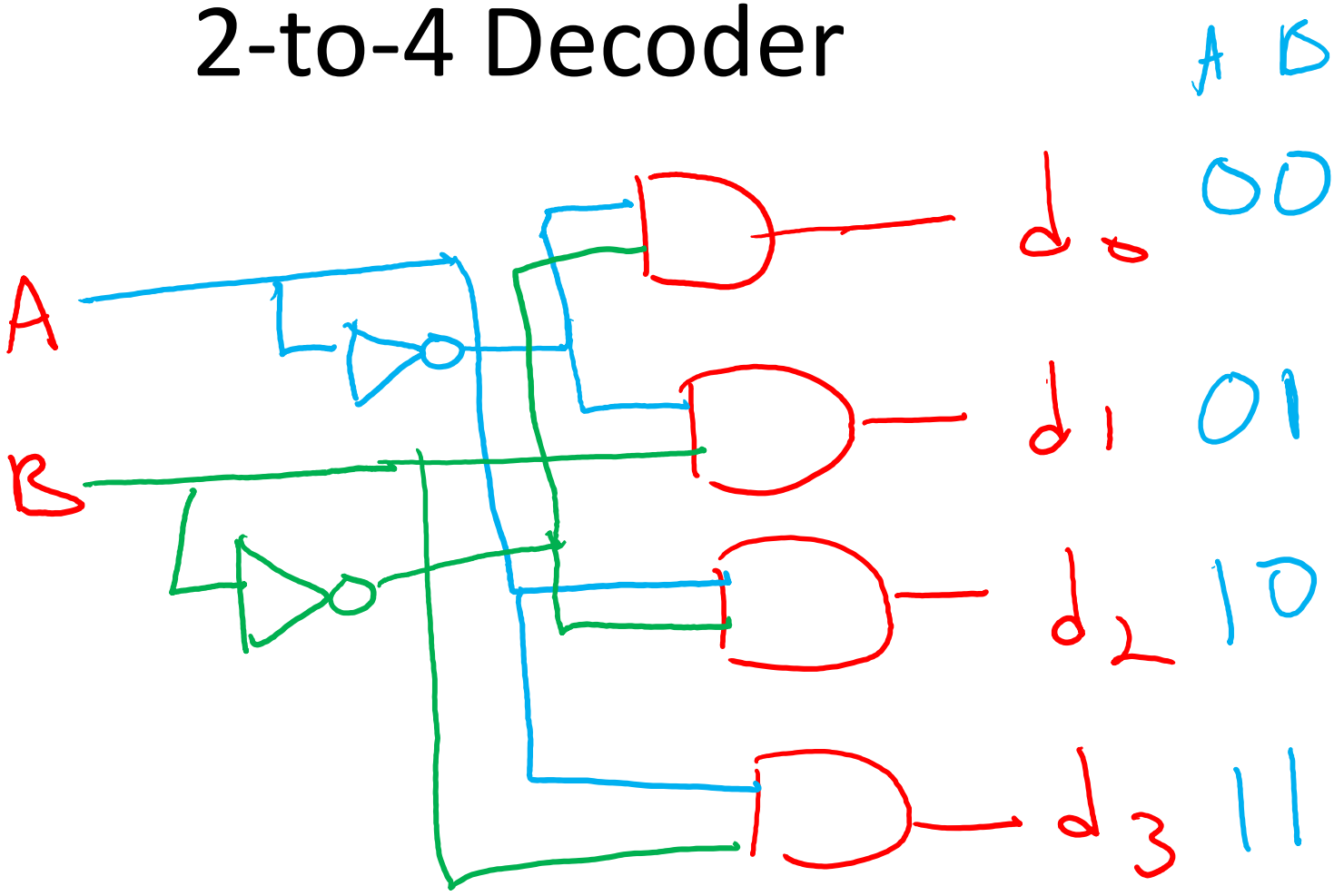
Creating a Decoder



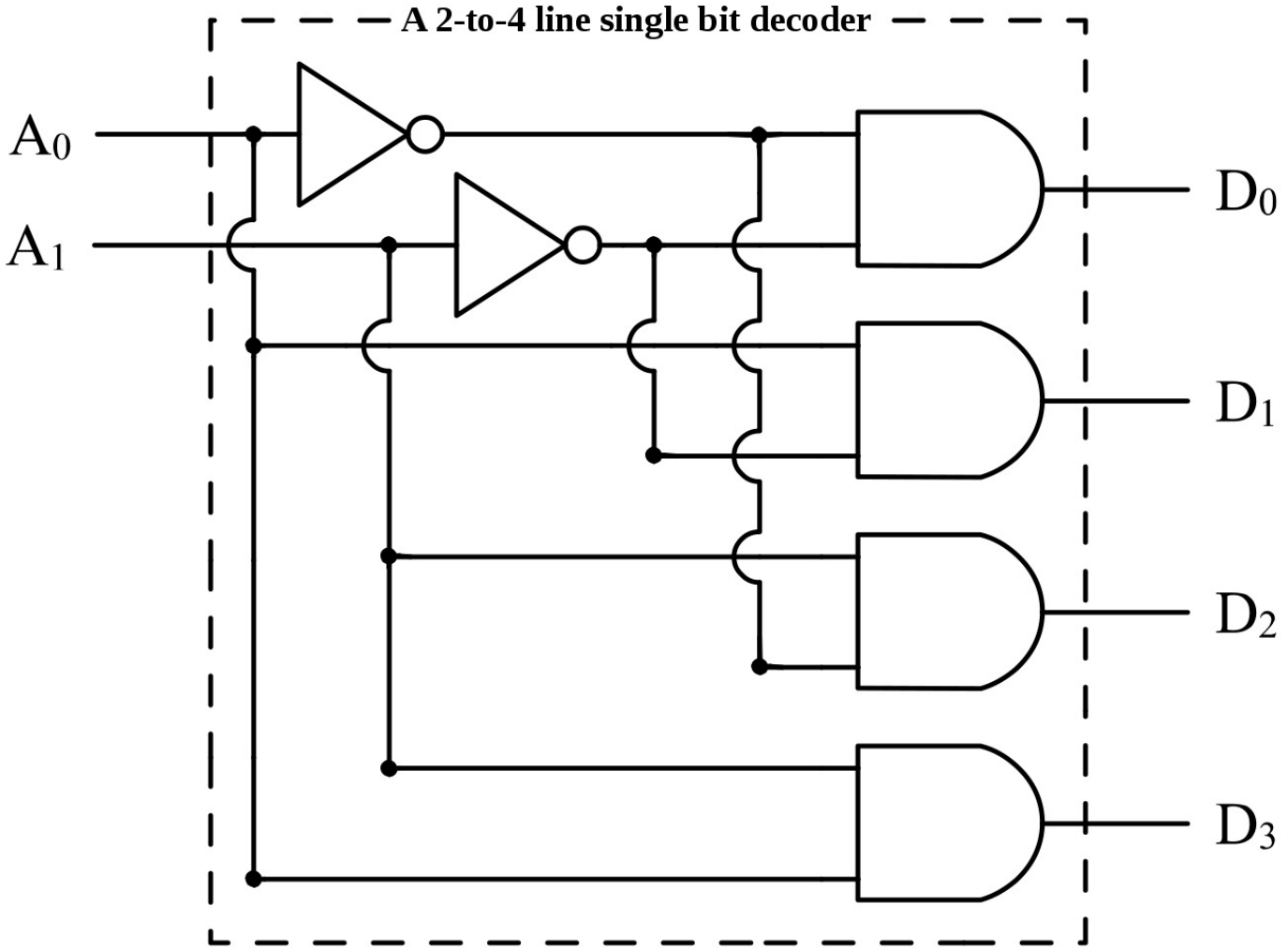
- Each input sends both its signal, and the inverse of its signal

