Digital Systems Lab

Lab 02 Experiment 1 – Part II & III

Fall 2023

Instructor: Ashkan Farhangi, Amirhossein Safari

Outline

- Lab Report
- Vivado Procedure (A-Z)
- Small Quiz (Bonus activity)

Lab performance

- You lab performance will be evaluated through the following:
  - 5 points pre-lab assignment (In-class activity)
    - Should be included to the lab report
  - 5 points lab performance
  - 15 points lab report

  Total of 25 points per experiment

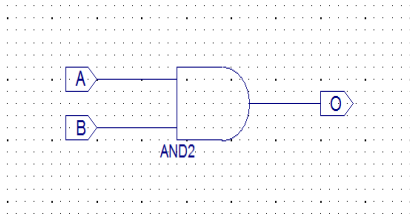
Lab Report

- Cover Page with Subject Name, Student Name, PID, Due Date

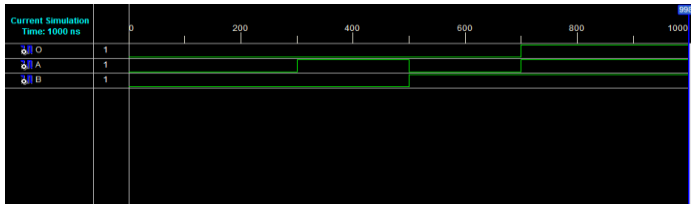
Item	Points
Pre-Laboratory Assignment	5.00
Design Completion	5.00
Heading	0.25
Objective	0.25
Block Diagram	0.25
Apparatus List	0.25
Procedure and/or Design Methodology	3.00
Design Specification Plan	2.00
Test Plan	2.00
Results Statement and Logic Schematic Diagram	4.00
Conclusion	3.00
TOTAL:	25.00

Lab Report

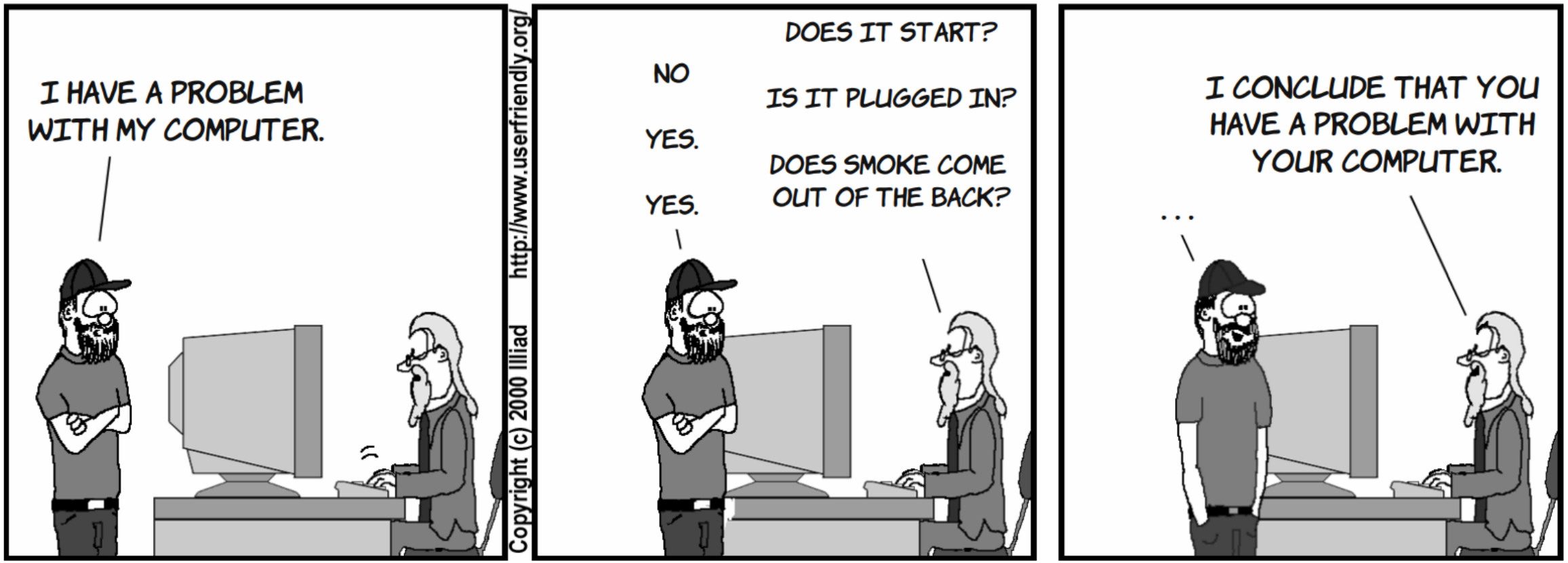
- Please look at the sample lab report
- What is needed on the report
  - Verilog program of Design
  - Verilog program of Simulation
  - Schematic diagram
  - Truth table
  - Timing diagram



