

ENGR 210 / CSCI B441

Introduction

Andrew Lukefahr

Course Website

fangs-bootcamp.github.io

Write that down!

Introduction

- Topics covered:
 - Boolean algebra and logic gates
 - Sequential Logic
 - State Machines
 - Serial Communication
 - Buses
 - Protocols

Prof. Lukefahr



Andrew Lukefahr, Assistant Professor

Email: lukefahr@iu.edu

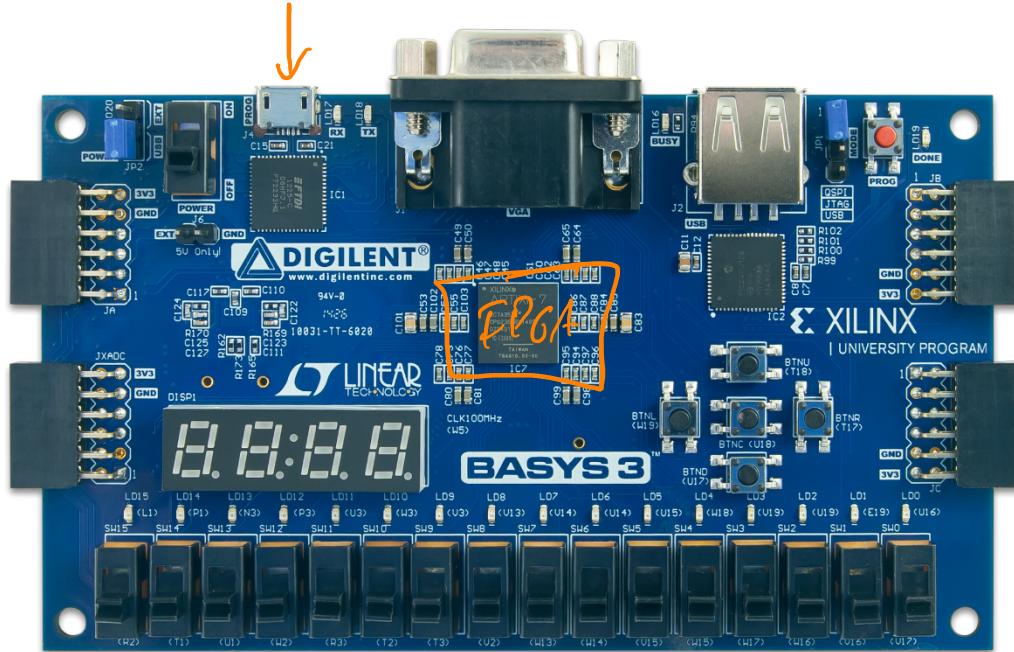
Navy Email: andrew.lukefahr@navy.mil

Research work on security for FPGA-based systems.

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
- More details later in a few minutes

