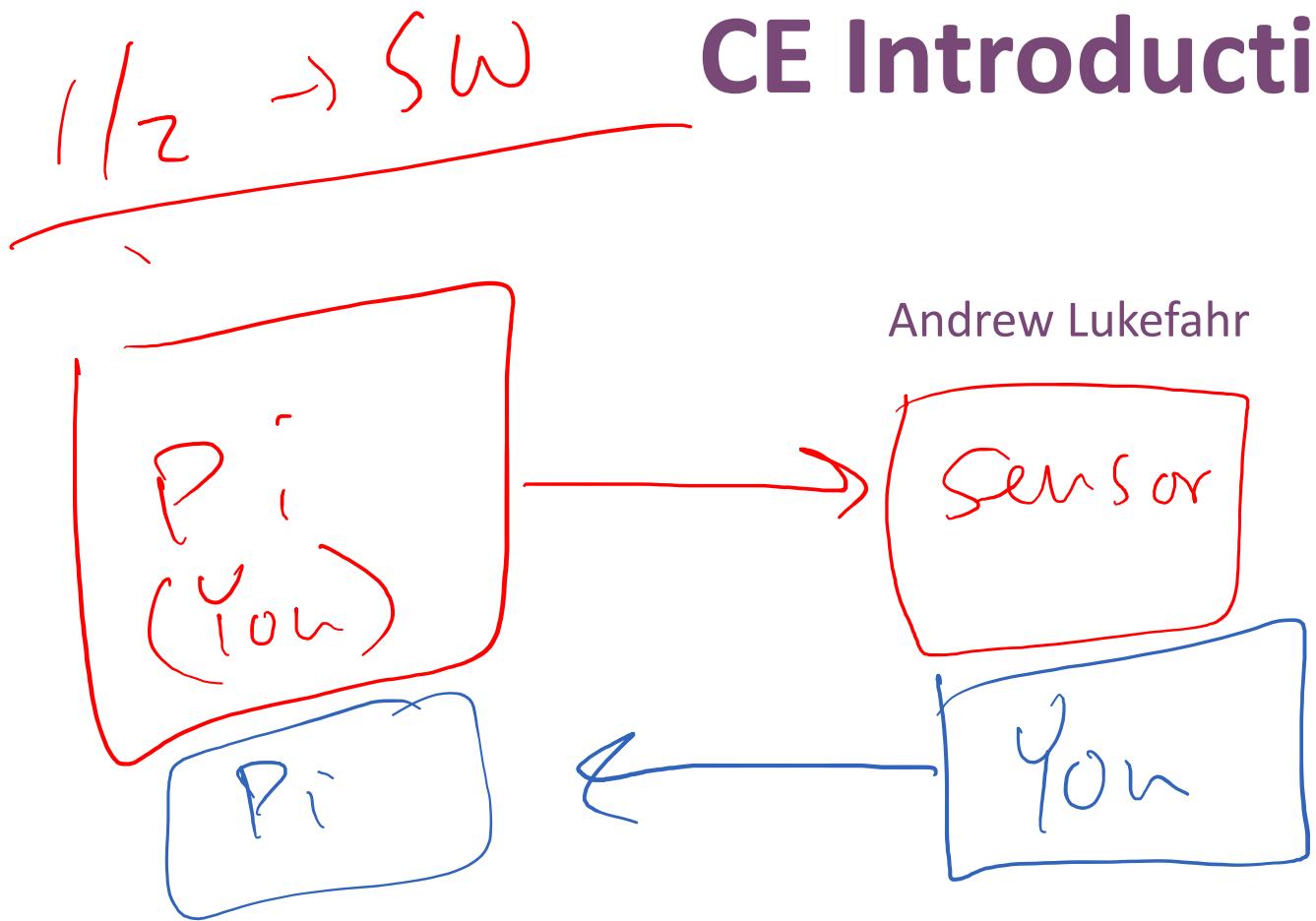


Test
Lay

CE Introduction



Introduction

- 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

Research work on security for FPGA-based systems.

official hours: right after class
② 4111

(post)

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.

