

ENGR 210 / CSCI B441

Test

# Introduction

Andrew Lukefahr

*Course Website*

enrgr210.github.io

Write that down!

# *Introduction*

- This is a 3-credit course introductions to organization and logic design of digital systems.
- Topics covered:
  - Boolean algebra and logic gates
  - Sequential Logic
  - State Machines
  - Serial Communication
    - Buses
    - Protocols

# *Prof. Lukefahr*



Andrew Lukefahr, Assistant Professor

Office: 2032 Luddy Hall

Email: [lukefahr@iu.edu](mailto:lukefahr@iu.edu)

Research work on security for FPGA-based systems.

# *Prof. Himebaugh*



Bryce Himebaugh, Clinical Assistant Professor

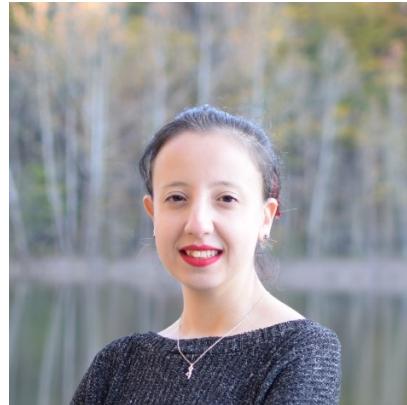
Office: 4146 Luddy Hall

Email: [bhimebau@iu.edu](mailto:bhimebau@iu.edu)

Research work on low-power embedded systems and IoT devices.

# TAs

- Merve Gokce Kurtoglu
- meggokce@iu.edu



- Malintha Fernando
- ccfernand@iu.edu



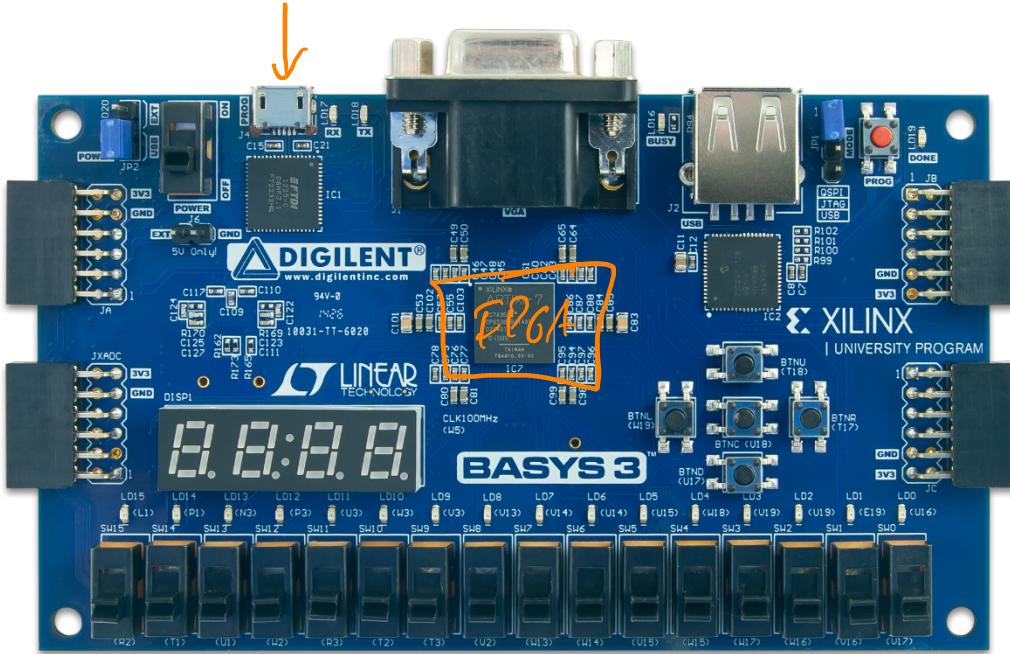
# *Projects*

- Initially assigned one week / due the next
- Gradually become multi-week
- Two Parts:
  - Submit Verilog code to the autograder
  - ~~Demonstrate functioning FPGA implementation to TA~~  
*Not for 3 weeks!*
- More details later in a few minutes

# *Submit Your Own Work*

- All submitted work must be your own
  - Not your buddy's
  - Not last semester's
  - Not the internet's
- I've done this before. I will catch you.
  - Ask last year's class

