EGR 234 – Digital Logic Design

Lab 7:

Modular Approach for Adder Implementation

Lab Report by: Christopher Parisi

Lab Partner: No Partner

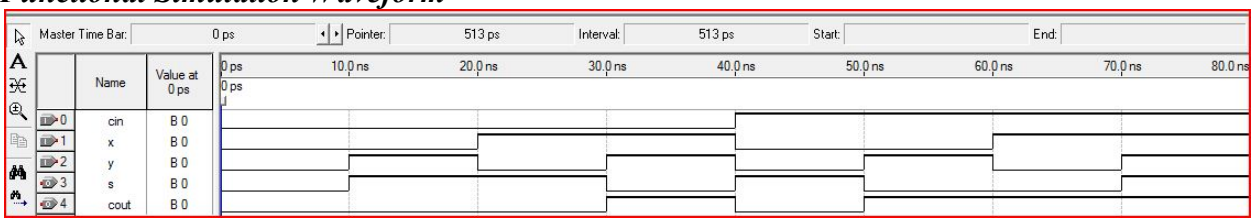
October 15, 2010

Exercise 1

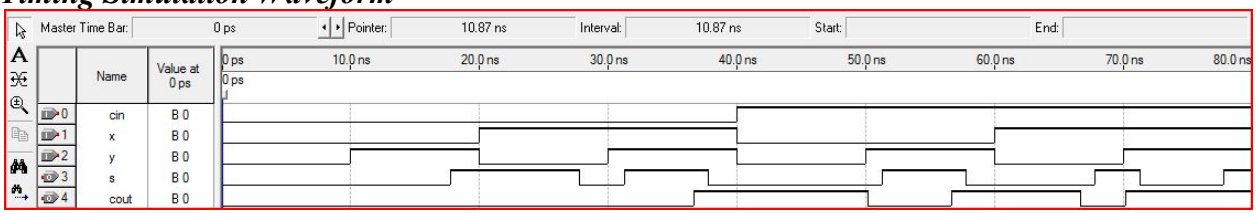
VHDL code for 1-bit full adder

```
------------------------------------------------------------------
-- 1-bit full adder
------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity fa is
port (x, y, cin : in std_logic;
                s, cout : out std_logic);
end fa;

architecture fa_arch of fa is
        signal input : std_logic_vector (2 downto 0);
        signal output : std_logic_vector (1 downto 0);
begin
        input <= cin & x & y;

with input select
        output <= "00" when "000",
                        "10" when "001",
                        "10" when "010",
                        "01" when "011",
                        "10" when "100",
                        "01" when "101",
                        "01" when "110",
                        "11" when "111",
                        "00" when others;

        s <= output(1);
        cout <= output(0);
end fa_arch;
```

