ECSE-323
Digital System Design

Lab #4 – ***Basic Calculator*** ***Fall*** 2018

Introduction .

In this lab you will develop a basic calculator which will be able to add or subtract two binary numbers and store the result for display.

The process will be to load the numbers in binary format into registers (one register per number) and then perform the operation. After the operation is completed the results are stored in another register and a fourth register will store if the resulting number is negative or if an overflow occurred

Learning Outcomes .

After completing this lab you should know how to:

• Implement a finite state machine which will respond to input, process data and provide a display of the output to a user

•Incorporate different elements (combinational logic, memory and controller unit) into a single functional device

•Be able to implement in VHDL and on the DEC board a basic add/subtract calculator.

Table of Contents .

This lab consists of the following stages:

1. Design the controller
2. Implement the controller
3. Implement the registers and Adder
4. Simulate the system
5. Port the system to the DEC board
6. Demonstrate the operational system

1. Design of the Controller .

The system has the following requirements:

A finite state machine controller will have the system perform the following operations:

Step 1) Two 8-bit binary integers are entered into registers R0 and R1 using the switches of the DEC board. One value entered at a time. The values of the numbers being entered should be displayed by the LEDs of the DEC board.

Step 2) Then the operation should be entered using the switches. The codes are: “00” add contents of R0 and R1, “01” Subtract R1 from R0, 01 Subtract R0 from R1, for each of the previous cases the result of the operation is stored in register R2 and if an overflow occurs then a “1” should be stored as the MSB of R3. If the result is zero then a “1” should be stored in the LSB of register R3. “11” reset registers and start step #1.

The contents of the registers R3 and R4 should be visible using the LEDs on the DEC board.

VHDL State Machine Description

-- Please use the text file from the course website:
--Sample State Machine VHDL description.
-- Use the sample file as a base to create the VHDL description for your machine.

```
library ieee;
use ieee.std_logic_1164.all;
entity sample_state is
port( clk, in_1, reset : in std_logic;
    out1 : out std_logic_vector(3 downto 0));
end sample_state;
```

```
 architecture moore_mod of sample_state is
                type state_type is (s0,s1,s2,s3); -- Declare states
                signal state : state_type;
                begin
                sample_state_process : process(clk, reset) --this will
define a clocked process FSM
                begin
                if reset = '1' then
                state <= s0; -- reset the systemt to start state
                elsif clk'event and clk='1' then --define rise clock edge
                case state is --This is similar to high level language
case statements
          when s0 => if in_1='1' then --Test for condition to change
state
             state <= s1;
          end if;
                  when s1 => if in_1='0' then
                     state <= s2;
                  end if;
                  when s2 => if in_1='1' then
                     state <= s3;
           end if;
                  when s3 => if in_1='0' then
                     state <= s0;
            end if;
end case; --All of the state transitions have been covered
end if;
end process; --End of the FSM definition
output_sys : process(state) --The output is a function of the state only
begin
case state is
when s0 => out1 <= "0000"; --This is the output for the state s0.
when s1 => out1 <= "0001";
when s2 => out1 <= "1111";
when s3 => out1 <= "0110";
end case;
end process;
end moore_mod;
```

You should modify the code to create a VHDL description the controller.

You should use the devices developed in the previous labs (registers, adder and subtractor) to complete the system.

You should demo the functions described above both by simulation as well as using the Altera board.

The TA can ask you to input any numbers and expect the system to provide the appropriate output.

The lab report, and all associated design files must be submitted, as an assignment to the WebCT site. Only one submission need be made per group (both students will receive the same grade!).

Combine all of the files that you are submitting into one *zip* file, and name the zip file gNN_LAB_3.zip (where NN is your group number).

Grade Sheet for Lab #4

Winter 2018.

Group Number:_______________.

Group Member Name:________________. Student Number:___________________.

Group Member Name:________________. Student Number:___________________.

Marks

□ 1. VHDL and schematic for the Controller (FSM) ____________________.

□ 2. VHDL and schematic for control signal logic________ ____________________.

□ 3. Demo of Addition of Two Numbers__(SIM)_________________________________

□ 4. Demo of Subtraction of Two Numbers _(SIM)___ ____________________.

□ 5. Demo of Addition of Two Numbers__(Altera) ____________________.

□ 6. Demo of Addition of Two Numbers__(Altera) ____________________.

□ 7. Demo of Overflow_Indication __________________ ____________________.

□ 8. Demo of Negative Numer indication________________ ____________________.

TA Signatures

Each part should be demonstrated to one of the TAs who will then give a grade and sign the grade sheet. Grades for each part will be either 0, 1, or 2. A mark of 2 will be given if everything is done correctly. A grade of 1 will be given if there are significant problems, but an attempt was made. A grade of 0 will be given for parts that were not done at all, or for which there is no TA signature.