EGR 234 – Digital Logic Design

Lab 8:

Modular Approach for a Double-Digit BCD Adder

Lab Report by: Christopher Parisi

Lab Partner: No Partner

October 22, 2010

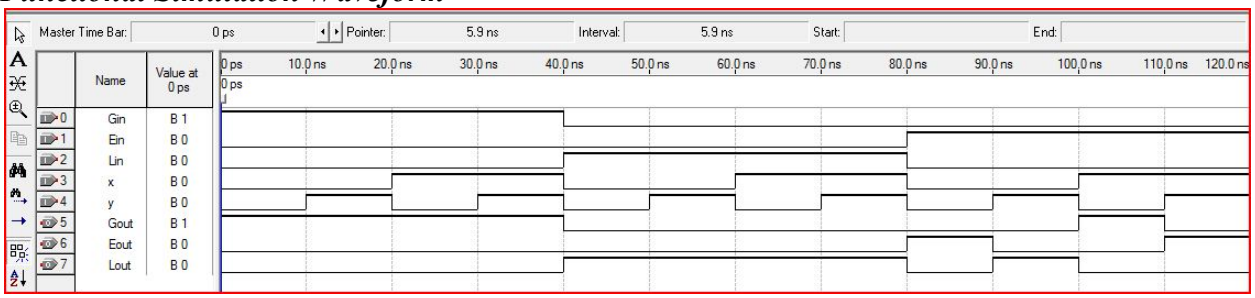
Exercise 1

VHDL Code for Single Bit Comparator Module

```
----------------------------------------------------------------
-- single bit comparator module
----------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;

entity bit_comp is
  port (Gin, Ein, Lin, x, y : in std_logic;
   Gout, Eout, Lout    : out std_logic);
end bit_comp;


architecture bit_comp_arch of bit_comp is
begin

  process(Gin, Ein, Lin, x, y)
  begin
    if   ( Gin = '1' and Ein = '0' and Lin = '0' ) then
                        Gout <= '1';
                        Eout <= '0';
                        Lout <= '0';
    elsif ( Gin = '0' and Ein = '0' and Lin = '1' ) then
                        Gout <= '0';
                        Eout <= '0';
                        Lout <= '1';
    elsif ( Gin = '0' and Ein = '1' and Lin = '0' ) then
                        if ( x > y ) then
                                Gout <= '1';
                                Eout <= '0';
                                Lout <= '0';
                        elsif ( x = y ) then
                                Gout <= '0';
                                Eout <= '1';
                                Lout <= '0';
                        else
                                Gout <= '0';
                                Eout <= '0';
                                Lout <= '1';
                        end if;
     else
                        Gout <= '0';
                        Eout <= '0';
```

```
                    Lout <= '0';
    end if;
  end process;
end bit_comp_arch;
```

Functional Simulation Waveform



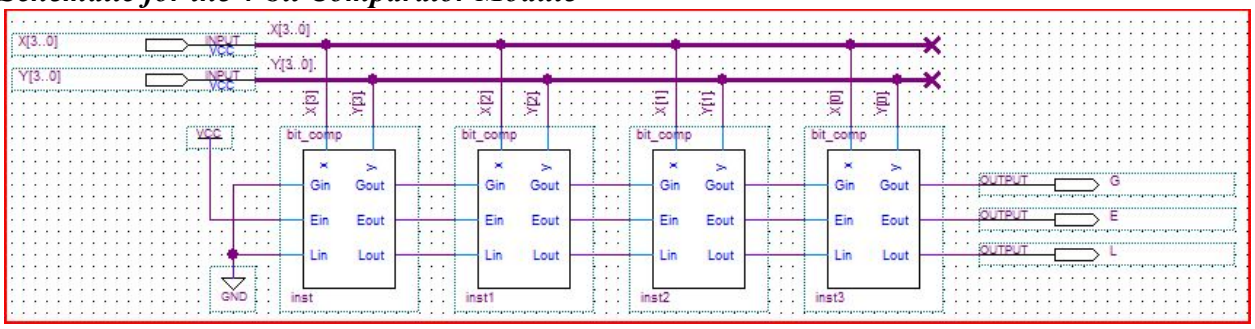
Truth Table for Single Bit Comparator Module

Gin	Ein	Lin	X	Y		Gout	Eout	Lout
1	0	0	0	0		1	0	0
1	0	0	0	1		1	0	0
1	0	0	1	0		1	0	0
1	0	0	1	1		1	0	0
0	0	1	0	0		0	0	1
0	0	1	0	1		0	0	1
0	0	1	1	0		0	0	1
0	0	1	1	1		0	0	1
0	1	0	0	0		0	1	0
0	1	0	0	1		0	0	1
0	1	0	1	0		1	0	0
0	1	0	1	1		0	1	0

