EEL 3701 – Digital Logic and Computer Systems Lab 2

Greg Bolling

UF ID 64911960

Problem Statement

The goal of the lab is to build two sequential circuits (a random logic one and a clock divider) and then build two combinatorial circuits (a washing machine and a seven segment decoder). The first two demonstrate that a signal is only valid at the rising edge of a clock. The next two are for building combinatorial circuits and showing them on the expansion board.

Design

Problem 1 Part 1 starts with a random logic gate. I chose a an XNOR gate to create A XNOR B as an output. The output is Logicout. The circuit is attached at the end of this in an appendix. The circuit updates Logicout when the clock rises. I added reset to the process list because of a Quartus warning (number 10492) that it wasn’t in the sensitivity list. The design had warnings but no errors in Quartus.

Logicout = A XNOR B

Truth Table Voltage Table

A B Logicout A B Logicout

F F T 0v 0v 5v

F T F 0v 5v 0v

T F F 5v 0v 0v

T T T 5v 5v 5v

Problem 1 Part 2 is a Clock Divider and then the A XNOR B circuit

This circuit divided the clock down to a 2 second intervale (f = 1/Tp or 0.5 Hz). To get from 4 Mhz down to 0.5 Hz I had to divide the frequency by 8,000,000. However, the frequency is really two periods; a high period and a low period. To get have of the period for each invert of the clock, I used 8,000,000/2 or 4,000,000 as a threshold for the counter to divide. This counted from 0,1,2,3,…,3,999,999 and back to 0. Converting 3,999,999 to hex, it was 0x3d08ff. the slow clock is high for 1 second and low for 1 second. I used this slow clock to run the same circuit in Problem 1 Part 1. The design had warning and no errors in Quartus.

Problem 2

I created the logic circuits for a washing machine. These circuits were optimized using a K-Map and resulted in

HeaterOn <= ((not TempRightLevel) and DoorIsClosed) and WaterrightLevel;

WaterValveOpen <= (not WaterRightLevel) and DoorIsClosed;

MotorOn <= (TempRightLevel and WaterRightLevel) and DoorIsClosed;

The design had warning and no errors in Quartus.

Problem 3

In this part, I implemented the combinational logic for a seven segment decoder. Using the tables, I created code using the process statement example and the table for turning on seven segments. The design had warning and no errors in Quartus.