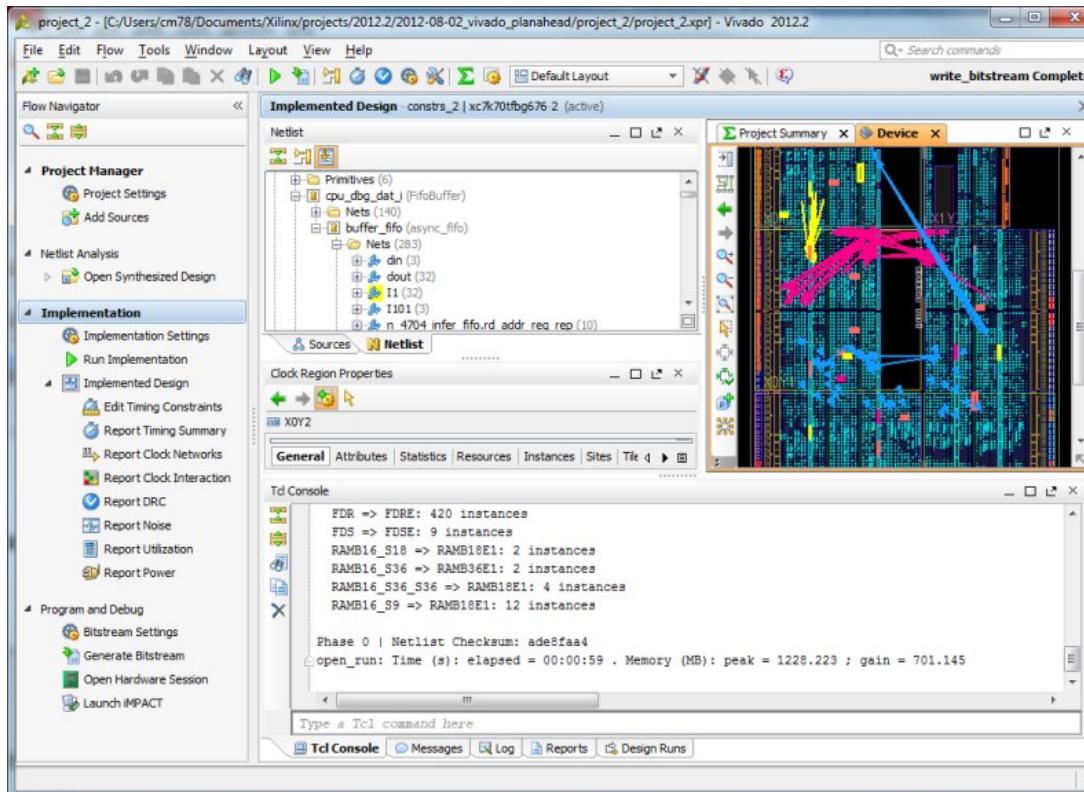
# Basys3 Boards



- Checkout board for semester ~~next time~~ in 3 weeks
- Programmed with Xilinx Vivado
  - Available in Luddy 4111 or download yourself

# Xilinx Vivado Design Suite

- Used to map Verilog code to an FPGA
- Professional tool with multiple steps



# *The Syllabus*

enrgr210.github.io  
Syllabus!

- It's on-topic for exams.
- You need to read it.

# Logic Gates

# *Review*

- Ask a series of (hopefully) review questions.
- If you have never seen this before, **that's ok, but let me know**
- I am happy to help review after class / in office hours.

# *Review Questions*

- What is 0x42 in binary? In decimal?
- What is -5 in 8-bit binary?
- Can you draw an AND gate? OR? NOT? NAND? NOR? XOR?
- Can you draw this circuit:  $D = A \cdot B + C$ ?
- What is the truth table for this:  $D = A \cdot B + C$ ?

## Review: Numbers

C same:  $x = 66$ ; // decimal

$x = \underline{0x42} = x = \underline{'h} 42; // hex$

0100 0010

$x = ?? \quad x = \underline{'b} 0100 0010; // binary$

- What is  $0x42$  in binary? In decimal?



66 66

$$\begin{array}{r} & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \times & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline & 0+64+0+0+0+0+2+0 & = & 66 \end{array}$$

decimal ✓

# Review: 2's Complement

- What is -5 in binary?

