PDE3411 Laboratory Worksheet 1

7/10/2019

Aim

The purpose of this lab is to introduce you to the basic FPGA design and programming tools. For the purposes of this lab you will first study the useful Vivado Design Suite and then implement a 2:1 Mux circuit using FPGA design flow.

Objectives

- Verify that the Xilinx tools are up and running
- Introduce you to the Xilinx Vivado Design Suite
- Become familiar with VHDL coding and use of the Vivado Simulator
- Be able to synthesize and implement designs to FPGAs using Vivado Design Suite.

Equipment & Tools

Xilinx Vivado Suite You can visit http://www.xilinx.com/products/design-tools/vivado.html and choose the version you want. You can download the Vivado Design Suite

Webpack for free, but if you have trouble installing it, let us know ASAP.

Vivado Simulator

Introduction

This is a step-by-step tutorial for building a 2:1 Mux in Xilinx Vivado, a Design Suite software that provides designers with the ability to code designs in a hardware description language such as VHDL or Verilog. The Vivado Design Suite also provides the ability to apply FPGA pin and timing constraints, analyse for errors and violations, and synthesize to generate configuration bit file formats for FPGAs.

By the end of this tutorial, you should be able to:

- Create a new design by VHDL coding.
- Verify the function of a design by behavioural simulation.
- Map a design to an FPGA device through placement and routing procedures.
- Estimate the performance of the design by timing analysis.
- Use the 2:1 MUX in this tutorial to implement and simulate a MUX in FPGA.

Step-by-step 2:1 MUX Design

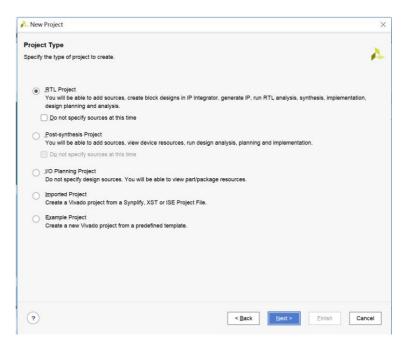
Tasks:

(1) Create a project file in Xilinx Vivado Project Navigator.

To launch the Project Navigator run Start → Programs → Xilinx Design Tools
 → Vivado 2019.1. Or, click the following icon on desktop.



Create a new project by clicking **Create Project** in Quick Start pane. You will get a **Create a New Project** wizard. Click Next, then give the project name and location of your choice, click Next. For the project type, choose the default **RTL project** then click Next (see figure below).



- In the Add Sources window, set the "Target Language" to VHDL and the "Simulator language" to Mixed.
- After setting the language options, click "Create File"

A New Project	×
Add Sources Specify HDL, netlist, Block Design, and IP files, or directories containing those files, to add to your project. Create a new source file on disk and add it to your project. You can also add and create sources later.	4
$ + = \pm =$	
Use Add Files, Add Directories or Create File buttons below	
Add Files Add Directories Create File	
Scan and add RTL include files into project	
Copy sources into project	
Add sources from subdirectories	
Target language: VHDL V Simulator language: Mixed V	
? <u>Seack</u> <u>Next</u> Einish Can	cel

• In the Create Source File dialog box, make sure the "File type" is set to VHDL and enter the name of your file.

• Again, it is always a good idea to use a descriptive name, in this case Mux21because this is a 2-1 multiplexor.

• <u>Click "OK" to close the dialog box and then "Next".</u>

+		
† 1	Create Source File	
	Create a new source file and add it to your project.	
	Ele type:	
	File name: Mux21 File location: Control Cont	
	? OK Cance	
Scan and add RTL include		

- The created file will appear in menu, click Next.
- We do not have any constraints yet, but in future projects, you may import and edit existing constraints files to save time. So click **Next** for now.

