

Digital Design Laboratory

Lab 1: Introduction to Xilinx ISE 8.2i

Purpose

In this introductory lab, you will learn how to:

- Design adders using a schematic diagram
- Simulate a logic circuit
- Use VHDL to program a circuit
- Implement a design onto an FPGA

Pre-lab and Background Information

Before coming to the lab, read the pre-lab section and answer all of the questions. Write the answers in your lab notebook for reference during the lab session.

1. A **half adder** (HA) is a circuit that adds two bits together and outputs a sum, S, and Carry, C, shown below in the following examples:

A	1	1
+ B	+ 0	+ 1
-----	-----	-----
C S	01	10

2. The first example adds 1 + 0, resulting in a sum S = 1 and carry C = 0. The second example adds 1 + 1, resulting in a Carry = 1 and Sum = 0. This can be accomplished with a few logic gates, shown in the figure below. If you are not familiar with logic gates, a [short introduction](#) is available on the web.

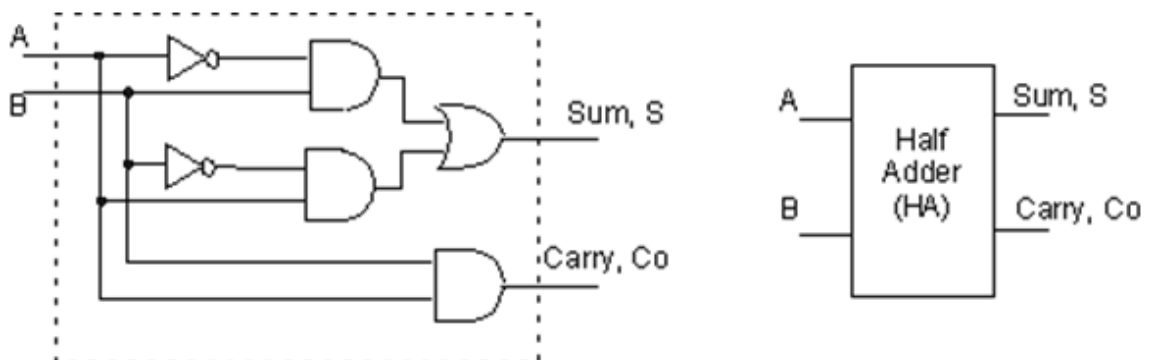


Figure 1: Schematic of a half adder circuit (HA)

- The HA circuit consists of three AND gates, two inverters, and one OR gate. You can easily verify for yourself that this circuit performs the desired function by applying an input gate and checking what the output is. Do this for all four possible combinations of the input signals A and B, that is $AB = 00, 01, 10, \text{ and } 11$. Fill out the table below and record in your notebook.

Table 1 for a Half Adder

Inputs		Outputs	
A	B	S	C
0	0		
0	1		
1	0		
1	1		

- Another type of adder is the **full adder**, which you will learn more about in the lab. The adder can be built from two half adders and an OR gate.

Pre-Lab Reading:

- Read the section on "[Lab Safety](#)."
- Read the section on "[Introduction to Xilinx ISE](#)" including design flow, project navigator, and device types.
- Read the [tutorial on Half Adder](#), including Schematic Entry, Simulation and Implementation (skim this; you should be familiar with it, but you will complete this entire procedure in lab)
- Read the tutorial on "[Basic Coding in VHDL](#)."

Pre-Lab Questions:

Before coming to lab, write the answers to the following questions in your notebook. Also complete the questions on Blackboard before Tuesday noon, the first day of lab.

- Questions on safety
- Questions about Xilinx

In-Lab Assignment
F. Ketterer Lab, 204 Moore

Parts and Equipment:

1. PC with Xilinx 8.2i software
2. [Virtex-II Development FPGA Board](#)

Procedure, Part I – Half Adder

Read the instructions carefully while doing the lab. **Take screenshots of all schematics, code, and test waveforms!** While doing the experiments, you may want to write down notes in your notebook. Your report and notebook must include the procedure you followed, as well as any observation and results. For an example of a notebook entry, click [here](#). For this part, please refer to the half-adder tutorial for more information.

