San Jose State University Department of Electrical Engineering EE178, Fall 2018, Crabill/Nguyen

Laboratory Assignment #3

Objectives

This lab is an introduction to the creation of basic constraints files with Xilinx Vivado 2016.2. Basic constraints include specification of pin placements and signaling standards, as well as timing specifications appropriate for designs fully synchronous to a single clock.

No new logic design concepts are presented in this lab. The goal of this lab is for you to become more familiar with the tools. Please read carefully, pay attention, and take your time. This lab is not a race to see who gets done first.

In order to receive credit for this lab, you must complete the tasks and submit your project to the instructor.

Bibliography

This lab draws heavily from documents on the Xilinx website <u>http://www.xilinx.com</u>. I would like to thank Xilinx for making this material available. This lab is effectively a customized introduction to Xilinx Vivado using Verilog-HDL and the Digilent Basys3.

Xilinx Design Constraints

Previously, you have been served basic implementation constraints to copy and paste into a newly created Xilinx Design Constraints file (hereafter referred to as an XDC file) added to your project. It is a common practice to re-use constraints in this manner, particularly for a known hardware platform like the Digilent Basys3.

As you gain confidence in specifying constraints, you may elect to leverage constraints you have used previously, even if they are not an exact match for your needs, and manually edit them in a text editor. However, Vivado provides a powerful capability to interactively create constraints and save them to an XDC file, a feature exceptionally useful to everyone. As a novice, perhaps you know nothing of the available constraints and their syntax – Vivado will guide you. As an expert, perhaps you need advanced capability to query elements in your design to which you will apply complex constraints – Vivado will assist you.

Referring to the previous assignments if necessary, open Vivado and create a new project. After you have created a new project, create a new design source named fsm.v and replace the automatically generated contents of the file with the following:

```
// File: fsm.v
// This is the top level design for EE178 Lab #3.
// The `timescale directive specifies what the
// simulation time units are (1 ns here) and what
// the simulator time step should be (1 ps here).
`timescale 1 ns / 1 ps
// Declare the module and its ports. This is
// using Verilog-2001 syntax.
```

```
module fsm (
  input wire pause,
  input wire restart,
  input wire clk,
  input wire rst,
  output reg [1:0] state,
  output reg odd,
  output reg even,
  output reg terminal
  );
  parameter [1:0] FIRST = 2'b11;
  parameter [1:0] SECOND = 2'b01;
  parameter [1:0] THIRD = 2'b10;
  always @(posedge clk or posedge rst)
  begin
    if (rst) state <= FIRST;
    else
    begin
      case(state)
        FIRST:
                 if (restart || pause) state <= FIRST;
                  else state <= SECOND;</pre>
        SECOND: if (restart) state <= FIRST;
                  else if (pause) state <= SECOND;
                  else state <= THIRD;</pre>
                 if (!restart && pause) state <= THIRD;
        THIRD:
                  else state <= FIRST;</pre>
        default: state <= FIRST;</pre>
      endcase
    end
  end
  always 0*
  begin
    odd = (state == FIRST) || (state == THIRD);
    even = (state == SECOND);
    terminal = (state == THIRD) && (restart || !pause);
  end
```

endmodule

You will notice this is a version of the FSM example discussed in class. Hopefully you understand its behavior, but its value in this assignment is the relative simplicity it provides. We will not be simulating the design, nor attempting to exercise it in hardware – but we will be applying constraints to it.

After you have saved and closed the new design source, create a new constraint source named fsm.xdc. Verify the new constraint file is empty, save it, and close it. You are now ready to work on constraints.

Pin Placement and Signaling Standard Constraints

Most designs have top-level ports that map to device pins. Prior to synthesis, enough information can be extracted from the elaborated design sources to facilitate assignment of top-level ports to FPGA device pins. The FPGA device pins are highly (but not infinitely) flexible, and as a result it is necessary to select analog characteristics of the FPGA device pins (e.g. signaling voltage, output drive strength, etc...)

concurrently with pin placement to ensure the desired combination can be correctly implemented. Under RTL Analysis in the Flow Navigator, click Open Elaborated Design. You may receive a notice that the design elaboration settings allow pin placement and constraint-related work, but can result in longer elaboration times. If you receive this notice, accept it to proceed with elaboration.

The elaborated design opens to an RTL Schematic as shown in Figure 1. RTL stands for register transfer level, a level of abstraction typically higher than the gate level output of a synthesis tool. Take a moment to explore the RTL Schematic and familiarize yourself with the buttons to scroll and zoom.



Figure 1: Elaborated Design Schematic

Now, switch from the Default layout to the I/O Planning layout, using the selection box that is also shown in Figure 1 at the top of the window.

The Package View of the device should appear as shown in Figure 2. This is a physical representation of the packaged device you selected during creation of the project. This is a ball grid array package, so it technically has balls, not pins, and they are identified by a row position (a letter, down the left edge) and a column position (a number, across the top edge). If you hover over an individual ball with the pointer, an informational note will appear to provide additional detail.

As you might expect, a number of pins are dedicated to power and ground. Other dedicated pins include those for special analog functions and exchanging programming information and status with the device. As far as programmable pins suitable for mapping to top-level design ports, these are shown as solid gray circles and solid cyan hexagons. Those shown as solid cyan hexagons are "clock capable" and have preferred internal connections to clocking resources.

Also notice the grouping of programmable pins suggested by the coloration of the heavy lines drawn around each group. These groups are called I/O banks, and the programmable pins in a given I/O bank must share certain characteristics such as the I/O power supply voltage.