A New Project	x
Add Constraints (optional) Specify or create constraint files for physical and timing constraints.	
Use Add Files or Create File buttons below	
Add Files Create File Create File	
? Enish Cance	•

- Here we need to select our device. You can use the search function, filters, or just scroll until you find our device: XC7A100tcsg324-1.
 - This FPGA is from the Xilinx Artix-7 family (XC7A100T).
 - The device is contained in a 324-pin csg324 package.
 - The speed grade of the part is "-1".
- Click Next

Default Part Choose a default Xili	nx part or board for	your project. This	s can be change	d later.		2
Select: 🔅 Parts 📓 I 4 Filter	Boards					
Product category:	General Purpose	▼ Speed	grade: All Rer	naining 🔻		
Eamily:	Artix-7		grade: All Ren	maining 🖛		
Package:	csg324 *					
		Reset A	Filters			
Search: Q		-				
Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	GÈ Tr
xc7a75tcsg324-2L	324	105	180	94400	0	0 ^
xc7a75tcsg324-1	324	105	180	94400	0	0
xc7a75ticsg324-1L	324	105	180	94400	0	0
xc7a100tcsg324-3	324	135	240	126800	0	0
xc7a100tcsg324-2	324	135	240	126800	0	0
xc7a100tcsg324-2L	324	135	240	126800	0	0
xc7a100tcsg324-1	324	135	240	126800	0	0
xc7a100ticsg324-1L	324	135	240	126800	0	0 +
4 111						

- The New Project Summary window lists information selected in the previous screens.
 If necessary, use the **Back** button to return to previous screens to make changes/corrections.
- Click **Finish** to open the Project Manager window.

🝌 New Project	
	New Project Summary
HLx Editions	() A new RTL project named 'Lab0' will be created.
	(j) 1 source file will be added.
	$\underline{\mathbb{A}}$ No Configurable IP files will be added. Use Add Sources to add them later.
	$\underline{\Lambda}$ No constraints files will be added. Use Add Sources to add them later.
	(i) The default part and product family for the new project: Default Part: xc7a100tcsg324-1 Product: Artix-7 Family: Artix-7 Package: csg324 Speed Grade: -1
EXILINX ALL PROGRAMMABLE.	To create the project, click Finish
?	<back next=""> Finish Cancel</back>

- (2) Create the VHDL model
 - Since we chose to create a new VHDL file, Vivado will automatically launch a wizard to assist in creating the entity and architecture structures that comprise a VHDL model. This can all be done by hand (and you can edit all of this later), but there is no reason not to take advantage of the wizard utility.
 - In Module Definition window, leave the Entity and Architecture names as their defaults.
 - Create three inputs (Direction "in"): Din0, Din1, Sel
 - Create two outputs (Direction "out"): Dout1, Dout2
 - Click **OK**

🔥 Define	Module					-	×	
For each MSB an	Define a module and specify I/O Ports to add to your source file. For each port specified: MSB and LSB values will be ignored unless its Bus column is checked. Ports with blank names will not be written.							
Module De	finition							
Entity	name:	Mux21					0	
Archi	tecture name:	Behavioral					0	
I/O P	ort Definitions							
+	Port Name	Direction	Bus	MSB	LSB			
- 1	Din1	in -	-					
1	Din0	in -	- 🔳					
+	Sel	in -	• 🔳					
	Dout1	out	•					
	Dout2	out	•					
		in ·	• 🔳					
?						OK	Cancel	

• Open your new VHDL file from the Sources window (double click on the Design Source name, Mux21)

le Edit Flow Tools Window	Lavout View Help											6	- Search	comman	\$
) ស មា 🗎 🐘 🗙 🔌 1		ult Layout 👻	X & X	3											Re
low Navigator ? «	1.9														?
、工庫	Sources		2-06	×	S. Pre	ject Su	mary	×							200
	< ≅ ≑ ≥ 8 ■ 2					ct Settin		~							
Project Manager	Design Sources (1)				- and the second		-								
R Project Settings	-@# Hux21 - Behavior	al (Mux21.vhd)				t name:	LAB								
Add Sources	Constraints Generation Sources (1)					t location:			/Labs/LA80						
Canguage Templates						t family:		tix-7							
🞐 IP Catalog					Project		_	7a 100tcsq	324-1						
The factor of th				_ U		odule nam	1.0	<u>x21</u>							
IP Integrator	Hierarchy Libraries Comple	Order				language									
Den Block Design	Land and the second sec	THINK				tor langua	sge: mo	000							
Generate Block Design	Source File Properties		? - 0 2	×	Synth	iesis					In	plement	ation		
- and the second design	$\leftarrow \rightarrow \boxed{5}$				Status	: Not	started				Sta	atus:	No	ot starte	đ
Simulation	Mux21.vhd				Messa	ges: No	errors or	warnings			ме	ssages:	No	errors o	or warnings
🚯 Smulation Settings 🔍	1				Part	xc7	a 100tcsg	9324-1			Pa	rt	xc	7a 100tc	sg324-1
Run Simulation	Enabled			11	Strates	ay: Vine	edo Synt	hesis Defa	ults		Str	ategy:	YE	vado Imp	lementation De
	Location: F:/ELEC4200/L	abs/LAB0/LAB0.srcs/sourc	es_1/new								Inc	remental			
RTL Analysis	General Properties				-				117					- · · ·	
🚳 Elaboration Settings	deneral ridge and								11						
🕨 🔂 Open Elaborated Design	Design Runs														? _ 🗆 🖻
Synthesis	Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Faled Routes	LUT	FF	BRAM	URAM	DSP	Start
👸 Synthesis Settings	🖀 🖻 🤿 synth_1	constrs_1	Not started												
Run Synthesis	🚔> impl_1	constrs_1	Not started												
	*														
Open Synthesized Design	14														
mplementation	30-														
Implementation Settings	44														
Run Implementation															
Dpen Implemented Desig															
Program and Debug	Td Console 🛛 💭 Message	s 🔄 Log 📄 Reports	Design	Runs											