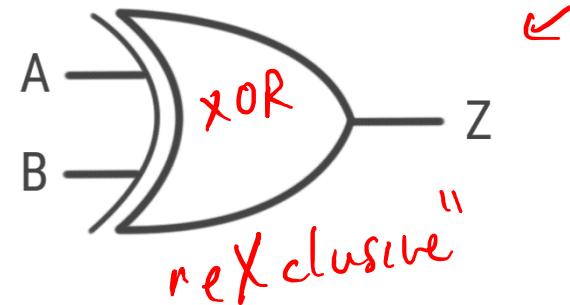
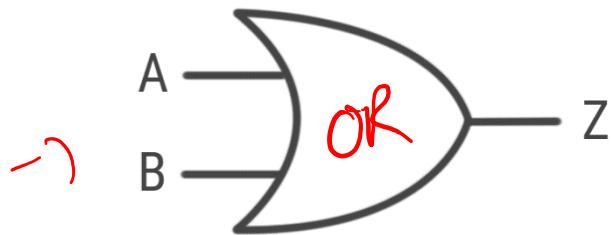
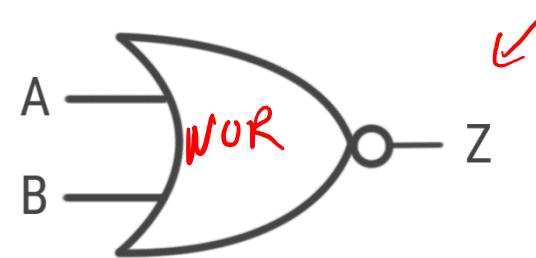
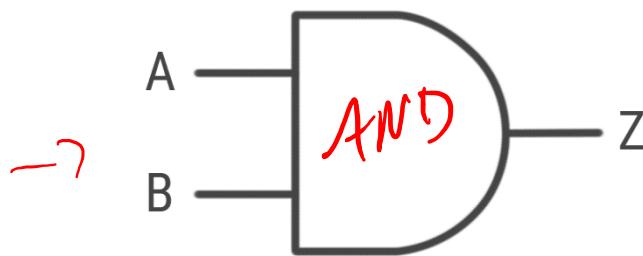
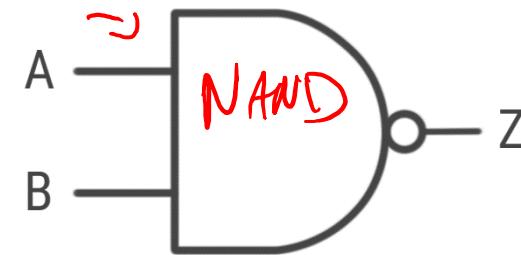
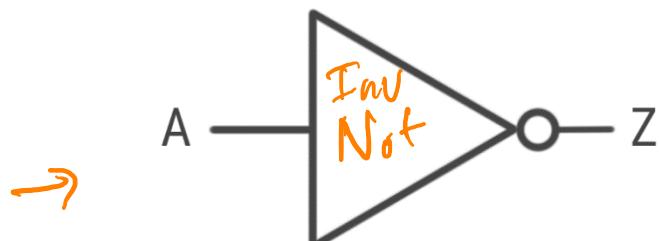
$$+5 = 1001$$