e Edit Flow Tools Window I	Layout View Help												Q+ Search commands
), 🖽 i in 🕫 🐚 🐘 🗙 🛷 (> > *	🌀 🔛 I/O Planning	- W	🔅 🔪 😓									Read
low Navigator ? «	Elaborated Design - xc7a35tcpg236-1 (a	ctive)											?
A II #	Device Constraints	2 - 0 6	× B Pa	kana v iškūasi	v v Historia	er v							2023
	2、二部()—			abye A groch									
Project Manager	Internal VRE# 0.6V 0.675V 0.75V						+				S S C		
😅 IP Catalog	0.9V						+ + +	+ + +	🗄 G 🔽	C 🔜 🛛	C S S 🕂 🤇		
IP Integrator Areate Block Design Doen Block Design			× ₽ N				+ +						
🍓 Generate Block Design									+ +		+ + +		
4 Simulation									÷ ÷	+ + -	+ + +	O ÷ O	
🚯 Simulation Settings									4	+ + -	+ + +		
Run Simulation							4	4	4 ±		+ +		
RTL Analysis			1 1 1					4	4	* * .	4 4 4		
Baboration Settings	🔒 Sources 👔 RTL Netist 🚵 Dev	ce Constraints	`					4	4	L 4	h h 1		
Elaborated Design Report Methodology	Source File Properties	? _ 🗆 🖻	×						* *		+ + +		
Report DRC	← → ¹ / ₂ ≥						4						
Report Noise	h fsm.xdc		_				4 C)	1					
Schematic	Enabled		^					4					
Synthesis	Location: C:/Projects/lab3/lab3.src	s/constrs_1/new	- 1 a a							- C			
🚯 Synthesis Settings	Type: XDC -		, i i				4		+ 6	+ C (
Run Synthesis	Size: 0.0 KB												
Open Synthesized Design	Modified: Today at 09:31:03 AM General Properties		• W										
 Implementation 	Properties Clock Regions		_										
Implementation Settings	1/O Ports												? _ O & ×
> Den Implemented Design	Name	Direction	Neg Diff Pair	Package Pin	Fixed Ban	k I/O Std	Vcco V	ref Drive Str	ength Slew Ty	pe Pull Typ	e Off-Chip Terr	mination IN_TERM	
	All ports (9)	0.5				defende di versione sito	- 1 800	12	- 0.00	- NONE			
Program and Debug Ritekeen Settings	G state[1]	OUT				default (LVCMOS18)	* 1.800	12	* SLOW	* NONE	 FP_VTT_50 		
Generate Bitriream	G state[0]	OUT			*	default (LVCMOS18)	* 1.800	12	* SLOW	* NONE	FP_VTT_50	*	
Conso Marchuses Manager	-D> dk	IN			*	default (LVCMOS18)	* 1.800			NONE	* NONE	*	
	- Cill even	OUT			*	default (LVCMOS18)	· 1.800	12	* SLOW	+ NONE	FP_VTT_50		
	bo D-	OUT			*	default (LVCMOS18)	1.800	12	* SLOW	* NONE	 FP_VTT_50 	*	
	✓ D- pause	IN			*	default (LVCMOS18)	* 1.800			NONE	* NONE	•	
	D restart	IN			*	default (LVCMOS18)	- 1.800			NONE	* NONE	-	
	-UP rst	IN				default (LVCMOS18)	* 1.800	12		NONE -	* NONE		
	1 10 THOS	001			-	uerouri (LVCMUS18)	- 1.800	14	- SLOW	- 14014C	< PP_V11_90		

Figure 2: Package Diagram for I/O Planning

If the list of I/O Ports at the bottom of the window is not already expanded as shown in Figure 2, expand it, and start assigning the top-level ports to pins. You can do this by dragging and dropping. Simply select the port symbol next to the I/O Port name, and drag it onto a ball in the Package View. Begin by assigning the clk port to package pin W5. This particular selection is dictated to you to ensure the clk port is assigned to a "clock capable" location, although any other "clock capable" location would be appropriate. In the I/O Ports list, update the other columns for the clk port to match what is shown in Figure 3, which shows the desired end result.

We will use LVCMOS33, a 3.3v LVCMOS standard. In general, we can put an LVCMOS33 pin anywhere we want, but at the point we have made I/O placements, the I/O banks in which the placements have been made will require specific I/O power supply voltages. There are rules that specify what I/O standards can co-exist in a bank, given a specific I/O power supply voltage. For example, you cannot put LVCMOS18, a 1.8v LVCMOS standard, into a bank with LVCMOS33, a 3.3v LVCMOS standard.

LAR INT 1005 VIEDOW L	associat Minus Hinto											O crash summark
	wor ⊼em ⊒eb s⊼onr ⊼em ⊒eb	ID 1/0 Barring	- W & > 10	D								Q+ search commands
	Elaborated Derion * - w7a2Dros226	(action)	1 × × × 1	-7								-
22 m	Device Coortrainty	2 🗆 🖉	X Manual	a la sultana a								 2 🗆 12
		0 0	Package X	Device X 34 Schematic	×		-					
Porject Hanager Porject Hanager Porject Settrops Stad Sarces Porjectars Porjectars	Dent to 1	r = U d	A Directory X (A Dir	Device X 3 Schematic	x							
Alborated Design Constraints Settings Synthesis Settings Sont Synthesis	I/O Port Properties ← → Solice Br ck Dectors: N PartBonpin: True Package pro: (WS Site type: 10,129,71,480C_34	? _ 0 2	х N P R T U V S						+ + + + C C + C C C C C			
Implementation	Properties Congue							Port: clk Direction: Input	(nartition nin)			
🔞 Implementation Settings	tin num							Net: clk	- man party			
Run Implementation Den Implemented Design	Name	Direction	Neg Diff Pair Package Pi	n Fixed Bank	I/O Std	Vcco V	ref Drive	BEL: PAD Site: W5 (fixed) Site type: IO_L12 Tile: RIOR33_X43	P_T1_MRCC_3	Off-Chip Termi	nation IN_TERM	7_00
Program and Debug	😂 🕀 🕄 state (2)	OUT			default (LVCMOS18)	 1.800 1.800 	12	Package pin: W5		* FP_VTT_50 * FP_VTT_50	-	
6 Bitstream Settings	- d state[0]	OUT		*	default (LVCMOS18)	* 1.800	12	U/O Bank: I/O Ba	nk 34	+ FP_VTT_50	*	
Generate ditstream	Galar ports (7)	IN	WS	• •	34 LVCMOS33*	3,300			NONE	NONE		
 Open naroware narlager 	G even	OUT		•	default (LVCMOS18)	▼ 1.800	12	▼ SLOW	 NONE 	FP_VTT_50	*	
	Bo Do De course	OUT			default (LVCMOS18) default (LVCMOS18)	 1.800 1.800 	12	* SLOW	 NONE 	 FP_VTT_50 NONE 	*	
	-D restart	IN			default (LVCMOS18)	* 1.800			NONE	* NONE		
	-D-rst	IN			default (LVCMOS18)	* 1.800	12	* 9.0W	NONE NONE	* NONE	:	
	terminal	(1)				- 1.04 H		T				

Figure 3: Design Clock Port Assigned to a Clock Capable Pin

Use the drag and drop feature to assign all the remaining I/O Ports to pins. In practice, you might have no freedom in your assignments if the board design for this device has already been created. On the other hand, you may find yourself providing pin assignments to a board designer, or possibly engaged in an interactive pin-planning discussion to yield an optimal result.