Functional Simulation Waveform



Timing Simulation Waveform



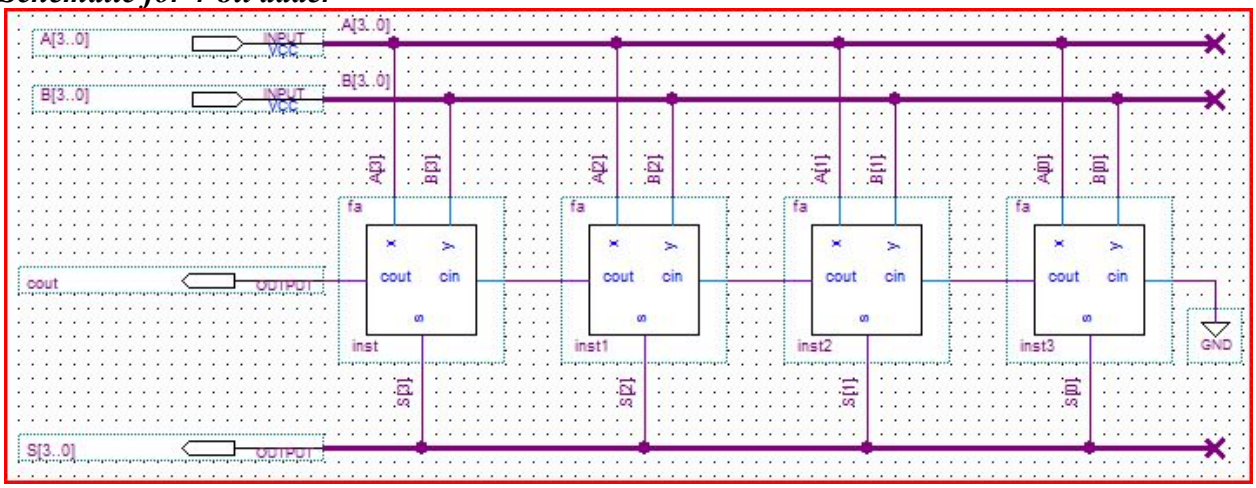
Truth Table

cin	x	y		s	cout
0	0	0		0	0
0	0	1		1	0
0	1	0		1	0
0	1	1		0	1
1	0	0		1	0
1	0	1		0	1
1	1	0		0	1
1	1	1		1	1

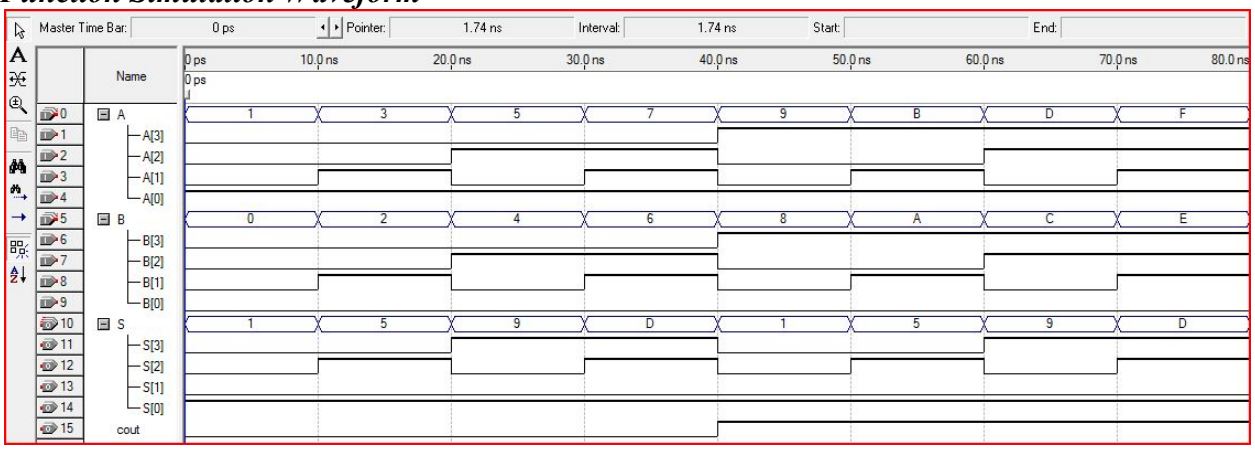
This truth table coincides with the waveform and shows the functionality of the 1-bit adder.

Exercise 2

Schematic for 4-bit adder

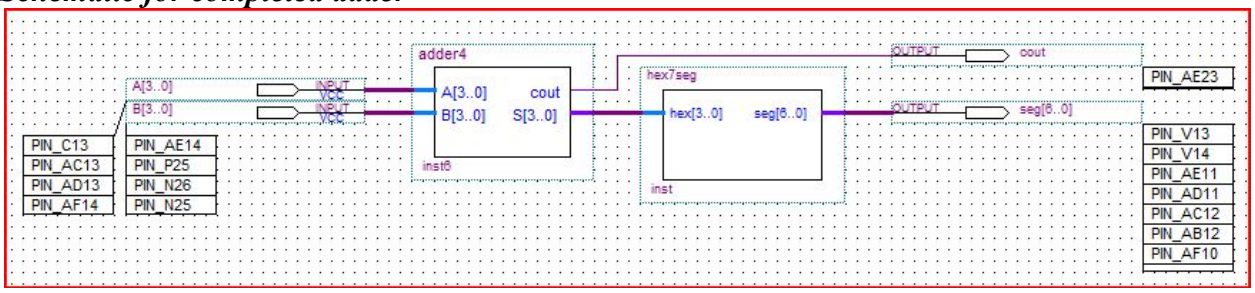


Function Simulation Waveform



Exercise 3

Schematic for completed adder



This schematic was then programmed into the DE2 board and tested. The results match up with the waveform from Exercise 2. This shows that the Modular Approach is capable of solving many problems. The modular approach is easy and organizes the problem making it easier to read and implement.