- 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



Truth Table

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

Minterm Equations

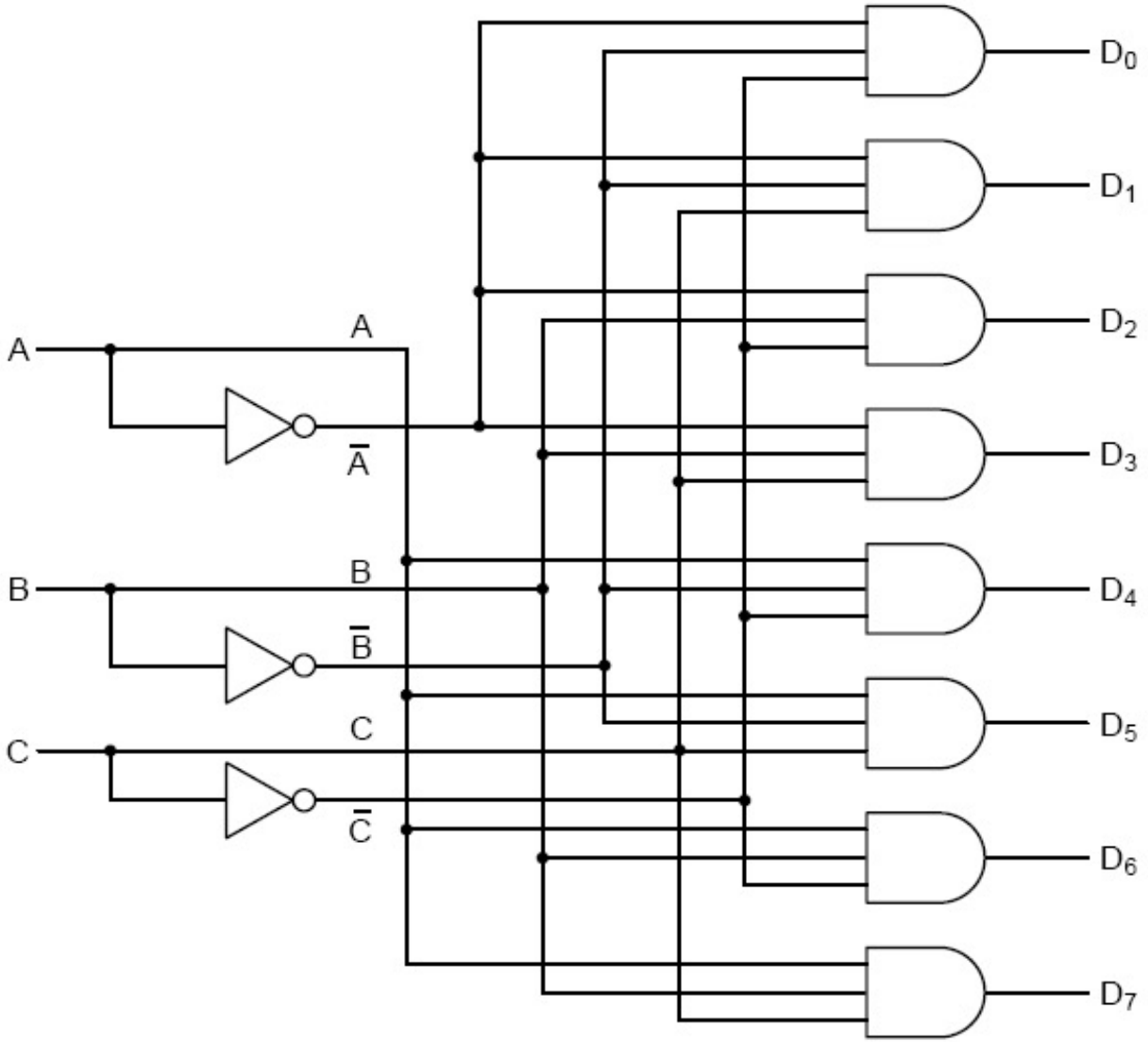
$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = \overline{A_1} \cdot A_0$$