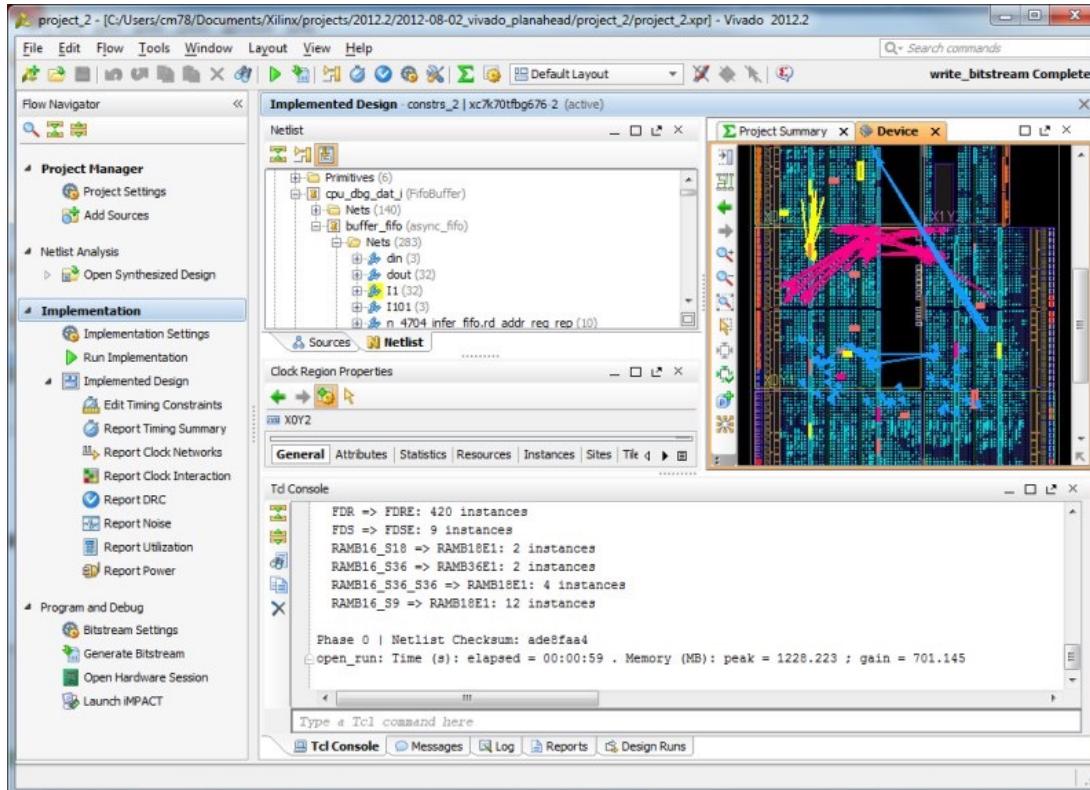
Basys3 Boards



- Checkout board for semester
- 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



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

A diagram showing the binary-to-decimal conversion of 0100 0010. The binary number is shown as $0100\ 0010$ with powers of 2 above it: $2^7, 2^6, 2^5, 2^4, 2^3, 2^2 + 2^1 + 2^0$. Below the binary number, the equation $0+64+0+0+0+2+0 =$ is written. To the right, a red circle contains the decimal value "66" with handwritten "decimal" above it and a checkmark below it. A large orange circle surrounds the entire calculation.

$$0100\ 0010$$
$$\begin{array}{r} 2^7 2^6 2^5 2^4 2^3 2^2 + 2^1 + 2^0 \\ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \\ \hline 0+64+0+0+0+2+0 = \end{array}$$

66 decimal ✓

Review: 2's Complement

- What is -5 in binary?

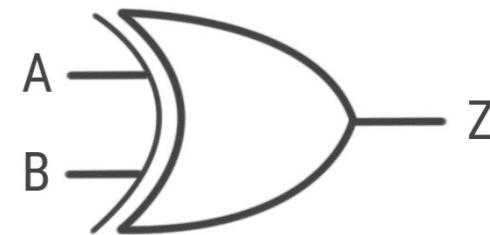
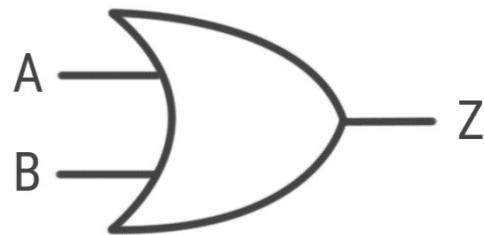
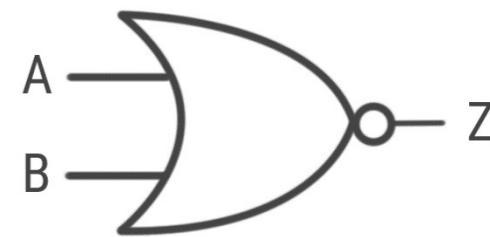
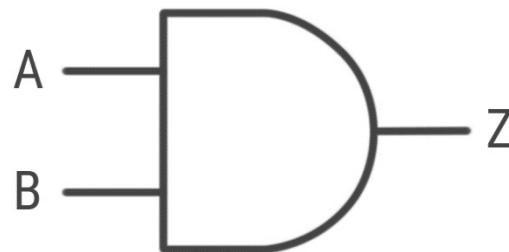
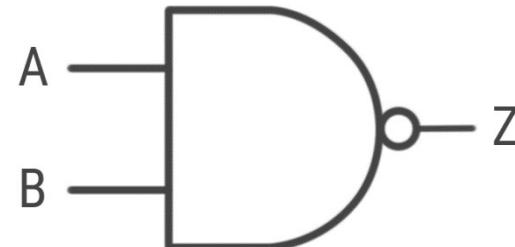
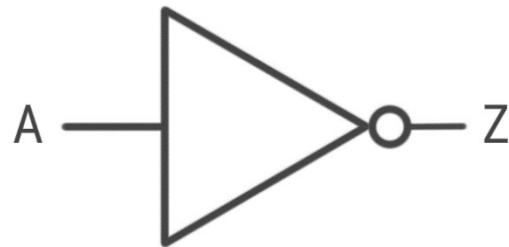
$$+5 = 1001$$

