LAB REPORT – 4

Title: Design a simple digital circuit called full adder, as well as learn to use Altera field-programmable gate array (FPGA) to design a schematic and simulate.

Purpose:

- The purpose of thus lab is to perform the arithmetic function on full adder
- Design the schematic of the full adder using Altera Quartus II software and to perform simulation
- Downloading the design onto DE2 Board and analyze the output of the given inputs

Requirements:

- Altera Quartus II 9.1p2 software
- DE2 board

Description:

Altera Quartus II 9.1p2 – It's a wed edition software which allow the user to analyze and synthesize the HDL designs, which enables the user to design the schematic model and simulate the design. Quartus implement the VHDL and Verilog for hardware description, vector waveform simulation and visual editing of logic circuits.

DE2 board: It's a board to test the circuit design on FPGA chip. FPGA will correspond to the input and outputs of the design.

FULL ADDER

Full adder is a type of adder which has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry designated as C_{in} . The output carry is designated as C_{out} and the normal output is designated as S. Below is the truth table of the full adder.

Input bit A	**Input bit B**	**Carry bit input C_{in}**	**Carry bit output C_{out}**	**Sum bit output 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

$$C_{OUT} = \overline{A}BC_{IN} + A\overline{B}C_{IN} + AB\overline{C_{IN}} + ABC_{IN}$$

Building Schematic using Altera Quartus II:

After working for more than 5 hours in lab we developed the schematic design of the full adder using Altera Quartus II software which help us to build the schematic design of the circuit and simulate the circuit.

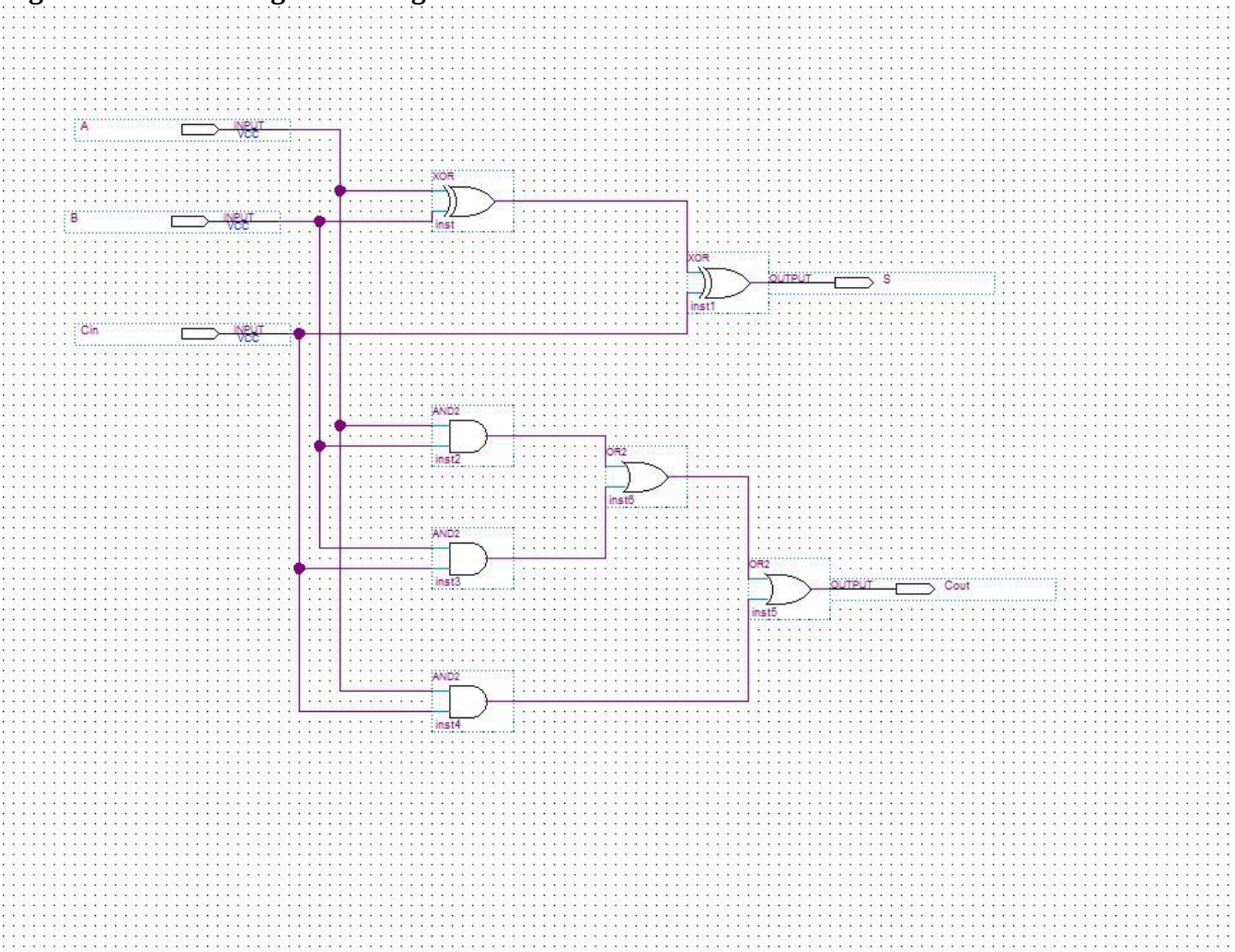
Step 1: Create a new schematic by choosing File New and selecting Block Diagram / Schematic File, and click OK. A new schematic window named Block1.bdf will appear. First, will place our logic gates by clicking on the Symbol Tool icon (shaped like an AND gate, OR gate, XOR gate).

