Lab 3 – Ripple Carry Adder

CS 2052 Computer Architecture

Dept. of Computer Science and Engineering, University of Moratuwa

Learning Outcomes

In this lab we will design a 4-bit Ripple Carry Adder (RCA). After completing the lab, you will be able to:

- design and develop a Half Adder, Full Adder, and a Ripple Carry Adder
- build more complex components using several basic components
- verify their functionality via simulation and on the development board

Introduction

The adder/subtractor unit is perhaps the most important element of an Arithmetic and Logic Unit (ALU) of a microprocessor. In this lab, we will also learn about hierarchical design where more complex components are built using many basic components. Those high-level components may be further combined to build even larger components. For example, we will use 2 Half Adders (Has) to build a Full Adder (FA) and multiple FAs are used to build the RCA. Later in the class and labs you will also learn how to extend your RCA to support subtractions.

Building the Circuits

We will first build an HA, then an FA, and finally the 4-bit RCA.

Step 1: Finding Boolean expression for HA and FA.

Write the truth table for a HA and FA. Simplify the HA and FA Boolean equations such that FA can be built using 2 HAs.

Step 2: Building HA.

Create a new project in Xilinx ISE Design Suite and name it as **Lab 3**. Build a HA using basic logic gates such as AND, OR, XOR, or NOT. Name your schematic file as **HA**. Label the inputs to HA as **A** and **B** and outputs as **S** and **C**.

Test the functionality of the HA by simulating the circuit. Name the Test Bench File as **TB_HA**.

Step 3: Creating an HA symbol.

Now let us create a macro symbol for future use of this half adder.

To create a macro circuit, select **Tools → Symbol Wizard** from the main menu. This will open up the Symbol Wizard dialogue box.

Under **Pin name source**, select **Using schematic** radio button. On the corresponding drop down list you should see **HA** listed.

Click **Next >** button. The next 2 dialogue boxes present general pin placement and symbol size options. You can modify those options depending on how you want the schematic symbol to appear. The final dialogue box gives you a preview of the symbol, given the settings you have selected. Click **Finish** to finalize the symbol.

Save changes to the current schematic editor session before closing the schematic file and return to the Project Navigator main window.

Step 4: Building FA.

Create a new schematic file and save it as **FA**. Now you should see **HA** listed as one of the symbols.

Using one or more HAs and basic logic gates build an FA. Label the inputs to FA as **A**, **B**, and **C_in** and outputs as **S** and **C_out**.

Test the functionality of the FA by simulating the circuit. Name the Test Bench File as **TB_FA**.

Create a new symbol and name it as **FA**.

Step 5: Build a 4-bit RCA.

Create a new schematic file and save it as **4_RCA**. Now build a 4-bit RCA using multiple FAs. Label the inputs to RCA as **A0-A3**, **B0-B3**, and **C_in**. Label the outputs as **S0-S3** and **C_out**.

Simulate the RCA and make sure it functions correctly. Name the Test Bench File as **TB_4_RCA**. It is not essential to try all the possible combinations of inputs. Instead, you should at least try the following input combinations:

- Consider the 6 digits of your index number. Then convert your index number to binary. Then take 4 Least Significant Bits (LSBs) and add them with the next 4 LSBs. Then try the next set of 4-bits (ignore any remaining bits). For example, suppose your index number is 123456R. Then its binary representation is 01 1110 0010 0100 0000 (ignoring the check digit). Then try 0000 + 0100 and 0010 + 1110.
- Also try 0101 + 1011 and 0111 + 1111
- Any 4 other unique combinations

Step 6: Connecting inputs and outputs.

Input and output pins are connected only to the top-level design which is the RCA. Therefore, first we have to set our RCA as the Top Design Unit. Select **RCA** from the Sources window. Then use **Source → Set as Top Model** from the main menu. Now we are ready to assign pins.

Connect switches SW0-SW3 as the inputs **A0-A3** and SW4-SW7 as the inputs **B0-B3**. Connect outputs **S0-S3** to LED LD0-LD3 and **C_out** to LED LD7. Connect **C_in** to ground.

Step 7: Test on BASYS2.

Generate the programming file (i.e., bitstream) and load it to the BASYS2 board.

Change the switches on the BASYS 2 and verify the functionality of your RCA (check the output of LEDs).

Demonstrate the circuit to the instructor and get the Lab Completion Log singed.

Step 8: Lab Report

You need to submit a report for this lab. Your report should include the following:

- Student name, index number, and group. Do not attach a separate front page
- State the assigned lab task in a few sentences
- Truth tables and steps involved in simplifying the Boolean expressions
- All schematic circuits (label figures as Annex 1, Annex 2, ...)
- All timing diagrams. Show all possible inputs for HA and FA. For 4-bit RCA provide at least the inputs listed under Step 5
- Discuss why some of the input combinations results in outputs that cannot be represented using LED LD0-LD3. Discuss the role of LD7.
- Other conclusions from the lab.

Submit the lab report at the beginning of the next lab.

Prepared By

- Dilum Bandara, PhD – Feb 26, 2014
- Updated on Sep 07, 2017