# The UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

### Comp 541 Digital Logic and Computer Design Fall 2014

## Lab #1: Getting Started

Issued Wed. 8/20/14; Due Wed. 8/27/14 (11:59pm)

This lab assignment consists of two parts. For the first part, detailed instructions, including screenshots of almost every step, are provided. Your task is to follow the instructions carefully, and make sure that your software is correctly installed and working properly. You will create a simple project, specify a simple design in the Verilog hardware description language, and simulate it using various inputs. In the second part, you will make a small modification to the design to add an extra output, re-simulate it, and show your results. Detailed instructions are provided below.

## PART 0: Software Installation

Please follow the link on the class website to download the Xilinx ISE WebPack from the Xilinx website. You will need to register with a user ID and password. Download the file (.tar), and uncompress it (if you need WinZip to uncompress it, you can obtain it for free from shareware.unc.edu). Double click on the file *xsetup* to launch the installation wizard, and follow on-screen prompts.

Once installation is complete, create a shortcut to the Project Navigator on your desktop, so it is easier to launch it. Following these steps:

Start Menu  $\rightarrow$  All Programs  $\rightarrow$  Xilinx ISE Design Suite 14.7  $\rightarrow$  ISE Design Tools  $\rightarrow$  Project Navigator (right click)  $\rightarrow$  Send to  $\rightarrow$  Desktop (create shortcut).

### Licensing

You will need the free ISE WebPack license. Follow these steps:

- First, make sure that Internet Explorer is set to be your default browser (you can change it, and restore it to your favorite browser after you have downloaded and installed your license file).
- Launch the Project Navigator by clicking on the desktop shortcut you just made. (Dismiss the Tip of the Day.)
- Obtain a license file by following either of these two steps:
  - Help  $\rightarrow$  Obtain a License Key  $\rightarrow$  Get Free ISE WebPack License; OR
  - Use your browser to go to <u>www.xilinx.com</u>  $\rightarrow$  Support  $\rightarrow$  Downloads and Licensing  $\rightarrow$  Get License File.
- Log in with your User ID and Password, and verify your contact information. Then, go to *Create a New License File* → check the box for *ISE WebPack License*, and follow on-screen instructions.
- The license file will be emailed to you as an attachment (*Xilinx.lic*). Save it to your desktop or downloads folder. Go back to the Project Navigator, and launch the License Manager (click *Help* → *Manage License*), click *Copy License*, browse to the license file you just downloaded from your email, select it and click *Open*. Now your copy of the software should be appropriately licensed, and you can start using it. Congratulations!

**Troubleshooting:** In case of problems with installing the software, submit a post to the Piazza discussion board. It is likely that other students are encountering the same issue. Seek help early instead of the day before the assignment is due!

## PART I: TUTORIAL

You are going to create a 3-input combinational circuit, with 1 output. The circuit is a single-bit *Full Adder*, with inputs *A*, *B* and *Carry*<sub>in</sub> (*Cin*), and the output is *Sum*. Note that a full adder has a second output, *Carry*<sub>out</sub>; you will ignore it in this part, but add it in Part II.

The procedure will be to do the following:

- Create a new project
- Add a Verilog file that describes the circuit
- Add a Verilog file that provides the inputs
- Simulate the circuit
- Observe the outputs and verify they are correct

For your reference, here is the circuit and Boolean equations for a Full Adder (from Comp411).



$$C_{out} = C_{in} (A \oplus B) + AB$$
$$Sum = C_{in} \oplus A \oplus B$$

## **Creating a Project**

Start by double-clicking on the Project Navigator icon on your desktop. This is what you will see.



You can close the InfoCenter window by clicking on the X below it.

To create a new project, select *File*  $\rightarrow$  *New Project*, which will bring up the following window:

> New Project	Wizard	×
Create New Pro	ect and type.	
Enter a name, locati N <u>a</u> me:	ns, and comment for the project	
Location: Working Directory:	C: (Users C: (Users	
Description:		
Select the type of to	p-level source for the project e:	
HDL		
More Info		Next Cancel

**Choose a project name.** This will also become a subdirectory of the location you choose next. Call it, say, Lab1.

Project Location: Browse and change the folder to some convenient place.

<u>NOTE:</u> Make sure there are **NO** spaces in the name and the path! This has caused a problem in the past.

Choose HDL (Hardware Description Languages).

New Project	Wizard	-
Create New Proj	iect	
Specify project location	and type.	
Enter a name, location	ons, and comment for the project	
Name:	Lab1	
Location:	D:\Comp541\Lab1	
Working Directory:	D:\Comp541\Lab1	2**
<u>D</u> escription:	1	
Select the type of to <u>T</u> op-level source typ HDL	p-level source for the project e:	
More Info		Next Cancel

# Click Next.

Sector Catting		
roject settings		
pecify device and project properties.		
elect the device and design now for the pr	oject	
Evaluation Development Board	None Specified	-
Product Category		
Family	Snartan3E	
Device	XC351200E	
Package	FG320	
Speed	-4	
Top-Level Source Type	HDI	
Synthesis Tool	XST (VHDL/Verilog)	
Simulator	ISim (VHDL/Verilog)	
Preferred Language	Verilog	
Property Specification in Project File	Store all values	
Manual Compile Order		
VHDL Source Analysis Standard	VHDL-93	

Click Next and then Finish.