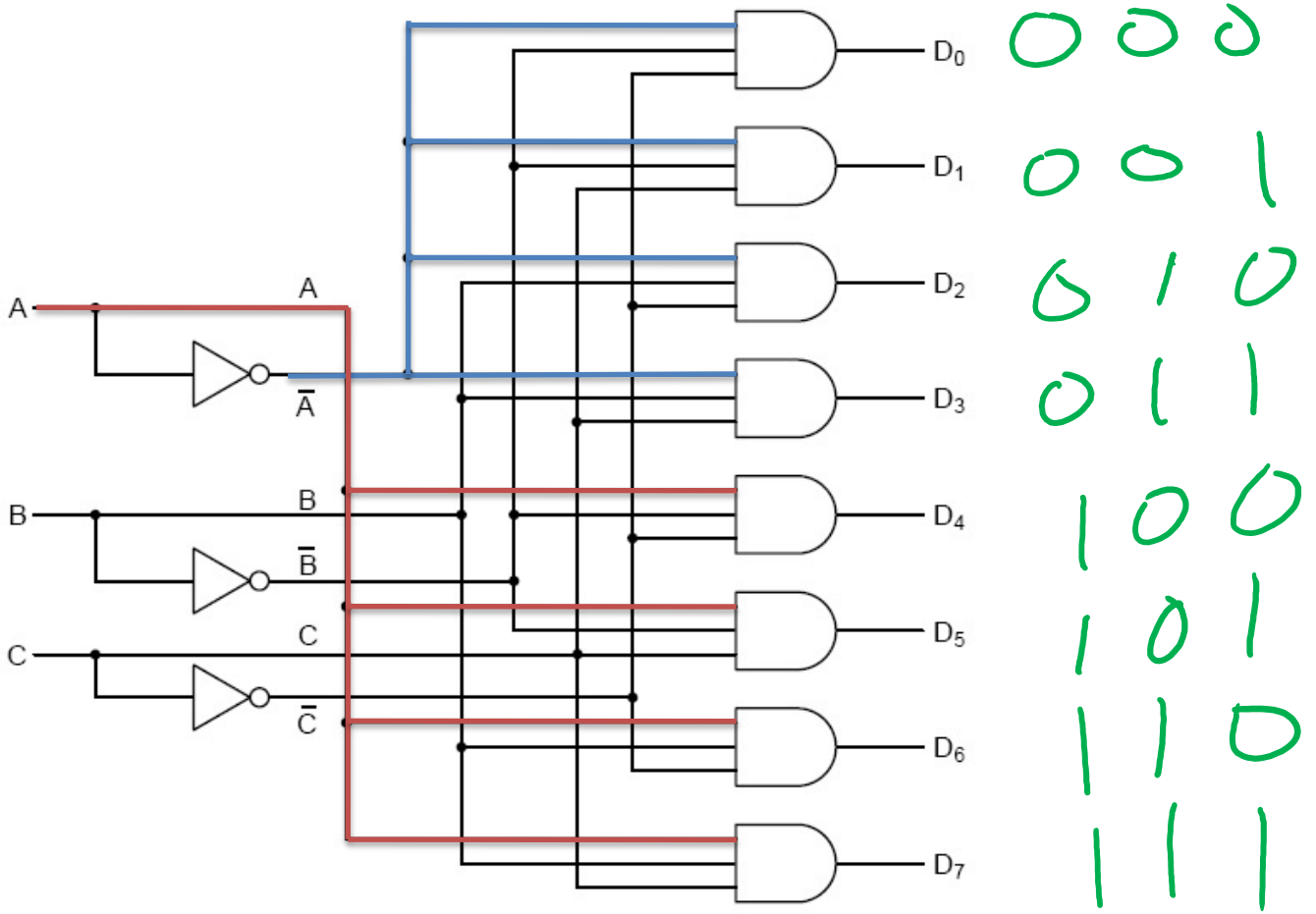
$$D_2 = A_1 \cdot \overline{A_0}$$

$$D_3 = A_1 \cdot A_0$$

