Experiment No. 6
High-Speed Adder
ECE 446

Peter Chinetti

October 7, 2014

Date Performed: September 28, 2014
Instructor: Professor Shanechi

1 Introduction

Although the ripple adder used in earlier labs is architecturally simple, the delay introduced through ripple propagation is problematic for larger word widths.

A solution has been designed to allow parallel addition more quickly. It introduces a carry and propagate signal for each bit, which are the result of operations on the previous bits.

The downside of this approach is the increase in complexity, and therefore transistor count, of the High-Speed Adder.

2 Procedure

a. Write VHDL to implement adder.

b. Assign pins to ports

c. Simulate

d. Program and Test

3 Equipment

- PC
- Spartan-3E development board

4 Code

4.1 Top-level Module

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity claadder is
  Port ( cin : in  STD_LOGIC;
         co : out  STD_LOGIC;
         A : in  STD_LOGIC_VECTOR (3 downto 0);
         B : in  STD_LOGIC_VECTOR (3 downto 0);
         S : out  STD_LOGIC_VECTOR (3 downto 0));
end claadder;

architecture Behavioral of claadder is

  signal ci : STD_LOGIC_VECTOR (2 downto 0);
  signal gi, pi, si, bi : STD_LOGIC_VECTOR (3 downto 0);

  component CLA
    Port ( g, p : in STD_LOGIC_VECTOR (3 downto 0);
           c : out STD_LOGIC_VECTOR (3 downto 0);
           c0 : in STD_LOGIC
         );
  end component;

  component propgen
    Port ( a : in  STD_LOGIC_VECTOR (3 downto 0);
           b : in  STD_LOGIC_VECTOR (3 downto 0);
           p : out  STD_LOGIC_VECTOR (3 downto 0);
           g : out  STD_LOGIC_VECTOR (3 downto 0)
         );
  end component;

  component adder
    Port ( a, b   : in  STD_LOGIC_VECTOR(3 downto 0);
           c   : in  STD_LOGIC_VECTOR(3 downto 0);
           sum : out  STD_LOGIC_VECTOR(3 downto 0)
         );
  end component;

begin

  cla_0 : CLA
  port map (
       g => gi,
       p => pi,
       c0 => cin,
       c(2 downto 0) => ci,
       c(3) => co
     );

  propgen_0 : propgen
  port map (
       a => A,
       b => bi,
```

```
        p => pi,
        g => gi
      );

  process (B, cin)
  begin
    if cin = '0' then
      bi <= B;
    elsif cin = '1' then
      bi <= not B;
    end if;
  end process;


  adder_0 : adder
  port map (
        a => A,
        b => bi,
        c(0) => cin,
        c(3 downto 1) => ci,
        sum => S
      );

end Behavioral;
```

claadder.vhd

4.2 CLA Module

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity CLA is
  Port ( g : in  STD_LOGIC_VECTOR (3 downto 0);
         p : in  STD_LOGIC_VECTOR (3 downto 0);
         c : out  STD_LOGIC_VECTOR (3 downto 0);
         c0 : in STD_LOGIC);
end CLA;

architecture Behavioral of CLA is

begin

  c(0) <= g(0) or (p(0) and c0);
  c(1) <= g(1) or (p(1) and (g(0) or (p(0) and c0)));
  c(2) <= g(2) or (p(2) and (g(1) or (p(1) and (g(0) or (p(0) and
    c0)))));
  c(3) <= g(3) or (p(3) and (g(2) or (p(2) and (g(1) or (p(1) and (
    g(0) or (p(0) and c0)))))));

end Behavioral;
```

CLA.vhd

4.3 Propagation Module

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity propgen is
  Port ( a : in  STD_LOGIC_VECTOR (3 downto 0);
         b : in  STD_LOGIC_VECTOR (3 downto 0);
         p : out  STD_LOGIC_VECTOR (3 downto 0);
         g : out  STD_LOGIC_VECTOR (3 downto 0));
end propgen;

architecture Behavioral of propgen is

begin

  g <= a and b;

  p <= a or b;

end Behavioral;
```

propgen.vhd

4.4 Test

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY claadder_test IS
  END claadder_test;

ARCHITECTURE behavior OF claadder_test IS

    -- Component Declaration for the Unit Under Test (UUT)

  COMPONENT claadder
    PORT(
          cin : IN  std_logic;
          co : OUT  std_logic;
          A : IN  std_logic_vector(3 downto 0);
          B : IN  std_logic_vector(3 downto 0);
          S : OUT  std_logic_vector(3 downto 0)
        );
  END COMPONENT;


   --Inputs
  signal cin : std_logic := '0';
  signal A : std_logic_vector(3 downto 0) := (others => '0');
  signal B : std_logic_vector(3 downto 0) := (others => '0');

   --Outputs
  signal co : std_logic;
  signal S : std_logic_vector(3 downto 0);

BEGIN
```

```
32
   -- Instantiate the Unit Under Test (UUT)
34   uut: claadder PORT MAP (
                cin => cin,
36              co => co,
                A => A,
38              B => B,
                S => S
40            );

42   -- Stimulus process
   stim_proc: process
44   begin
       -- hold reset state for 100 ns.
46     wait for 10 ns;

48     A <= "0001";

50     wait for 10 ns;

52     B <= "0001";

54     wait for 10 ns;

56     A <= "0010";

58     wait for 10 ns;

60     A <= "1111";

62     wait for 10 ns;

64     A <= "0111";
       cin <= '1';
66
       wait for 10 ns;
68
       B <= "0111";
70
       wait for 10 ns;
72
       B <= "0000";
74     A <= "0000";

76     wait;
   end process;
78
END;
```

claadder_test.vhd

5 Conclusions

The purpose of this lab was achieved. A high-speed adder was built and tested. Operation was verified through simulation and physical implementation.