You have now created a new project, although an empty one. You should see this:



# Adding a new Verilog source file.

In the toolbar to the left, click the top icon (which says "New Source" if you hover over it). Alternatively, right-click on *xc3s1200e-4fg320*, and choose "New Source").

New Source Wizard	×
Select Source Type Select source type, file name and its location. IF (CORG Generator & Architecture Wizard) Schematic System Generator Project User Document Verilog Module Verilog Module VHOL Library VHOL Library VHOL Package VHOL Test Bench Embedded Processor	Ejle name: lab1_part1 Logaton: D:\Comp541\Lab1 
More Info	Next Cancel

On the left panel, select Verilog Module, and fill in the file name as shown. Click Next.

New Source Wizard						
Define Module						
Specify ports for module.						
Module name lab1_part1						
Port Name	Directio	on	Bus	MSB	LSB	-
A	input	-				
В	input	•				
Cin	input	-				
Sum	output	-				
	input	•				
	input	•				
	input	•				
	input	•				
	input	•				
	input	•				
	input	-				

Specify the inputs and outputs and click Next, then Finish.

	Nev	v Source W	izard				×
	Summary Project Navig Add to Project Source Direct Source Name: Module name: Port Definition	/ ator will creat :Yes :Yes CComp :Alb1_part1. lab1_part1. lab1_part1. B S: A B Cin Sum	ate a new skeleton 0541)(ab1 le V Pin Pin Pin Pin Pin	source with the foll	input input input input output	ns.	
(	More Info					Einish	Cancel

The editor opens and shows you a skeleton of a Verilog file. Fill in the code shown:

```
module lab1_part1(
input A,
input B,
input Cin,
output Sum
);
```

assign Sum = Cin ^ A ^ B;

endmodule

Note that '^' is the Verilog symbol for the exclusive-or (XOR) operation.

ISE Project Navigator (O.87xd) - D:\Comp541\Lab1\L	ab1.xise - [lab1_part1.v]	
<u>File Edit View Project Source Process</u>	ools <u>W</u> indow La <u>v</u> out <u>H</u> elp	_ & ×
🗠 📬 🗶 🗊 🖓 👹 📲 🐔 🚺	» 🏓 🔎 🛞 🖉 🖂 🦳 🖻 🗖 🗁 🥬 🛠 🕨 🗴 🛠	
Design       ↔ □ ♂ ×         View:       ● □ molementation         Implementation       ● □ molementation         Hierarchy       □ □ molementation         □ □ molementation       ● □ molementation         □ □ molementation<	<pre>1 'timescale ins / ips 2 ////////////////////////////////////</pre>	
No Processes Running	<pre>12 // Description: 13 // 14 // Dependencies: 15 // 16 // Revision: 17 // Revision 0.01 - File Created 19 // Additional Communication</pre>	Е
Processes: lab1_part1       Processes: lab1_part1       Design Summary/Reports       Design Utilities       Design Utilities       Design Otimities       Design Otimities    <	<pre>19 // 20 ////////////////////////////////</pre>	
🍃 Start 🚉 Design 🖺 Files 🚺 Libraries 📑	lab1_part1.v 🔀 Design Summary 🗵	
Console		↔□♂×
↓ INFO:HDLCompiler:1845 - Analyzing V ↓ INFO:ProjectMgmt - Parsing design r	erilog file "D:/Comp541/Lab1/lab1_part1.v" into library work ierarchy completed successfully.	×
Console R Errors A Warnings R Eind in	File: Results	•
	T THAT THAT THE T	Ln 29 Col 7 Verilog

#### Hit Save.



## **Simple Simulation**

In order to simulate your circuit, you will need to specify a set of inputs (which change over time). To do so, again click "New Source", but this time select *Verilog Test Fixture* from the left pane. Fill in the file name as shown, and click *Next*.

New Source Wizard      Select Source Type      Select Source Type      BMM File     Minore Perfinition and Connection File     Imploementation Constraints File     P (CORE Generator & Architecture Wizard)     M File     Schematic     System Generator Project     Using Module     Verling Module     Verling Module     VHDL Library     VHDL Library     VHDL Mackage     VHDL Test Bench     Embedded Processor	Ele name: ab1_test Logation: D:\Comp541\Lab1
More Info	Next Cancel

Click Next again.



