ENGR-E 210 ENGINEERING CYBER-PHYSICAL SYSTEMS

01/14/2020

Introduction

- This is a 3-credit course provides an introduction to organization and logic design of digital systems.
- Topics covered:
- Boolean algebra and logic gates
- Hardware building blocks
- Architecture and control
- Implementation

Course Website

engr210.github.io

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: $D = A \cdot B + C$?

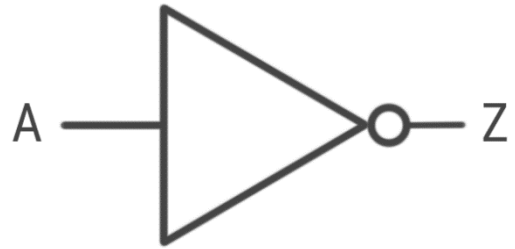
Review: Numbers

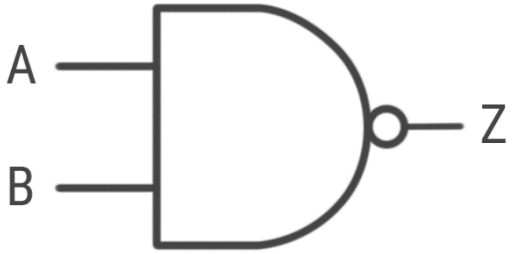
- What is 0x42 in binary? In decimal?

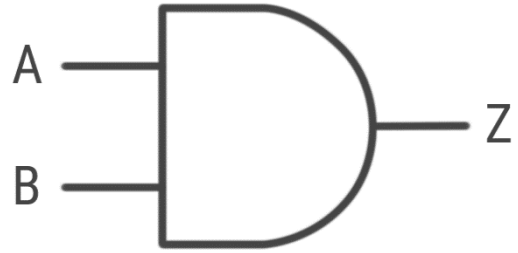
Review: 2's Compliment

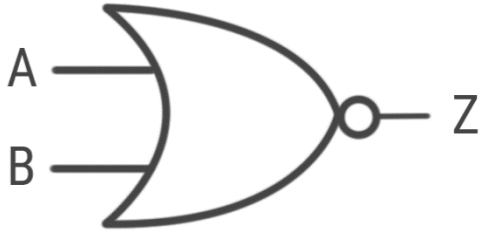
- What is -5 in binary?

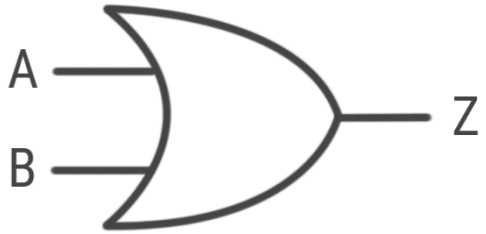
Review: Logic Gates

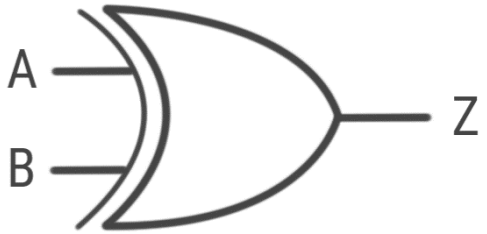












Review: Boolean Equation

- What circuit is this?

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

Review: Truth Table

- What is the truth table for this?

