

# VHDL Assignment #4: Design and Simulation of Adder Circuits

In this lab, you will implement a 4-bit ripple-carry adder in VHDL using both behavioral and structural descriptions.

## 1 Learning Outcomes

After completing this lab you should know how to

- Synthesize logic functions
- Use CAD tools and VHDL to implement ripple-carry adders
- Perform functional simulation

## 2 Prerequisites

Before starting this lab you should

- Read the posted lecture slides (Lecture #13) and the corresponding textbook sections related to adder circuits.
- Be familiar with the information in previous coding assignments.
- Read Sections 5.5.2, 5.5.3, and 5.5.4.

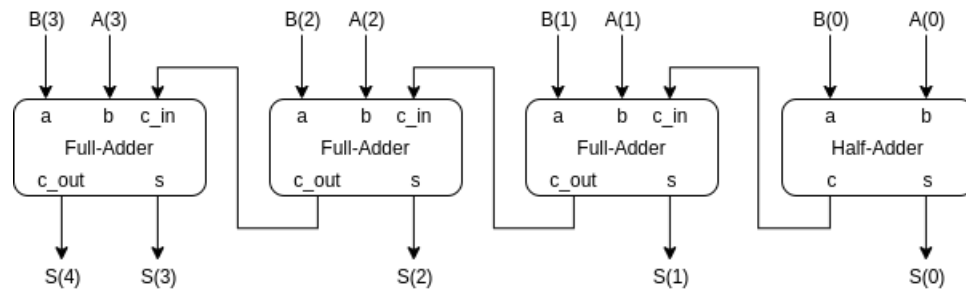
If you need any help regarding the lab materials, you can

- Ask the TA for help during lab sessions on office hours.
- Refer to the text book. In case you are not aware, Appendix A “VHDL Reference” provides detailed information on VHDL.
- You can also refer to the tutorial on Quartus and ModelSim provided by Intel ([click here for Quartus](#) and [here for ModelSim](#)).

It is highly recommended that you first try to resolve any issue by yourself (refer to the textbook and/or the multitude of VHDL resources on the Internet). Syntax errors, especially, can be quickly resolved by reading the error message to see exactly where the error occurred and checking the VHDL Reference or examples in the textbook for the correct syntax.

### 3 Structural Description of a 4-bit Ripple-Carry Adder

In this section, you will implement a structural description of a 4-bit ripple-carry adder out of basic addition components: half-adders and full-adders.



#### 3.1 Structural Description of a Half-Adder in VHDL

A half-adder is a circuit that takes two binary digits as inputs, and outputs the result of the addition of the two digits in the form of sum and carry signals. The carry signal represents an overflow into the next digit of a multi-digit addition. As a first step to build a structural description of a 4-bit ripple-carry adder, *implement a structural description of the half-adder in VHDL* using the following entity definition for your VHDL code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity firstname_lastname_half_adder is
    Port (a      : in  std_logic;
          b      : in  std_logic;
          s      : out std_logic;
          c      : out std_logic);
end firstname_lastname_half_adder;
```

After you have described your structural style of the half-adder in VHDL, you are required to test your circuit. Write a testbench code and perform an exhaustive test for your VHDL descriptions of the half-adder.

#### 3.2 Structural Description of a Full-Adder in VHDL

You will now implement a full-adder circuit. Unlike the half-adder, a full-adder adds binary digits while accounting for values carried in (from a previous stage addition). Write a structural VHDL description for the full-adder circuit *using two instances of the half-adder circuit that you designed earlier*. Use the following entity declaration for your structural VHDL description of the full-adder:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity firstname_lastname_full_adder is
    Port (a      : in  std_logic;
          b      : in  std_logic;
          c_in   : in  std_logic;
          s      : out std_logic;
          c_out  : out std_logic);
end firstname_lastname_full_adder;
```

After you have described your circuit in VHDL, write a testbench code and perform an exhaustive test for your VHDL descriptions of the full-adder.

### 3.3 Structural Description of a 4-bit Ripple-Carry Adder (RCA) in VHDL

After you have implemented both the full-adder and the half-adder circuits in VHDL, implement a 4-bit carry-ripple adder *using instances of these basic components (i.e., full-adder and half-adder components)*. Write a structural VHDL code for the 4-bit RCA using the following entity declaration:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity firstname_lastname_rca_structural is
    Port (A          : in  std_logic_vector (3 downto 0);
          B          : in  std_logic_vector (3 downto 0);
          S          : out std_logic_vector (4 downto 0));
end firstname_lastname_rca_structural;
```

Note that S(4) contains the carry-out of the 4-bit adder. After you have described your circuit in VHDL, write a testbench code and perform an exhaustive test for your VHDL descriptions of the 4-bit RCA.

## 4 Behavioral Description of a 4-bit RCA in VHDL

In this part, you are required to implement the 4-bit RCA using behavioral description. One way to obtain a behavioral description is to use arithmetic operators in VHDL (i.e., "+"). Write a behavioral VHDL code for the 4-bit RCA using the following entity declaration for your behavioral VHDL description:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity firstname_lastname_rca_behavioral is
    Port (A          : in  std_logic_vector (3 downto 0);
          B          : in  std_logic_vector (3 downto 0);
          S          : out std_logic_vector (4 downto 0));
end firstname_lastname_rca_behavioral;
```

After you have described your circuit in VHDL, write a testbench code and perform an exhaustive test for your VHDL description of the 4-bit RCA.

## 5 Deliverables and Grading

**Read this section extra carefully!**

You are required to submit the following files through myCourses:

- A report that includes the answers to the questions in Section 6. Use the report template (provided in .docx format) that is attached to the assignment. Make sure that all figures are clearly visible and contain relevant information (*e.g.*, axes and relevant signal names). Save your report in a .pdf format (include your name in the filename).
- All design files (.vhd)
- All schematic files (.bdf)
- All testbench files (.vht)

Per the VHDL assignments submission policy, please note

- For partially submitted assignments, *i.e.*, some of the files (design/simulation/report) are missing, the penalty is 25% of the full mark of the assignment.
- For assignments where all the design/simulation files are missing, *i.e.*, only a report was submitted, the penalty is 50% of the full mark of the assignment.
- For not submitted assignments, *i.e.*, all files are missing, the penalty is 100% of the full mark of the assignment.

## 6 Questions

1. Report the number of pins and logic modules used to fit your designs on the FPGA board. (1 mark)

	Half-Adder	Full-Adder	4-bit Ripple Carry Adder	
	Structural	Structural	Structural	Behavioral
Logic Utilization (in ALMs)				
Total pins				

2. Provide the VHDL code that you wrote to instantiate the half-adders in Part 3.2 and the full-adders in Part 3.3. (1 mark)
3. Show representative simulation plots of the half-adder circuit for all the possible input values. (1 mark)
4. Show representative simulation plots of the full-adder circuit for all the possible input values. (1 mark)
5. Show representative simulation plots of both the behavioral and structural 4-bit ripple-carry adder descriptions for all the possible input values. (1 mark)