For this assignment, you may select pin locations randomly, or strive for a theme such as "close together" or "spread out" (this is a tutorial, we are experimenting). You will notice that Vivado won't let you drag and drop when the result would cause an I/O power supply voltage conflict. In this case, you may need to change the pin I/O standard before you drag and drop. Figure 4 shows one possible final result – yours may differ.

A random or thoughtless assignment is not a good idea for a real project, because the performance of a design is affected by its pinout – and once you commit to a board layout based on a pinout, it costs time and money to correct bad choices in the pinout.

File Edit Flow Tools Window L	ayout View Help													Q - Search o	ommands
😂 🔛 📾 🕫 🖬 🐘 🗙 🏘	> > > > > > > > > > > > > > > > > > >	🖸 🧑 🔛 I/O Planning	- X & 1	k ©											Read
Flow N Save Constraints (Ctrl+S) «	Elaborated Design * - xc7a35tcpg236	1 (active)													? >
9, 22 #	Device Constraints	2 - 0 6	× El Package	v Bloom v Hill	chanatic V										? 🗆 🖒 ×
	2、22 命 图 -		+	A COUNCE A DAT	JOIGHOIC A					0 10 11	12 12 14 15	10 17 10 10			
Project Hanager Reg Project Hanager Ad Sources Language Terolates P Language Terolates P Language Pointegrator Create Book Design Reventate Block Design Committee Block Design Smulation	Internal V607 0.677 0.6779 0.6779 0.6779 0.6779 0.797 0.079		<mark>311, (), (), (), (), (), (), (), (), (), ()</mark>						+ G C						
🚱 Simulation Settings (Run Simulation			*							÷ + ÷	++				
# RTL Analysis			Pn K						T	1 4 1			1		
Baboration Settings	& Grunner D DTI Natiet @ D	auica Constrainte	·							* * *	P P				
Elaborated Design Report Methodology Report DRC Report Noise	I/O Port Properties ← → ☆ k ≪ terminal	? _ 🗆 🖻	× N P						* *	+++	÷ ÷				
Synthesis Synthesis Synthesis Settings Run Synthesis b Open Synthesized Design	Name: terminal Direction; OUT Partison pin; True Package pin; U1 Site type; IO_J.3N_T0_DQS_34 General Properties; Configure Configure		т U V W							<pre>C C + C C C C C C</pre>					
 Implementation 	Properties Clock Regions		_												
Implementation Settings	1/O Ports														? _ 🗆 ど ×
Open Implemented Design	Name	Direction	Neg Diff Pair Pack	age Pin Fixed	Bank I/O	Std	Vcco Vr	ef Drive Stre	ngth Slew Typ	e Pull Type	Off-Chip Term	nation IN_TERM			
Program and Debug Bitstream Settings Generate Bitstream		OUT OUT OUT	U3 U2	:	34 LVC 34 LVC 34 LVC	740533* 740533* 740533*	 3.300 3.300 3.300 	12 12 12	 SLOW SLOW SLOW 	NONE NONE NONE	 FP_VTT_50 FP_VTT_50 FP_VTT_50 	*			
Open Hardware Manager			W5 R2 R3	: *	34 LVC 34 LVC 34 LVC	MOS33* MOS33* MOS33*	* 3.300 * 3.300 * 3.300	12 12	+ SLOW + SLOW	NONE NONE NONE	 NONE FP_VTT_50 FP_VTT_50 	•			
	By pause By restart By restart By rst G terminal	IN IN OUT	T1 T2 T3 U1		34 LVC 34 LVC 34 LVC 34 LVC	MOS33* MOS33* MOS33* MOS33*	 3.300 3.300 3.300 3.300 	12	* SLOW	NONE NONE • NONE	 NONE NONE FP_VTT_50 	*			
From the second state for the second data	Td Console 🗋 🗩 Messages 🗌 🗔	Log 🗋 Reports 🗎 🖨 Desig	n Runs 🗌 🔎 Package Pir	s D I/O Ports											

Figure 4: Remaining Design Ports Assigned to Other Pins

When you are done, click on the Save Constraints button in the toolbar, also shown in Figure 4. A dialog box will appear, asking if you want to save the constraints to a new XDC file, or to an existing XDC file in the project. Save the constraints to the fsm.xdc file that's already in the project.

At this time, close the elaborated design by clicking on the "X" in the light blue Elaborated Design title bar.

Fully Synchronous Single Clock Timing Constraints

Timing constraints involve the specification of delays between objects in the design. It is only after the completion of synthesis that all objects in the design are known. Under Synthesis in the Flow Navigator, click Run Synthesis, and when it completes, Open Synthesized Design. You can review the reports later, if you wish.

There are a variety of constraints you can interactively apply to the synthesized design. For interactive timing constraint entry, click on Edit Timing Constraints in the Flow Navigator. This is listed under Synthesis in the Flow Navigator, but you may have to expand the Synthesized Design options to see it.

For synchronous design, timing analysis is fundamentally asking one question: For each signal path that exists from a flip-flop output, possibly through combinational logic, to a flip-flop input, can the successful transfer of information be guaranteed? This type of timing analysis is called static timing analysis because it involves evaluation of signal path delays without regard to circuit activity.

If you have a complex design with lots of paths, this question may need to be posed millions of times to yield a complete result. Nobody has time (or wants) to do this task by hand, which is why static timing analysis tools exist.

Review the FSM structure shown in Figure 5. As mentioned, our interest is in the analysis of paths from flip-flops to flip-flops. Two assumptions are made:

- Inputs to the circuit originate from synchronous flip-flops, although we can't "see" them
- Outputs from the circuit terminate at synchronous flip-flops, although we can't "see" them

With these assumptions, it is possible to analyze all the paths that exist. However, for paths involving inputs and outputs, the fact that we can't "see" everything complicates analysis because it may be necessary to account for additional delays.