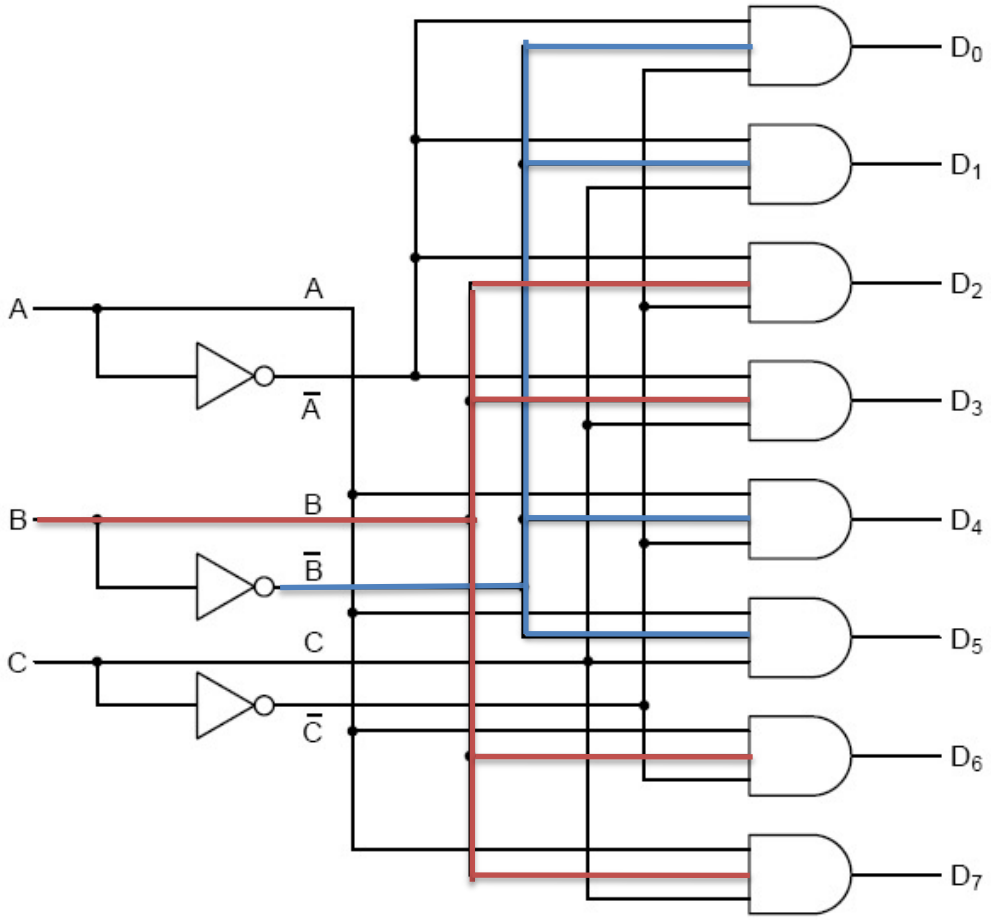
3-to-8 Decoder



3-to-8 Decoder, A is our MSB



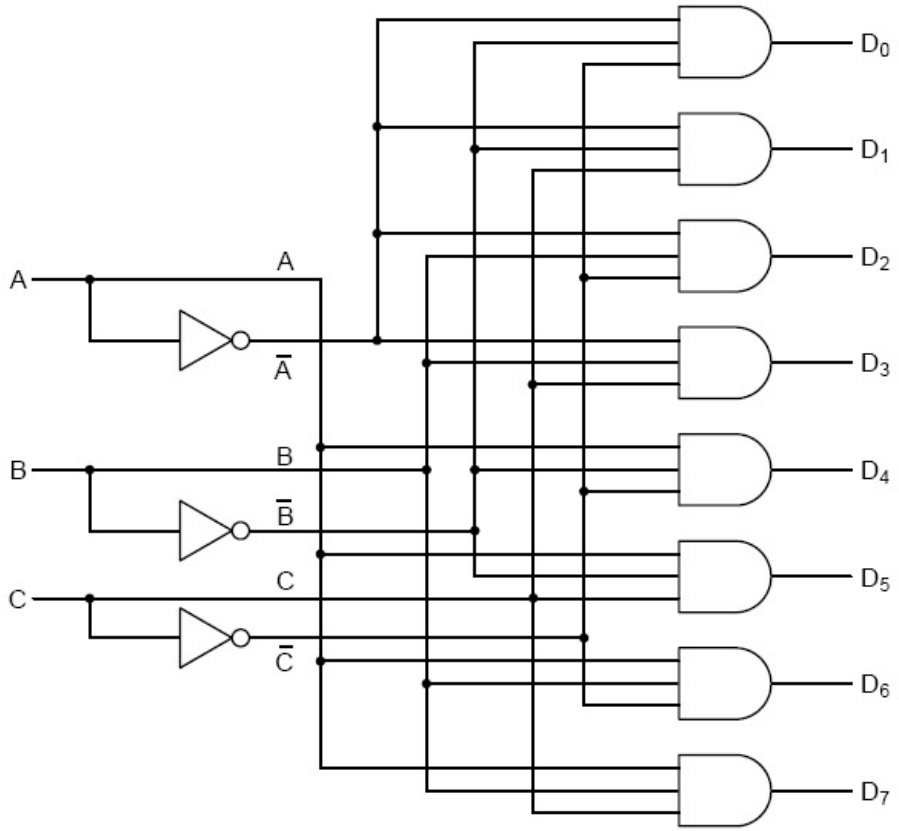