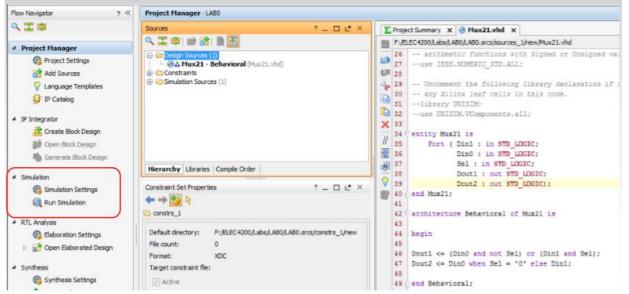
- Scroll down and add the following two lines of code between the Begin and End statements of the architecture section of the model.
 - Dout1<=(Din0 and not Sel) or (Din1 and Sel);
 - Dout2<=Din0 when Sel = '0' else Din1;
- After saving, we can then set up the simulator.

AB0 - [F:/ELEC4200/Labs/LAB0/	A80.xpr] - Vivado 2016.2			
Ele Edit Flow Tools Window	Lagout Yew Help	Q. Search	commands	
😂) 📾 💷 🔛 🏠 🗙 👂 🕨	🔹 🍓 🐝 ∑ 🤪 😬 Default Layout 💿 💌 🎉 🗞 🍾 🧐		Ready	
Flow Navigator 7 <<	Project Manager LABO		7 X	
Q. 🞞 🗰	Sources ? _ C L* ×	E Project Summary X Mux21.vhd * X	D C X	
# Project Hanager	🔍 🎞 🗰 📾 📾 🖪	P:/ELEC4200/Labs/LAB0/LAB0.arcs/sources_1/new/Mux21.vhd		
Project Settings Ad Sources Ad Sources Lange Templates Protegrator Brocket Block Design Centre Block Design Generate Block Design Generat	Constantial Constanti	<pre>24 estimatic functions with Signed or Unsigned values 27est IESE.NEWERC_FFL.ALL: 42</pre>	 Mux models 1st statemen logic equation 	it:
 RTL Analysis Baboration Settings Baboration Bettings Sign Open Elaborated Design Synthesis Settings Run Synthesis Bon Synthesis de Design 	If Drabled * Location: PLBLEC-1000/Labs/L480/L480.arcs/sources_Unew Type: VHCK: Library: dL_default6 Stet: 1.118 Ceneral Properties	<pre>4 4 5 5 6 1 5 6 1 5 6 1 5 6 1 5 6 1 5 6 1 5 6 1 6 1</pre>	2 nd statemer "behavior"	nt:
Diperentation Diperentation Diperentation Diperentation Diperentation Diperented Design Constrained Design Diperented Design	Degraf.Are Name Constants Status With Physical Constants Status With Physical Constants I Netstated Physical Constant I Netstated Physical Constant Physical Constants I Netstated Physical Constants Physical Physical	1765 WHS THS THOS FadedRoutes LLT FF 86444 LBAHE COP Start	P _ □ L ² × Bigened	

