Lab 4

Ly Nguyen

February 6, 2022

Programmable Devices

Introduction

In this lab, we practiced the use of components through the creation of a full adder circuit and by creating 4-bit and 8-bit adders in different ways.

Equipment and Method

Programs:

Quartus Prime Version 20.1.1 Build 720 11/11/2020 SJ Elite Edition

ModelSim SE-64 2021.2

Procedure:

Follow the lab handout as written. For building 4-bit and 8-bit adders, follow the lecture slides on components where it outlines how to implement an adder.

Results

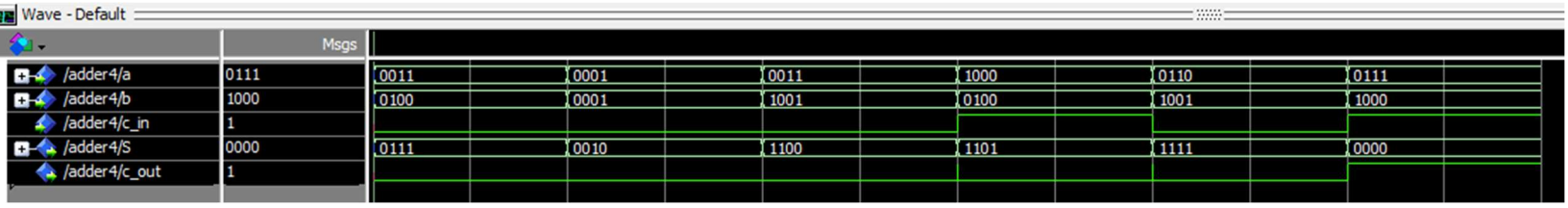
Adder4.vhd:

```
1   LIBRARY IEEE;
2   USE IEEE.std_logic_1164.ALL;
3   ---------------------------------------------
4   ENTITY adder4 IS
5   PORT  (a, b: IN std_logic_vector (3 DOWNTO 0);
6          c_in: IN std_logic;
7          S: OUT std_logic_vector (3 DOWNTO 0);
8          c_out: OUT std_logic);
9   END adder4;
10  ---------------------------------------------
11  ARCHITECTURE structure OF adder4 IS
12     COMPONENT fulladder IS
13        PORT (x, y, c_in: IN STD_LOGIC;
14              sum, c_out: OUT STD_LOGIC);
15     END COMPONENT;
16     SIGNAL C: std_logic_vector (4 DOWNTO 0);
17  BEGIN
18     C(0) <= c_in;
19     c_out <= C(4);
20     FA0: fulladder PORT MAP (a(0), b(0), C(0), S(0), C(1));
21     FA1: fulladder PORT MAP (a(1), b(1), C(1), S(1), C(2));
22     FA2: fulladder PORT MAP (a(2), b(2), C(2), S(2), C(3));
23     FA3: fulladder PORT MAP (a(3), b(3), C(3), S(3), C(4));
24  END ARCHITECTURE;
```

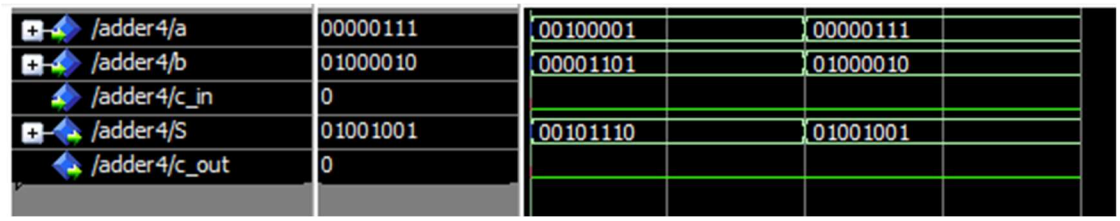
Exercise 2:

C:/Users/lnguyen5/Desktop/lab 4/

```
Ln#
 1   radix symbolic
 2   force c_in 0
 3   force a x"3"
 4   force b x"4"
 5   run
 6   force a x"1"
 7   force b x"1"
 8   run
 9   force a x"3"
10   force b x"9"
11   run
12   force c_in 1
13   force a x"8"
14   force b x"4"
15   run
16   force c_in 0
17   force a x"6"
18   force b x"9"
19   run
20   force c_in 1
21   force a x"7"
22   force b x"8"
23   run
```



Exercise 3:



References

Lecture slides and lab handout