1. First, create a folder *C:\users\your_name* on the C drive in which you will save your project.
2. Start with the [Xilinx tutorial](#) on the lab web pages to get familiar with creating a new project, using the schematic entry and simulator. **Name your project MYFA (not MYHA) and your first schematic MYHA.**
3. First you will build a half adder as shown in the figure below. This is the same circuit as used in the tutorial.

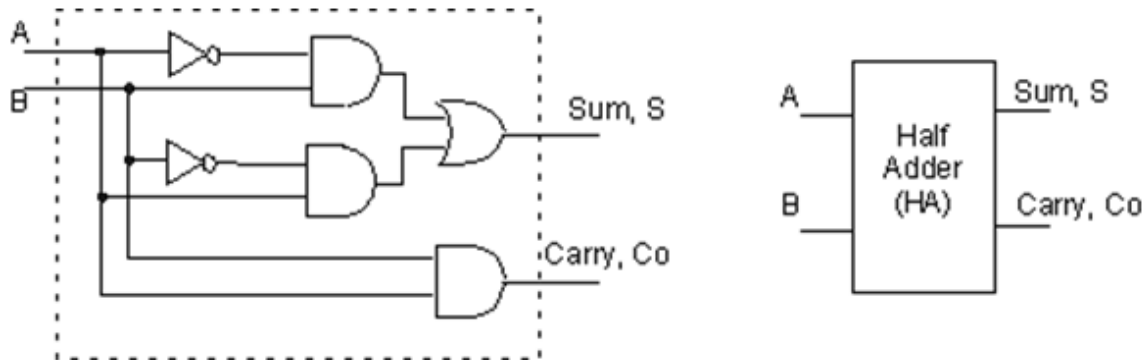


Figure 2: Schematic and symbol of a half adder (HA)

4. When finished with the schematic of the HA, follow through with the instructions on design and behavioral simulation of the circuit. Verify that the circuit works properly: referring to the waveform, fill out the table similar to Table 1 in your lab notebook (you also need to repeat this table in your lab report, with a brief discussion).

After verifying that the circuit works properly, take a snapshot of the simulation results and circuit schematic for inclusion in your report. When finished with the XOR circuit, continue with the next instructions.

5. Now that you have learned how to use Xilinx to create a half adder, we will explore the operation of the full adder.

Procedure, Part II – Full Adder

1. The **full adder** is a circuit very similar to the half adder. This adds three input bits: A, B, and Ci (called a Carry-In). The outputs are the same: S and Co (called a Carry-Out). Below are some examples of how the full adder works:

A	1	1	1
B	0	1	1
+ Ci	+ 0	+ 0	+ 1
-----	-----	-----	-----
Co S	01	10	11

2. Fill out the table below for the full adder in your notebook, **based on the logic of adding the terms A, B, and Ci.**

Table 3 for a Full Adder (based on logic)

Inputs			Outputs	
A	B	Ci	S	Co
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