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

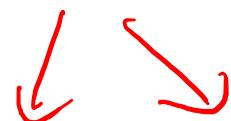
- What is 0x42 in binary? In decimal?
- What is -5 in 8-bit binary?
- Can you draw an AND gate? OR? NOT? NAND? NOR? XOR?
- Can you draw this circuit: $D = A \cdot B + C$?
- What is the truth table for this: $D = A \cdot B + C$?

(fix notes)

Review: Numbers

- What is 0x42 in binary? In decimal?

0x42 ∈ hex



bin → 0100 0010 ⇒

$$0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$
$$0 + 64 + 0 + 0 + 0 + 0 + 2 + 0$$

Review: 2's Compliment

- What is -5 in binary?

$$(-5) = -(+5)$$

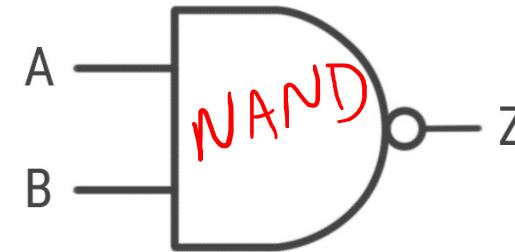
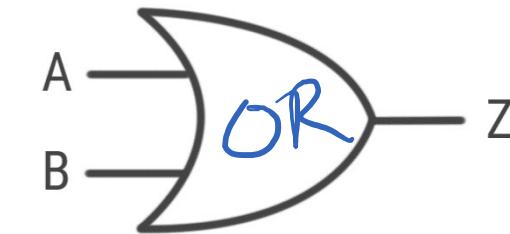
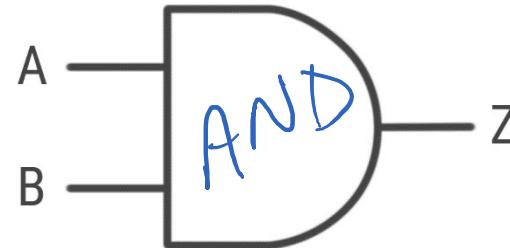
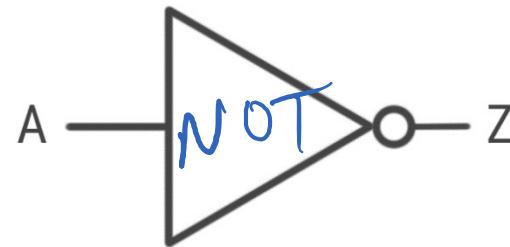
$$+5 = 0101$$

$$\sim(5) = 1010$$

$$\sim(5)+1 = \underline{\underline{1011}} \quad (4\text{ bit})$$

$$1111 \quad 1011 \quad (8\text{ bit})$$

Review: Logic Gates



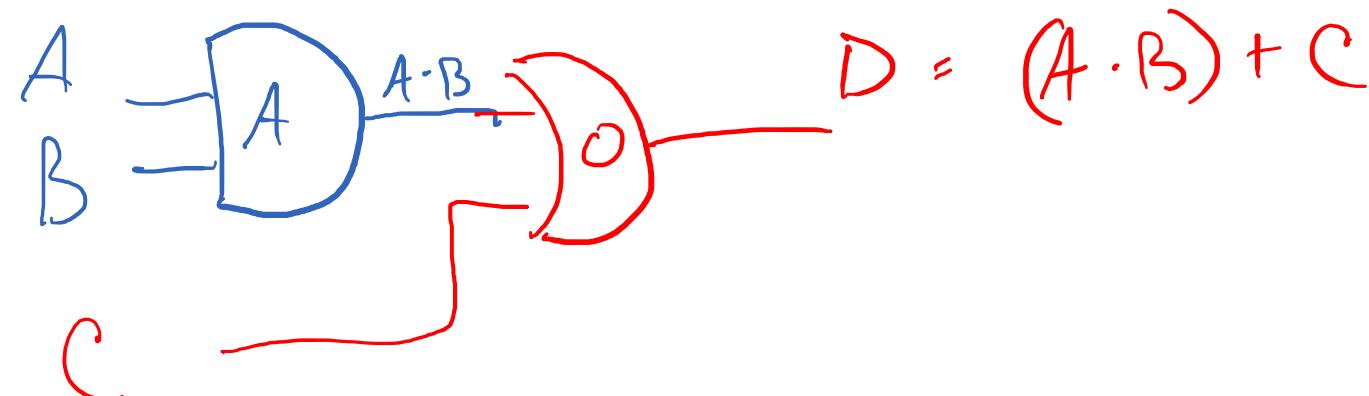
Review: Boolean Equation

- What circuit is this?