Step 2: Using the Orthogonal Node Tool ( ) to wire the gates together. Click and drag to connect the pins to gates and the two gates together.

Step 3: Use the existing input terminals for A, B, and C_{in}, and add an output terminal for C_{out}. The symbols we used to draw logic gates are as follows: and2, or2.

Below is the designed completed schematic including the logic gates i.e. XOR, AND, OR gates.

Fig.1 Schematic Design of the logic circuit of Full Adder



$$C_{OUT} = BC_{IN} + AC_{IN} + AB$$

Simulation:

After designing the schematic model of the full adder we simulate the design in HDL such as Verilog HDL using ModelSim simulator. Following are the steps followed to simulate the design:

Step 1: We simulate the design using ModelSim. ModelSim expects a description of a circuit in a hardware description language (HDL) such as Verilog. To convert our schematic to Verilog, we opened the schematic and chose File - Create / Update, Create HDL Design File for Current File. Chose Verilog HDL.

Step 2: Now fire up ModelSim SE 10.1b from the Tools, Run Simulation Tool pulldown. We got a message indicating that there was no path to ModelSim. To correct this problem, we chose **Tools / Options /EDA tool options / Paths** and specified the path for ModelSim-Altera as **C:/Program Files/Altera/13sp1/modelsim_ase/Win32aloem** .Then recompiled (**Processing / Start Compilation**) and start up ModelSim with **Tools / Run Simulation Tool / Gate-Level Simulation**.

Step 3: Chose **Compile / Compile All** to compile the Verilog code into a form that ModelSim can simulate. Watch for and correct errors in the transcript pane. Then chose **Simulate / Start Simulation**.

Step 4: When the simulator starts, ModelSim will open more panes including sim and Objects that help us select signals for the waveform viewer. In the objects window, we'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.