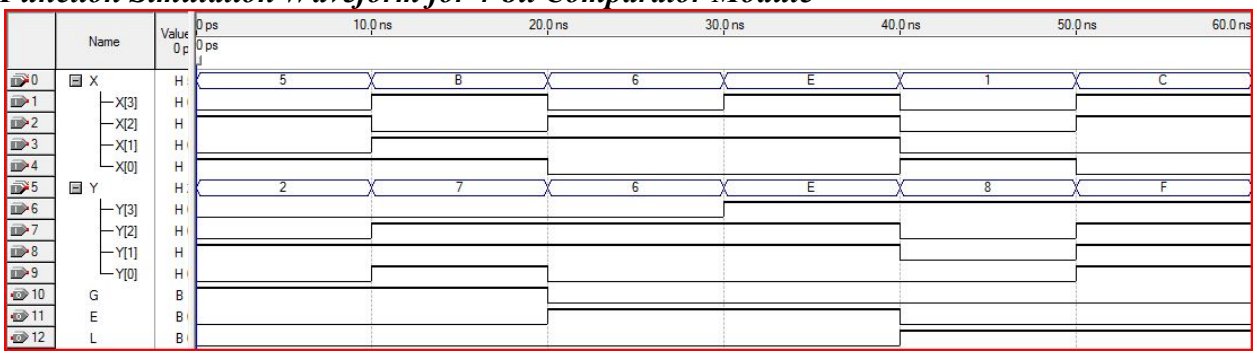
This table accurately interprets the functional simulation and proves the VHDL code works.