$$D = \underline{\underline{(A \cdot B)}} + C$$

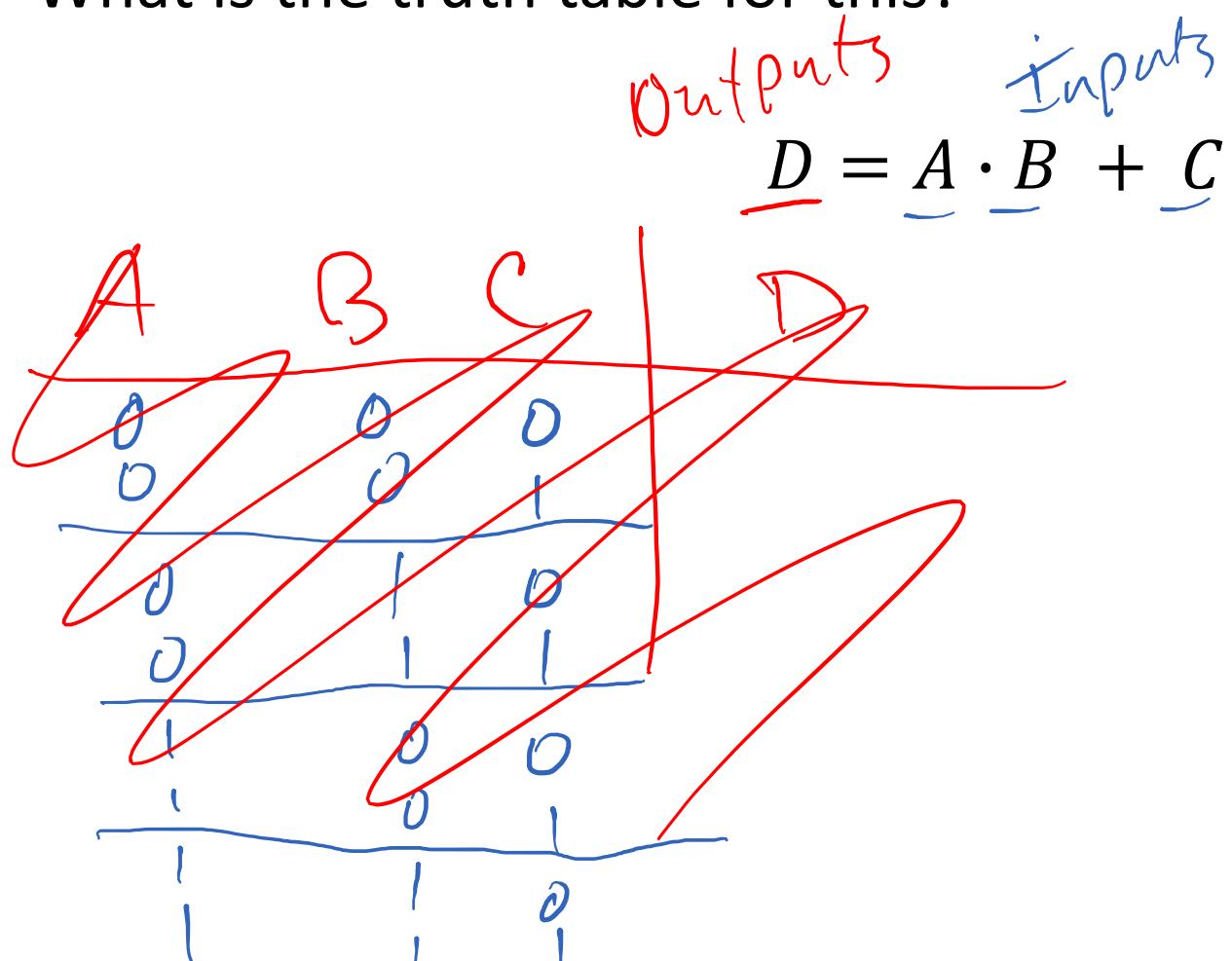
" D is A and B or C "

AND OR
↓ ↓



Review: Truth Table

- What is the truth table for this?