Figure 5: Common FSM Structure

Although we assume the inputs to the circuit are originating at flip-flops, there may be combinational logic delay and wire delay on the paths. Similarly, although we assume the outputs from the circuit are terminating at flip-flops, these paths may also have additional delay. To account for these possibilities, four different path types exist:

- 1. Clock (period) constraints, covering paths from flip-flops in the design to flip-flops in the design.
- 2. Input delay constraints, covering paths from flip-flops outside the design which we cannot "see", through inputs, to flip-flops in the design.
- 3. Output delay constraints, covering paths from flip-flops in the design, through outputs, to flipflops outside the design which we cannot "see".
- 4. Path delay constraints, covering paths from flip-flops outside the design which we cannot "see", through inputs to outputs, to flip-flops outside the design which we cannot "see".

As shown in Figure 6, expand the available constraint types in the Timing Constraints window and double click on Create Clock. A series of dialog boxes begin to guide you through the creation of the clock constraint.

Iab3 - [C:/Projects/Iab3/Iab3.xpr] - Vivado 20' Ela Edit Elan Taola Wardan Langut Via	16.2		0.000	- 0 ×
Image: A market of the second sec	🐑 👷 🖉 🕲 🕲 🛞 💥 ∑ 🔯 🕾 Default Layout 🔹 🛒	<u>ک ک</u>	M [*] search	Synthesis Complete
Flow Navigator ? «	Synthesized Design - xc7a35tcpg236-1 (active)			? ×
۹. 🞞 🗰	Netist ? _ D & ×	E Project Summary X @ Device X A Timing	a Constraints X	0 C ×
Project Manager	X H	Z # 18. 64	Create Clock	
® Project Settings ⊗ Add Sources ♥ Language Templates ₽ IP Catalog	uil fam → Contest (21) → Contest Color (17) → Contest Color (17)	Codds (0) Costs Clock (0) Costs Clock (0) Costs Clock (0) Set Clock Latency (0) Set Clock Latency (0) Set Clock Coupe (0)	Person Clock Name Period (m) Rise AL (m) Fell AL (m) Add Clock Source Objects Source File Scoped Cel Current Instance Could cick to create a Create Clock constraint	
IP Integrator Create Block Design Provide Block Design Provide Block Design Generate Block Design		Set Clock Sense (0) Set Input Atter (0) Set External Delay (0) Set External Delay (0) Set External Delay (0)		
Simulation Simulation Settings Que Run Simulation		Output (0) -Set Output Delay (0) -Set Output Delay (0) -Assertions (0) -Set Data Check (0) -Exceptions (0) -Set Check (0)		
RTL Analysis Rabyzating Settings		-Set False Path (0)		
Open Elaborated Design	& Sources 😫 Netlist	All Constraints		
Report Methodology	Descention 2 D v3 Y	Q Destine Comment	curve de la	
Report DRC		Command	Scopes Cer	
Report Noise		cate		
🚰 Schematic				
4 Synthesis		τ.		
🚯 Synthesis Settings		1		
Run Synthesis	Select an object to see properties	-		
4 🔛 Synthesized Design				
S Constraints Wizard				
Set Up Debug			Apply Cancel	
Report Timing Summary	Tcl Console			? _ 🗆 🖻 ×
Report Clock Hetmorks Report Clock Interaction Report Dick Interaction Report Dick Report Dick Report Dick Report Utilization Report Utilization Report Power Schematic	DITO: [NetList 27-22] Thisli Transformation complete DITO: [NetList 27-22] Thisli Transformation complete DITO: [NetList 27-22] Thisle was created with MV DITO: [NetList 27-22] Thisle was	<pre>ted in 0 CFU seconds ado 2014.2 optimization rrai/resulfam.xdc] rrai/comstre_l/new/fsm.xdc] yt </pre>		*
 Implementation 	C Press of Tail assessed have			>
Contraction Settings	Tcl Console C Messages C Log Reports Design Runs			
Create Clock				

Figure 6: Create Clock – Constraint Creation

The first dialog asks you to provide a friendly nickname for your clock. You will later need to identify where the clock is coming from and sometimes that name be unfriendly or unwieldy, so the opportunity to create a nickname is useful.

In addition to the nickname, the clock waveform parameters need to be specified. A 100 MHz clock with a 50% duty cycle rises at 0 ns, falls at 5 ns, and has a total period of 10 ns. The choice of clock waveform in this exercise is arbitrary, but in a real design it would be based on the actual clock needed for the implementation. Copy the settings shown in Figure 7, and then click on the "…" to specify the source objects.

Creates a clock source objects,	object. The created clock is applied to t but give a clock name, a virtual clock is	he specified source objects. If you do not spe created.	cify
Clock <u>n</u> ame:	my_only_clock		e
ource objects:			
Naveform			
Period:	10 🜩	ns	
<u>R</u> ise at:	0 🖨	ns	
Eall at:	5 🜩	ns	

Figure 7: Create Clock - Provide Name and Waveform, then Identify Source

In this design, the clock signal is coming from a top level port, so it happens to have a friendly name and is easy to locate. However, this might not always be the case. For this design, though, use the settings shown in Figure 8 to find design ports that are inputs.

ecify the po	rts, pins, or nets which	are the source of the	specified	COCK.	
in <u>d</u> names o Options	of type: I/O Ports 🔻]			
	NAME	▼ contains	Ŧ	•	
AND -	DIRECTION	▼ is	Ŧ	IN - +-	
<u>R</u> egular o	expression Ignore	: case			
<u>R</u> egular (<u>)</u> f Objects: esults	expression Ignore	case		End	
<u>R</u> egular (2f Objects: esults =ound: 0	expression Ignore	case	[End Selected: 0	 ↓ <u>2</u>
_ <u>R</u> egular ı 2f Objects: esults =ound: 0 C	Expression Ignore	case	# # +	End Selected: 0 Use the buttons on the left to move items to this list.	

Figure 8: Create Clock – Find Ports

After you click Find, the results should list all the input ports to the design as shown in Figure 9. You need to select the design clock, clk, from the list of ports.

	or a, pina, or nets whi	ch are the :	source of the specifi	d dock.	
in <u>d</u> names ptions	of type: I/O Ports	•			
	NAME	•	contains	• • 6	3 + -
AND 🔻	DIRECTION	Ŧ	is	• IN •	+-
<u>R</u> egular	expression Ignor	re case			
Begular f Objects:	expression Ignor	re case		End	
Regular If Objects: esults iound: 4	expression Ignor	re case		Eind Selected: 0	13
Regular f Objects: esults iound: 4 ik bause estart st	expression Ignor	re case		End Selected: 0 Acves the selected items to the right to move	L2 e items to this list.