$$\begin{array}{r} 0110 \\ + \quad 1 \\ \hline (-5) \quad 0111 \end{array}$$

Signed Values

→ 1010 negative  
0010 → positive

# Review: Logic Gates

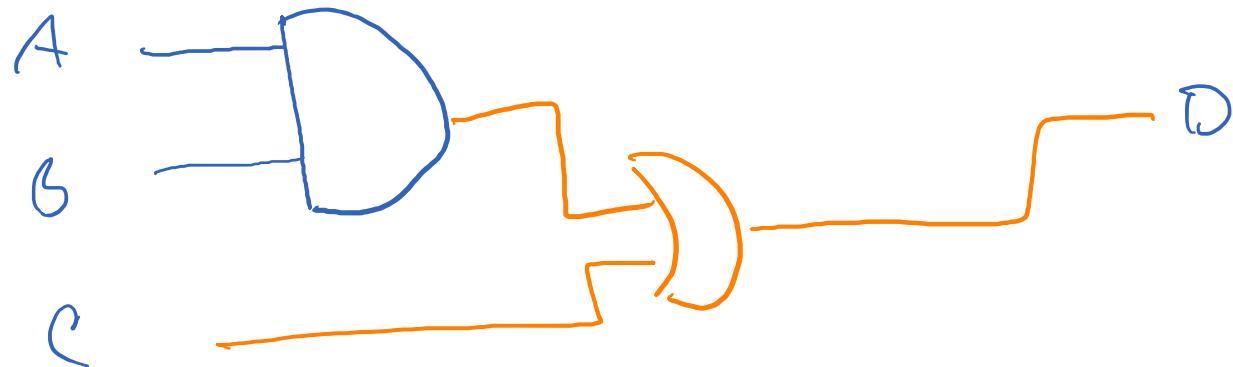


# Review: Boolean Equation

- What circuit is this?

$$D = (A \cdot B) + C$$

D is A "and" B or C



# Review: Truth Table

- What is the truth table for this?

$$D = (A \cdot B) + C$$

Truth Table:

A	B	C	D
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Boolean Expressions:

$$D = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C} + A \cdot B \cdot C$$

# NOT (Inverter)

Not ( $\sim$ )  $\Rightarrow$  bit flip

Not (!)  $\Rightarrow$  logical

- Math:

- $Z = \bar{A}$  ↵

- Code:

- $Z = \sim A$  ↵

- Schematic

A	Z
0	1
1	0

- Math:
  - $Z = \bar{A}$
- Code:
  - $Z = \sim A$
- Schematic

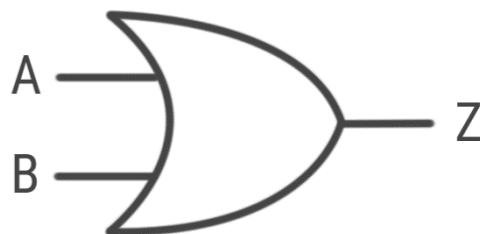
$$S = 0101$$

$$\text{if}(\neg S) = (0101 = 0000) = \text{false}$$

$$\text{if}(\sim S) = 0101 \rightarrow \underline{1010} = \text{true}$$

# OR

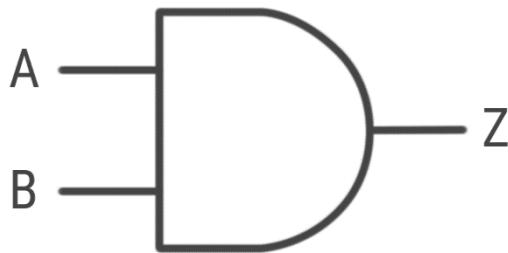
- Math:  $Z = A + B$
- Code:  $Z = A \mid B$
- Schematic