Exercise 2

Schematic for the 4-bit Comparator Module

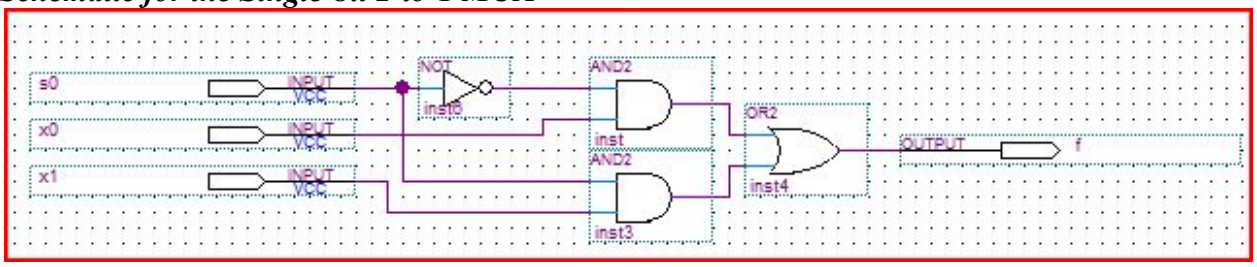


Function Simulation Waveform for 4-bit Comparator Module

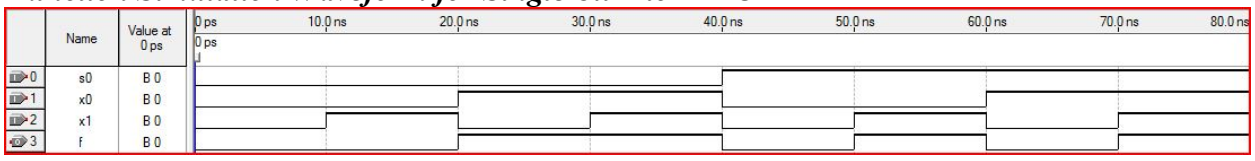


Exercise 3

Schematic for the Single-bit 2-to-1 MUX



Function Simulation Waveform for Single-bit 2-to-1 MUX

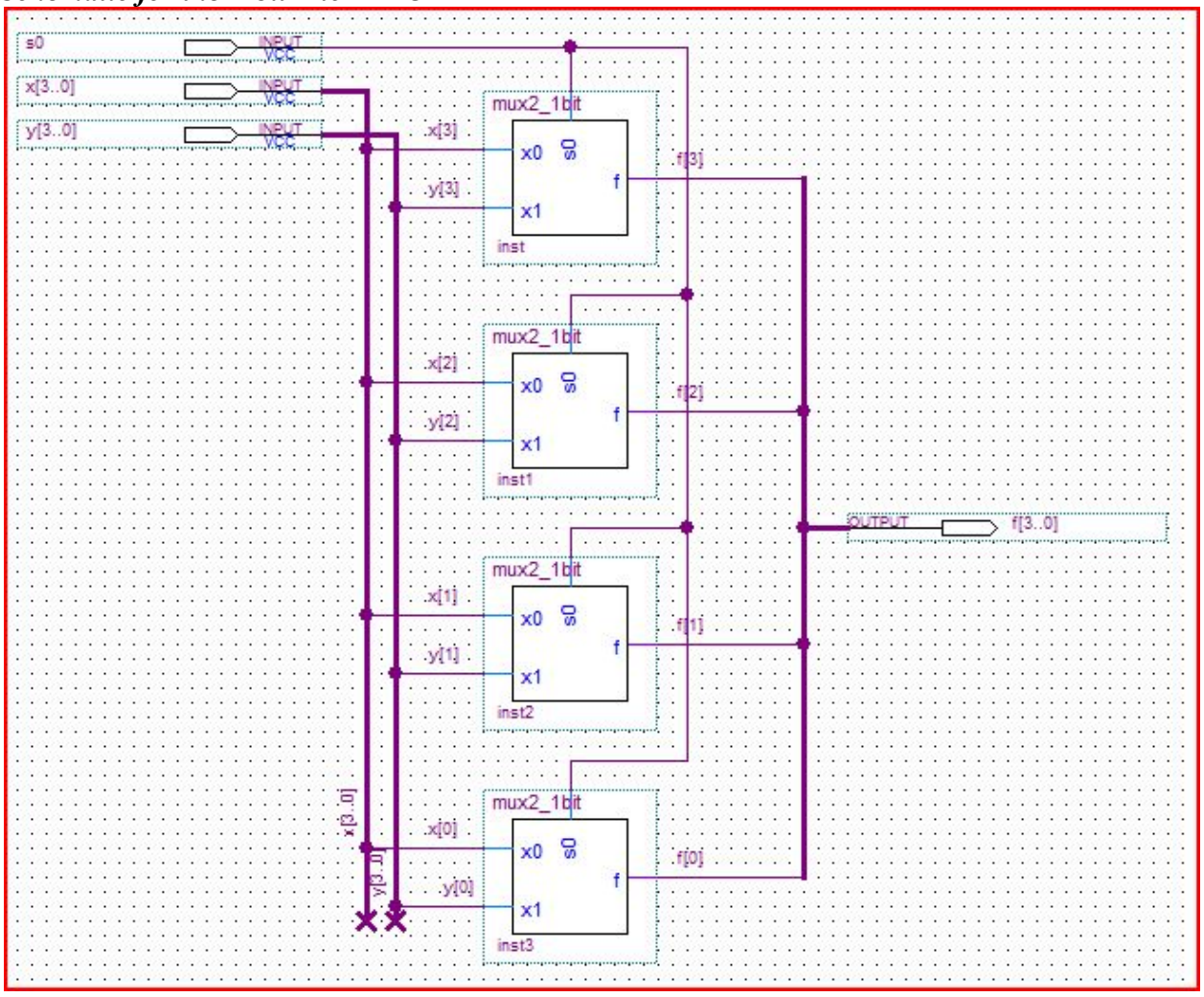


Truth Table for Single-bit 2-to-1 MUX

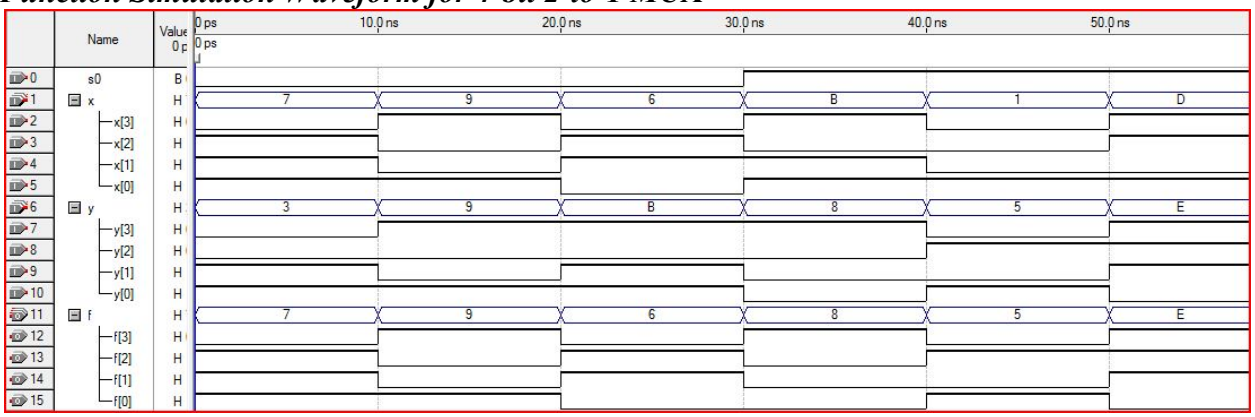
s0	x0	x1		f
0	0	0		0
0	0	1		0
0	1	0		1
0	1	1		1
1	0	0		0
1	0	1		1
1	1	0		0
1	1	1		1