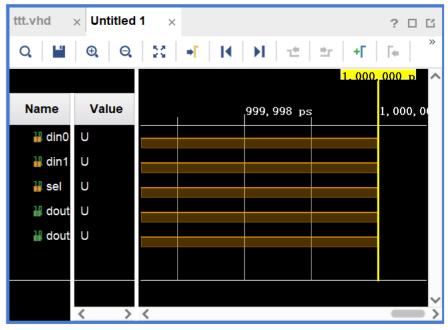
• Double click on **Settings** in **Project Manager**, in the pop out window, click on **Simulation** in **Project Setting**. In the **Target Simulator** drop-down menu, you will have choice of different simulators. We will use the default Vivado simulator.

ject Settings	Simulation Specify various settings asso	ciated to Simulatic	n	1
General Simulation	Target simulator:	Vivado Simulato	r	~
Elaboration	Simulator language:	Mixed		~
Synthesis	Simulation set:	🗅 sim 1		~
Implementation Bitstream IP	Simulation top module name:			
l Settings	Compilation Elaboration	n Simulation	Netlist Advanced	
Project				
P Defaults	⊻erilog options:			
Board Repository	<u>G</u> enerics/Parameters opt	ions:		
Source File	xsim.compile.tcl.pre			
Display	xsim.compile.xvhdl.nos	ort	\checkmark	
VebTalk	xsim.compile.xvlog.nos			
Help	xsim.compile.xvlog.rela			
Text Editor	xsim.compile.xvhdl.rela			
Brd Party Simulators	xsim.compile.xsc.more_	options		
Colors	xsim.compile.xvlog.mor			
Selection Rules				×
Shortcuts Strategies Nindow Behavior	Select an option above to s	ee a description of	ī it	

• Click on "Run Simulation" in the Flow Navigator, and then Run "Behavioral Simulation" in the popup menu.



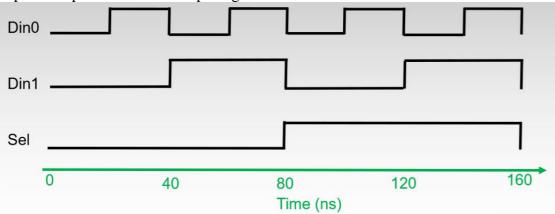
- •Note that we have not yet provided any signal inputs, so this "simulation" did not provide any useful information.
- When the simulator is finished launching you should see the following window. If you do not, call tutors to set up the proper view windows.
- We will set up the desired simulation in the next steps.



- Design verification requires that you stimulate the inputs and observe the outputs. You are to stimulate the inputs with all possible input combinations and observe each output to verify its correctness.
- This is called "exhaustive testing". It is suitable for a simple circuit (such as the mux) but is not practical for large circuits with many inputs.
- Instead of using testbench, we use an easy, alternative way to provide the repeatitive clocks to the input signals to test our design.
- Now right click on one of the input signals, in the popup menu, choose **Force Clock**. In the popup menu, set the values as shown below.

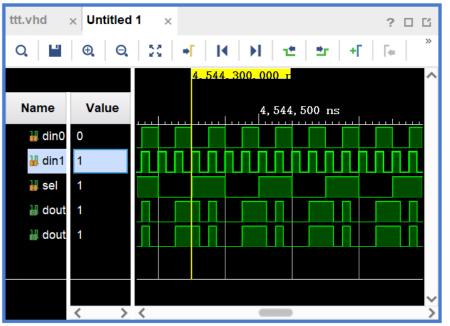
A Force Clock: /ttt/din0		×
	rce the signal to a constant value. Assignments or any previously applied constant or clock force	4
Signal name:	/ttt/din0	
<u>V</u> alue radix:	Binary	~
Leading edge value:	0	\otimes
<u>T</u> railing edge value:	1	\otimes
Starting after time offset:	Ons	\otimes
<u>Cancel after time offset:</u>		
Duty cycle (%):		50 🌲
<u>P</u> eriod:	100ns	\otimes
(?)	ок	ancel

- When you set the input signal values in the above menu, you can set the period of clock of your choice.
- For the 2-1 MUX we have 3 inputs (Din0, Din1, and Sel) where each input can be either logic high (1) or logic low (0).
- • There are 2^3=8 possible input combinations. While we could stimulate each combination individually, an easier approach would be to clock stimulator, as we did just now, to generate the input timing diagrams below.
- Repeat this process to all the input signals.