SW0	SW1	LED0
Off	Off	Off
Off	On	Off
On	Off	Off
On	On	On



How to deal with Error in Vivado

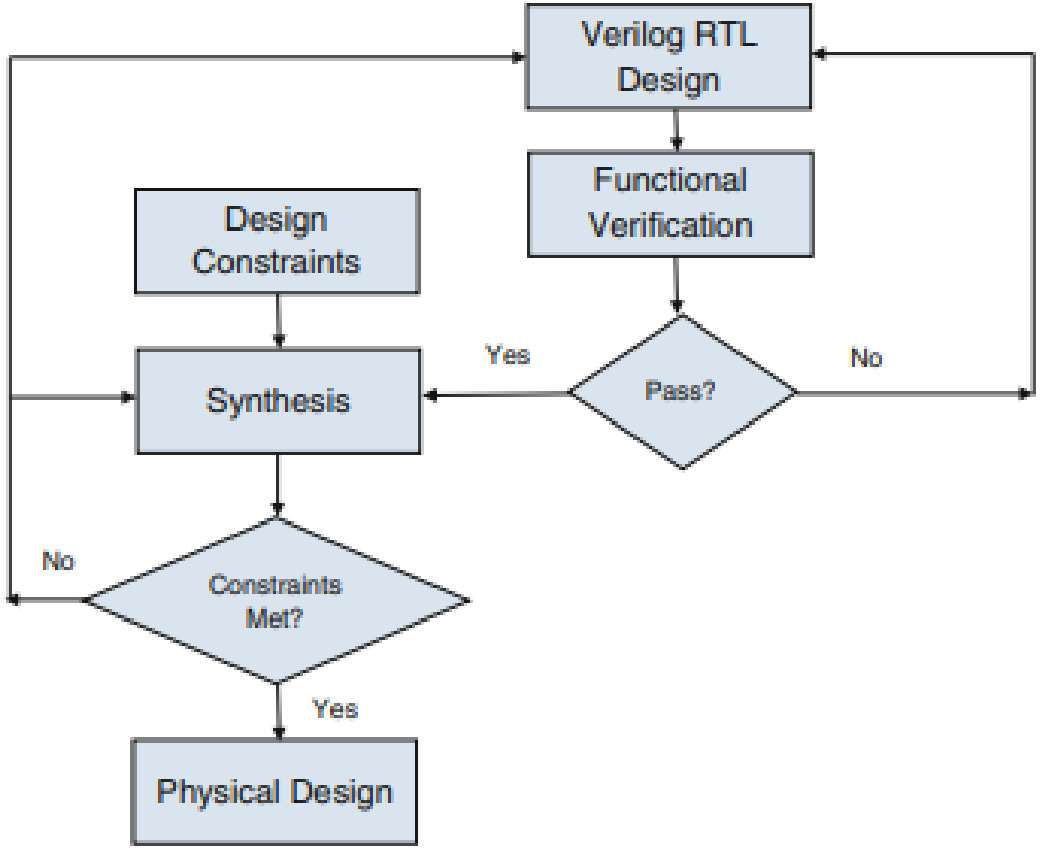

I HAVE A PROBLEM WITH MY COMPUTER.
Copyright (c) 2000 Illiad http://www.userfriendly.org/
DOES IT START?
NO
IS IT PLUGGED IN?
YES.
DOES SMOKE COME OUT OF THE BACK?
YES.
I CONCLUDE THAT YOU HAVE A PROBLEM WITH YOUR COMPUTER.
. . .

Steps to handle Error

- Increase our knowledge of the reason behind why we take certain actions in Vivado
  - Prevention of them appearing again and again
- Look for messages, hints that program tells us
  - Learn to debug your program