Step 5: To add the wave position, right-click on the names in the object pane and choose Add to wave. Finally, we apply the inputs. In the transcript pane at the bottom, type

Force sim: /lab1_LM/A 0

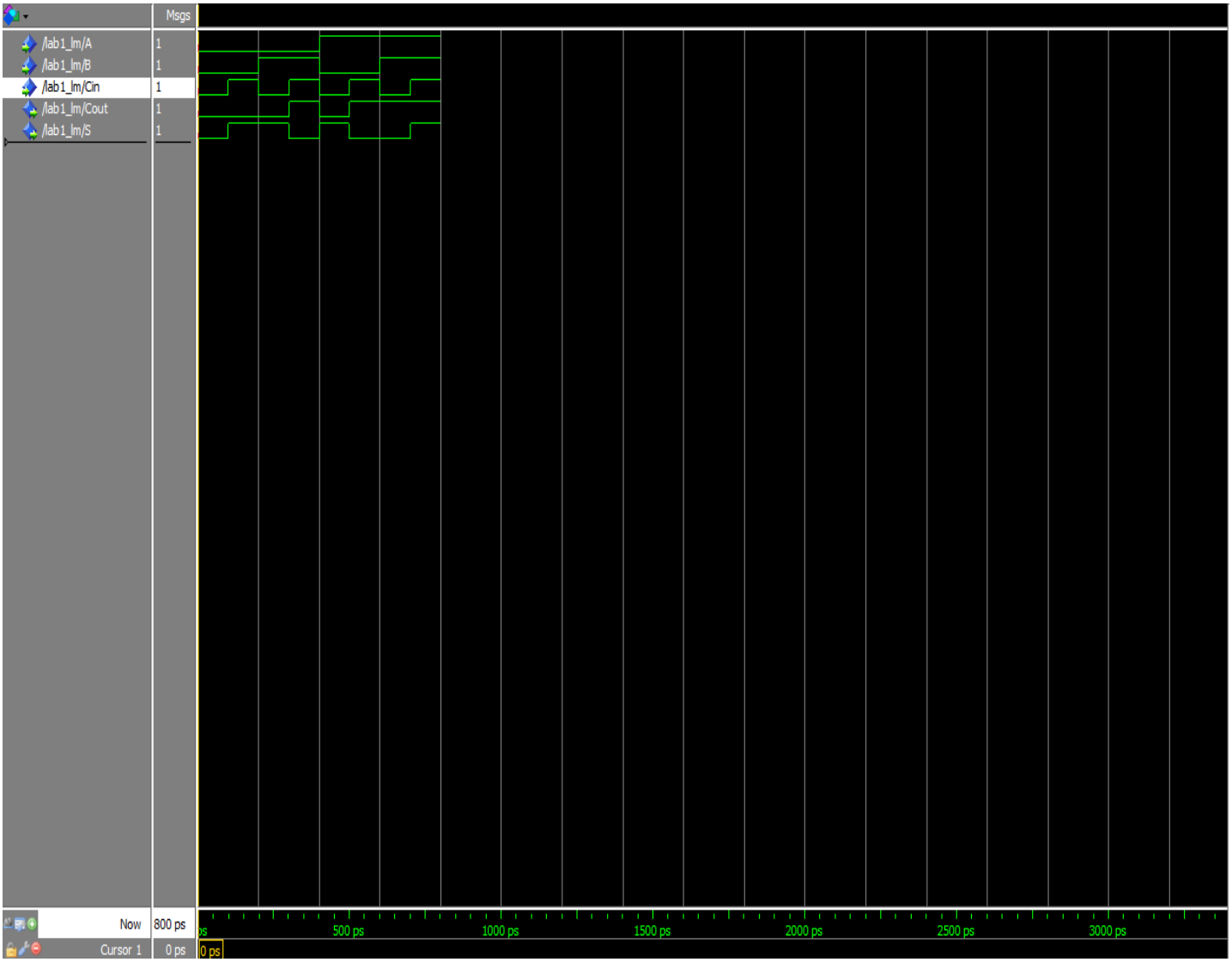
Force sim: /lab1_LM/B 0

Force sim: /lab1_LM/Cin 0

Step 6: We continued with the eight other patterns of inputs to check our truth table.