The Truth Table matches up with the functional waveform and shows the 2-to-1 MUX is functioning correctly.

Schematic for the 4-bit 2-to-1 MUX



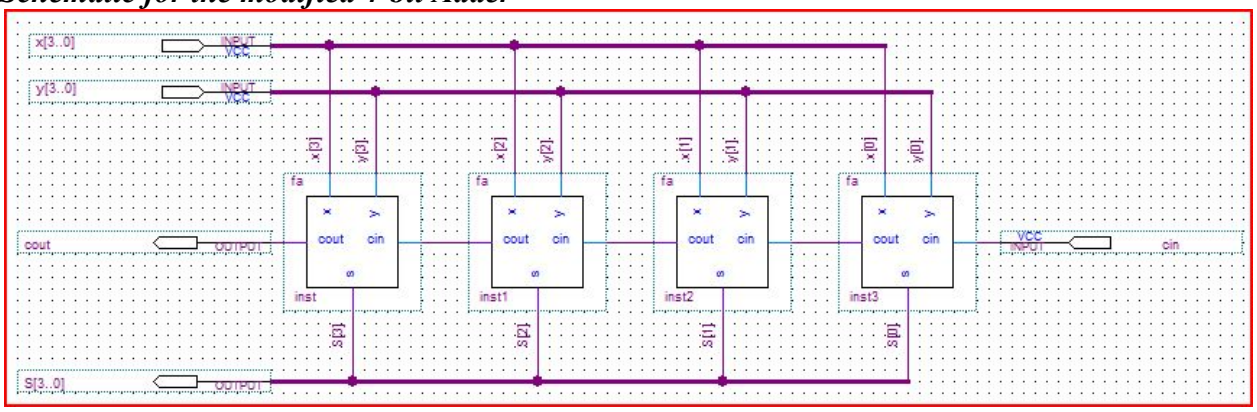
Function Simulation Waveform for 4-bit 2-to-1 MUX



The outputs for the 2-to-1 MUX are correct which shows the MUX is working properly.