3. The full adder can be implemented using two half adders and an OR gate as shown in the logic circuit below.

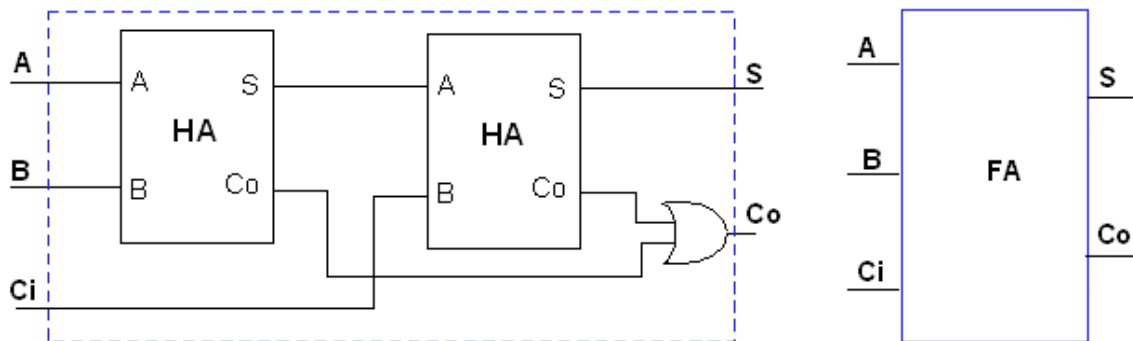


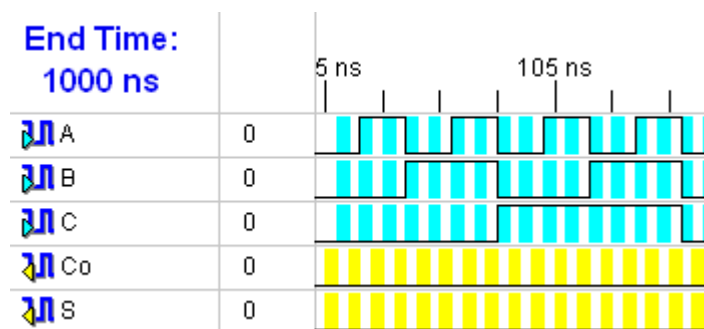
Figure 3: Schematic and symbol of a full adder circuit using two half adders.

3. To verify that this circuit indeed implements a FA, fill out the table below, using the definition of the HA. Include this result in your lab notebook. You can simply append Table 3 to compare the two sets of results.

Table 4 for a Full Adder (based on schematic)

Inputs (Full Adder)			Outputs (Logic)		Outputs (Schematic)	
A	B	Ci	S	Co	S	Co
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

- You will build the full adder using the circuit of Fig. 3. First you need to create a schematic symbol for the half adder circuit. Call it "MYHA". This circuit becomes a **macro** (or module) that can be used later on. To create a circuit symbol, select the MYHA file in the Sources Window. In the Processes window, double click on "Create Schematic Symbol" under the Design Utilities process. This will add a new symbol with the same name MYHA.
- Add a new schematic (add new source) and call it MYFA. Make this the top level module. With two half adders and one OR gate, you can now complete the schematic of the full-adder.
- When you finish the schematic, do a behavioral simulation of the full adder. To generate the input values, you can use the Pattern Wizard (right click on the waveform and select Set Value: Pattern Wizard). Define the Pulse so that it looks like the figure below. Notice that all 8 possible combinations of the inputs are shown. Verify that it works properly according to the table 2.



- Now you can do a Post-Place & Route simulation of the full adder. This will give the exact timing information (delays of the outputs). To do the timing simulation, select Post-Route Simulation from the drop down menu. In the Process window, select Simulate Post-Place & Route. Zoom in on the waveform and use the markers to determine the delay of the output signal S when the input A goes from 0 to 1 (B=C=0). How much is the delay of Co after both A and B go high?

8. Download the design to the board and verify the operation. Give a demo to the TA who will help verify the operation.

Procedure, Part III –Half Adder in VHDL

1. Now we will use VHDL to construct a Half Adder. VHDL is a programming language used to code logic circuits, as an alternative to schematics. Read the [Basic Coding in VHDL](#) tutorial and construct the HA in code. Using the Figure 2, translate the schematic into logical code. For a refresher on logic gates read the [short introduction](#) to gates.

You only have to verify the logic by waveform simulation, not FPGA implementation. Make sure to take a snapshot of your code and simulation before moving on.

2. Before leaving the lab, archive your project and store it on your seas account or on a memory stick. Now, delete your project from the computer. To archive a project, select Project > Archive in the Project Navigator Window. This will create a zip file of all files, and directories needed for your project. To restore the archived project you will first need to unzip it outside the ISE program.

Hand-In (At the start of the next lab)

You have to hand in a report that contains the following (See [guidelines](#) for reports):

1. Course title, Lab number, Lab title, Your names, and date
2. Section on the pre-lab showing the filled tables. Answers submitted online do not have to be repeated here.
3. Brief description of the experiment including the goals and theory
4. Circuit schematics (**screenshots**, include your name)
5. Waveform simulations (**screenshots**)
6. Discussion of the results indicating proper function, using the waveform and completed table 4.
7. Conclusion

Tips on writing your report:

1. You can provide a table of values based on the waveform. By referring to the time periods for each value, you can explain how the simulator provided the correct results (comparing it to the pre-lab truth tables).
2. You may find it easier to walk through the procedure.
3. Name your image files clearly so you can organize them when writing your report.

The lab report is an important part of the laboratory. Write it carefully, be clear, and well-organized. It is the only way to convey that you did a great job in the lab. It is preferred (but not necessary) that you type the report. The report is due at the start of the next lab session.

Updated January 3, 2011.