Following waveform of the simulation is formed using inputs A, B , Cin

Fig.2 Simulation Waveform Diagram



The following simulation output of S and Cout verifies that the schematic circuit designed is correct and working correctly after compilation.

DE2 Board Implementation:

After the design is simulated using ModelSim correctly, we close the ModelSim and return back to Quartus II software. Our next goal was to download the circuit onto a DE2 test board to test it on the FPGA chip. The pins on the FPGA will correspond to the inputs and outputs of our design. We need to assign the pins so that we can use switches to control the inpouts and LEDs to display the outputs.

We downloaded the CSV file to rename the inputs on the board. We renamed the inputs A, B,Cin to SW[0],SW[1],SW[2], respectively. Renamed the outputs S and Cout to LEDR [0], LEDR [1], respectively. Saved the schematic.

Connecting the DE2 board:

- We plugged in the power adapter, attached it to the board
- Connected the DE2 board to the computer USB cable port. The cable must go into the leftmost USB jack on the board labeled BLASTER
- Switch S9 should be in the run position

Testing the DE2 Board:

- Toggle the Switch SW[0], SW[1], SW[2] to different eight input patterns
- Check the output in LEDR[0] and LEDR[1]
- For every combination of the inputs the LEDR [0] and LEDR [1] will glow when the output is 1 and if the output is 0 for both LEDR [0] and LEDR [1] it will not glow.

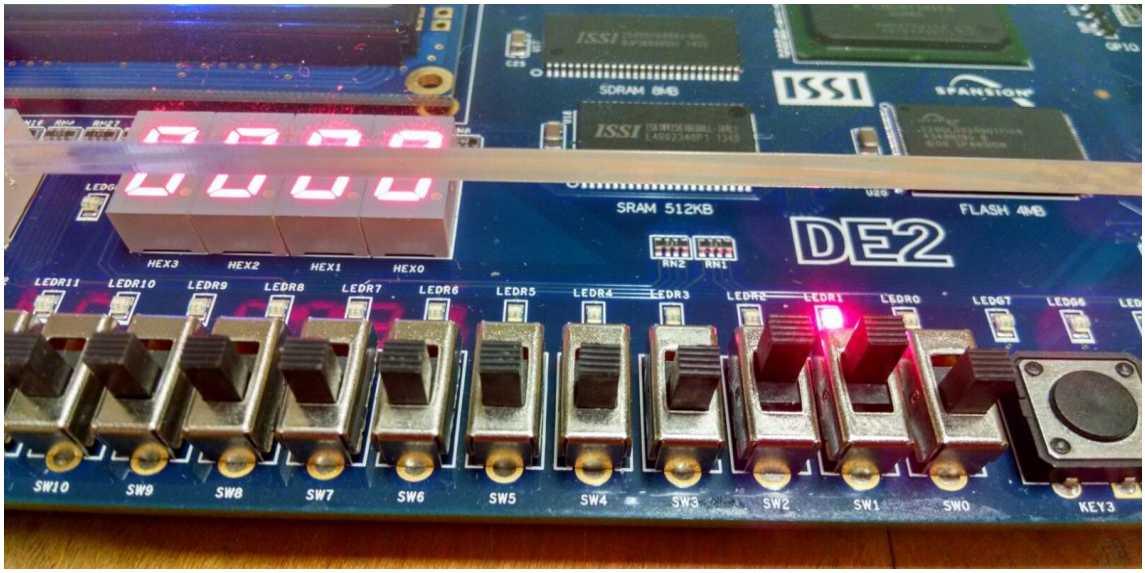


Fig.3 DE2 Board

Below is the truth table which shows the output S and Cout

Input bit SW[0]	Input bit SW[1]	Carry bit input SW[2]	Carry bit output LEDR[1]	Sum bit output LEDR[0]
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

ARNIKA VISHWAKARMA (@01367603)

Lab partners – Pravalika, Lina Mi