Jyresa Mae M. Amboang
2020-2591
Laboratory Submitted: 03/17/23

Laboratory Activity No. 1
Full Adder

Purpose: In this laboratory activity you will learn to use the Quartus II, ModelSim, and FPGAs to create a full adder. Implementing and simulating the full adder in the Altera DE2-115 and breadboard.

Overview: In this laboratory activity, you will learn how to build and simulate Field Programming Gate Arrays (FPGAs) to create a Full-Adder design. The schematic for the circuit will then be created using logic gates. By simulating the operation of your full adder, we can finally ensure that your design is accurate. Also, you should be able to use Quartus® II Web Edition Design Software, Version 13.1 after completing this laboratory.

Theory: A full adder is a combinational arithmetic circuit of X-OR gates, OR gates, and AND gates that adds three inputs to produce two outputs. The first two inputs are A and B and the third input is called carry (C-IN). Carry-out is shown as C-OUT and normal output is shown as S, which is SUM. A full adder circuit can be constructed by adding two half adders with an OR gate. The first half-adder circuit has two single-bit binary inputs A and B. As we proved earlier, two outputs are produced: SUM and Carry Out. The SUM output of the first half adder circuit is also provided to the input of the second half adder circuit. We provided a carry-in bit from the other input in the second half of the order circuit. SUM out and carry bit are also provided here. This SUM output is the final output of the full adder circuit. On the other hand, the carry of the first half adder circuit and the carry of the second adder circuit are further provided in the OR logic gate. After the logical OR of the two carry outputs, we get the final carry from the full adder circuit.

Design

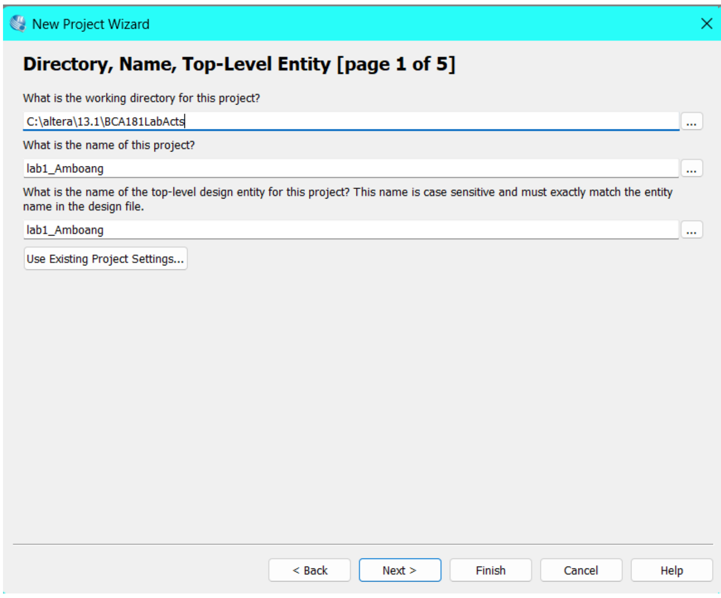
A partially completed truth table for a full adder is given in Table below. The table indicates thevalues of the outputs for every possible input and thus completely specifies the operation of a fulladder. As is common, the inputs are shown in binary numeric order. The values for *S* (sum) are given, but the C_{out} (carry out) column is left blank. **Complete the table by filling in the correct values for C_{out} so that adders connected, as in the table, will perform valid addition.**

Inputs			Outputs	
C_{in}	*B*	*A*	C_{out}	*S*
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1

1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

SCHEMATIC

This laboratory activity we will be using the Altera Quartus II 13.1 Web Edition softwareIn the Getting Started Window, click on Create a New Project. In the New Project Wizard, set the working directory to a good place in your laboratory home directory. For example, if your laboratory directory is mapped to the H drive, choose H:\bca181\lab1_Surname, where Surname is your Surname. Name the project lab1_Surname. We'll be using the Altera DE2-115 development board, which contains Cyclone-IV E as the target device. From the list of available devices, choose the device called EP4CE115F29C7 that is used on the board.



New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

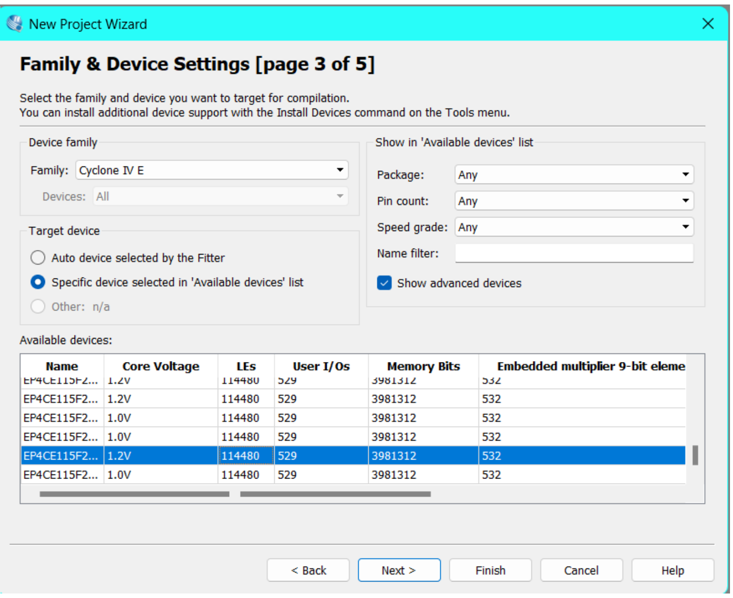
C:\altera\13.1\BCA181LabActs

What is the name of this project?

lab1_Amboang

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

lab1_Amboang



New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

Device family

Family: Cyclone IV E

Devices: All

Target device

○ Auto device selected by the Fitter

● Specific device selected in 'Available devices' list

○ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Speed grade: Any

Name filter:

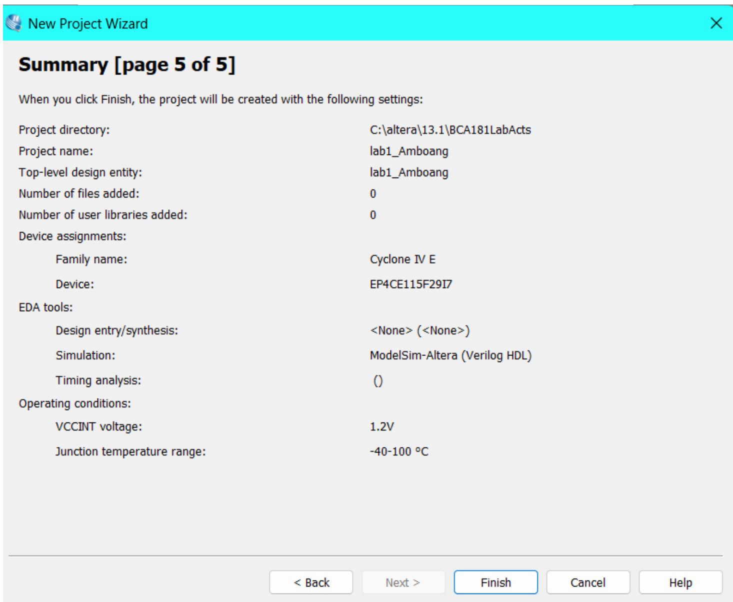
☑ Show advanced devices

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit eleme
EP4CE115F2...	1.2V	114480	529	3981312	532
EP4CE115F2...	1.2V	114480	529	3981312	532
EP4CE115F2...	1.0V	114480	529	3981312	532
EP4CE115F2...	1.0V	114480	529	3981312	532
EP4CE115F2...	1.2V	114480	529	3981312	532
EP4CE115F2...	1.0V	114480	529	3981312	532