A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

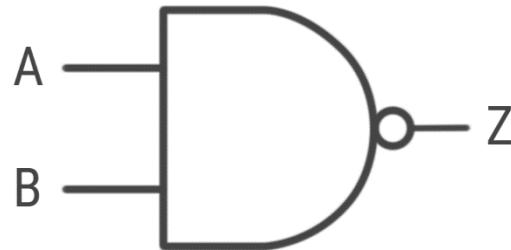
# AND

- Math:  
•  $Z = A \cdot B$
- Code:  
•  $Z = A \& B$
- Schematic



A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

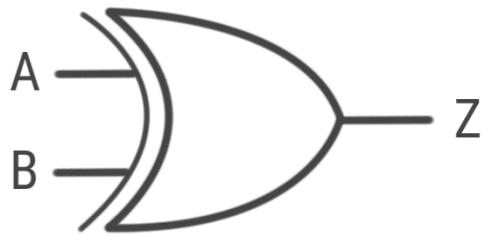
## Other Gates: NAND



$$\cancel{Z} = \neg(A \wedge B)$$

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

## Other Gates: XOR



$$z = A \wedge B$$

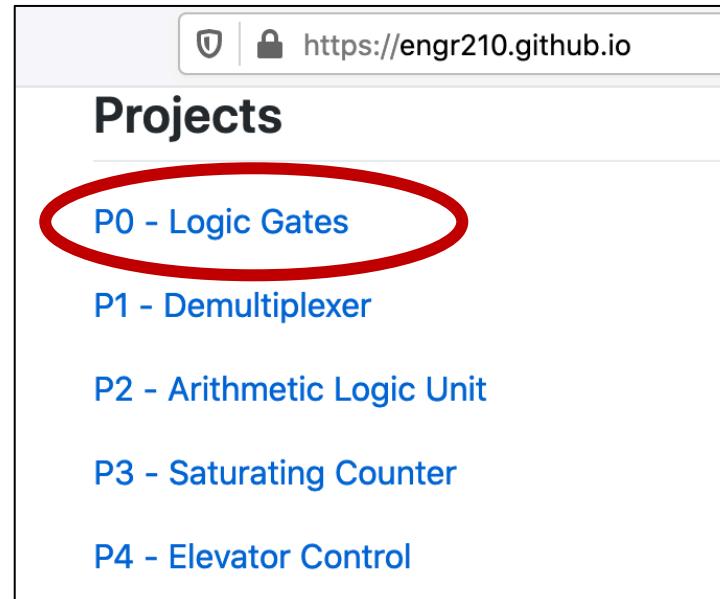
↑  
shift + 6

$$z = A \oplus B$$

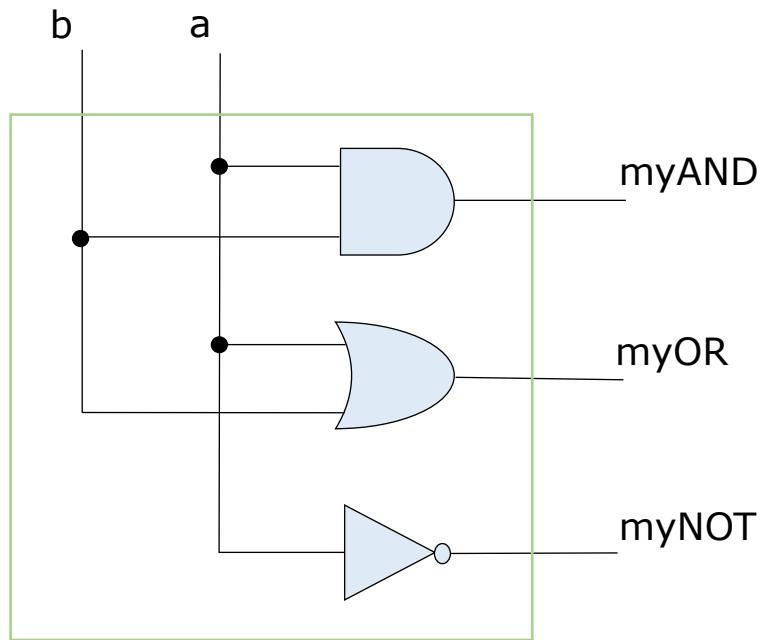
A	B	Z

# *Project 0: Logic Gates*

- This is a ‘demo’ lab.
- It is to provide you a reference design.
- It is NOT DUE!



# *Project 0: Schematic*



# *Project 0: Verilog Logic Operators*

Verilog logic operators:

AND: & ✓

OR: |

NOT: ~

Example:

```
assign myAND = a & b;
```

The constraint file should create the following mapping of input and outputs to the switches and LEDs on the Basys3 board:

<u>Signal</u>	<u>Basys3 input</u>	<u>Signal</u>	<u>Basys3 output</u>
a	sw0	myAND	led0
b	sw1	myNOT	led1
		myOR	led2

*Basys3*

# *Deliverable #1: Autograder*

- Log on to the autograder:  
<http://autograder.sice.indiana.edu>
- Log on with your '**username@iu.edu**' account
- Select '**Engr210.s21**' -> '**P0**'
- Upload:
  - **top.v** : This is your top-level Verilog source file
  - **top\_tb.v**: This is your top-level Verilog **TestBench**

# *Deliverable #1: Autograder*

Courses - B441/E315⚙️ - P1 ⚙️

Hi, Andrew! (lukefahr@iu.edu)  
[Sign out](#)

Submit      My Submissions      Student Lookup

Group members:

lukefahr@iu.edu

submissions are in the queue.