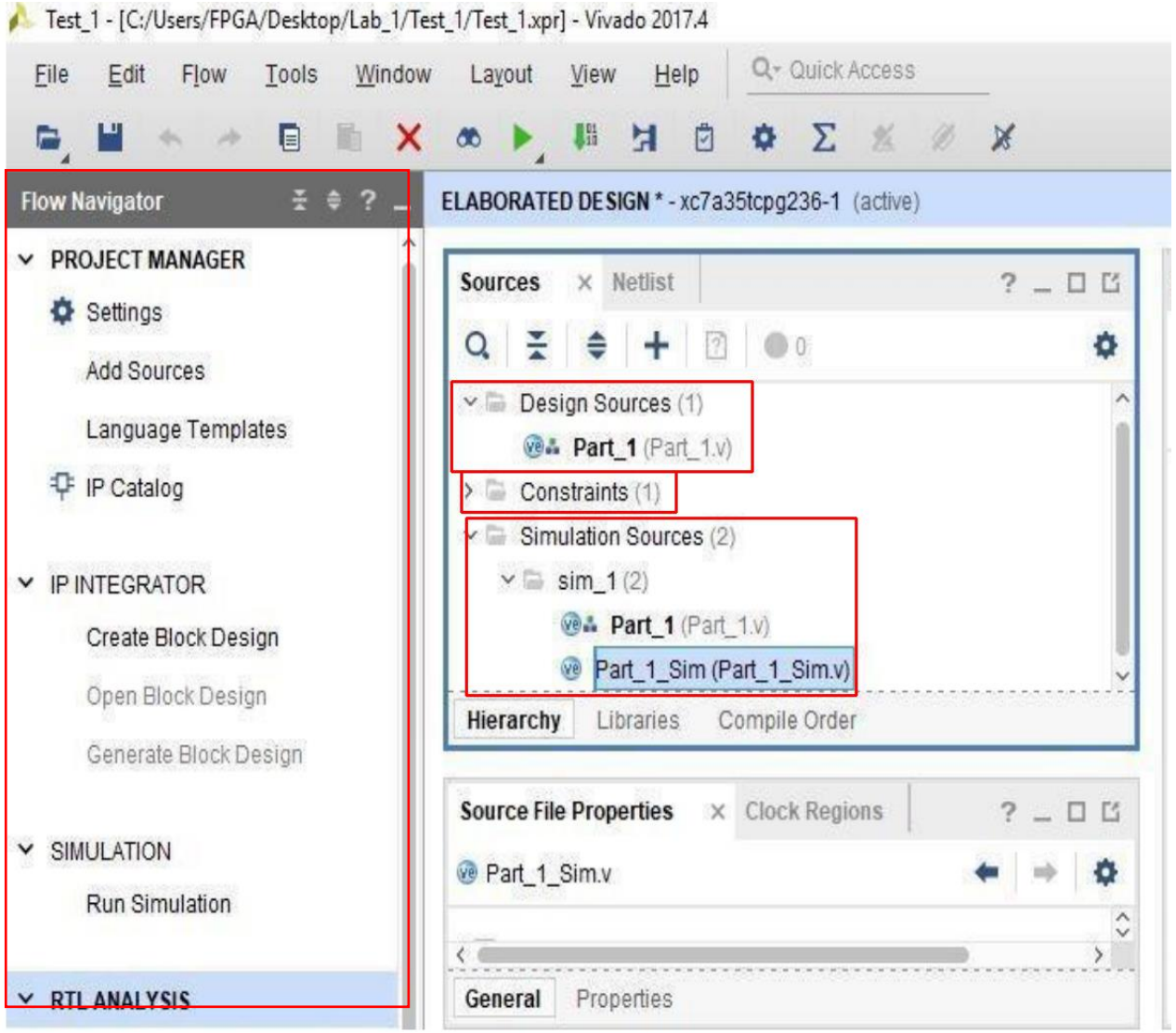
Vivado Procedure

- **R**egister **T**ransfer **L**evel (RTL) design
  - RTL uses registers connected by Boolean equations.
- “Synthesis”
  - Gate-level representation
- “Implementation”
  - Applying constraints of design
    - Logical
    - Physical
    - Timing constraints
- “Generate Bit-Code”
  - Transfer your code into the board
  - This part is optional