Fig 1. Getting Started with Altera Quartus II

Fig 2. Choose the device family and a specific device

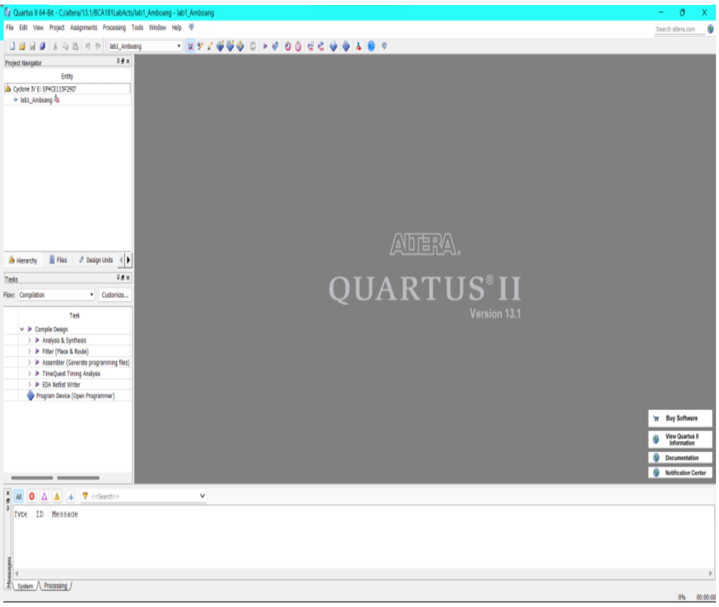


New Project Wizard

Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

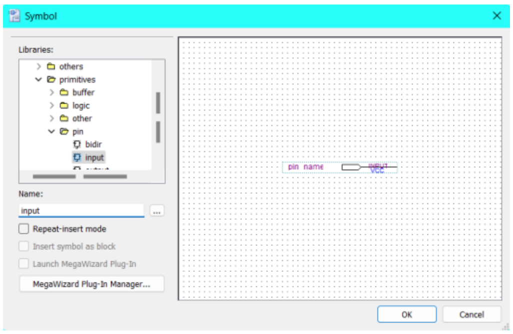
Project directory:	C:\altera\13.1\BCA181LabActs
Project name:	lab1_Amboang
Top-level design entity:	lab1_Amboang
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	Cyclone IV E
Device:	EP4CE115F29I7
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	ModelSim-Altera (Verilog HDL)
Timing analysis:	()
Operating conditions:	
VCCINT voltage:	1.2V
Junction temperature range:	-40-100 °C



Quartus II 64-Bit - C:/altera/13.1/BCA181LabActs/lab1_Amboang - lab1_Amboang

Fig 3. Summary of the Project Settings

Fig 4. The Quartus II display for the created project

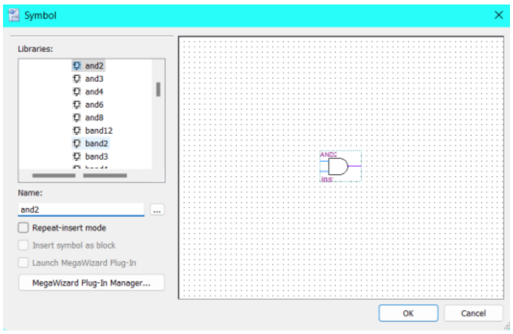


Name: input

☐ Repeat-insert mode

☐ Insert symbol as block

☐ Launch MegaWizard Plug-In



Name: and2

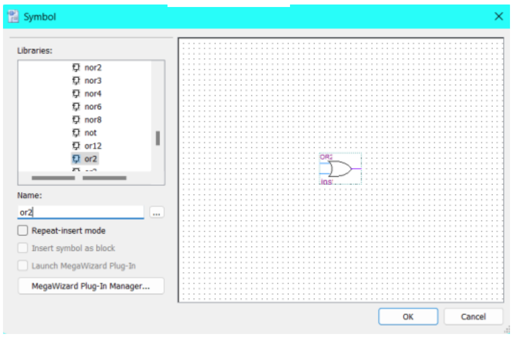
☐ Repeat-insert mode

☐ Insert symbol as block

☐ Launch MegaWizard Plug-In

Input

And

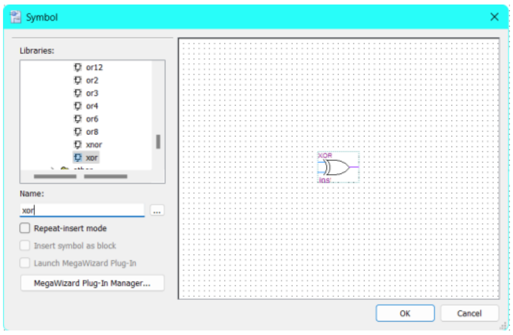


Name: or2

☐ Repeat-insert mode

☐ Insert symbol as block

☐ Launch MegaWizard Plug-In



Name: xor

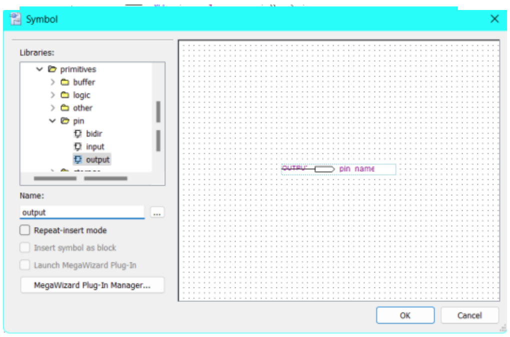
☐ Repeat-insert mode

☐ Insert symbol as block

☐ Launch MegaWizard Plug-In

Or

XOR



Name: output

☐ Repeat-insert mode

☐ Insert symbol as block

☐ Launch MegaWizard Plug-In

Output

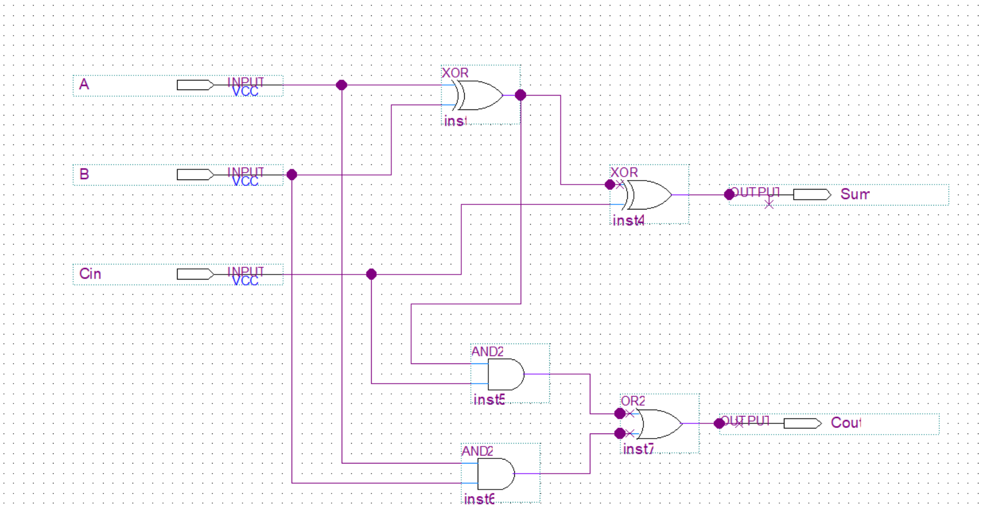
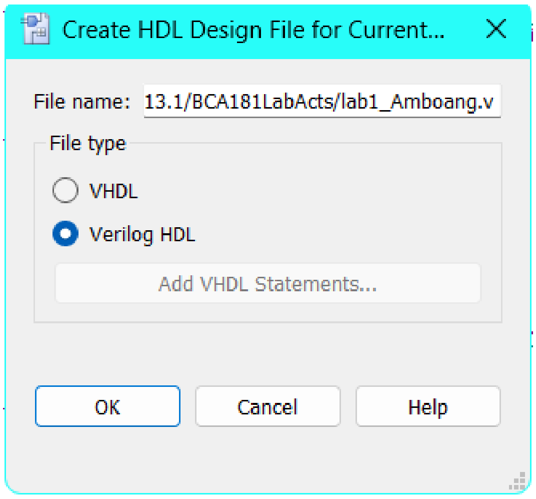


Fig 5. The Quartus II display the Schematic Diagram



Create HDL Design File for Current...