In the top left of the main window, switch to *Simulation* view. Double click on *lab1\_test*, and in the editor that opens, paste in the following code within the *initial begin* and *end* block.

initial begin // Initialize Inputs A = 0; B = 0; Cin = 0;

// Wait, say, 10 ns before inputs start changing
#10;

// Add stimulus here
// Inputs change every 1 ns, going from 000 to 111
#1 {A, B, Cin} = 3'b001;
#1 {A, B, Cin} = 3'b010;
#1 {A, B, Cin} = 3'b100;
#1 {A, B, Cin} = 3'b100;
#1 {A, B, Cin} = 3'b101;
#1 {A, B, Cin} = 3'b110;
#1 {A, B, Cin} = 3'b111;
// Wait another 5 ns, and then finish simulation
#5 \$finish;
end

Click on *lab1\_test*, make sure *Simulation* radio button is selected...



And then double-click Simulate Behavioral Model in the pane below.



# The simulator ISim will launch:

ISim (O.87xd) - [lab1_test.v]		_	
File Edit View Simulation Wine	dow Layout Help		_ 8 ×
🗋 🌶 🖬 🖕 🕺 🗈 🗅 🗙 🕲	<b>⋈</b> ⋈ 🕅 🕅 🕅 🗠 🖬	🗆 🖻 🎤 K?	P 20 ア 20 10 10 10 10 10 10 10 10 10 10 10 10 10
Instances and Processes ↔ □ ♂ ×	Objects ↔ □ ₽ ×	40	.Sum (Sum)
	Simulation Objects for Initial_43_0	41	);
	ччжччч 🕼	42	
Instance and Process Name D		43	// Initialize Inputs
V 📕 lab1_test la	Object Name Value	45	A = 0;
Diffial 43.0		46	B = 0;
		47	Cin = 0;
	1 Cin 1	48	
		49	<pre>// wait, say, 10 ns before inputs start changing #10.</pre>
		51	#10,
		% 52	// Add stimulus here
		3 53	<pre>// Inputs change every 1 ns, going from 000 to 111</pre>
		54	#1 {A, B, Cin} = 3'b001;
		55	$\#1 \{A, B, Cin\} = 3'b010;$
		57	$\#1$ {A, B, Cin} = 3 b011, $\#1$ {A, B, Cin} = 3'b100:
		- 58	#1 {A, B, Cin} = 3'b101;
			#1 {A, B, Cin} = 3'b110;
		60	#1 {A, B, Cin} = 3'b111;
		61	// White searching from and other finish simulation
		63	#5 Sfinish:
		64	
< <u> </u>		<	F F
🛃 Instanc 🛗 Memory 📔 Source	۰ III ا	Defa Defa	ult.wcfg 🗵 📄 lab 1_test.v 🗵
Console			+□₽×
WARNING: Please use Xilinx License Configurat	ion Manager to check out a full ISim license.		A
WARNING: ISim will run in Lite mode. Please ref	er to the ISim documentation for more information	on the differences b	etween the Lite and the Full version.
Time resolution is 1 ps			
Simulator is doing circuit initialization process.			E
Finished circuit initialization process.	1 lab 1 test v <sup>a</sup> l inc 62		
ISim>	That test.v Line 05		-
Console Compilation Log 🗕 I	Breakpoints 🛛 🕅 Find in Files Results 🖬 Se	earch Results	
			Sim Time: 22,000 ps Ln 63 Col 1 Verilog

Click on the waveform window (*Default.wcfg*).



Click on the "Zoom to fit" icon.





You will now see the resulting waveforms of all the inputs as well as the output!

		22.000 ns
Name	Value	0 ns  10 ns  20 ns
🗓 Sum	1	
16 А	1	
16 в	1	
1 Cin	1	

This is the end of Part I!

## PART II: EXERCISE

Modify your work from PART I to add the second output to your circuit: *Carry*<sub>out</sub>. This requires two simple changes to your Verilog description from Part I:

- Adding another output (*Carry*<sub>out</sub>) to the module's interface
- Adding another continuous assignment (using keyword *assign*) to describe the Boolean equation for computing *Carry*<sub>out</sub>.

For a complete list of Verilog operators, please refer to the online Verilog reference (linked from the course website), Section 11.

Rename the module to **lab1\_part2**. It will look as follows (fill in the blanks):

```
module lab1_part2(
    input A,
    input B,
    input Cin,
    output Sum,
    output ...
  );
    assign Sum = Cin ^ A ^ B;
    assign ... = ...
```

### endmodule

Follow all of the steps as in Part I to perform a simulation, and observe the waveforms. Make sure the carry output is correct, along with the sum.

*What to submit:* A screenshot of the *ISim* window for Part II only, clearly showing the 3 input and 2 output waveforms.

*How to submit:* We will be using electronic submission (save a tree!). Please submit your work by email as follows:

- Send email to: comp541submit-cs@cs.unc.edu
- Attach the simulator screenshot using the filename waveforms.png (or other appropriate extension)
- Submit your work by 11:59pm on Wednesday, August 27.