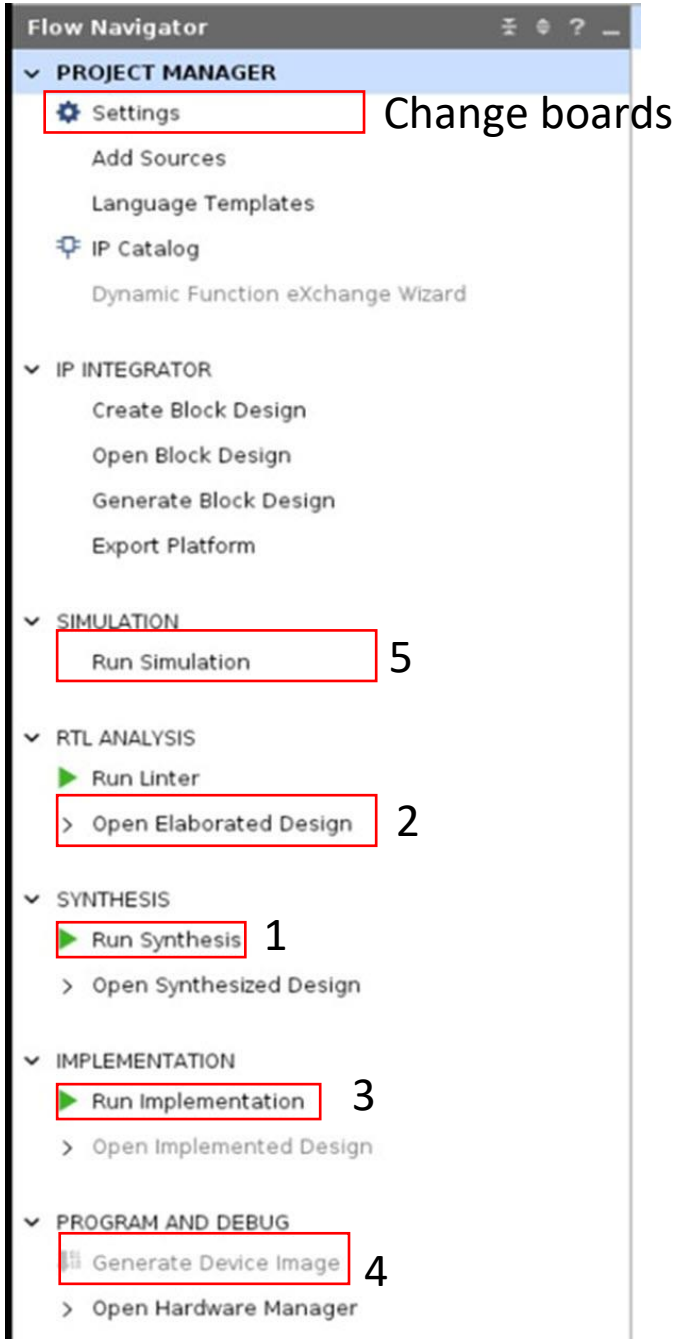
Overview of Vivado

- Flow Navigator
- Sources
  - Design Code
    - Part_1.v
    - ANDGate.v
  - Constraints
    - I/O ports
  - Simulation code
    - Part_1_Sim.v
    - ANDGate_Sim.v



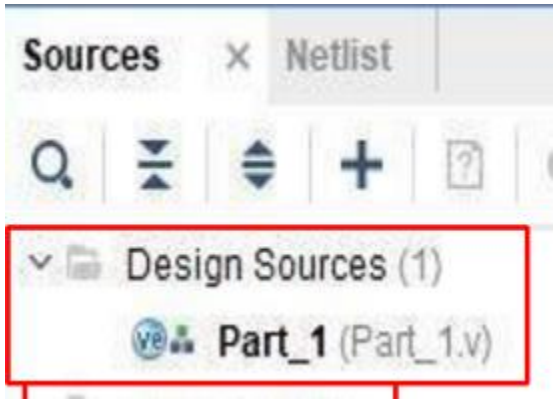
Flow Navigator (S-RIG-S)

- "**S**ynthesis"
  - Gate-level representation
- **"R**TL ANALYSIS"
  - Schematics
- "**I**mplementation"
  - Applying constraints of design
    - Logical
    - Physical
    - timing constraints
- "**G**enerate Bit-Code"
  - Transfer your code into the board
- "**S**imulation"
  - Simulate all possible combination of inputs



Design code

- Design Code




```
`timescale 1ns / 1ps

module Part_1(
input wire Inp_1,
input wire Inp_2,
output wire Outp
);

assign Outp = Inp_1 & Inp_2;
endmodule
```

Remember these names.
You call them later in Simulation_code

Board

- Page 94 of Lab Manual
  - Appendix D
- Have this page “open” when lab starts

APPENDIX D
BASYS 3 BOARD

	PIN #	I/O Std
Switches		
SW0	V17	LVCMOS33
SW1	V16	LVCMOS33
SW2	W16	LVCMOS33
SW3	W17	LVCMOS33
SW4	W15	LVCMOS33
SW5	V15	LVCMOS33
SW6	W14	LVCMOS33
SW7	W13	LVCMOS33
SW8	V2	LVCMOS33
SW9	T3	LVCMOS33
SW10	T2	LVCMOS33
SW11	R3	LVCMOS33
SW12	W2	LVCMOS33
SW13	U1	LVCMOS33
SW14	T1	LVCMOS33
SW15	R2	LVCMOS33
LEDs		
LD0	U16	LVCMOS33
LD1	E19	LVCMOS33
LD2	U19	LVCMOS33
LD3	V19	LVCMOS33
LD4	W18	LVCMOS33
LD5	U15	LVCMOS33
LD6	U14	LVCMOS33
LD7	V14	LVCMOS33
LD8	V13	LVCMOS33
LD9	V3	LVCMOS33
LD10	W3	LVCMOS33
LD11	U3	LVCMOS33
LD12	P3	LVCMOS33
LD13	N3	LVCMOS33
LD14	P1	LVCMOS33
LD15	L1	LVCMOS33
Push buttons		
BTNU	T18	LVCMOS33
BTND	U17	LVCMOS33
BTNR	T17	LVCMOS33
BTNL	W19	LVCMOS33
BTNC	U18	LVCMOS33

Constraints

- Package Pin
  - Chooses the input/output of your board

- I/O Standard Matters
  - Your board works on 3.3V
    - You must choose LVCMOS33 as I/O Std