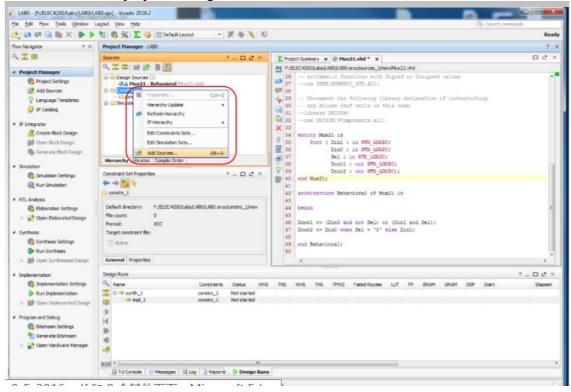
- Now we need to run the simulation using the simulation controls.
- Run the simulation for 10ms by typing 10ms into the "Run Duration" box and then click "Run For". Your output should look similar to this. Use the zoom controls as needed to make the waveform fill the window.





- Examine your results and ensure that they accurately match the operation of a 2-1 MUX for both Dout1 and Dout2.
- If you observe any incorrect results during simulation, go back and debug your circuit. Do not proceed any further until your simulation produces the correct results.

- (3) Add constraints to your design for implementing in FPGA
 - We need to define a "constraints file" that defines which signals (Din, Dout, Sel) go to which pins on the FPGA.
 - Now close the **Simulation Window** if you have not done so. Right click on the "Constraints" folder in the project manager and select "Add Sources".



• This opens the "Add Sources Wizard". Make sure "Add or create constraints" is selected and click "Next".

Add Sources		×
HL _x Editions	Add Sources This guides you through the process of adding and creating sources for your project Add or greate constraints Add or create design sources Add or create simulation sources	
£ XILINX。		
?	< Back Next > Einish	Cancel

• Click "Create File" and in the dialog box that opens, name your constraints file. As always, use a descriptive name, like "LAB0_Constraints". Then click "OK".

Add Sources	
Add or Create Constraints Specify or create constraint files for physical and timing constraint to add to your project.	
Specify constraint set: 🚞 constre_1 (active) 💌	
t + + +	
Use Add Files or Create File buttons below	Create Constraints File
Use Add thes or Create the outlans below	Create a new constraints file and add it to your project
	Eile type:
Add Files Create File	File name: LAB0_Constraints
Cogy constraints files into project	File location: 🔂 <local project="" to=""> 🔻</local>
<pre></pre>	? OK Cancel

• Click "Finish" to add your file to the project. Add Constraints

🝌 Add Sources	×
Add or Create Constraints Specify or create constraint files for physical and timing constraint to add to your project.	4
Specify constraint set: Constrs_1 (active) ✓	
+, - + + Location	- 1
lab0_constraint.xdc <local project="" to=""></local>	
Add Files Create File	
Copy constraints files into project	1
? <gack< td=""> Next> Einish</gack<>	Cancel

• Open your newly added file by double clicking on it from the Project Manager.
PROJECT MANAGER - project_2

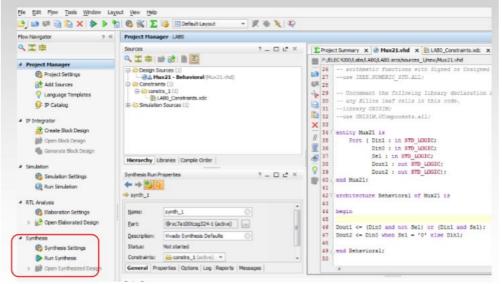
Sources	? _ 🗆 🖒 X
Q 素 ♦ + ፼ ● 0	¢
✓	1
Iab0_constraint.xdc	
Hierarchy Libraries Compile Order	
Constraint Cat Dranatian	
Constraint Set Properties	? _ O 🛙 X
Constraint Set Properties	? _ □ ⊑ ×
	? _ □ ≦ × ← → ✿

• Add the following lines in the constraints file, renaming the ports to match those in your design. Save after you are done.

```
set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMOS33 } [get_ports { din0}];
set_property -dict { PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [get_ports { din1}];
set_property -dict { PACKAGE_PIN M13 IOSTANDARD LVCMOS33 } [get_ports { sel}];
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { dout0}];
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { dout1}];
```