Figure 9: Create Clock – Select Port Sourcing Clock

Promote the clock port to the list of selected objects by clicking on the button with the arrow pointing to the right. The result should look like Figure 10. Then, set the selection to return to the previous dialog.

city the po	orts, pins, or nets whic	ch are the source of the :	pecified	ск.	
n <u>d</u> names o otions	of type: I/O Ports	·			
	NAME	▼ contains	Ŧ	⊗ + -	
AND 🔻	DIRECTION	▼ is	Ŧ	N + -	
] <u>R</u> egular	expression Ignor	e case			
] <u>R</u> egular f Objects: sults	expression Ignor	e case		Eind	
] <u>R</u> egular f Objects: sults pund: 3	expression Ignor	e case		Eind	13
] <u>R</u> egular f Objects: sults sund: 3 ause estart st	expression Ignor	e case		End ected: 1	
] <u>R</u> egular f Objects: sults ound: 3 ause estart st	expression Ignor	e case	# # +	End ected: 1	

Figure 10: Create Clock - Add Port to Selection List and Accept

When you return to the dialog shown in Figure 11, no further editing is necessary. Review the settings for consistency, and notice the "command" shown at the bottom of the dialog. This is the actual constraint that is generated from the settings. There is no need to copy and paste it; Vivado will add all your new constraints into the XDC file when we have completed constraint entry. Accept the constraint, which will return you to the window shown previously in Figure 6.

ource objects, l	object. The created clock is applied to the out give a clock name, a virtual clock is (e specified source objects. If you do not specify reated.
llock <u>n</u> ame:	my_only_clock	6
ource objects:	[get_ports dk]	8
Vaveform		
Period:	10 🜩	ns
Rise at:	0 🜩	ns
Fall at:	5 🜩	ns

Figure 11: Create Clock – Accept Constraint

Next, create an input delay constraint. We are going to create a single constraint for all inputs, although it is possible to create different constraints for each input. The input delay constraint dialog is shown in Figure 12.

Specify input delay for por	ts or pins relative to a clock edge.		
Clock:			
Delav value:	0	ns	
Delay Value Options			
Delay value is <u>r</u> elative De <u>l</u> ay value already ir Rise/Fall	to clock edge:	rise v None v	
Delay value	specifies rising 👻 delay		
Min/Max			
Delay value	specifies min 🔻 delay (shortest path)		
Add delay inform	ation to the existing delay (no overwrite)		
	som of the existing deay (no over mile)		

Figure 12: Input Delay – Identify Clock, Ports, and Specify Delay

First, click on the "…" to the right of the clock name textbox. This allows you to search for the design clock related to the input delay constraint. Figure 13 shows how to find clocks in your design, you should only find one, it will be the clock you identified while creating the clock constraint, and it will be listed by its nickname. Promote the clock to the selection list and accept the selection. You will return to the main input delay dialog.

acify the clock that the delay is re	lative to			
eery the clock that the delay is re	auve to.			
ind names of type: Clocks 💌				
NAME 👻	contains 👻 *		⊘ +	
<u>R</u> egular expression <u>Ignor</u>	e case			
Regular expression Ignor	e case			
Regular expression Ignor	e case	Eind		
Regular expression Ignor	e case	Ejnd		
	e case	Eind Selected: 1		
_ Regular expression Ignor f Objects: esuits ;ound: 0	e case	Eind Selected: 1		12
Regular expression Ignor f Objects: esuits found: 0 Use the buttons on the right	e case	Eind Selected: 1		12
Regular expression Ignor f Objects: esuits found: 0 Use the buttons on the right ist.	e case	Eind Selected: 1		12
Begular expression Ignor f Objects: esults iound: 0 Use the buttons on the right: list.	e case	End Selected: 1 my_only_dock		19
Begular expression Ignor f Objects: suits found: 0 Use the buttons on the right: list. ommand: Get docks ***	e case	Eind Selected: 1 my_only_dock		

Figure 13: Input Delay – Identify Clock and Accept

Next, click on the "…" to the right of the objects name textbox. This allows you to search for the design ports related to the input delay constraint. Figure 14 shows how to find ports in your design, you should select pause and restart ports only, and promote them to the selection list. Accept the list of selected ports.

Although the clk port is listed in the port search results, it is not (usually) a data signal so you would not want it in the list of selected ports.

The rst port is also listed in the port search results, it is an asynchronous reset. If we assume the source of it is truly asynchronous, we can ignore the timing on this signal because we have no idea when it will assert or for how long it will assert, and it won't be possible to guarantee any particular behavior through a timing constraint. Note, however, that sometimes designers drive asynchronous control signals from synchronous sources such as flip-flops, and may wish to analyze the timing by adding a constraint.

👃 Specify De	elay Objects					×
Specify the list	of ports to which the delay val	ue will be assigned.				2
Fin <u>d</u> names o Options	f type: I/O Ports 🔻					
	NAME	contains 🔹	*	8	+-	
AND 🔻	DIRECTION -	is 👻	IN 🔻		+-	
☐ Regular e Of Objects: Results	xpressionIgnore case		Eind			п
Found: 2			Selected: 2			↓ <u>₽</u>
rst		t + + +	pause restart			↑ ↓ ↑
<u>C</u> ommand:	get_ports {pause restart}				8 🗆 E	numerate
				Set	Append	Cancel

Figure 14: Input Delay – Identify Ports and Accept

When you return to the dialog shown in Figure 15, specify the delay value as 5 ns, as shown. This delay value is how you express the delay that cannot be "seen" from the source flip-flops outside the design up to the input ports of the design. The remainders of the paths from the input ports to flip-flops inside the design are visible to the timing analysis tool. The 5 ns is arbitrary, but a fair sharing of the 10 ns clock period between our design and some other invisible design that cannot be "seen".

Review the settings for consistency, and notice the "command" shown at the bottom of the dialog. This is the actual constraint that is generated from the settings. There is no need to copy and paste it; Vivado will add all your new constraints into the XDC file when we have completed constraint entry. Accept the constraint, which will return you to the window shown previously in Figure 6.