APPENDIX D
BASYS 3 BOARD

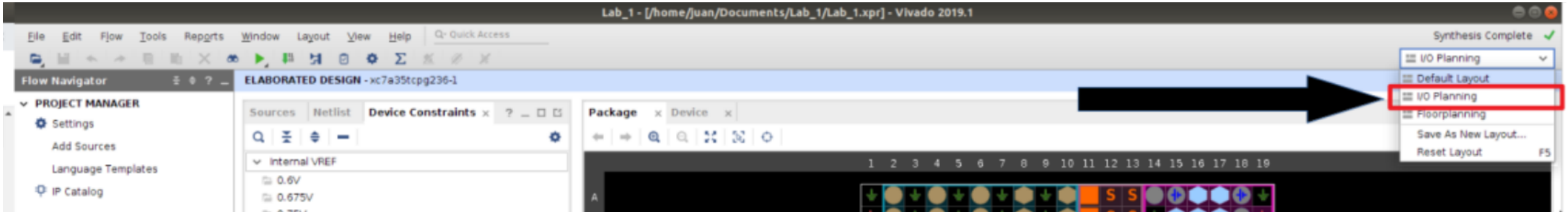
	PIN #	I/O Std
Switches		
SW0	V17	LVCMOS33
SW1	V16	LVCMOS33
SW2	W16	LVCMOS33
SW3	W17	LVCMOS33
SW4	W15	LVCMOS33
SW5	V15	LVCMOS33
SW6	W14	LVCMOS33
SW7	W13	LVCMOS33
SW8	V2	LVCMOS33
SW9	T3	LVCMOS33
SW10	T2	LVCMOS33
SW11	R3	LVCMOS33
SW12	W2	LVCMOS33
SW13	U1	LVCMOS33
SW14	T1	LVCMOS33
SW15	R2	LVCMOS33
LEDs		
LD0	U16	LVCMOS33
LD1	E19	LVCMOS33
LD2	U19	LVCMOS33
LD3	V19	LVCMOS33
LD4	W18	LVCMOS33
LD5	U15	LVCMOS33
LD6	U14	LVCMOS33
LD7	V14	LVCMOS33
LD8	V13	LVCMOS33
LD9	V3	LVCMOS33
LD10	W3	LVCMOS33
LD11	U3	LVCMOS33
LD12	P3	LVCMOS33
LD13	N3	LVCMOS33
LD14	P1	LVCMOS33
LD15	L1	LVCMOS33
Push buttons		
BTNU	T18	LVCMOS33
BTND	U17	LVCMOS33
BTNR	T17	LVCMOS33
BTNL	W19	LVCMOS33
BTNC	U18	LVCMOS33

Tcl Console | Messages | Log | Reports | Design Runs | Package Pins | I/O Ports

	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
ll ports (3)											
Scalar ports (3)											
Inp_1	IN		V17	✓	14	LVCMOS33*	3.300				NONE
Inp_2	IN		V16	✓	14	LVCMOS33*	3.300				NONE
Outp	OUT		U16	✓	14	LVCMOS33*	3.300		12	SLOW	NONE

Common Question

- How do I access I/O Ports menu?
- After implementation it would open automatically
- If not Run Implementation first (If you have not already)
  - The follow the step below:



Simulation Code

- In simulation, you call your design code
- You assign simulation variables to the original
- Common pitfalls
    - Unit Under Test (UUT)
        - Your Design code
            - “Name” of the Verilog code you wrote
- Use the original variable names left part
- Use the simulated variable names right part
- There is no comma at the end

```
`timescale 1ns / 1ps
module Part_1_Sim(
    );
    reg Inp_1_t;
    reg Inp_2_t;
    wire Outp_t;
```

Same as your Design code Part_1.v

```
    Part_1 UUT(
    .Inp_1(Inp_1_t),
    .Inp_2(Inp_2_t),
    .Outp(Outp_t)
    );

    initial
    begin
        Inp_1_t=1'b0;
        Inp_2_t=1'b0;
    end

    always #5 Inp_1_t=~Inp_1_t;
    always #10 Inp_2_t=~Inp_2_t;
endmodule
```