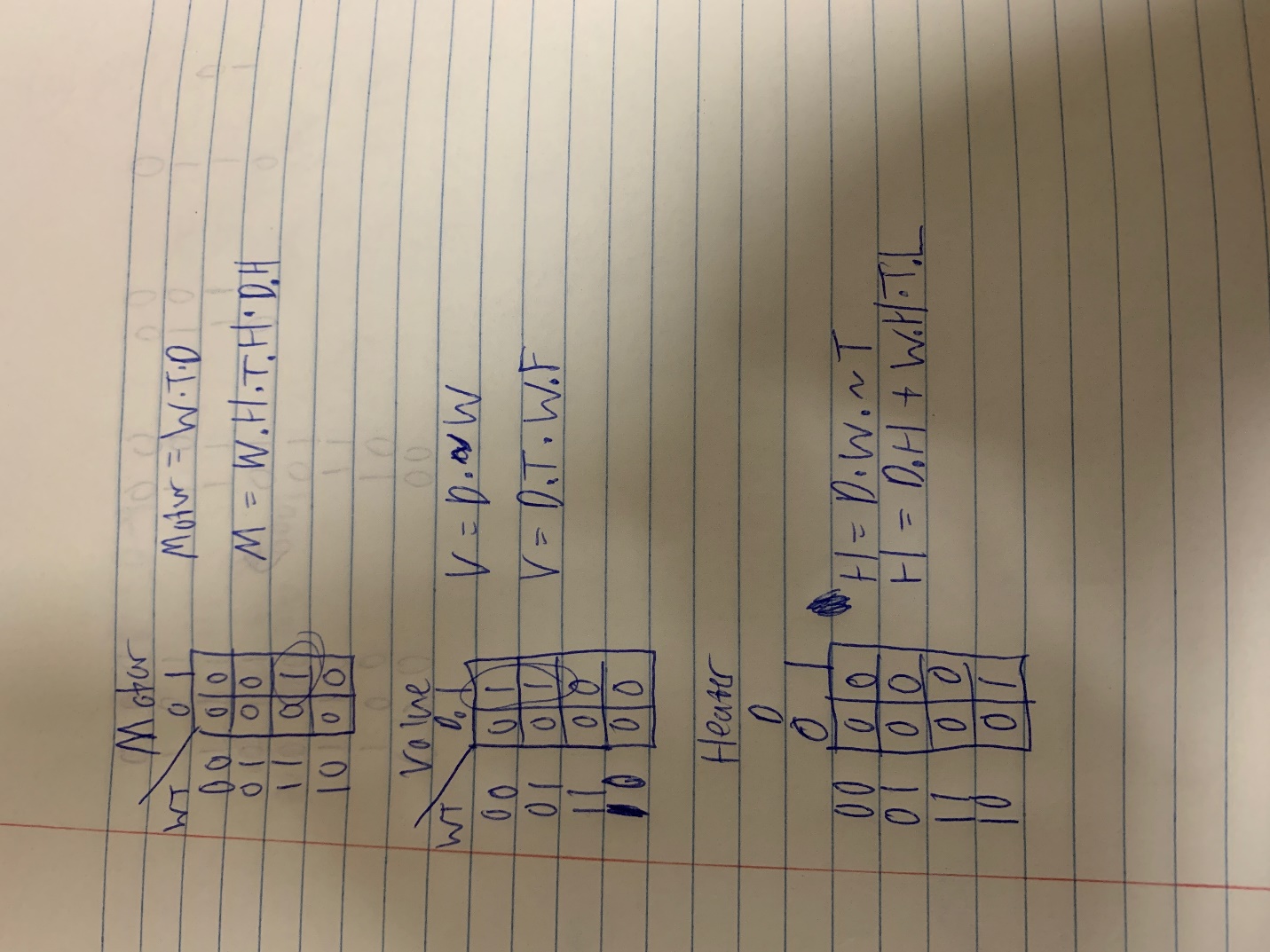
Implementation

I will assign pins in a way that I might be able to wire up all the circuits. The first two circuits used some pins. Then add more for washing machine and finally seven segment.

Conclusions

Appendix

Picture(s)



VHDL Files

---------------------------------------

-- Author: Greg Bolling

-- Date: February 3, 2020

-- Class; EEL 3701

-- Assignment: Lab 02 svnseg

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

---------------------------------------

library IEEE;

use IEEE.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use IEEE.std\_logic\_unsigned.all;

entity Lab03e is

port

(

binary : in std\_logic\_vector(3 downto 0);

a\_to\_g : out std\_logic\_vector(6 downto 0)

);

end Lab03e;

architecture arch\_Lab03e of Lab03e is

begin

P1: process(binary)

begin

case (binary) is

when "0000" => a\_to\_g <= "0000001"; -- display 0 => '1' is off and a '0' is on

when "0001" => a\_to\_g <= "1001111"; -- display 1 => '1' is off and a '0' is on

when "0010" => a\_to\_g <= "0010010"; -- display 2 => '1' is off and a '0' is on

when "0011" => a\_to\_g <= "0000110"; -- display 3 => '1' is off and a '0' is on

when "0100" => a\_to\_g <= "1001100"; -- display 4 => '1' is off and a '0' is on

when "0101" => a\_to\_g <= "0100100"; -- display 5 => '1' is off and a '0' is on

when "0110" => a\_to\_g <= "0100000"; -- display 6 => '1' is off and a '0' is on

when "0111" => a\_to\_g <= "0001111"; -- display 7 => '1' is off and a '0' is on

when "1000" => a\_to\_g <= "0000000"; -- display 8 => '1' is off and a '0' is on

when "1001" => a\_to\_g <= "0001100"; -- display 9 => '1' is off and a '0' is on

when others => a\_to\_g <= "1111111"; -- display nothing

end case;

end process;

end arch\_Lab03e;