$$\begin{array}{r} 0110 \\ + \quad 1 \\ \hline (-5) \quad 10111 \end{array}$$

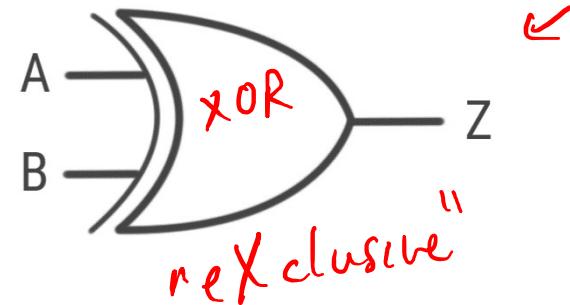
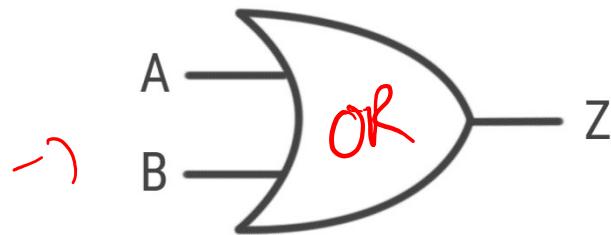
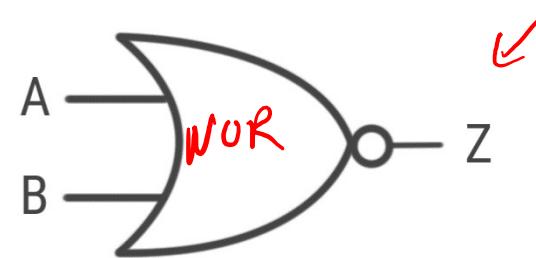
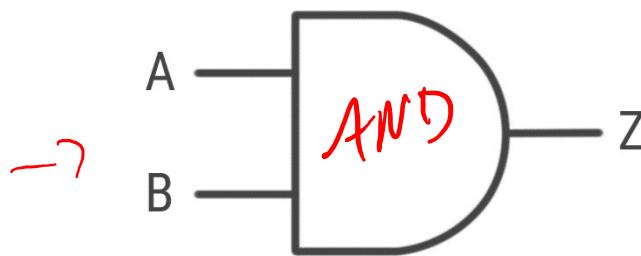
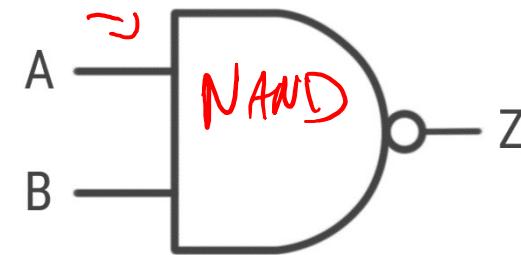
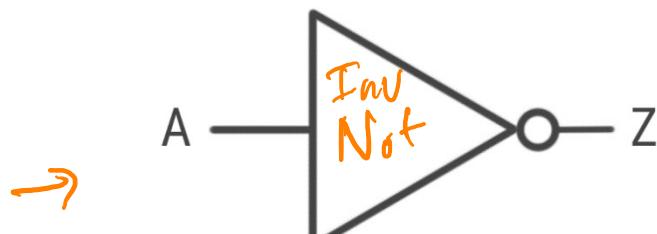
Signed Values

→ 1010 negative
0010 → positive

Review: Logic Gates



Review: Logic Gates



Review: Boolean Equation

- What circuit is this?