3-to-8 Decoder, B



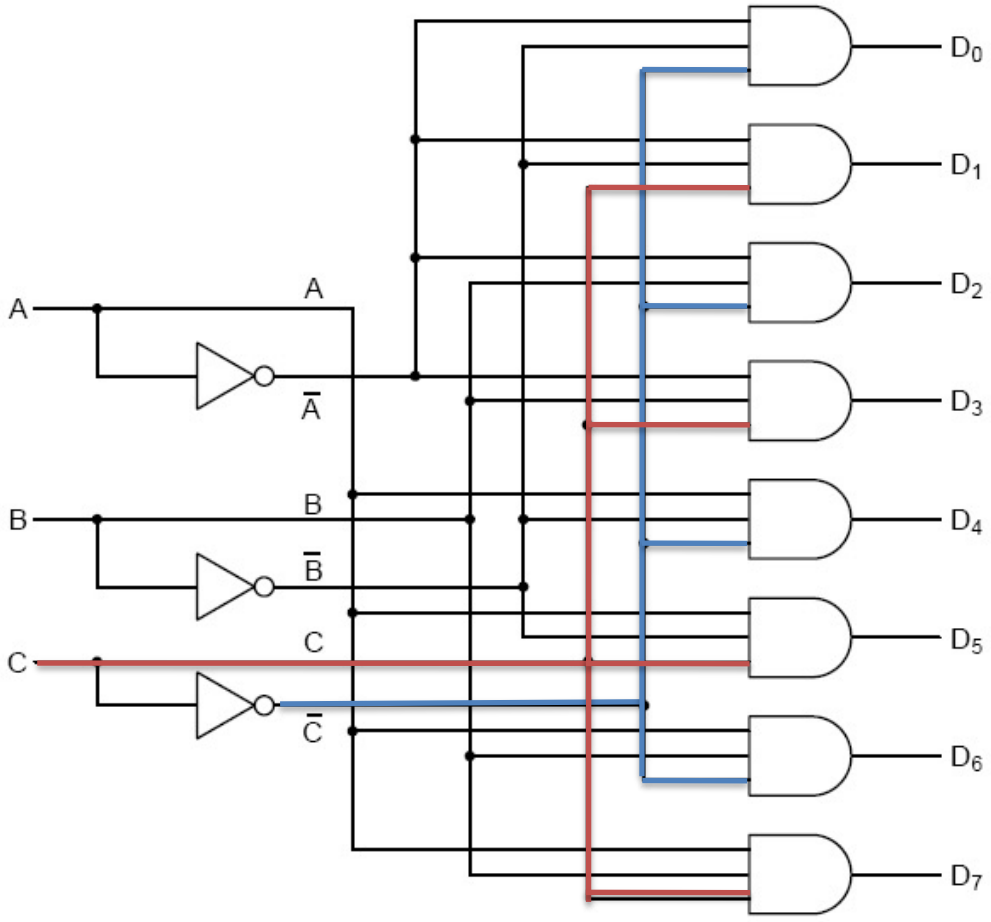
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