---------------------------------------

-- Author: Greg Bolling

-- Date: February 3, 2020

-- Class; EEL 3701

-- Assignment: Lab 02 WashingMachine

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

---------------------------------------

library IEEE;

use IEEE.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use IEEE.std\_logic\_unsigned.all;

entity Lab03d is

port

(

TempRightLevel : in std\_logic;

WaterRightLevel : in std\_logic;

DoorIsClosed : in std\_logic;

HeaterOn : out std\_logic;

WaterValveOpen : out std\_logic;

MotorOn : out std\_logic

);

end Lab03d;

architecture arch\_Lab03d of Lab03d is

begin

HeaterOn <= ((not TempRightLevel) and DoorIsClosed) and WaterrightLevel;

WaterValveOpen <= (not WaterRightLevel) and DoorIsClosed;

MotorOn <= (TempRightLevel and WaterRightLevel) and DoorIsClosed;

end arch\_Lab03d;

---------------------------------------

-- Author: Greg Bolling

-- Date: February 3, 2020

-- Class; EEL 3701

-- Assignment: Lab 02 LED Blink

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

---------------------------------------

library IEEE;

use IEEE.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use IEEE.std\_logic\_unsigned.all;

entity Lab03b is

port

(

A : in std\_logic;

B : in std\_logic;

reset : in std\_logic;

clock : in std\_logic;

LED : out std\_logic

);

end Lab03b;

architecture arch\_Lab03b of Lab03b is

signal count : std\_logic\_vector(27 downto 0); -- max value is 2^27-1 or 134,217,727, min value is 0

signal led\_output : std\_logic;

signal slow\_clock : std\_logic := '0';

begin

P1 : process(clock, reset)

begin

if(reset = '1') then

slow\_clock <= '0';

count <= X"0000000";

elsif (clock'event and clock = '1') then

if (count = X"3d08ff") then -- exactly 3,999,999 so counter goes 0, 1, 2, ..., 3,999,999 and repeats every half second

count <= X"0000000";

slow\_clock <= not slow\_clock; -- inverts every half to output 2 hz

else

count <= count + 1;

end if;

end if;

end process;

--put your code here drive the LED

--Make sure that your combinational circuit

--depends on the slow clock

LED <= led\_output;

P2 : process(slow\_clock, reset)

begin

if(reset = '1') then

led\_output <= '0';

elsif (slow\_clock'event and slow\_clock = '1') then -- this slow\_clock cycle is 1/2 hz

led\_output <= A xnor B;

end if;

end process;

end arch\_Lab03b;

---------------------------------------

-- Author: Greg Bolling

-- Date: February 3, 2020

-- Class; EEL 3701

-- Assignment: Lab 02 Random Clock

-- Class: 12368

-- Section 7441

-- PI Name: Savvas Ferekides

---------------------------------------

library IEEE;

use IEEE.std\_logic\_1164.all;

use ieee.numeric\_std.all;

use IEEE.std\_logic\_unsigned.all;

entity Lab03 is

port

(

A : in std\_logic;

B : in std\_logic;

clock : in std\_logic;

reset : in std\_logic;

Logicout : out std\_logic -- meaning Make your output dependent on the clock might be combine

); -- with a and b or it might mean wait to next clock egde to output

end Lab03;

architecture arch\_Lab03 of Lab03 is

begin

P1: process(clock, reset)

begin

if (reset = '1') then

Logicout <= '0';

elsif (clock'event and clock = '1') then -- this is a clocked logic signal of A xnor B

Logicout <= A xnor B;

end if;

end process;

end arch\_Lab03;