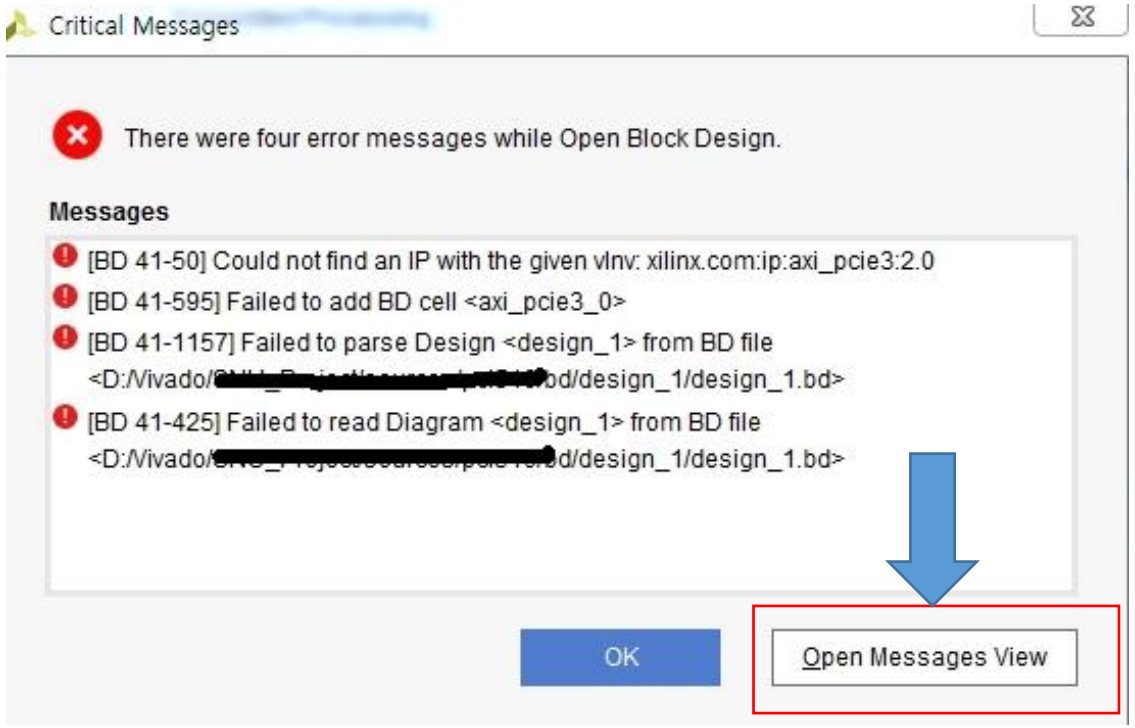
Original

Simulation

No Comma

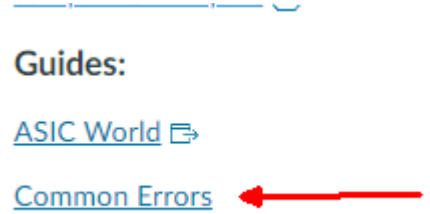
Steps to handle Error

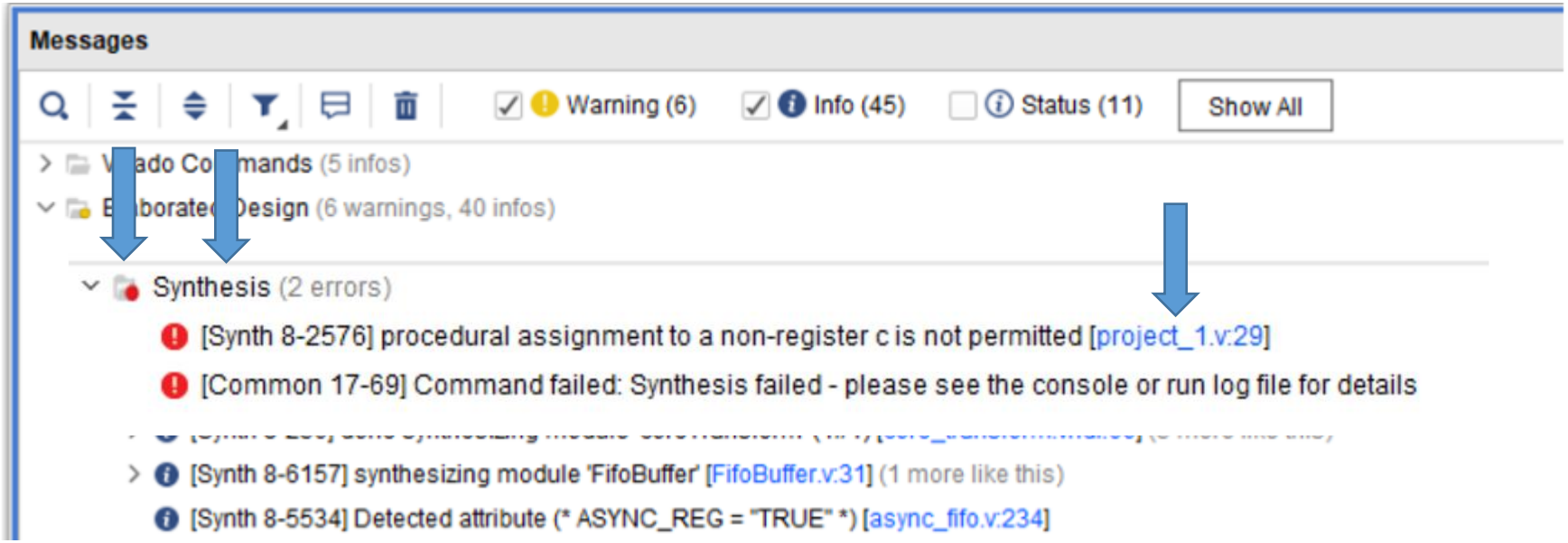
- Increase our knowledge of the reason behind why we take certain actions in Vivado
  - Prevention of them appearing again and again
- Look for messages, hints that program tells us
  - Learn to debug your program