Truth Table

	A	B	C	D
(0)	0	0	0	0
(1)	0	0	1	1
(2)	0	1	0	0
(3)	0	1	1	1
⋮	⋮	⋮	⋮	⋮
(6)	1	0	0	0
(7)	1	0	1	1
⋮	⋮	⋮	⋮	⋮

NOT

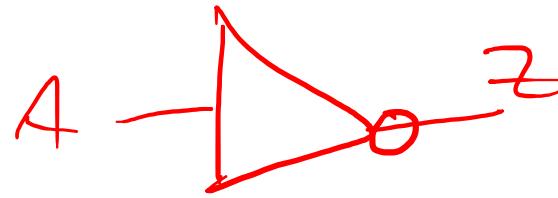
- Math:

- $Z = \bar{A}$ *Bar Above A*

- Code:

- $Z = \sim A$

- Schematic



A	z
0	1
1	0

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

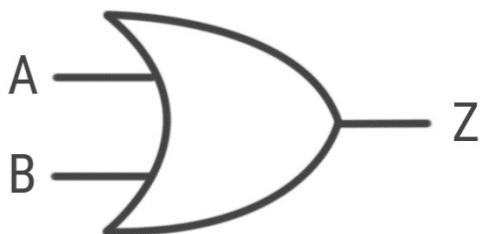
$$\begin{aligned} A &= 0100 \\ \sim A &\Rightarrow 1011 \\ !A &= 0 \end{aligned}$$

$!A$
logical Not

$\sim A$
bitwise Not

OR

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



A	B	Z

AND

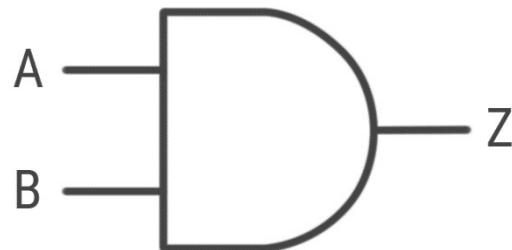
- Math:

- $Z = A \cdot B$

- Code:

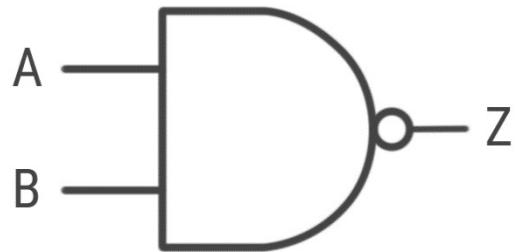
- $Z = A \& B$

- *Schematic*



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

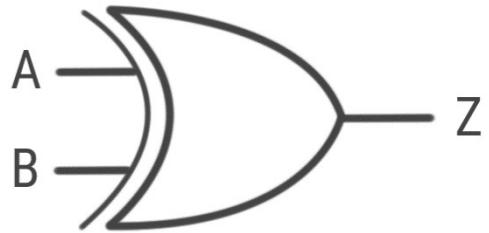
Other Gates: NAND



$$Z = \neg(A \wedge B)$$

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

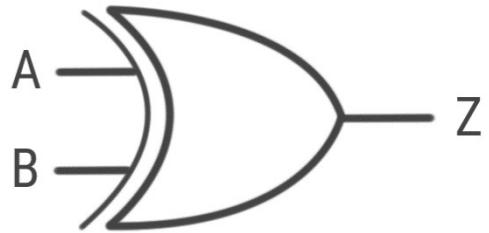
Other Gates: XOR



$$Z = A \Delta B$$

A	B	Z

Other Gates: XOR



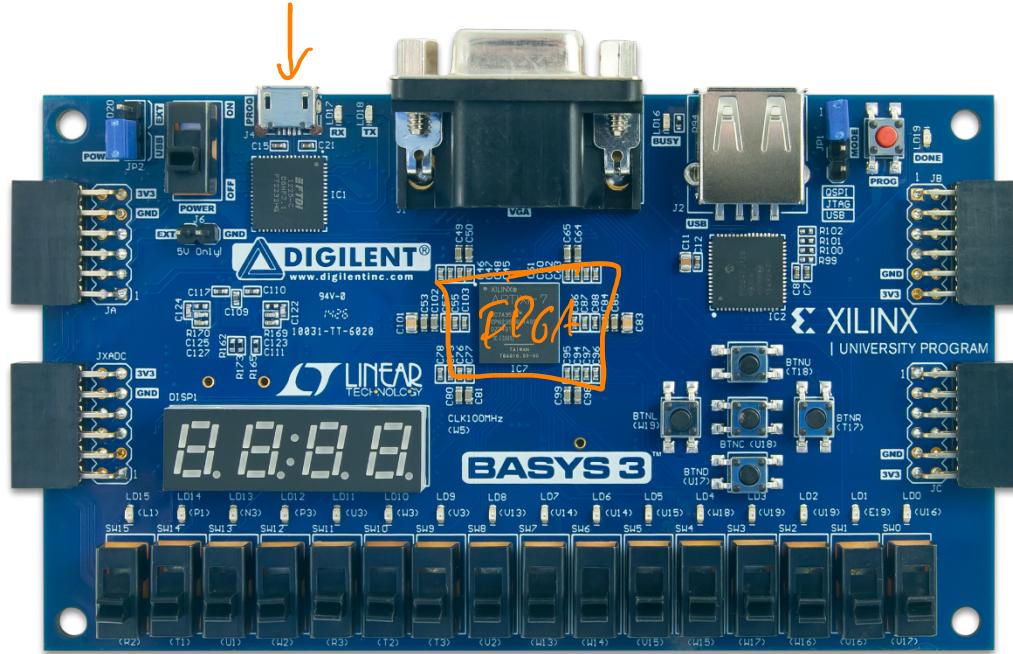
$$z = A \wedge B$$

↑
shift + 6

$$z = A \oplus B$$

A	B	Z

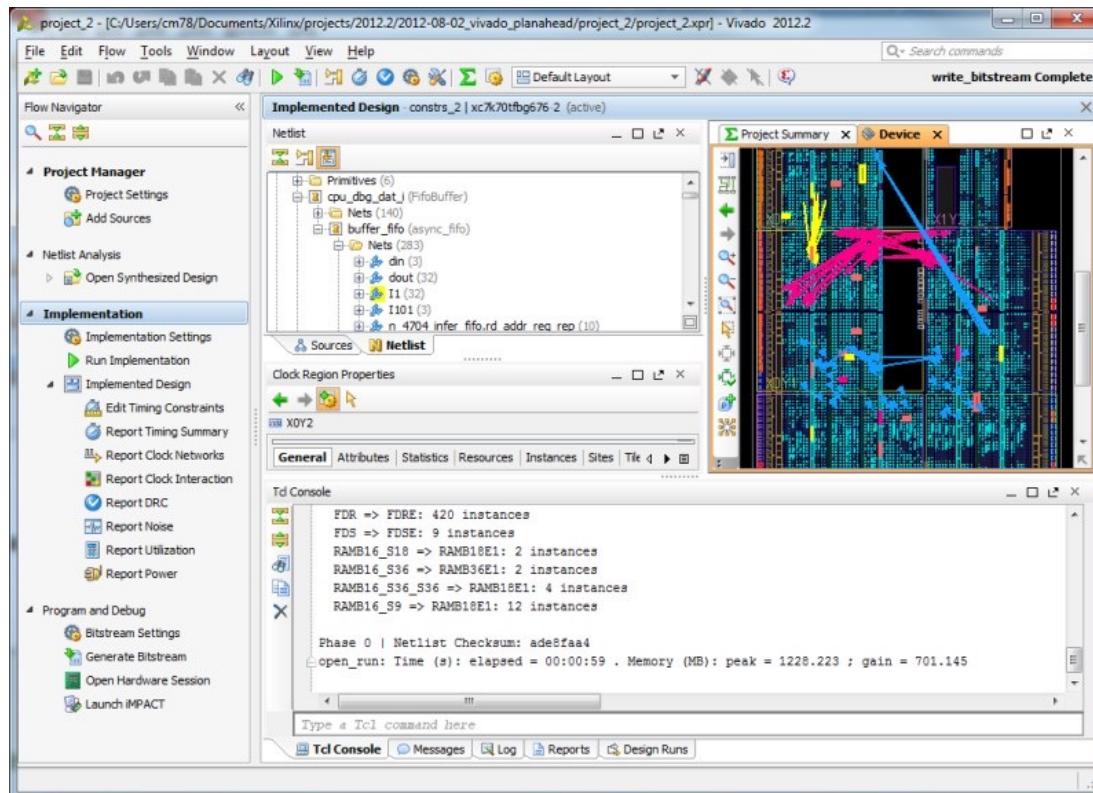
Basys3 Boards