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

Lab Sessions

- In laboratory will use:
- Xilinx Vivado
- Digilent Basys 3 FPGA board

Projects

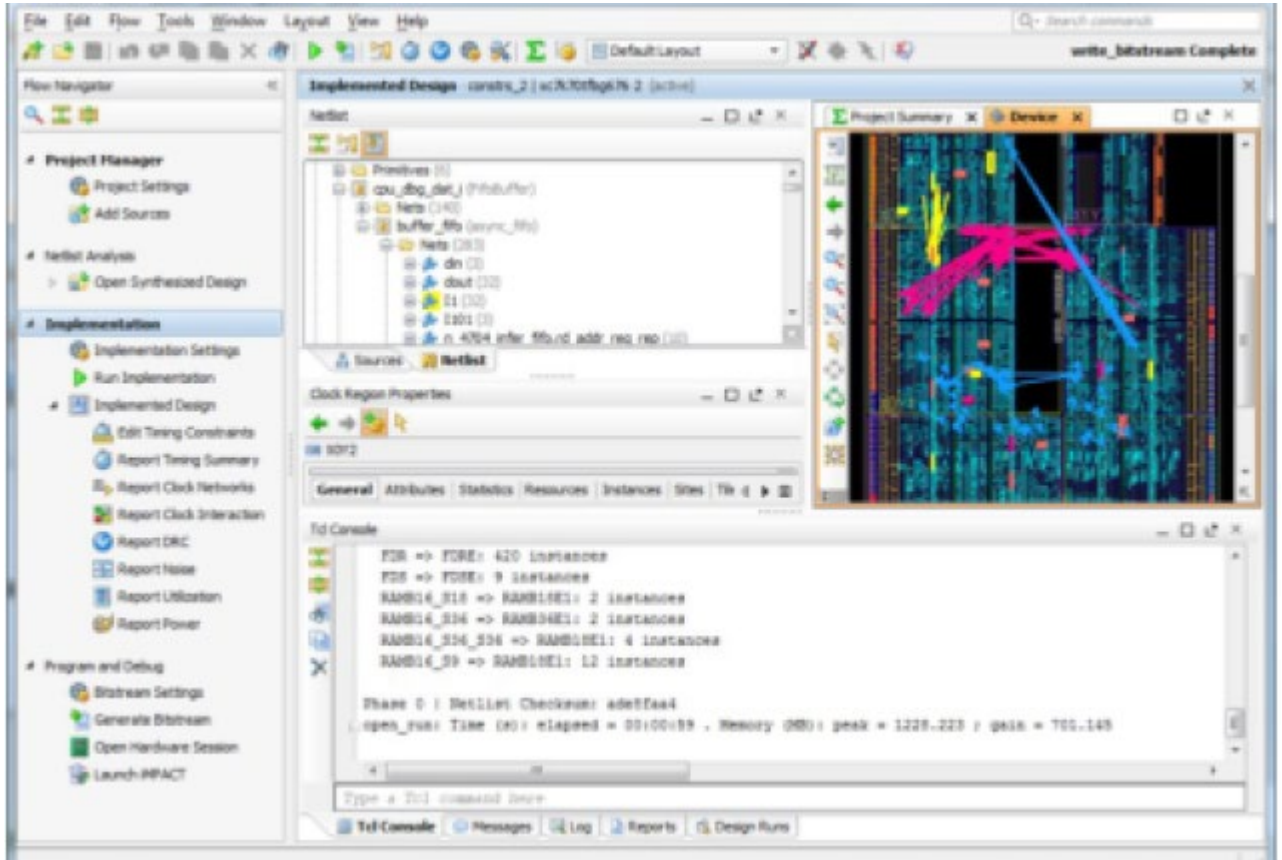
- Assigned every week and due the next week.
- Some projects will be completed over multiple weeks.
- Submit two parts:
- Submit Verilog code to the autograder
- Demonstrate functioning FPGA implantation to TA

Basys3 Boards

- Checkout board for semester on Friday
- Programmed with Xilinx Vivado

Xilinx Vivado Design Suite

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



Project 0: Blink an LED

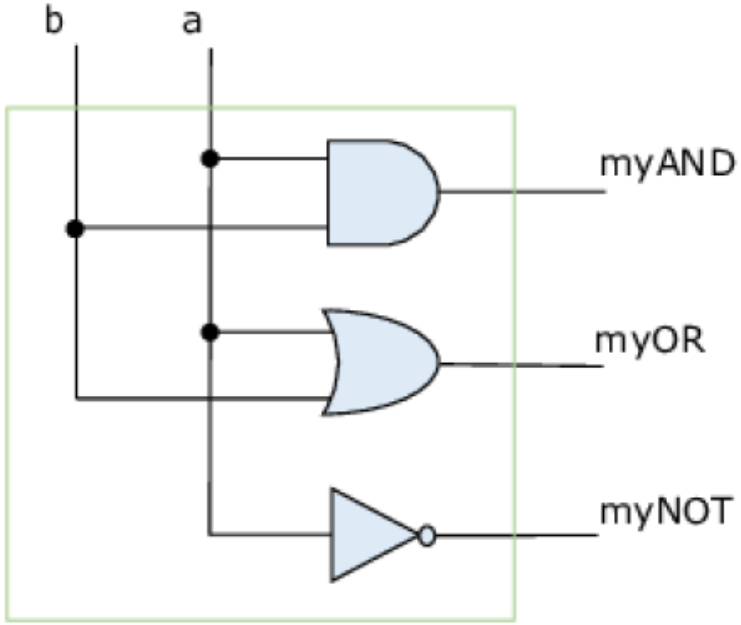
- This is a ‘demo lab’
- It lets you blink an LED with a switch
- It is to provide you a reference design
- It is not Due!

Project 01: Logic Gates

- Task:
- Design and implement a simple digital system on an FPGA circuit. This involves the following steps: writing the Verilog source code, using Vivado to synthesize the design, and implanting your design on an FPGA.
- In this lab project you are asked to design and implement a digital system with two inputs (a,b) and three outputs (myAND, myNOT,myOR).
- Deliverables:
- Submit your Verilog code to the autograder
- Demonstrate the FPGA implantation of your design.

Project 01: Schematic

- You can modify the program of Lab 00:
- Change the input and the output signals,
- Add more Verilog code to define the outputs.



Project 01: 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:

Signal	Basys3 input	Signal	Basys3 output
a	sw0	myAND	led0
b	sw1	myNOT	led1
		myOR	led2

Deliverable #1: Autograder

- Log on to the autograder:

  http://autograder.sice.indiana.edu

- Log on with your ‘username@iu.edu’ account

- Select ‘Engr210.s20’ -> ‘P1’

- Upload:
  - top.v : This is your top-level Verilog source file
  - top_tb.v: This is your top-level Verilog **T**est**B**ench

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

- Details of Verilog
- Synthesis process