File name: 13.1/BCA181LabActs/lab1_Amboang.v

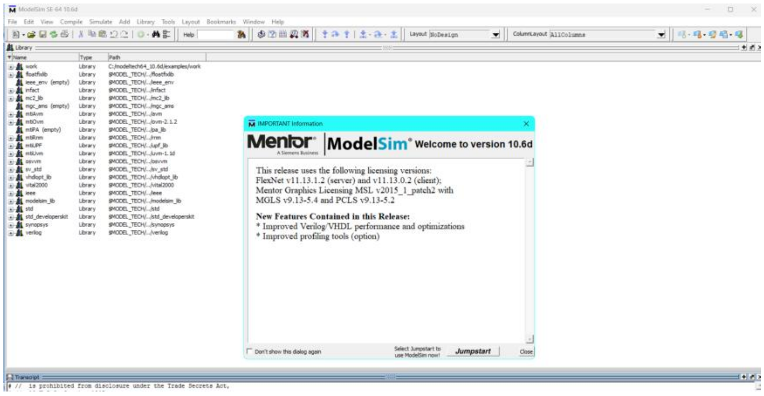
File type

○ VHDL

● Verilog HDL

Fig 6. Create HDL Design File for Current File

Now fire up ModelSim SE 10.0d from the Windows start menu. Maximize the ModelSim window when it opens. If prompted, you may wish to associate file types with ModelSim but do not want to use Jumpstart.



IMPORTANT Information

Mentor ModelSim Welcome to version 10.6d

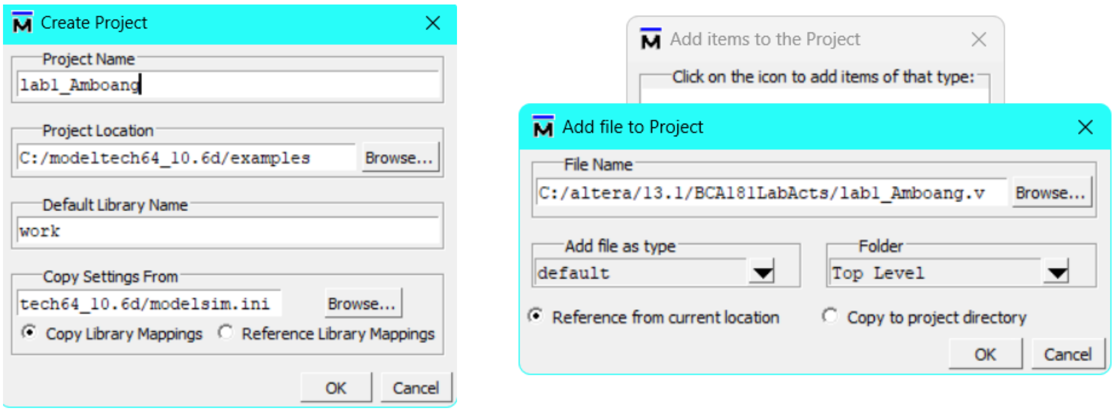
This release uses the following licensing versions:
FlexNet v11.13.1.2 (server) and v11.13.0.2 (client);
Mentor Graphics Licensing MSL v2015_1_patch2 with
MGLS v9.13-5.4 and PCLS v9.13-5.2

New Features Contained in this Release:
* Improved Verilog/VHDL performance and optimizations
* Improved profiling tools (option)

Don't show this dialog again
Select Jumpstart to use ModelSim now!
Jumpstart
Close
Layout NoDesign
ColumnLayout AllColumns

Fig .7 Opening ModelSim Application

Choose File → New → Project. Name the project lab1_Surname and put it in the directory where you are working (e.g. H:/bca181/lab1_Surname). Accept the default library name of "work." Then click "Add Existing File" and add lab1_Surname.v.



Create Project

Project Name: lab1_Amboang

Project Location: C:/modeltech64_10.6d/examples

Default Library Name: work

Copy Settings From: tech64_10.6d/modelsim.ini

● Copy Library Mappings ○ Reference Library Mappings

Add items to the Project

Click on the icon to add items of that type:

Add file to Project

File Name: C:/altera/13.1/BCA181LabActs/lab1_Amboang.v

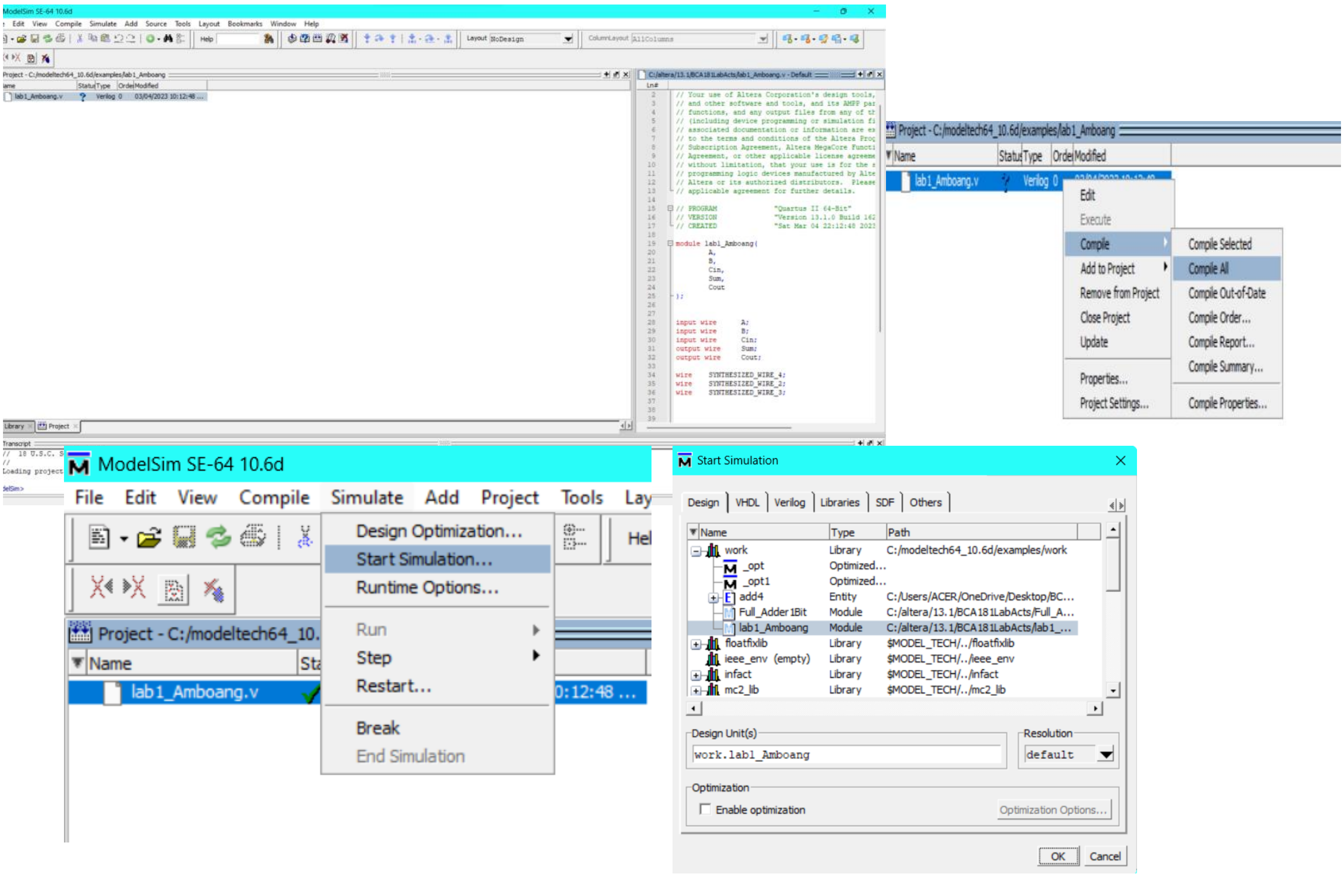
Add file as type: default

Folder: Top Level

● Reference from current location ○ Copy to project directory

Fig .8 Creating Project File ModelSim

You should see lab1_ Surname.v in the ModelSim project pane. Double-click on it to view the inputs and outputs and the wires of the schematic diagram. Choose Compile → Compile All to compile the Verilog code into a form that ModelSim can simulate. Then choose Simulate → Start Simulation and choose lab1_Surname as your module to simulate. Uncheck “enable optimization” because it sometimes hides information that is useful during debugging.



C:/altera/13.1/BCA181LabActs/lab1_Amboang.v - Default

```
// Your use of Altera Corporation's design tools,
// and other software and tools, and its AMPP par
// functions, and any output files from any of th
// (including device programming or simulation fi
// associated documentation or information are ex
// to the terms and conditions of the Altera Prog
// Subscription Agreement, Altera MegaCore Functi
// Agreement, or other applicable license agreeme
// without limitation, that your use is for the s
// programming logic devices manufactured by Alte
// Altera or its authorized distributors.  Please
// applicable agreement for further details.

// PROGRAM          "Quartus II 64-Bit"
// VERSION          "Version 13.1.0 Build 162
// CREATED          "Sat Mar 04 22:12:48 2023

module lab1_Amboang(
        A,
        B,
        Cin,
        Sum,
        Cout
);


input wire      A;
input wire      B;
input wire      Cin;
output wire     Sum;
output wire     Cout;

wire    SYNTHESIZED_WIRE_4;
wire    SYNTHESIZED_WIRE_2;
wire    SYNTHESIZED_WIRE_3;
```

Name	Type	Path
work	Library	C:/modeltech64_10.6d/examples/work
_opt	Optimized...	
_opt1	Optimized...	
add4	Entity	C:/Users/ACER/OneDrive/Desktop/BC...
Full_Adder1Bit	Module	C:/altera/13.1/BCA181LabActs/Full_A...
lab1_Amboang	Module	C:/altera/13.1/BCA181LabActs/lab1_...
floatfixlib	Library	$MODEL_TECH/../floatfixlib
ieee_env (empty)	Library	$MODEL_TECH/../ieee_env
infact	Library	$MODEL_TECH/../infact
mc2_lib	Library	$MODEL_TECH/../mc2_lib

Design Unit(s): work.lab1_Amboang

Resolution: default

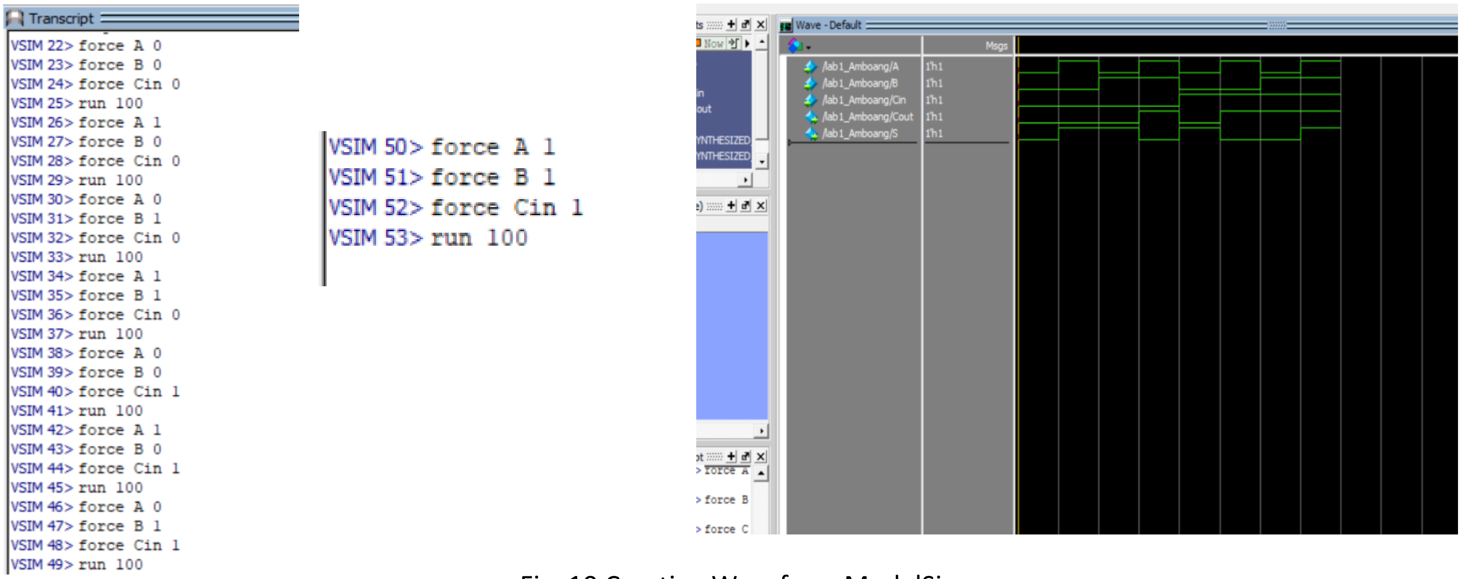
Optimization

□ Enable optimization

Optimization Options...

Fig .9 Compiling and Simulation ModelSim

In the objects window, you’ll see all the inputs, outputs, and internal wires. Shift-click to select them all. Then right-click and choose Add → To Wave → Selected Signals. A Wave pane will pop up with the signals. Typing the inputs in the transcript pane at the bottom and then you should see all the inputs and outputs on the Wave pane:



Transcript

```
VSIM 22> force A 0
VSIM 23> force B 0
VSIM 24> force Cin 0
VSIM 25> run 100
VSIM 26> force A 1
VSIM 27> force B 0
VSIM 28> force Cin 0
VSIM 29> run 100
VSIM 30> force A 0
VSIM 31> force B 1
VSIM 32> force Cin 0
VSIM 33> run 100
VSIM 34> force A 1
VSIM 35> force B 1
VSIM 36> force Cin 0
VSIM 37> run 100
VSIM 38> force A 0
VSIM 39> force B 0
VSIM 40> force Cin 1
VSIM 41> run 100
VSIM 42> force A 1
VSIM 43> force B 0
VSIM 44> force Cin 1
VSIM 45> run 100
VSIM 46> force A 0
VSIM 47> force B 1
VSIM 48> force Cin 1
VSIM 49> run 100
```

```
VSIM 50> force A 1
VSIM 51> force B 1
VSIM 52> force Cin 1
VSIM 53> run 100
```

Fig .10 Creating Waveform ModelSim

Altera DE2-115:

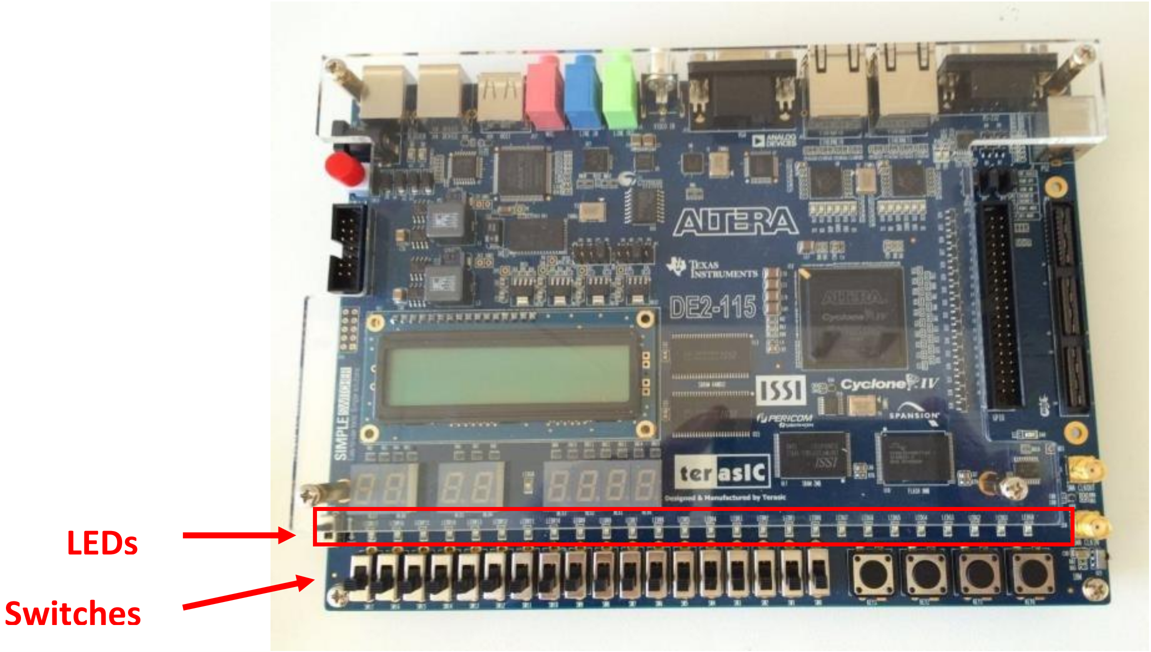


Fig .11 DE2-115 board (image courtesy Terasic Inc.)

FPGA (labeled Cyclone IV E) in the middle of the board as well as 18 switches (on the bottom of the board) and 18 LEDs just above the switches (see Figure 11). You will map the inputs A, B, and Cin to the three right-most switches (called SW [0], SW [1], and SW [2], from right to left. The output S and Cout will connect to the right-most LED (light emitting diode), called LEDR [0] and LEDR [1].

Since we have finished the circuit of our project, we need to assign the input and output pins to switches and LEDs on the FPGA board. This will be the true test to see if the circuit works. To use this file, choose Assignments → Pin Planner. A new window will appear with all the input and output pins you've placed in the circuit. Grab your FPGA manual for your device's pin assignment (https://usermanual.wiki/Pdf/DE2115UserManual.833210098/html). The pin assignment should go through all of your switches and LEDs.

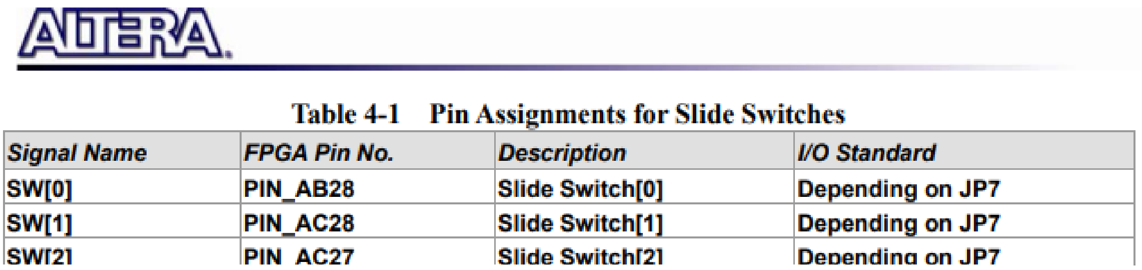


Table 4-1 Pin Assignments for Slide Switches

Signal Name	FPGA Pin No.	Description	I/O Standard
SW[0]	PIN_AB28	Slide Switch[0]	Depending on JP7
SW[1]	PIN_AC28	Slide Switch[1]	Depending on JP7
SW[2]	PIN_AC27	Slide Switch[2]	Depending on JP7

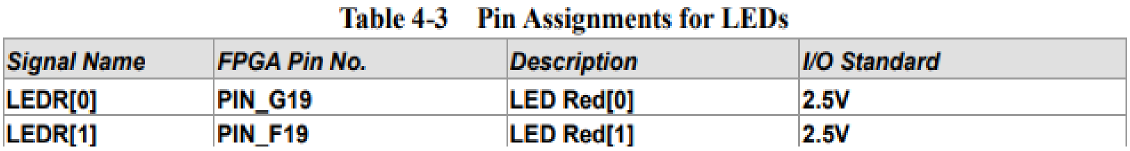
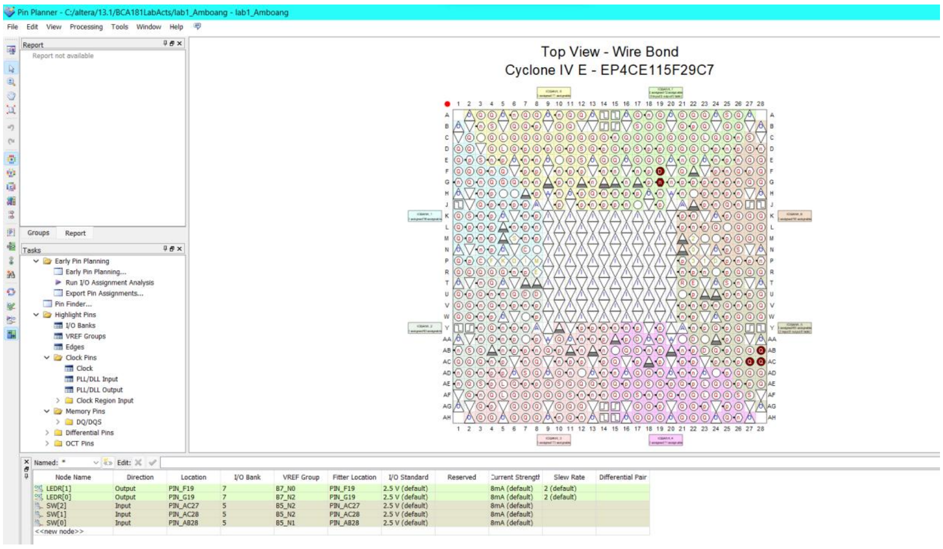


Table 4-3 Pin Assignments for LEDs

Signal Name	FPGA Pin No.	Description	I/O Standard
LEDR[0]	PIN_G19	LED Red[0]	2.5V
LEDR[1]	PIN_F19	LED Red[1]	2.5V



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
LEDR[1]	Output	PIN_F19	7	B7_N0	PIN_F19	2.5 V (default)		8mA (default)	2 (default)	
LEDR[0]	Output	PIN_G19	7	B7_N2	PIN_G19	2.5 V (default)		8mA (default)	2 (default)	
SW[2]	Input	PIN_AC27	5	B5_N2	PIN_AC27	2.5 V (default)		8mA (default)		
SW[1]	Input	PIN_AC28	5	B5_N2	PIN_AC28	2.5 V (default)		8mA (default)		
SW[0]	Input	PIN_AB28	5	B5_N1	PIN_AB28	2.5 V (default)		8mA (default)		
<<new node>>										

Fig .12 Cyclone IV E FPGA device pins

You’ll need to assign the pins so that you can use switches to control the inputs and LEDs to .display the outputs. Altera provides a file describing how the various circuits on the DE2-115 board are connected to the FPGA. To use this file, choose Assignments → Import Assignments.