🍌 Set Input De	lay	×
Specify input dela	ay for ports or pins relative to a clock edge.	4
<u>Clock:</u>	[get_clocks "*"]	
Objects (ports):	[get_ports {pause restart}]	
Delay <u>v</u> alue:	5 🜩 n:	s
Delay Value Opti	ons	
Delay value	is <u>r</u> elative to clock edge:	rise 👻
Deļay value Rise/Fall	already includes latencies of the specified clock:	None 💌
Del	lay value specifies rising 👻 delay	
Min/Max		
Del	ay value specifies min 👻 delay (shortest path)	
Add del	av information to the existing delay (no overwrite)	
	-,	
Command: set	t_input_delay -clock [get_clocks "*"] 5.0 [get_ports -	{pause restart}]
Re <u>f</u> erence	Reset to Defaults	OK Cancel

Figure 15: Input Delay – Specify Delay and Accept

Next, create an output delay constraint. We are going to create a single constraint for all outputs, although it is possible to create different constraints for each output. The output delay constraint dialog is shown in Figure 16.

Specify output delay	for ports or pins relative to a clock edge.			1
<u>D</u> lock:				
Objects (ports):				
Delay <u>v</u> alue:	0 🜩	ns		
Delay Value Options				
Delay value is <u>r</u> el	lative to clock edge:	rise	*	
Deļay value alrea	ady includes latencies of the specified clock:	None	Ŧ	
Rise/Fall				
Delay v	alue specifies rising v delay			
Min/Max				
Min/Max	alue specifies min 👻 delay (shortest path)			
Min/Max	alue specifies min 👻 delay (shortest path)			
Min/Max Delay v Add delay in	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	value specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			
Min/Max	alue specifies min v delay (shortest path) formation to the existing delay (no overwrite)			

Figure 16: Output Delay – Identify Clock, Ports, and Specify Delay

First, click on the "…" to the right of the clock name textbox. This allows you to search for the design clock related to the output delay constraint. Figure 17 shows how to find clocks in your design, you should only find one, it will be the clock you identified while creating the clock constraint, and it will be listed by

its nickname. Promote the clock to the selection list and accept the selection. You will return to the main output delay dialog.

Specify Clock			×
ipecify the clock that the delay is relative to.			À
Fin <u>d</u> names of type: Clocks 👻 Options			
NAME • Contains • (*		6 +	
Regular expression Ignore case Of Objects:			
Results	Eind		
Found: 0	Selected: 1		↓2
Use the buttons on the right to move items to this list.	↓		* * *
Command: get_docks ***			C Enumerate
		ОК	Cancel

Figure 17: Output Delay – Identify Clock and Accept

Next, click on the "…" to the right of the objects name textbox. This allows you to search for the design ports related to the output delay constraint. Figure 18 shows how to find ports in your design, you should select all the output ports, and promote them to the selection list. Accept the list of selected ports.

ecity the lis	t of ports to which th	ie delay valu	e will be assigned	Ι.		i
in <u>d</u> names i iptions	of type: I/O Ports	•				
	NAME	•	contains	*	© + -	
AND -	DIRECTION	-	is	▼ OUT ▼	+-	
Degular						
<u>R</u> egular Objects: esults	expression Igno	ore case		Ejnd		
<u>R</u> egular of Objects: esults ound: 0	expression Igno	ore case		Eind Selected: 5 even		↓ <u>2</u>
<u>R</u> egular of Objects: esults found: 0 Use the list.	expression I Igno	t to move iter	ms to this	Eind Selected: 5 ↔ even odd state[0] state[1] terminal		12
<u>Regular</u> of Objects: esults cound: 0 Use the list.	expression I Igno	t to move iter	ms to this	Eind Selected: 5 odd state[0] state[1] terminal t		<u>\$</u>

Figure 18: Output Delay – Identify Ports and Accept

When you return to the dialog shown in Figure 19, specify the delay value as 5 ns, as shown. This delay value is how you express the delay that cannot be "seen" from the output ports of the design up to the flip-flops outside the design. The remainders of the paths from the flip-flops inside the design to the output

ports are visible to the timing analysis tool. The 5 ns is arbitrary, but a fair sharing of the 10 ns clock period between our design and some other invisible design that cannot be "seen".

Review the settings for consistency, and notice the "command" shown at the bottom of the dialog. This is the actual constraint that is generated from the settings. There is no need to copy and paste it; Vivado will add all your new constraints into the XDC file when we have completed constraint entry. Accept the constraint, which will return you to the window shown previously in Figure 6.

Specify output d	Jelay for ports or pins relative to a clock edge.	4
<u>Clock:</u>	[get_docks "*"]	8
Objects (ports):	[get_ports -filter { NAME =~ "*" && DIRECTION == "OUT" }]	8
Delay <u>v</u> alue:	5 🖨 ns	
Delay Value Opti	tions	
Delay value	e is <u>r</u> elative to clock edge:	
Delay value	e already includes latencies of the specified clock: None 💌	
Rise/Fall		
	elav value specifies rising 💌 delav	
Min/Max		
De	elay value specifies min 👻 delay (shortest path)	
	lay information to the eviation delay (as suspential)	
- Add dei	say information to the existing delay (no overwrite)	
-	I I I I I I I I I I I I I I I I I I I	Internet Internet
Command: pu	ut_delay -clock [get_clocks ***] 5.0 [get_ports -filter { NAME =~ *** && DIF	RECTION == "OUT" }]

Figure 19: Output Delay – Specify Delay and Accept

At this point, you should see the three new constraints you have entered, listed under All Constraints in the Timing Constraints window, as shown in Figure 20. Below the main menu bar, you will find a small disk icon to save your constraints; click on it.



Figure 20: Save Timing Constraints

As you have created constraints, the synthesis results may now be out-of-date, and Vivado will display an Out of Date Design warning. This is because some constraints have an effect on the synthesis results. For this exercise, we will completely re-run synthesis to ensure all constraints have been applied during synthesis. Dismiss this message, but accept any subsequent message to confirm saving of the new constraints.

At this time, close the synthesized design by clicking on the "X" in the light blue Synthesized Design title bar.