(4) Implement your design in FPGA

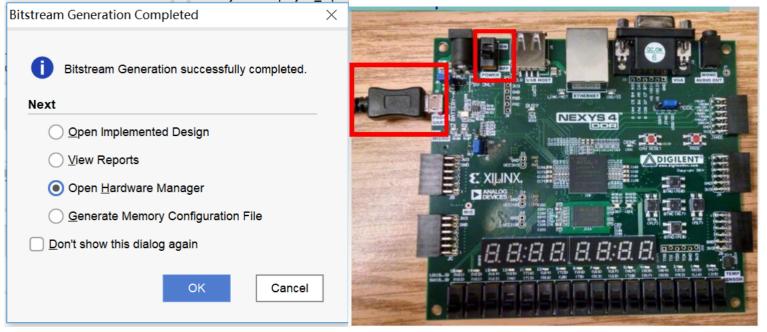
- Now you are going to have Vivado "synthesize" your design into generic digital logic components.
- In the Flow Navigator, click "Run Synthesis".
- This may take several minutes to run, so be patient.
- A dialog box will open on completion, either select "Run Implementation" or hit "Cancel".



- Click on "Implement Design". This maps the synthesized design to the hardware, and routes I/O ports to the pins specified in the constraints file.
- Again, this will take several minutes to run.
- A dialog box will open on completion, either select "Generate Bitstream" or hit "Cancel".

j un 44 liji liji 🗙 🕨 🎙	🚳 🐝 🔼 🧐 🖾 Defailt sput 🔷 🗶 😓	Synthesis Comple
w Navigator 9 40	Project Manager - LA00	?
X #	Sources ? = D L ² × Derect Summary X @ Mus2Lubd X D LABO_Constraints.ude X	000
Project Hanager	🔍 🎞 🕸 📑 🛃 🔚 🚺	
Project Settings Add Sources Impunge Templates P Catalog P Catalog	B: Bocgenshaves () IB (Ander, 1997 visc) (BLOADE, 1997 visc) (BLOA	
💏 Create Block Design 📬 Open Block Design 🍓 Generate Block Design	· /	
Simulation	Hierandhy Ubranis Comple Order 2	
Simulation Settings	source He Properties 7 = L1 to X	
🔍 Run Simulation	← ⇒ 💁 k ■ LARD Constraints sold	
RTL Analysis		
 Elaboration Settings Open Elaborated Design 	[ℤ Drabled #	
	Location: Fr;REC400(Labo).480(JA80, won(constra_1)/www Type: NDC	
Synthesis	Sac: 0.449	
 Synthesis settings Run Synthesis 	Madfied: Today at 13-49:24 PM	
> Den Synthesized Design	General Properties	
Implementation	Design Runs	7 - D 2 ×
(b) Implementation Settings	Name Constraints Status WHG THS WHS THIS TP/US Failed Routes UUT IFF SRAM URAM OSP Start	Elapsed Stra
Run Implementation Deen Implemented Design	Image: membra and me	:+9 PM 02:01:25 Vive Vive
Program and Debug	H	
Cenerate Bitstream		
	e	

- With the design now implemented, it is time to generate the bitstream, or the file that will be downloaded to the FPGA.
- Click "Generate Bitstream".
- A dialog box will open on completion, either select "Open Hardware Manager" or hit "Cancel".
- Plug in the USB cable between a PC USB port and the USB port on the FPGA board to access the JTAG programming module.
- Flip the board power switch from OFF to ON



- Click on "Open Target" => "Auto Connect" to connect to your Nexys 4 board.
- If you encounter any errors, verify that the board is plugged in and powered on.
- Click on "Program Device", and the device name. The bitstream file will appear, then clock on "Program".
- Verify correct operation of the circuit using the switches and observe the output on the LEDs. If you encounter any bugs, attempt to figure out what went wrong before asking for help.
- Ensure that you apply all possible input combinations, as you did in simulation, and verify that the outputs match the simulation results.
- After verifying the circuit is correct, call the tutors over and demonstrate the circuit. GOOD LUCK!

(5) Clean up!

- Turn off the board, unplug the USB cable, put them back in their box, and return the box to the tutors.
- Close Vivado, any other open programs, and save all files.
- Don't forget to log out of your machine and take any USB drives with y