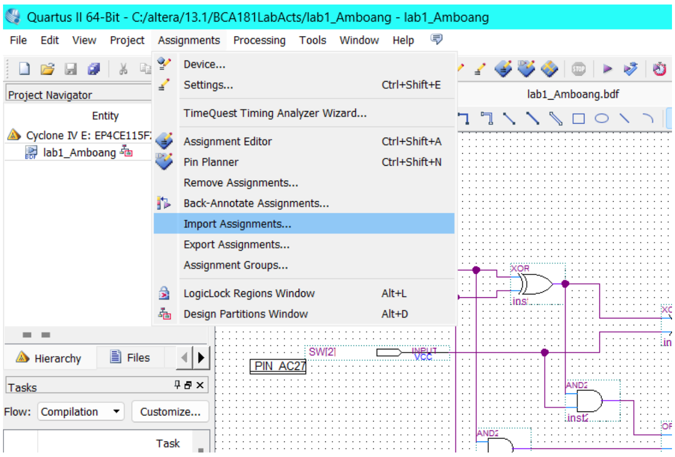
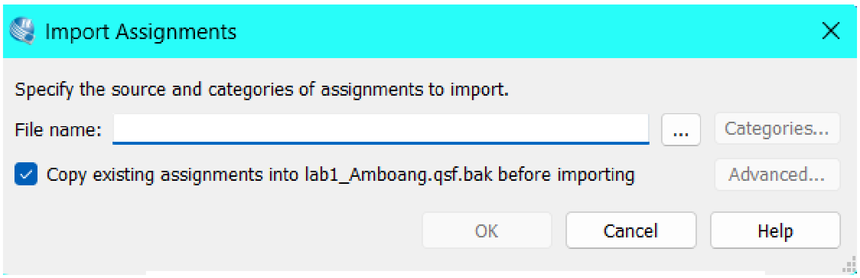


Fig .13 Importing Pin Assignments

Now click on the browse button (…) as shown in Figure 14.



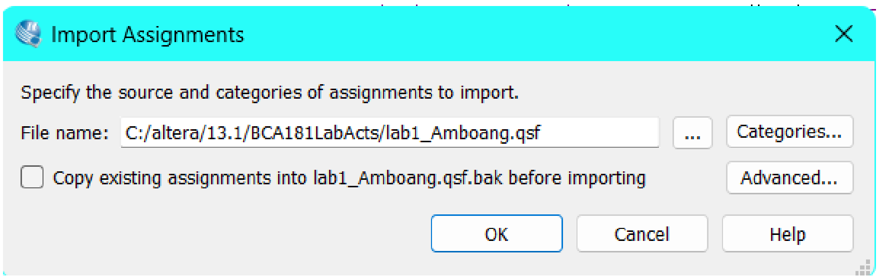
Specify the source and categories of assignments to import.

File name:

☑ Copy existing assignments into lab1_Amboang.qsf.bak before importing

Fig .14 Import Assignments

Browse to where you placed lab1_amboang.qsf (Quartus Settings File). You don't need to keep a backup of the existing pin assignments, so uncheck the box next to Copy existing assignments into Project1… and click OK, as shown in Figure 15.



Specify the source and categories of assignments to import.

File name: C:/altera/13.1/BCA181LabActs/lab1_Amboang.qsf

☐ Copy existing assignments into lab1_Amboang.qsf.bak before importing

Fig .15 Selecting the .qsf (pin assignment) file

Now recompile your design.

After completing these steps, you can download your design to the DE2-115 FPGA board. First, connect the DE2-115 board to your computer and turn it on. Connect your computer and board with a USB cable and plug the power cord into a power outlet. Power on the board by pressing the red power button once. The LEDs should light up in a pattern and the hexadecimal value 0-f should appear on the 7-segment display just above the LEDs. Now click on Tools → Programmer in Quartus II (see Figure 16).

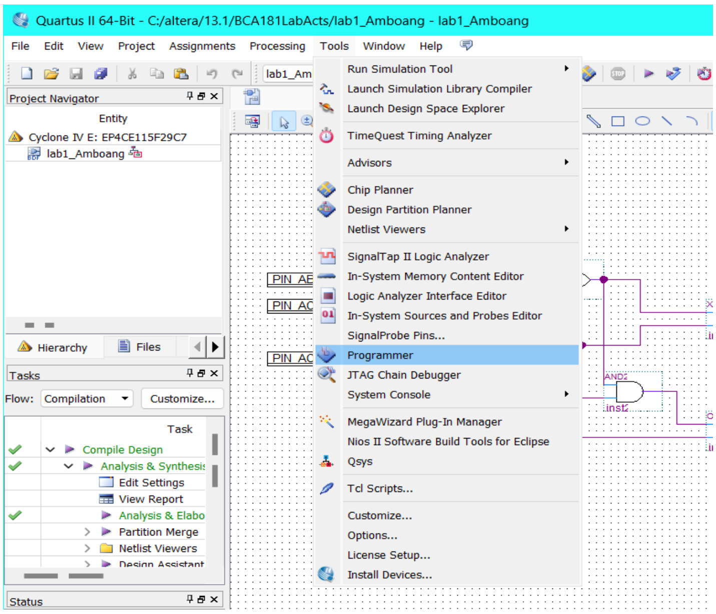
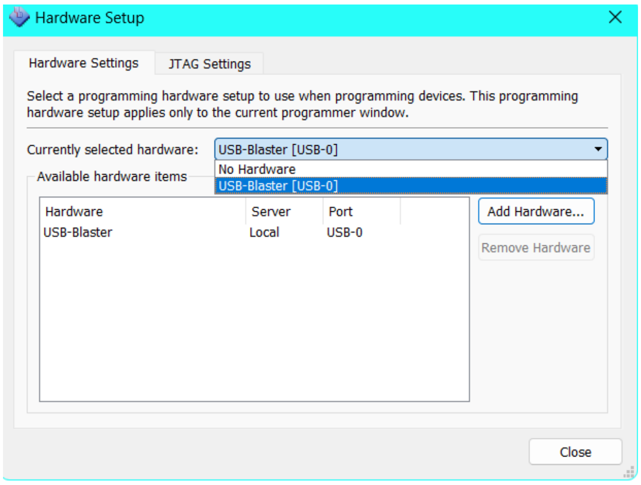


Fig .16 Programming the DE2-115 board with the MIPSfpga system

Then click on USB-Blaster [USB-0] under currently selected hardware (see Figure 17). If no USB-Blaster option is available, the driver is not installed, or the cable is not connected to your computer, etc. The lab1_Amboang.sof (SRAM Object File) contains the information to configure the Cyclone IV E FPGA on the DE2-115 board so that it implements your design. Now click on Start, which will configure the FPGA (see Figure 17).