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



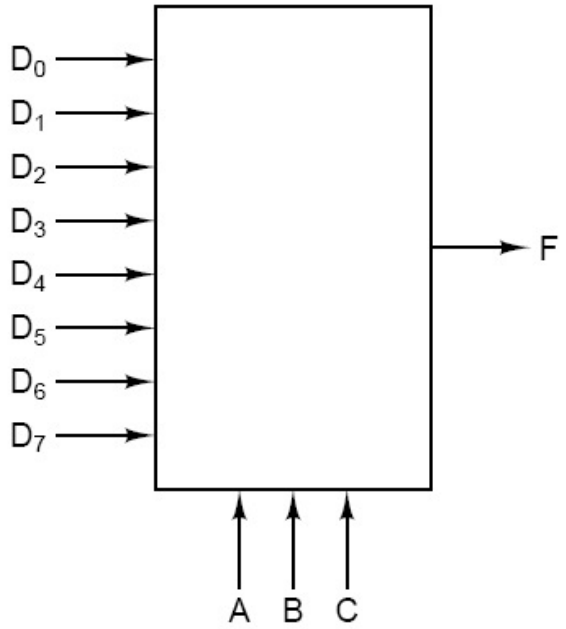
3-8 Decoder, C



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

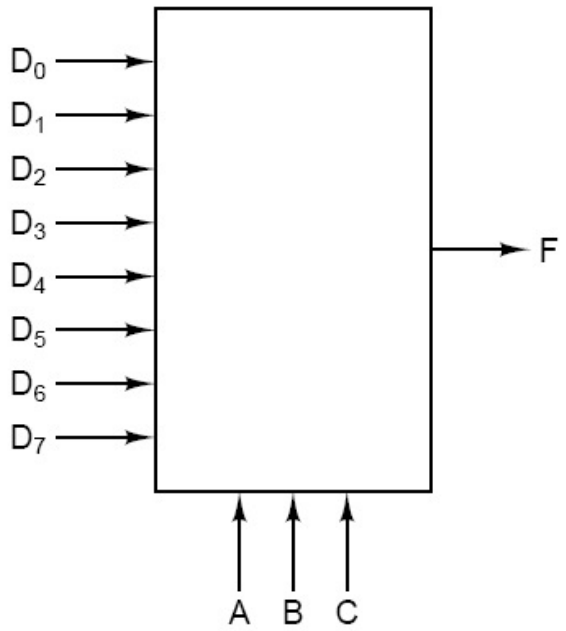
Multiplexer

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



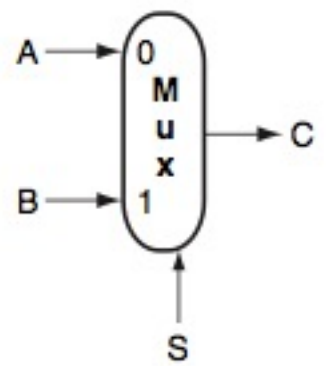
Multiplexer

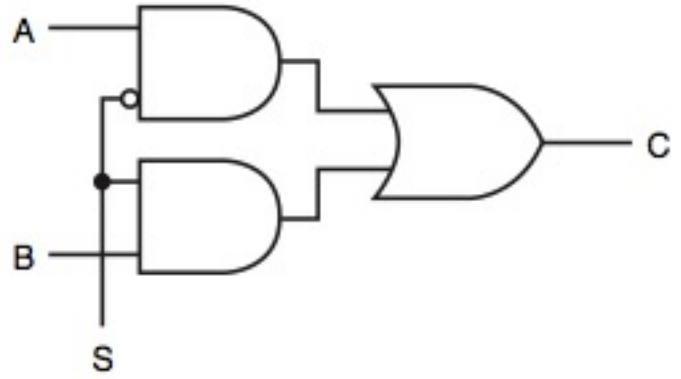
- Lines $D_0,\ldots,D_7$ are the data input lines and F is the output line.
- Lines A, B, and C are called the select lines. They are interpreted as a three-bit binary number, which is used to choose one of the D lines to be output on line F.

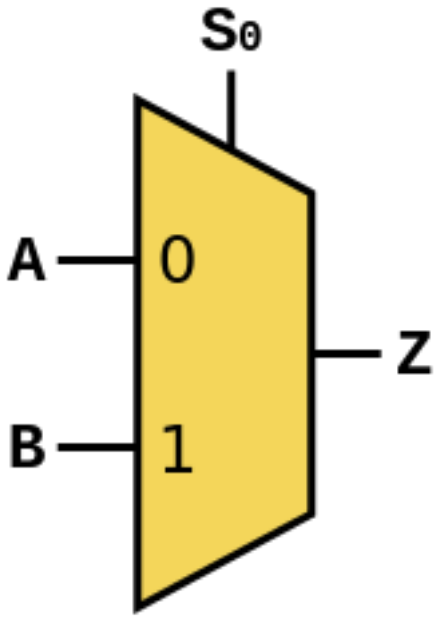


2-to-1 Multiplexer

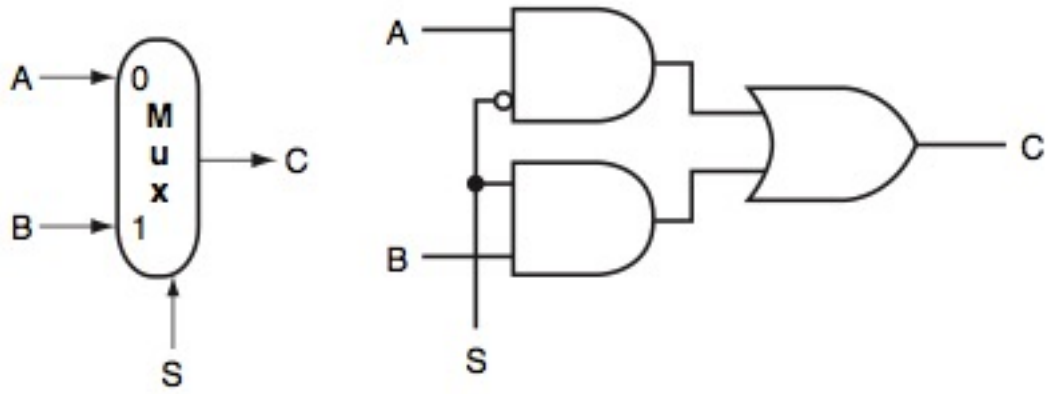
- S is the selector
- 0 selects A; 1 selects B
- Common circuit symbol