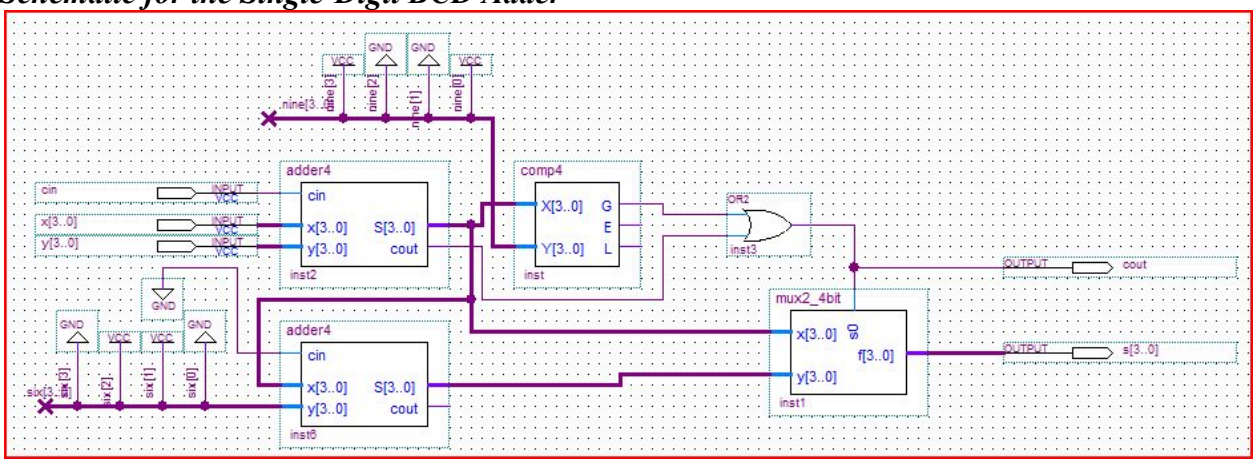
Exercise 4

Schematic for the modified 4-bit Adder

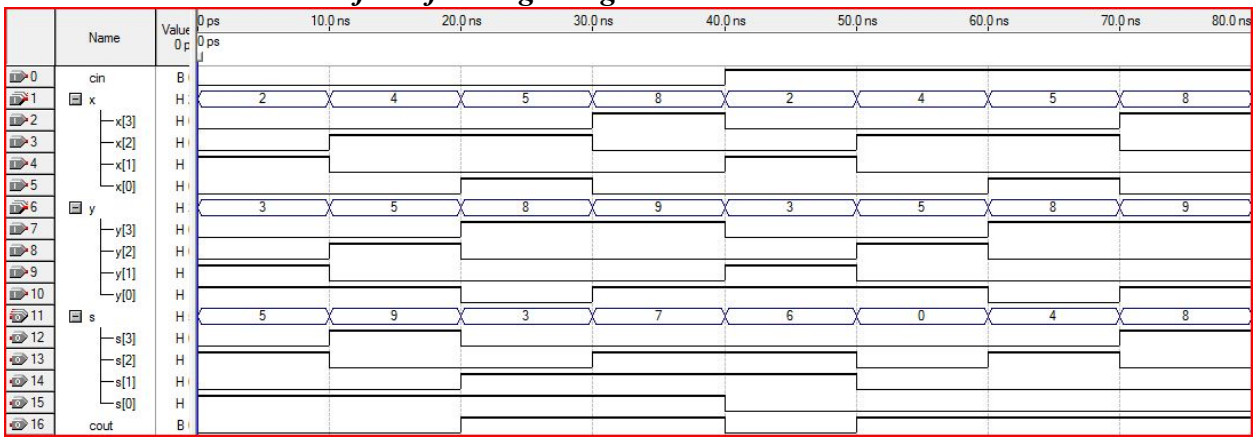


Exercise 5

Schematic for the Single-Digit BCD Adder



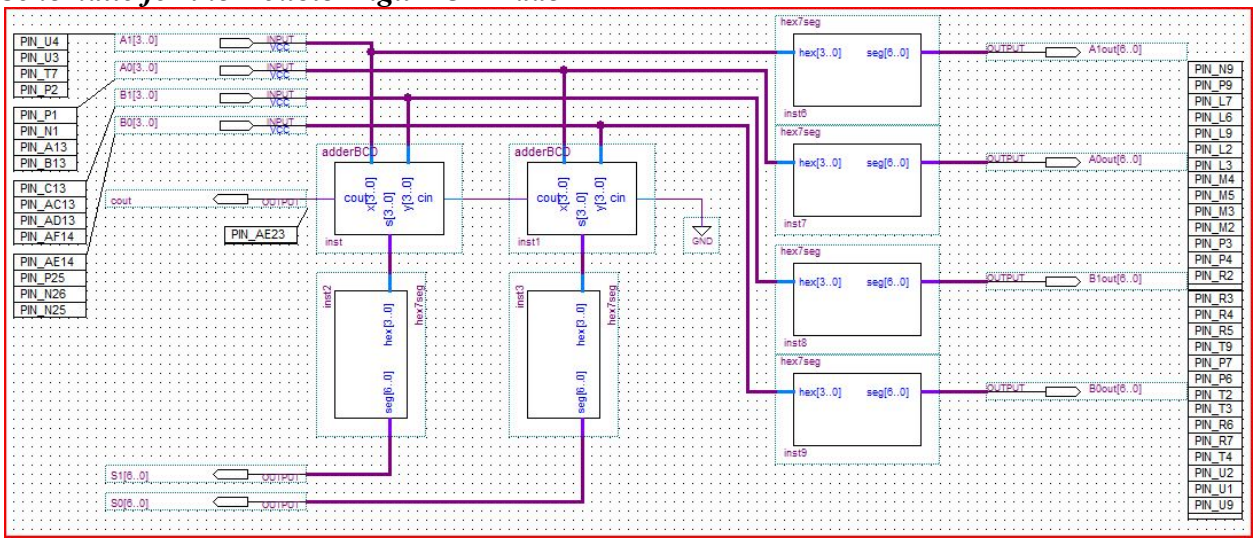
Function Simulation Waveform for Single-Digit BCD Adder



The waveform shows the correct output for the various cases. This shows the single-digit BCD adder is functional.

Exercise 6

Schematic for the Double-Digit BCD Adder



The BCD adder was tested on the DE2 board and functioned as it should have. This shows each module is functional and the overall Double Digit BCD Adder is also correct.