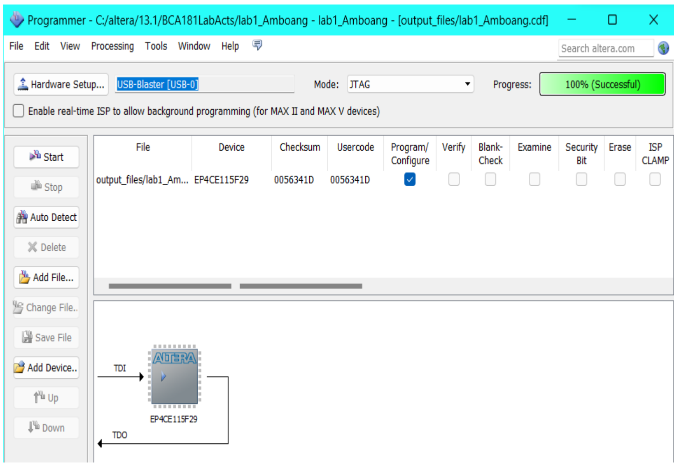
Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: USB-Blaster [USB-0]

Available hardware items

Hardware	Server	Port
USB-Blaster	Local	USB-0

Fig .17 Setting currently selected Hardware



Programmer - C:/altera/13.1/BCA181LabActs/lab1_Amboang - lab1_Amboang - [output_files/lab1_Amboang.cdf]

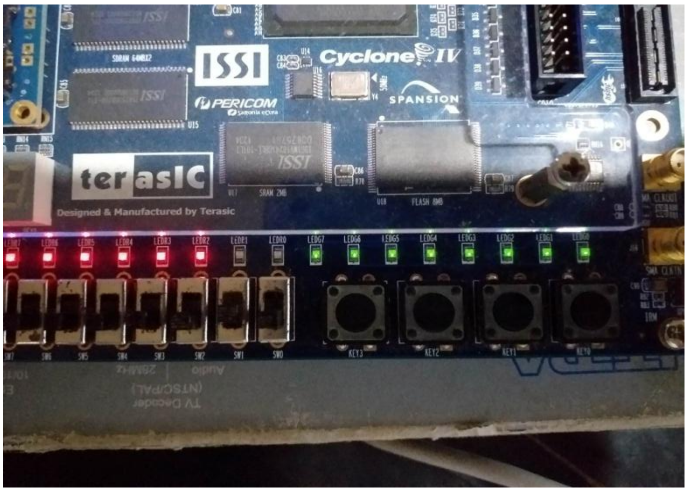
Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress: 100% (Successful)

□ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

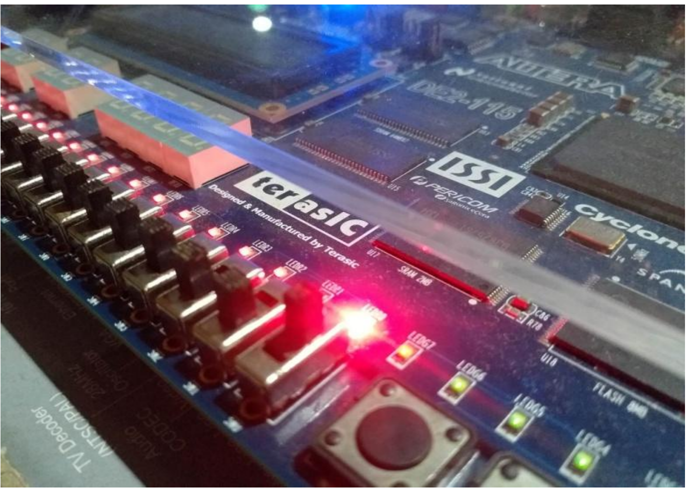
File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
output_files/lab1_Am...	EP4CE115F29	0056341D	0056341D	☑	□	□	□	□	□	□

Fig .18 Hardware Setup…

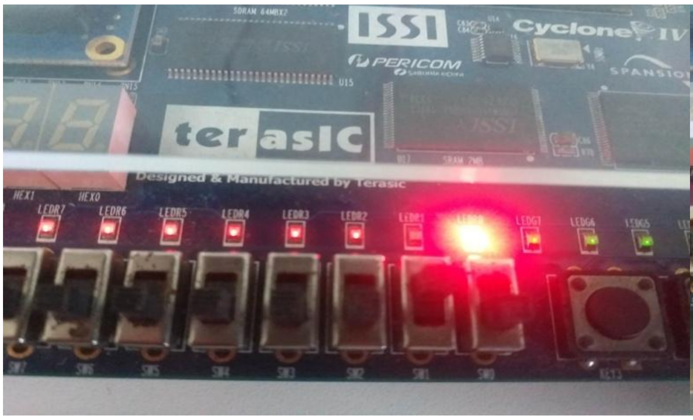
After clicking the Start button, wait a few seconds and the design will download to the DE2-115 board. Test all combinations of inputs and see what the output (rightmost LED) does. For example, the combination A, B, Cin = 000 requires all switches to be in the down position. Make sure the rightmost LED0 and LED1 are not light up (that is, the output is 0). Then toggle Cin to 1 (that is, toggle the rightmost switch up). Now the 001 combination is tested: the LED should light up (that is, the output should be 1 in LED0) as shown in Figure 19. Test all input combinations and verify that the outputs match the desired truth table.



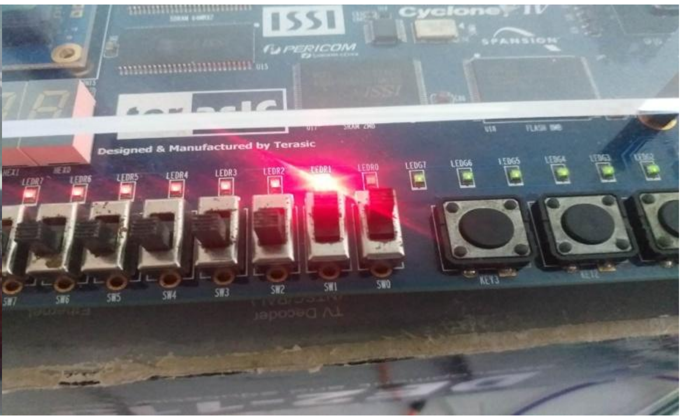
SW0 (A): 0 SW1(B):0 SW2 (Cin):0
LEDR0(Sum):0 LEDR(Cout):0



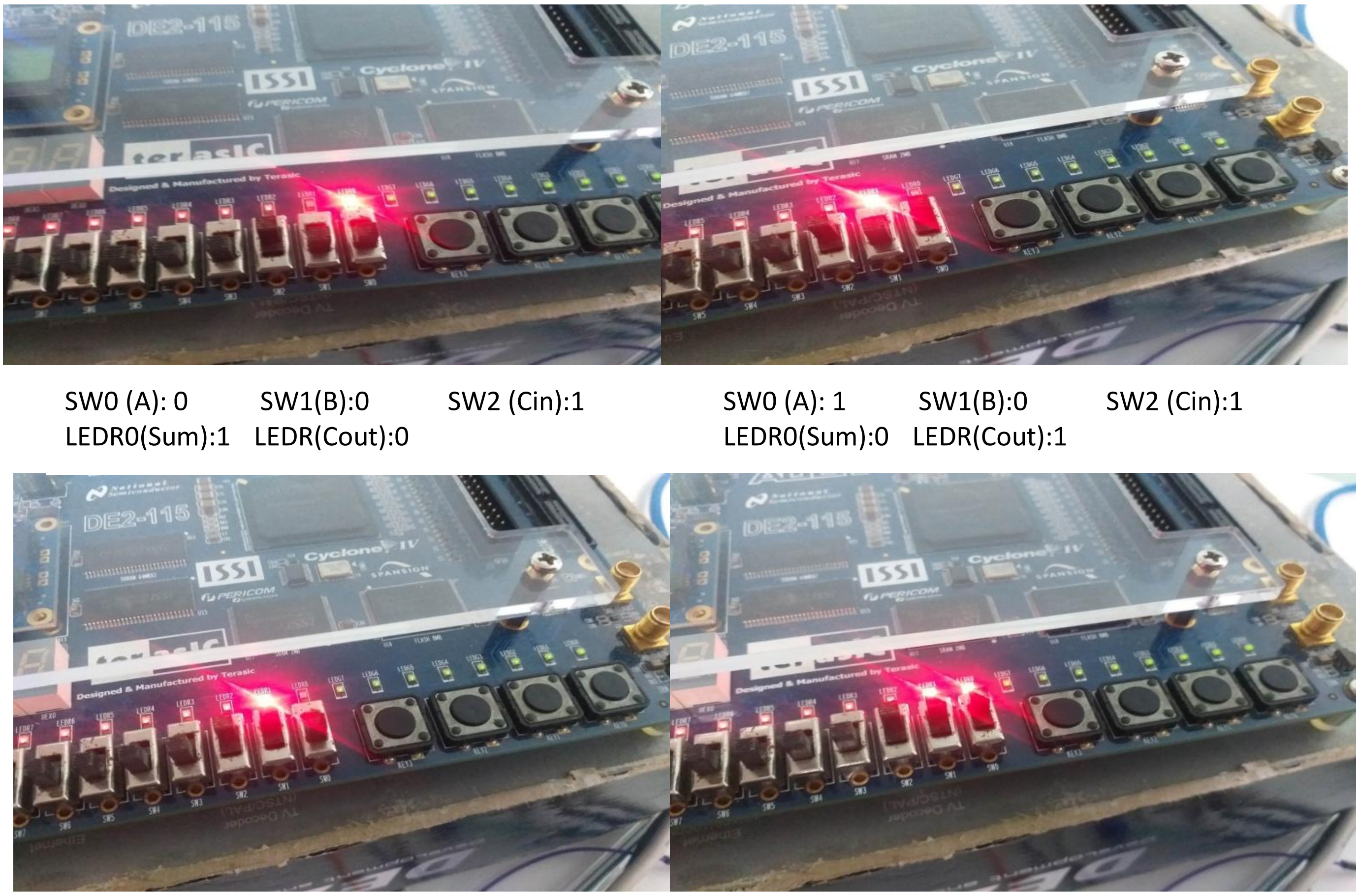
SW0 (A): 1 SW1(B):0 SW2 (Cin):0
LEDR0(Sum):1 LEDR(Cout):0