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



A. 0

B. 1

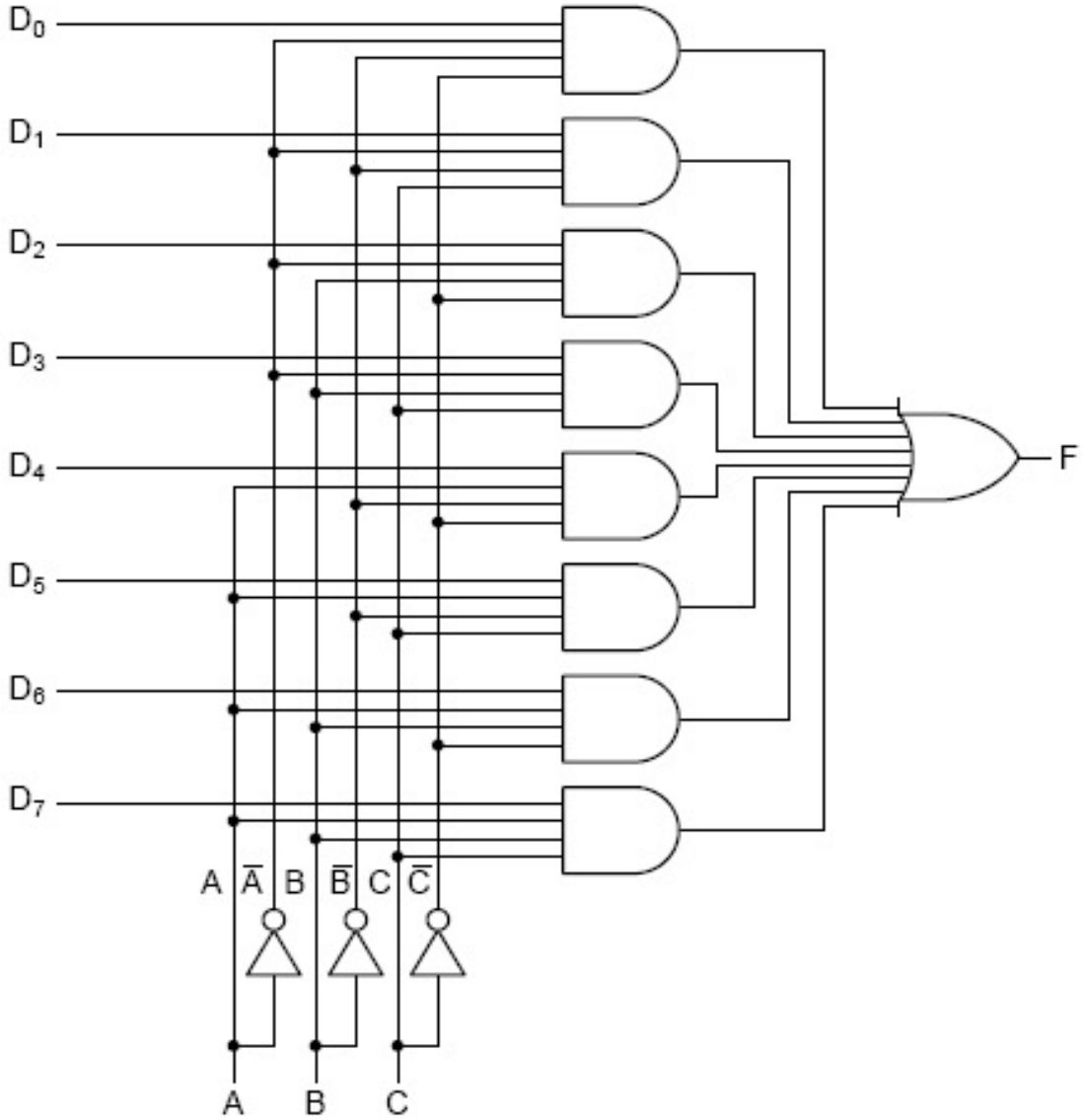
C. Unclear

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

8-to-1 multiplexer



Implementing Functions

- A 2^n-to-1 multiplexer can be used to implement an arbitrary Boolean function of n variables, by associating each input line of the multiplexer with a row of the truth table for the function.

A	B	F
1	1	1
1	0	0
0	1	1
0	0	1

To create a multiplexer for this function, if A B C are our controls, and we want to output F, what should D_0, …, D_7 be?

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

A. 0,0,0,0,1,1,0,1

B. 1,1,1,1,0,0,1,0

C. 0,0,0,1,0,0,1,1

D. 0,0,0,0,1,1,1,1

Reading

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