Drop files here  
- or -  
[Upload from your computer](#)

Files to submit	Size

[Submit](#)

# Deliverable #1: Autograder

## Courses - B441/E315⚙️ - P1 ⚙️

Hi, Andrew! (lukefahr@iu.edu)

[Sign out](#)

[Submit](#)

[My Submissions](#)

[Student Lookup](#)

### Final Graded Submission:

02 August 2018 02:50:29 PM 20/20

### All Submissions:

02 August 2018 02:50:29 PM 20/20

23 July 2018 01:14:12 PM 20/20

19 June 2018 04:38:43 PM 20/20

19 June 2018 03:51:47 PM 20/20

19 June 2018 03:42:58 PM 20/20

19 June 2018 03:41:20 PM 20/20

19 June 2018 03:36:32 PM 20/20

19 June 2018 03:31:10 PM 20/20

19 June 2018 03:29:53 PM 20/20

19 June 2018 03:11:37 PM 20/20

19 June 2018 03:10:19 PM 20/20

19 June 2018 02:55:52 PM 20/20

19 June 2018 02:54:29 PM 20/20

19 June 2018 02:50:42 PM 20/20

19 June 2018 02:32:44 PM 20/20

19 June 2018 02:29:37 PM 20/20

19 June 2018 01:29:15 PM 20/20

19 June 2018 01:22:53 PM 20/20

19 June 2018 01:21:46 PM 20/20

Submitted by [lukefahr@iu.edu](#) on **02 August 2018 02:50:29 PM**

### Grading status:

Everything is finished grading!

**Score: 20/20**

[top\\_tb.v](#)

[top.v](#)

Adjust feedback:

### Student Test Suites

Suite Name	Student Tests	Score
------------	---------------	-------

#### TopLevel



9/9

Setup:

Bugs exposed: 3

- top\_bug01
- top\_bug02
- top\_good

Student tests summary:

The rest of your test cases were run against buggy implementations:

- top\_tb.v

### TopLevel

Test Case	Passed	Score
-----------	--------	-------

#### Setup



#### Simulation



11/11

Command	Exit Status	Stdout	Stderr
sim		<input type="button" value="Output"/>	<input type="button" value="Output"/>

## *Deliverable #2: FPGA Demo*

- Use Vivado to:
  - Synthesize your design
  - Program the FPGA
- Verify + ... (no points ☹ )
- ... (the points ☺ )

**No demos for the first 2 projects!**

**COVID:**

# Why 2 Deliverables?

- Encourage testing
  - We give you points for
- Check correctness
  - Automate

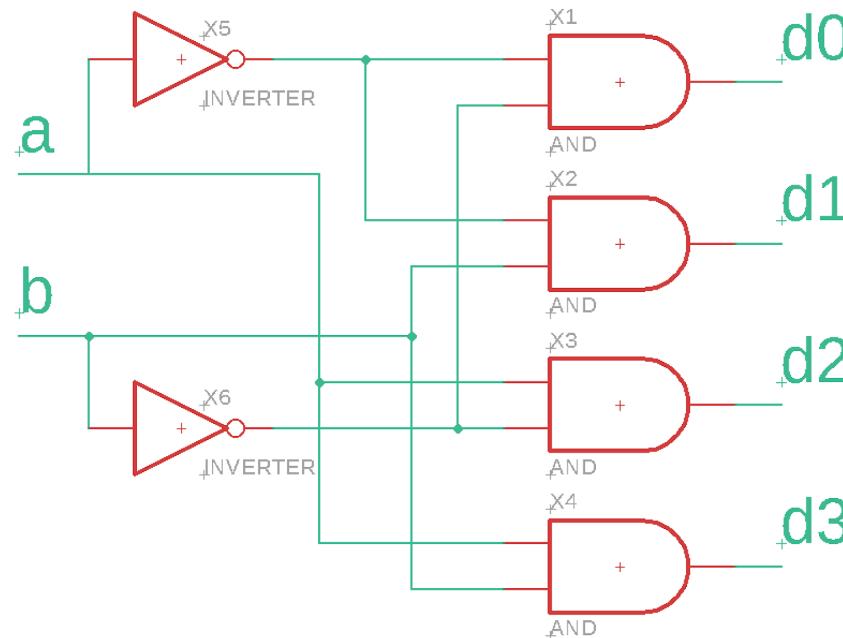
**No demos for the first 2 projects!**

COVID: os

Debug time  
is slow. Don't until you are 100% sure your  
code works.

# Project 1: Demultiplexer

- Create a 3-to-8 demultiplexer in Vivado.
- Here's a 2-to-4 demultiplexer example.



# Next Time

- Truth Tables

Labs  
Zoom Link  
Slack doesn't work