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

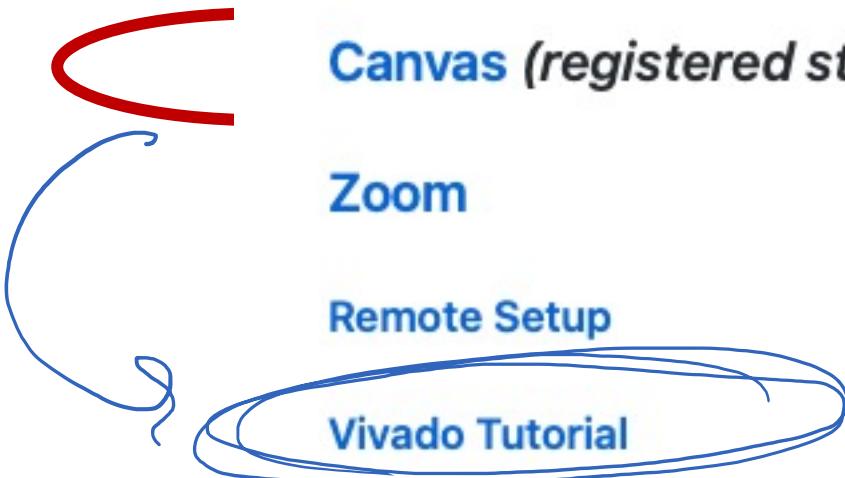
Xilinx Vivado Design Suite

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



Vivado Tutorial

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



Quick Links

[Syllabus](#)

[Downloads](#)

[Autograder \(registered students only\)](#)

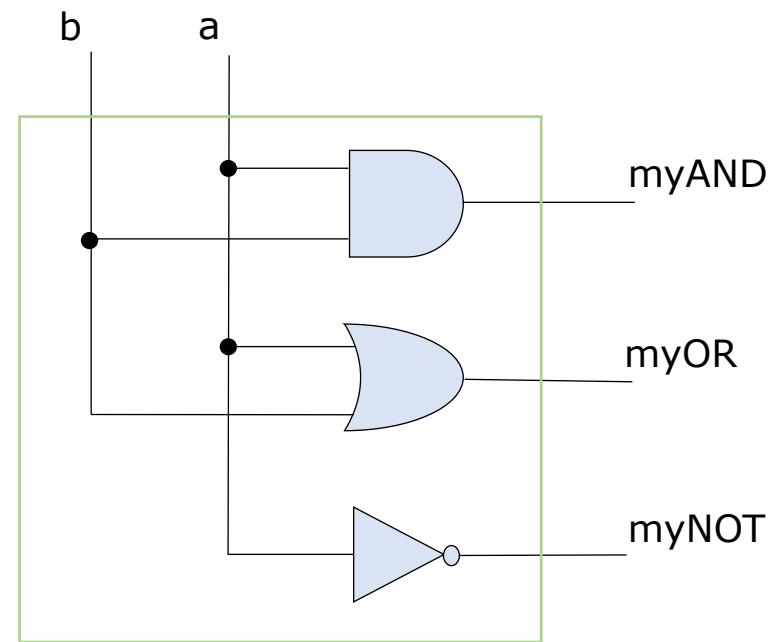
[Canvas \(registered students only\)](#)

[Zoom](#)

[Remote Setup](#)

[Vivado Tutorial](#)

Tutorial Schematic



Verilog Logic Operators

Verilog logic operators:

AND: &

OR: |

NOT: ~

Example:

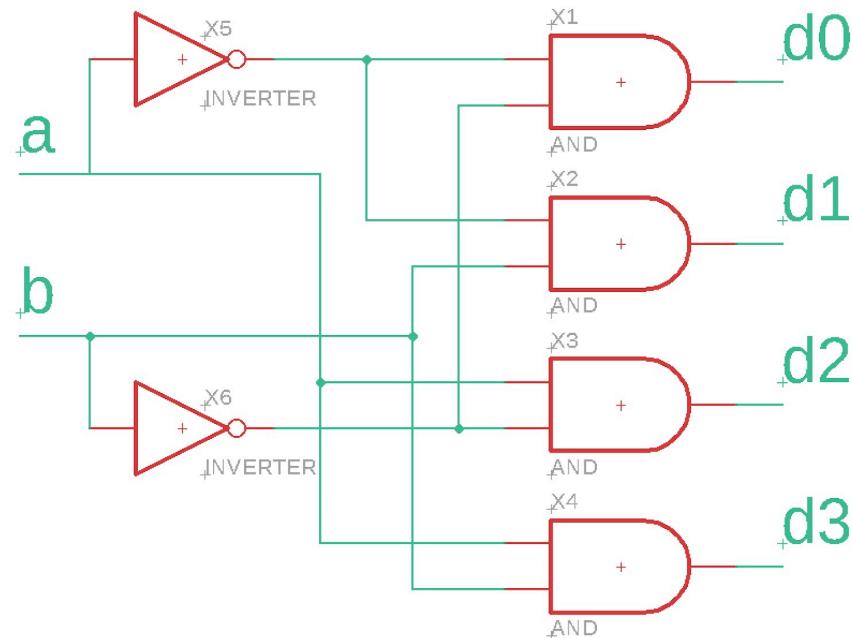
```
assign myAND = a & b;
```

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

Verilog Project 1: Demultiplexer

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



Deliverable #1: Autograder

~~NOT~~

lucy.indiana.edu

- Log on to the autograder:

<http://autograder.sice.indiana.edu>

- Log on with your '**username@iu.edu**' account

- Select '**Engr210.s22**' -> '~~PO~~' **P1**

- Upload: **top.sv**

- **top.v**: This is your top-level Verilog source file

- **top_tb.v**: This is your top-level Verilog **TestBench**

top-tb.sv

Deliverable #1: Autograder

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

Courses - B441/E315⚙️ - P1 ⚙️

Submit My Submissions Student Lookup

Group members:

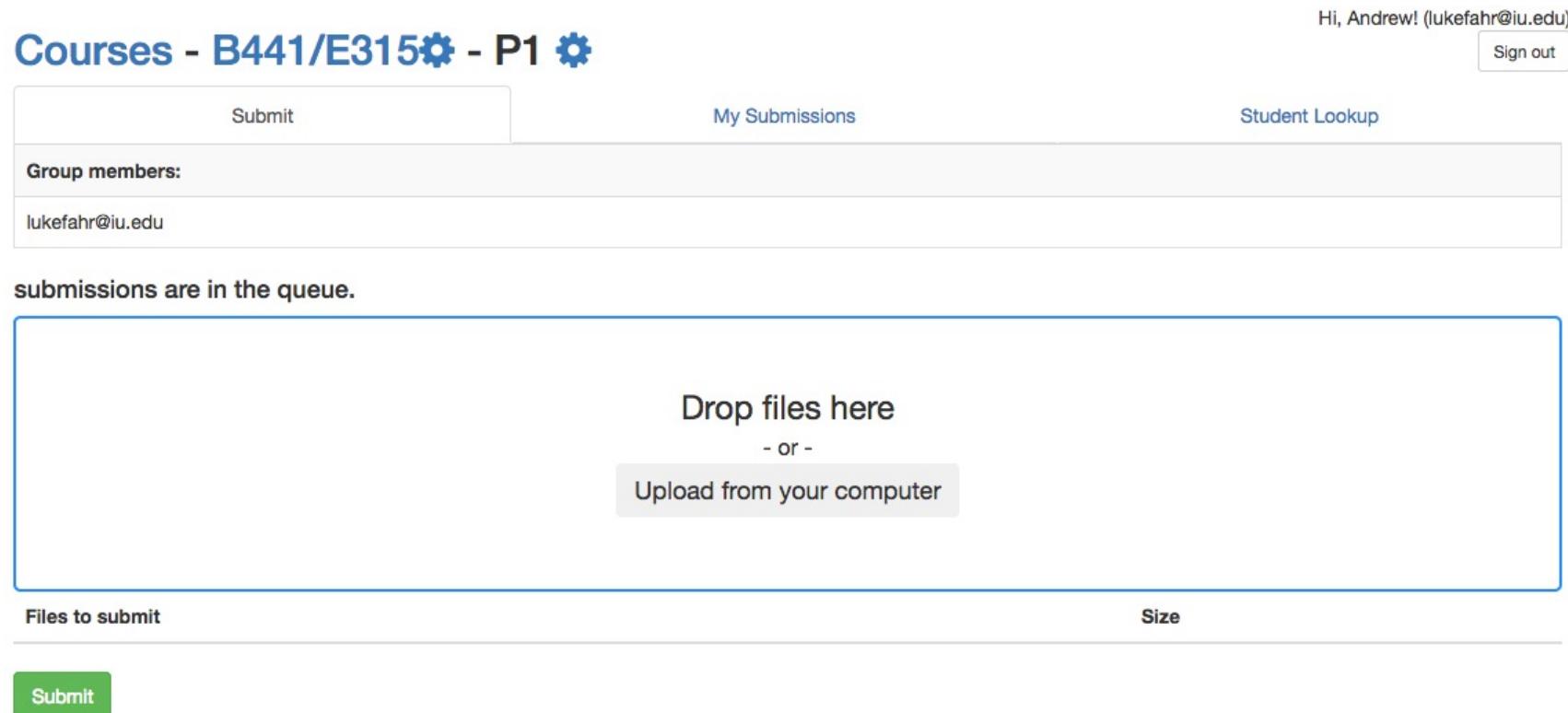
lukefahr@iu.edu

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: ✓ Stdout Stderr

Bugs exposed: 3 Stdout Stderr

- top_bug01
- top_bug02
- top_good

Student tests summary: Validity Check Stdout Validity Check Stderr

Get test names stdout Get test names stderr

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
---------	-------------	--------	--------

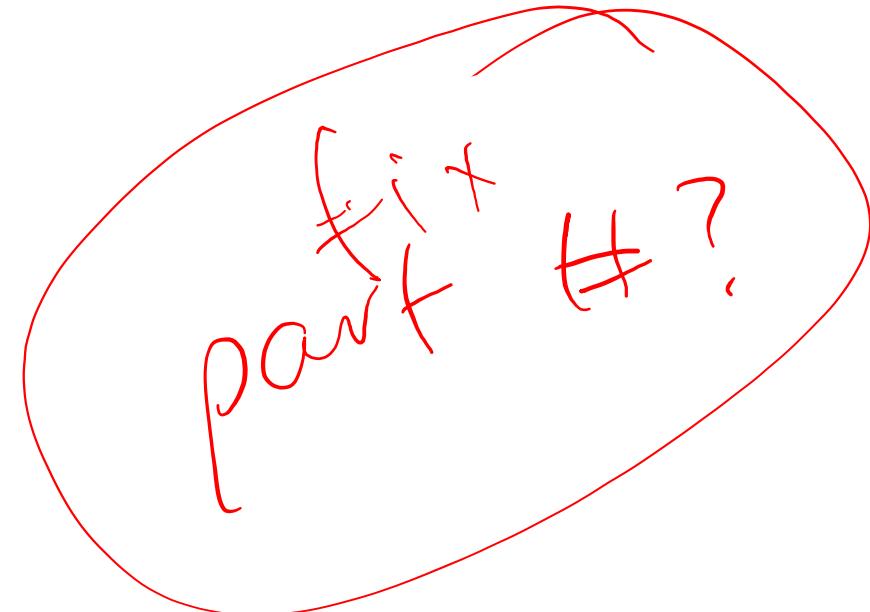
sim	✓ 0	✓ Output	Output
-----	---	--	--

Deliverable #2: FPGA Demo

- Use Vivado to:
 - Synthesize your design
 - Program the FPGA
- Verify to yourself it works. (no points ☹)
- Do a **demo for a TA**. (the points ☺)

Why 2 Deliverables?

- Encourage testing
 - We give you points for good testbenches
- Check correctness
 - Automatically checks for bugs
- Reduce your debug time
 - Synthesis is slow. **Don't** until you are 100% sure your code works.



Next Time

- Truth Tables