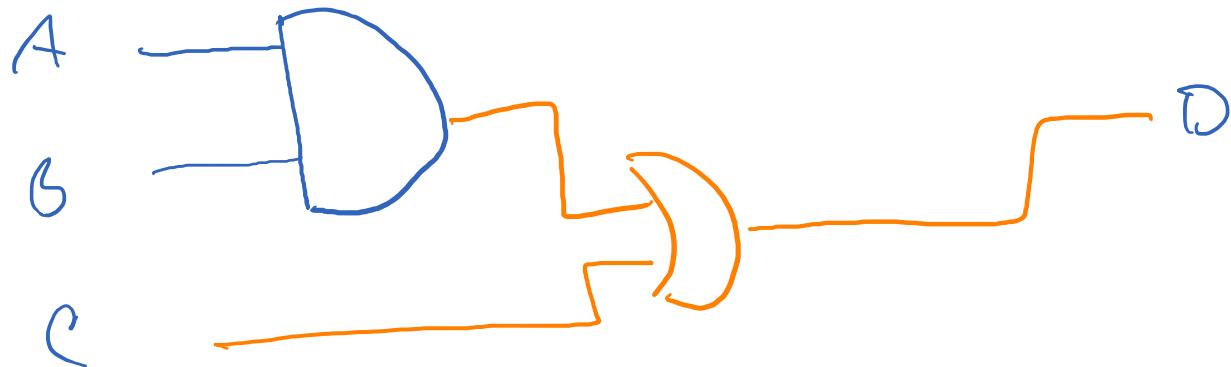
$$D = (A \cdot B) + C$$

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?

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

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

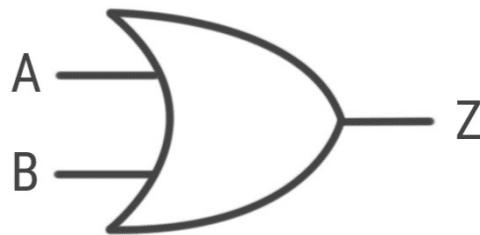
$$S = 0101$$

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

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

OR

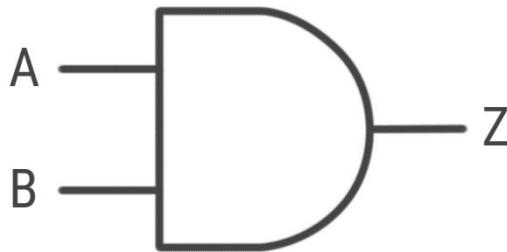
- Math:
 - $Z = A + B$
- Code:
 - $Z = A \mid B$
- Schematic