SW0 (A): 0 SW1(B):1 SW2 (Cin):0
LEDR0(Sum):1 LEDR(Cout):0



SW0 (A): 1 SW1(B):1 SW2 (Cin):0
LEDR0(Sum):0 LEDR(Cout):1



SW0 (A): 0 SW1(B):0 SW2 (Cin):1
LEDR0(Sum):1 LEDR(Cout):0

SW0 (A): 1 SW1(B):0 SW2 (Cin):1
LEDR0(Sum):0 LEDR(Cout):1

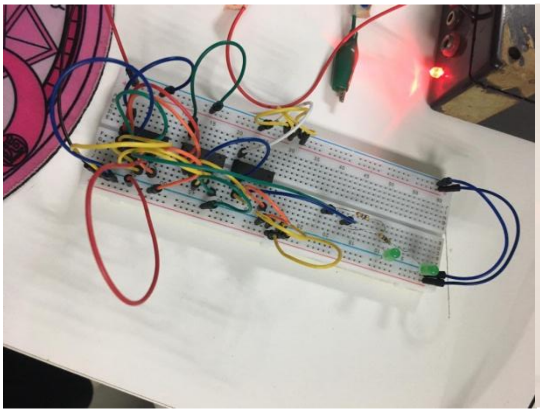
SW0 (A): 0 SW1(B):1 SW2 (Cin):1
LEDR0(Sum):0 LEDR(Cout):1

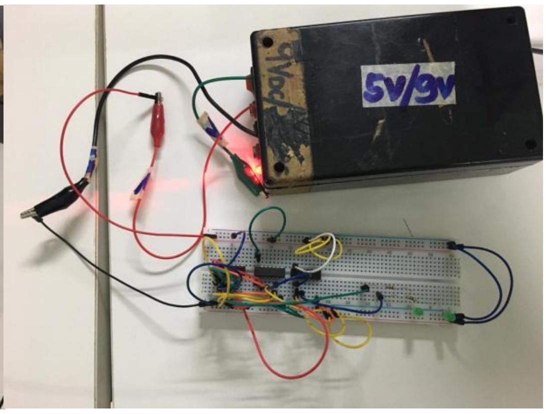
SW0 (A): 1 SW1(B):1 SW2 (Cin):1
LEDR0(Sum):1 LEDR(Cout):1

Fig .19 Simulating DE2-115

Breadboard Simulation:

In this part of the lab activity, we will implement the design in hardware 7486(X0R Gate IC), 7480(AND Gate IC), and 7432(0R Gate IC) series chips, commonly sold in 14-pin dual in-line packages (DIP). We will need a power supply capable of producing 5V/9V DC and breadboard in making the 1-bit full adder design. Test all input combinations and verify that the outputs match the desired truth table in toggling our jumper wires for the specific light indicators in the design thus has two resistors and LED's corresponds to the sum and carry out of the truth table





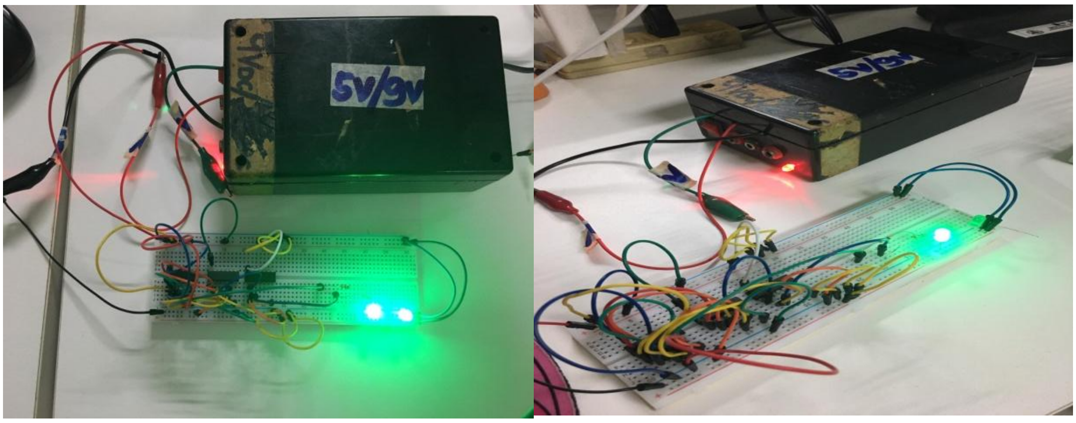


Fig .20 1-Bit Full Adder Breadboard Simulations

What did you learn or take away from performing this laboratory activity?

In this laboratory activity I have learn to simulate in the DE2-115 Altera board with the use of an application of Quartus II and ModelSim. Before the simulation, you have to review the logic gates in making a schematic diagram such as the Full Adder with a corresponding truth table to support to simulate the DE2-115 and breadboard simulation. In creating a schematic diagram with use of design tools of Quartus II in the design logic and Verilog design file, after that compile, then assign the pin assignments that are based in FPGA manual for the device pin's assignment for the hardware setup of our Altera board, the Quartus II Compiler can select any pin on the selected FPGA to act as input and output. However, DE Series boards have hardwired connections between FPGA pins and other components on the board, and make sure that your PC or laptop has USB blaster installed in order for the connection of hardware setup be detected. Check that the board is turned on and press the red power button; the blue Power and LEDs should turn on. Then click the Start button on the hardware setup window, wait a few seconds and your design will be downloaded onto the DE2-115 board. Toggle SW0, SW1, and SW2 through the eight possible patterns in the truth table. Check that LEDR0 and LEDR1 display the correct combination. Also, I learn to use the other design tool which the ModelSim it is a verification and simulation tool for Verilog file to verify the correctness of our design logic in Quartus II, using ModelSim as your primary debugging tool to debug your processor designs by the input and observe the output. For this purpose we can verify that the circuit is working properly by comparing it to the output. Test all possible input combinations and observe the output for each combination.