Reset Design Runs, Re-Run with Constraints

Open the fsm.xdc constraint file in the text editor. You originally added a blank file, now it is populated with the constraints you applied. Add the following additional constraints manually and then save the file:

```
set_property CFGBVS VCCO [current_design]
set property CONFIG VOLTAGE 3.3 [current design]
```

The completed constraints should look similar to Figure 21. In the bottom window, select the Design Runs tab; right click on the synthesis run. The context menu provides an option to Reset Runs. Reset the synthesis run. Resetting the synthesis run allows you to re-run it from scratch.

🍌 lab3 - [C:/Projects/lab3/lab3.xpr] - Vivado 2	016.2		– ø ×
Elle Edit Flow Tools Window Layout V	jew Help		Q,+ Search commands
😂 i in 🕫 🐚 🦍 🗡 🔈 隆	😽 ∑ 🧔 🗄 Default Layout 🔷 🛒 🐥 🍾 🖏		Synthesis Out-of-date mice.info
Flow Navigator ? «	Project Manager - lab3		? ×
🔍 🎞 📾	Sources ? _ 🗆 🖒 ×	E Project Summary X fism.xdc X	0 2 ×
A	🔍 🎞 🛱 📾 🐮 🗎 🛃	C:/Projects/lab3/lab3.srcs/constrs_1/new/fsm.xdc	
Project Hanager	Design Sources (1)	1set_property IOSTANDARD LVCMOSS3 [get_ports clk]	^
Controject betangs		2 set_property PACKAGE_PIN WS [get_ports clk]	
Acto sources	⊖ constraints (1) ⊖ constra_1 (1)	Set property IOSIANDAND LACHOSS (ge_ports (state[1])) a det property IOSIANDAD LACHOSS (get_ports (state[1]))	
	fsm.xdc (target)	5 set property IOSTANDARD LVCMOS33 [get ports even]	
gr ir Calaby	⊕ in findatori sources (1) ⊕ in sim 1 (1)	6 set_property IOSIANDARD LVCMOSS3 [get_ports odd]	
# IP Integrator		<pre>[4] 7 set property IOSTANDARD LVCMOS33 (get_ports pause)</pre>	
🕂 Create Block Design		Set property IOSTANDAD LYCHOSS (ge_ports retart)	
Dpen Block Design		// 10 set property IOSTANDARD LVCMOS33 [get_ports terminal]	
Senerate Block Design		11 set_property PACKAGE_PIN U3 (get_ports (state[1]))	
		3 12 set property PACKAGE PIN U2 (get_ports (state[0]))	
 Simulation 		V Mast property PACKAGE VIN 83 (get ports even)	
Simulation Settings		15 set property PACKAGE PIN T1 (get_ports pause)	
Run Simulation		16 set_property PACKAGE_FIN T2 [get_ports restart]	
4 PD Andurie	Nierarchy Ubraries Comple Order	17 set property PACKAGE PIN T3 (get_ports rst)	
Cit Makyas		10 sec propercy switching of [dec borcs cemines]	
4 Chara Enhanted Design	Synthesis Run Properties ? _ C L* ×	20 create_clock -period 10.000 -name my_only_clock -waveform (0.000 5.000) [get_ports clk]	
		<pre>21set_input_delay -clock (get_clocks *] 5.000 (get_ports {pause restart)}</pre>	
C Report Methodology	v≰ synth_1	22 set_output_delay -clock (get_clocks *] 5.000 [get_ports -filter { NAME "*" ss DIRECTION "COT" }]	
M REPORT DRC	Name and 1	24 set property CFGWS VCCD [current design]	
Crill Schematic	Baue. Bhun't	25 set property CONFIG VOLTAGE 3.3 [current design]	
# Synthesis	Part: @xc7a35tcpg236-1 (active)	26	
🚯 Synthesis Settings	Description: Vivado Synthesis Defaults		
Run Synthesis	Status: Synthesis Out-of-date		
4 😴 Open Synthesized Design	Constraints: a constrs_1 (active) -		
So Constraints Wizard	Run directory: C:/Projects/lab3/lab3.runs/synth_1		
A Edit Timing Constraints	Synthesis Run Properties Ctrl+E		
Set Up Debug	X Delete Delete		
Report Timing Summary	General P Make Active		, ×
Beport Clock Networks	Change Run Settings		
Beport Clock Interaction	Design Runs Save As Strategy		? _ 🗆 🕑 ×
C. Report Methodology	Name Open Run	WNS TNS WHS THS TPWS PaledRoutes LUT FF BRAM URAM DSP Start Elapsed Strategy	Part Description
Report DBC	Launch Runs	Fidate 2 2 0 0 0 9/17/16 10:00 AM 00:00:38 Wivado Synthesis Defaults (Vivado 5	ynthesis 2016) xc7a3Stcpg236-1 Wvado Synthesis Default
Report Literation	N Reset Runs	Wado Imperientation Defaults (W	ado implementation 2016) xc7a35tcpg236-1. Default settings for impl
(D) Papert Preser	Launch Next Step		
Commenter	Reset to Previous Step: synth_design		
	Porce Up-to-Date		
4 Implementation	4 Generate Ristream		
🚯 Implementation Settings	-> Denivel on		
Run Implementation	Cie Unspikey Log		
Open Implemented Design	Tri Car Display Messages		,
~	Create Dury	***	
Reset the selected run(s)	Oran Run Directory		
	open tel secold yn		

Figure 21: Inspect XDC File, Reset Design Runs

When you select to reset a design run, you are prompted for confirmation. Accept the confirmation request and be sure to delete the generated files in the working directory.

In the Flow Navigator, click on Run Implementation. Since you have reset the design run, this will re-run all of the design flow steps up through implementation. When the implementation run has completed, open the implemented design. You can review the reports later, if you wish.

Reviewing Implementation Results

The initial presentation of the timing results for the implemented design, shown in Figure 22, has a navigation tree on the left and a high level summary on the right. Expand the navigation tree, keeping the Design Timing Summary selected. Failures are indicated by the use of red text. The navigation tree indicates that failures exist on intra-clock paths of the declared clock signal my_only_clock, and those failures are setup failures.

Positive slack on a timing constraint is the portion of the constraint that has no "strain" on it, indicating the actual timing of the path being constrained exceeds the requirement by some amount. This is generally what you would consider a satisfied or passing constraint. If you achieve positive slack on all constraints, it is called timing closure.