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

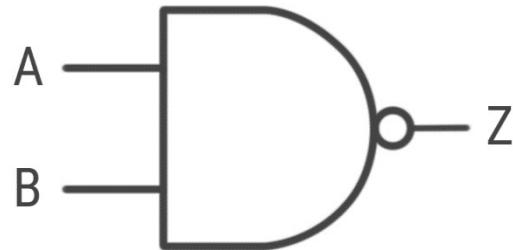
AND

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



A	B	Z

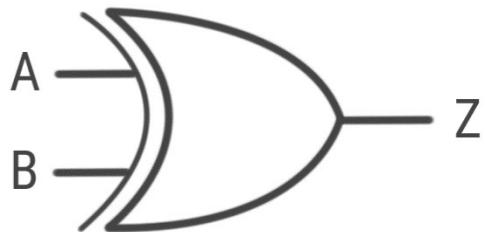
Other Gates: NAND



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

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

Other Gates: XOR



$$z = A \wedge B$$

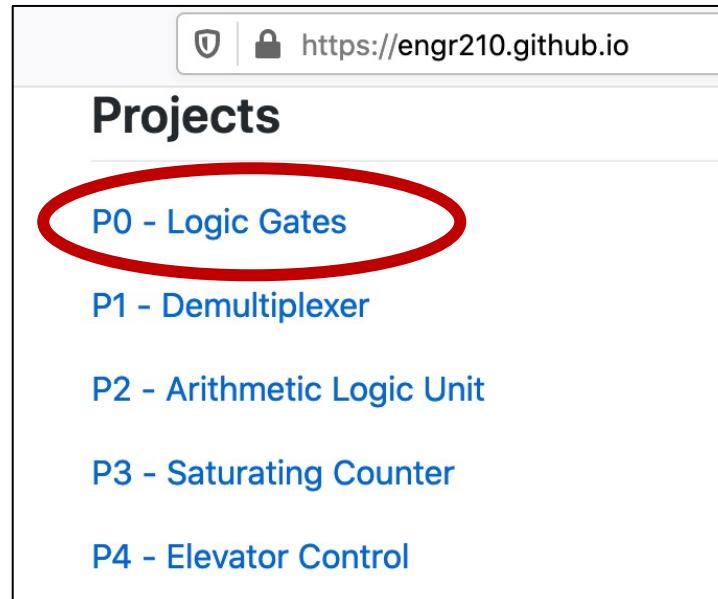
↑
shift + 6

$$z = A \oplus B$$

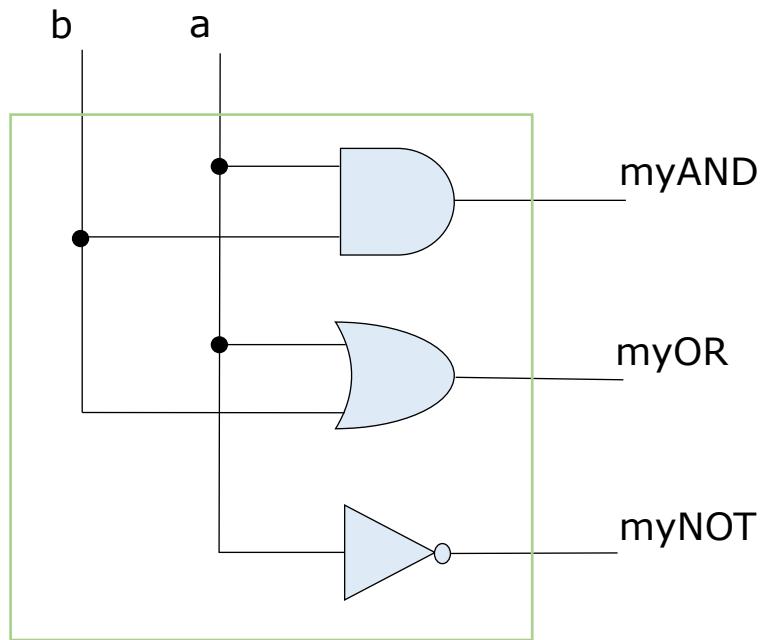
A	B	Z

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

Deliverable #1: Verilog Code

- Send completed / tested code to Chris Sozio
 - Christopher.sozio@navy.mil
- Chris will grade and offer feedback.

Deliverable #2: FPGA Demo

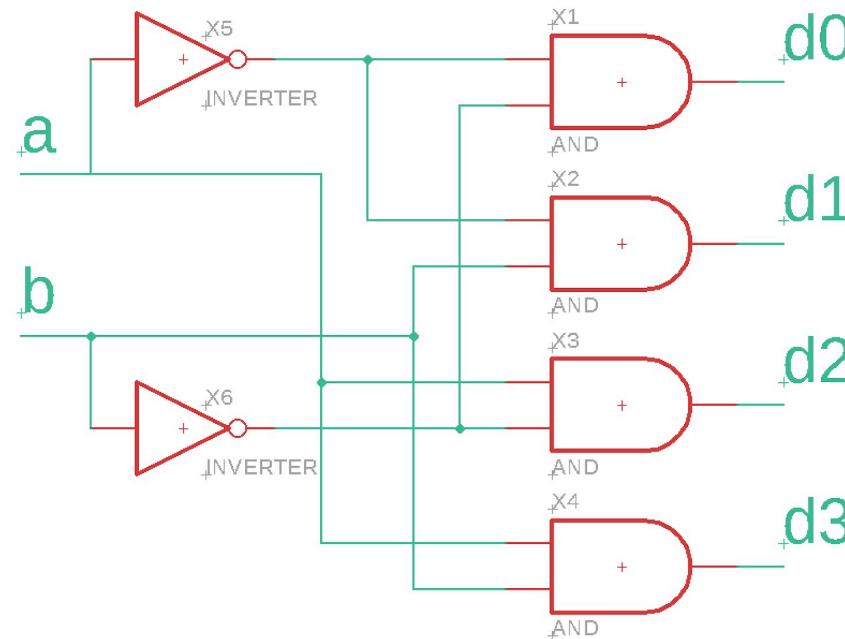
- Use Vivado to:
 - Synthesize your design
 - Program the FPGA
- Verify to yourself it works.
- Do a **demo for Chris.**

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.

Project 2: Demultiplexer

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



Next Time

- Truth Tables