Messages

- Learn to read messages window
  - Give the location (file), line (code) and type of problem
- Use the “Common Error” PDF file in webcourses
  - Keep this document “open” when lab starts (3 Pages only)



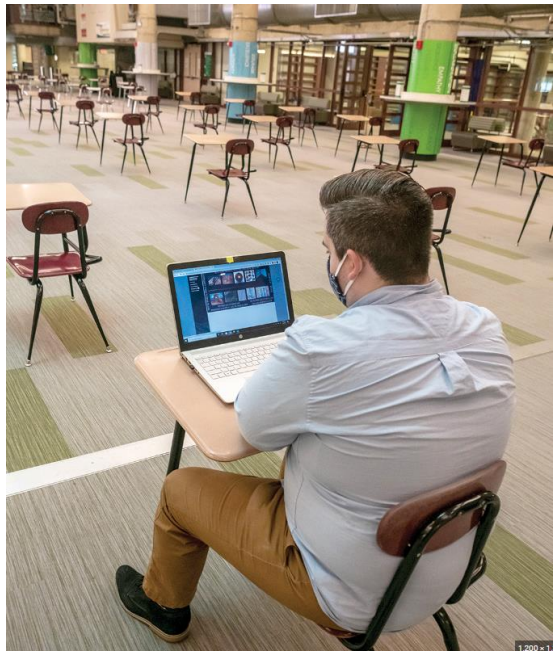


Numbers in Verilog

- 4’b0010
  - 4 bit binary 0010
- 8’h3f
  - 8 bit hex = 0011 1111
- 4’d7
  - 4 bit decimal = 0111

Collaboration: True of False?

- In this lab you are encouraged to “pair up” to check on each others progress
  - Share the knowledge you learn or discover
  - Help each other out with common bugs
  - Real-world FPGA projects are often complex and requires “teamwork”
  - However, keep the conversation short and technical (keep quiet so other students close to you wont lose their focus)



Conclusion

- Open the board and common errors pages
  - Added in the next pages
- Review the flow navigator and pitfalls to refresh FGPA design fundamentals
- Learn to collaborate effectively

Board

APPENDIX D
BASYS 3 BOARD

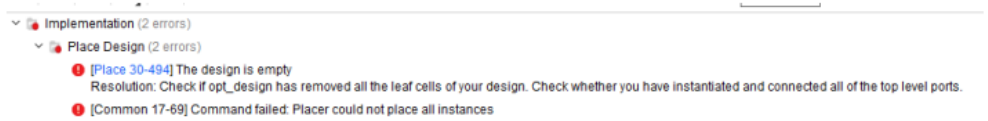
	PIN #	I/O Std
Switches		
SW0	V17	LVCMOS33
SW1	V16	LVCMOS33
SW2	W16	LVCMOS33
SW3	W17	LVCMOS33
SW4	W15	LVCMOS33
SW5	V15	LVCMOS33
SW6	W14	LVCMOS33
SW7	W13	LVCMOS33
SW8	V2	LVCMOS33
SW9	T3	LVCMOS33
SW10	T2	LVCMOS33
SW11	R3	LVCMOS33
SW12	W2	LVCMOS33
SW13	U1	LVCMOS33
SW14	T1	LVCMOS33
SW15	R2	LVCMOS33
LEDs		
LD0	U16	LVCMOS33
LD1	E19	LVCMOS33
LD2	U19	LVCMOS33
LD3	V19	LVCMOS33
LD4	W18	LVCMOS33
LD5	U15	LVCMOS33
LD6	U14	LVCMOS33
LD7	V14	LVCMOS33
LD8	V13	LVCMOS33
LD9	V3	LVCMOS33
LD10	W3	LVCMOS33
LD11	U3	LVCMOS33
LD12	P3	LVCMOS33
LD13	N3	LVCMOS33
LD14	P1	LVCMOS33
LD15	L1	LVCMOS33
Push buttons		
BTNU	T18	LVCMOS33
BTND	U17	LVCMOS33
BTNR	T17	LVCMOS33
BTNL	W19	LVCMOS33
BTNC	U18	LVCMOS33

Common Errors

This document illustrates some common Vivado errors and their solutions.

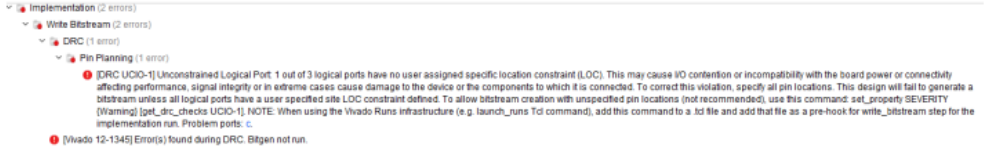
(i) **Design, Synthesis & Implementation errors:**

Error # 1: The design is empty during implementation



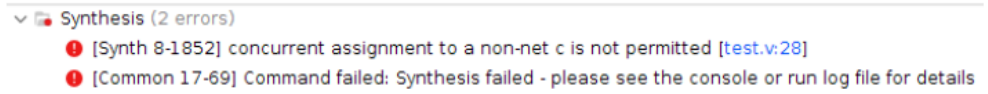
Solution : Your design most likely doesn't have any outputs defined (i.e. all the ports are defined as inputs)! Please check the code and define the outputs for the design.