Negative slack, on the other hand, is generally undesired and indicates the path being constrained is not meeting the requirement by some amount. If you have more than one path under analysis, the term worst negative slack (sometimes referred to as WNS) is the negative slack of the greatest magnitude and may be a useful guide to where the design needs work to achieve timing closure. The term total negative slack (sometimes referred to as TNS) is the sum of all negative slacks in the design, and may be a useful guide to how much the design needs work to achieve timing closure.



Figure 22: Timing Summary, Constraints Failing

As shown in Figure 23, in the navigation tree under Intra-Clock Paths for my_only_clock, select the failing Setup category. This will display a list of paths in this category. The list of paths may be sorted by several characteristics. By default they are sorted in ascending order of slack (meaning, large negative slacks at the top of the list, because these show where the design needs improvement).

Also notice, in Figure 23, that selection of a path in the list generates arrows on the device view to provide insight into the physical path. You can zoom and scroll to look at these arrow diagrams more closely. When you are evaluating why a path is failing its constraint, understanding its physical path may provide insight into the failure – or it may not. Please note your path may not match what is shown in the figure.

🍌 lab3 - [C:/Projects/lab3/lab3.xpr] - Vivado 21	016.2												– ø ×
Ele Edit Flow Tools Window Layout V	iew Help											Q	iearch commands
😅 🔟 1 to 🕫 📴 🏢 🗙 🍕 🖻 🕨	🌯 🖽 🥝 🗘 🥝 🗞 💥 🖸 🦉 🔛 🕬	ultLayout 👻 🎉 🧃	k 🖹 🤤									Implement	ation Complete, Failed Timing
Flow Navigator ? «	Implemented Design - xc7a3Stcpg236-1 (active)											? X
의 🎞 🗰	Netlist	? = 🗆 🖻 ×	D Project Summa	ry 🗙 🍥 De	vice 🗙 🗈 fsm.xdc 🗙								? 🗆 🖻 ×
Project Manager Optimized College	i Si Mi	^											^
Add Sources	Image: Berney State (21) </th <th></th> <th>And the st</th> <th></th>											And the st	
Language Templates IP Catalog	- Cli ck_IBUF_inst (IBUF) - Cli even_OBUF_inst (OBUF) - Cli even_OBUF_inst i 1 (LUT2)		ু নায় 🕻	6								citi Da	
IP Integrator Create Flock Derive	- dd_OBUF_Inst (CBUF) - dd_pause_IBUF_Inst (IBUF)											PAD	
Den Block Design	rst_IBUF_inst (IBUF)	~										1083310X1V460	
Simulation Gisculation Settings	Path Properties	? = 🗆 🕹 ×	الله الله الله الم والم الله الله الم المالية الم	0 0 0 0 0 0 0 0	2000 0000								
Run Simulation	Path 1		*										
 RTL Analysis 	Name Path 1	î					/					T1	
Baboration Settings	Slack <u>-7.785ns</u>	ad human and a dash. faire	🦉 🕹 🖥 🚺			/	-					2401	
Open Baborated Design	Destination I terminal (output port di	cked by my_only_dock {r	🔁 🖻 🔋 🛙	1~64	3 🛯 🖆 8							10833 (X1744)	
Report Methodology	Path Group my_only_dock		×	<u>ا ت ا</u>									
Report brise	Requirement \$0.000ns (my_only_clock ris	e@10.000ns - my_only_ck		0 0 0								U1	
Schematic	Data Path Delay 7.750ns (logic 5.090ns (65.)	674%) route 2.660ns (34.		2 4				_					
	< C C C C C C C C C C C C C C C C C C C	>		김 영영 중								(SE33(X)	
Synthesis	General Properties Report Cells Nets Net	Segments Options	<						,				э К
Dun Cunthasir	Timing - Timing Summary - ingl 1												2 0 2 1
Open Synthesized Design		Intra-Clock P	the area only clock	- Sahan									
S. Constraints Wizard	(i) This is a same report	y C Mana	o 1 Lavala	High Execut	From To	Total Dalay	Logic Dalay	Net Delay	Dan inement Service Clark	Destination Clock	Europeiro		
📇 Edit Timing Constraints	- General Information	2 Roth 1	-7.785	ngirranos	3 restart terminal	7,750	5.090	2.660	10.000 my only dod	my only dock	Encepton		
🐳 Set Up Debug	-Timer Settings	Path 2	-6.491 2		5 state_reg(0)/C even	6.298	4.091	2.207	10.000 my_only_ded	my_only_clock			
Report Timing Summary	-Design Timing Summary	Path 3	-6.109 1		6 state_reg[1]/C state[1]	5.915	3.960	1.956	10.000 my_only_dod	c my_only_clock			
Report Clock Networks	Cook Summary (1) B-B Check Timing (1)	Path 5	-6.043 1		6 state_reg[1]/C odd	5.849	3.975	1.875	10.000 my_only_dod	my_only_dock			
Report Clock Interaction	Intra-Clock Paths	Path 6	5.036 2		3 restart state_reg[1]/D	1.303	0.467	0.836	10.000 my_only_dod	my_only_clock			
C Report Methodology	my_only_dock	Path 7	5.090 2		3 restart state_reg[0]/D	1.255	0.466	0.789	10.000 my_only_dod	k my_only_dock			
Report DRC	-Hold 0.686 ns (7)												
Report Noise	-Pulse Width 4.500 ns (11)												
Report Utilization	Inter-Clock Paths												
(Report Power	-User Ignored Paths												
Schematic	- Unconstrained Paths												
 Implementation 	Timing Summary - impl_1												4 0 00
Implementation Settings	🔠 Tcl Console 🛛 🗇 Messages 🔂 Log 🔒	Reports 🔑 Package Pins 🚯	Design Runs 🛛 🗐 Pi	mer 🍐 Tim	ing								
h													

Figure 23: Setup Paths, Worst Offender, Graphical Illustration

Double click on a path in the timing summary list to open a detailed analysis of that path. Figure 24 provides an example of a detailed analysis. We will discuss these reports in class.



Figure 24: Worst Offender, Timing Detail

Laboratory Hand-In Requirements

Once you have completed this exercise, prepare for the submission process. Within four hours of the scheduled class time on the due date, you are required to submit your entire project directory in the form of a compressed ZIP archive. Use WinZIP to archive the entire project directory, and name the archive lab3_nguyen_tan.zip. For example, if I were to make a submission, it would be lab3 crabill eric.zip. Then email the archive to the instructor. Only WinZIP archives will be accepted.