Error # 2: The design contains unconstrained logical ports



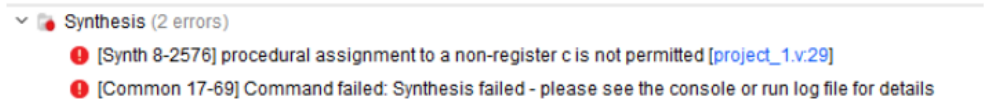
Solution: The inputs and outputs of the design are not constrained (i.e. not mapped to the ports of the FPGA board). Go to RTL Analysis > I/O Planning > I/O tab to map the ports and then save the constraints and the re-run the bitstream generation process.

Error #3: Concurrent Assignment not permitted



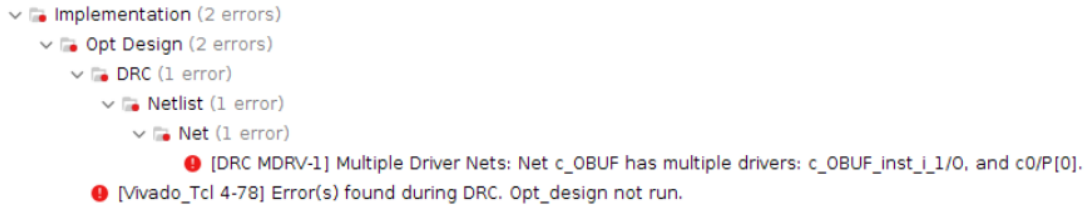
Solution: The variable (In this case, 'c') is declared as a register, but is still assigned concurrently through the use of 'assign' statement. To solve this issue, the variable needs to be either declared as a wire or its value needs to be assigned within sequential logic (i.e. inside an 'always' block).

Error # 4: Procedural Assignment not permitted



Solution: The variable (In this case, 'c') is declared as a wire, but is still procedurally assigned (within an 'always' block). To solve this issue, the variable needs to be either declared as a reg or its value needs to be set directly with 'assign' statement.

Error #5: Multiple driver nets not permitted



Solution: A register is assigned a value within multiple sequential/always blocks (Multi-driven port). The register needs to be written to in only one of the procedural blocks instead. In other words, all the assignments to that register should be moved within one procedural block.

(ii) **Syntax Errors:**

Vivado points out the line where the syntax error is, but it usually won't provide useful information about the error itself. Some of the common mistakes include but are not limited to :

1. Using semicolon instead of comma inside module declaration,
2. Putting comma after the last I/O declaration within the module parenthesis,
3. Not using a semicolon after every line of code,
4. Wrong use of operators (for example: using a~&b instead of ~(a&b)),
5. Not using (') sign correctly when assigning numerical values to a register etc.

(iii) **Simulation Errors:**

If running the testbench simulation results in an error (stops at compile/elaborate step) and error message includes something like "more information is available at "C:\User\.....\xvlog.log" or "more information is available at "C:\User\.....\elaborate.log" , first open the aforementioned log file in the specified directory from the computer. The file would typically have the description of the error in the testbench file. Some of the common errors might include:

1. The (name of the port) port was not found.

   Solution : Most likely, the name of the I/O port in the instantiation doesn't match the original name of the ports declared in the design file. Make sure to include the exact same I/O name during instantiation as Verilog is case-sensitive. For example : If an input is named Outp in the module declaration of a design file, then its name in the testbench instantiation should be in the form of .Outp() and not .outP() or .outp().

2. The module was not found.

   Solution : Make sure you use the exact same name for the module in design file and testbench.

3. Non-register value should not be assigned procedurally or non-net value should not be assigned concurrently.

   Solution : Make sure to map only the reg type variables to the inputs and only the wire type variables to the outputs of the instantiated module. Also, wire type variables should not be initialized in the testbench.

   If the simulation starts but the output signals in the simulation exhibit a blue bar, then the testbench-specific internal signals are not correctly mapped to I/O of the design file. Make sure to check the testbench code, especially the module instantiation to fix this issue.

Experiment 1: Gates & Verilog

- Objectives: Developing logic gates in FPGA platform
  - AND, NAND, OR, XOR, NOT
- Part I: Implement and Simulate AND gate (Last Week)
- Part II: Implement NAND, OR, XOR, NOT gate (Today)
- Part III: Two input – five output logic circuit (Today)
  - Show us your board only for part III and screenshots for part II

Reminder

- Lab Report for Experiment 1 is due next week (Night before the lab 11:59 PM)

Reference

- Fundamentals of Logic Design by Charles H. Roth, Jr. 7th edition