Lab Experiment No. 08:

Traffic Light Controller in VHDL

By: Alexander Lukens (A20417036)
Instructor: Dr. Jafar Saniie
ECE 441-001
Lab Completion Date: 06/18/2020

Acknowledgement: I acknowledge all of the work (including figures and codes) belongs to me and/or persons who are referenced.

Signature:____Alexander Lukens____

I. Introduction

A. Purpose

The main purpose of this experiment is to introduce students to synchronous logic design in VHDL using the Xilinx Vivado software ecosystem. Students will design a traffic light system that operates automatically or takes user input. Students must derive the controller logic using a state diagram for the system. Furthermore, students will synchronize the circuit with the internal clock signal of Xilinx FPGA development boards.

B. Background

Xilinx is a technology company based in California, USA that creates field programmable gate array (FPGA), system-on-chips (SoCs), and other semiconductor products. They compliment their hardware designs with a robust software environment for designing digital logic circuits using System Verilog and VHSIC-Hardware Description Language (VHDL). This allows silicon engineers to rapidly develop and test digital logic designs without physically designing them on a breadboard or waiting for dedicated, application specific silicon to be produced.

Synchronous circuits refer to circuits that operate around a global "clock" signal. This clock signal is utilized to "sync" logic operations throughout a circuit such that all logic devices transition logic states simultaneously. This is important because logic operations rely on the output of other components, therefore it is necessary to have valid values available simultaneously for the circuit to respond correctly.

II. Lab Procedure and Equipment List

A. Equipment

Equipment required for this lab includes the following:

- Xilinx Artix-7 based FPGA
- PC with Xilinx Vivado software

B. Procedure

Students will follow the instructions outlined in the ECE 742 - FPGA Lab Manual for experiment 8. This includes designing and implementing a traffic signal system for a single intersection. Two variations of the circuit will be produced: one that transitions states based on a timer, and another that transitions states based on the input of two "vehicle sensors". Both circuits will use a 1-Hz clock signal derived from the 100MHz clock signal of the Xilinx FPGA using a clock-divider component.

III. Results and Analysis

A. Discussion

When designing the light signal system without buttons, I used a clock divider circuit obtained from an online library

Clock Divider 2.1

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

entity Clock_Divider is
port ( clk,reset: in std_logic;
clock_out: out std_logic);
end Clock_Divider;

architecture bhv of Clock_Divider is

signal count: integer:=1;
signal tmp : std_logic := '0';

Begin
        process(clk,reset)
        Begin
        if(reset='1') then
        count<=1;
        tmp<='0';
        elsif(rising_edge(clk)) then
        count <=count+1;
        if (count = 50000000) then
        tmp <= NOT tmp;
        count <= 1;
        end if;
        end if;
        clock_out <= tmp;
end process;

end bhv;
```

The state controller utilizes two variables, “state” and “next_state”, along with the count shown by a “timer” variable to determine what the traffic signals will display. This timer is incremented once every clock cycle until a limit is reached in the state controller.

In order to incorporate this design into the Xilinx FPGA, a third script was created, and a constraints file was utilized to assign the correct pin numbers for the RGB leds and clock signal on the FPGA board. This script also utilizes the clock divider in order to produce a 1-Hz signal for use with the state controller. The clock divider and state controller are added to this script by using VHDL “components”.

For the light signal system that uses buttons, the design of the state controller was modified slightly such that the state would only progress if one of the “vehicle sensors” was depressed. This was imitated on the FPGA by the use of push-buttons. An important portion of the state controller modifications is shown below.

```
when s0 =>

  if (timer = 10) then
  next_state <= s1; --if timer at 10 seconds, advance to next state
  timer_reset <= '1';

  elsif(SensorE ='1' OR SensorW = '1') then
  next_state <=s0; --if sensor asserted stay in s0, but do not reset timer
  timer_reset <='0';
  else
  next_state <= s0; --if no sensor asserted stay in s0 and reset timer
  timer_reset <='1';
  end if;
```

In the FPGA control script, the state controller and clock divider are both instantiated, and the RGB LEDs for the traffic signals are controlled by the outputs of the state controller. An excerpt from the script is shown below.

```
divider : clk_divider port map (clk => CLK100MHZ, s0 => clk_s0, s1 => clk_s1,clk_out => final_clk);
--clk divider set to produce 1 Hz clock signal
Controller : TrafficButton port map (clk => final_clk, SensorW => btn(3), SensorE => btn(0), ns_r => ns_r,
        ns_y => ns_y, ns_g => ns_g, ew_r => ew_r, ew_y => ew_y, ew_g => ew_g);

      led0_r <= ns_r OR ns_y;
      led0_g <= ns_g OR ns_y; --LEDs controlled based on output from state controller
      led3_r <= ew_r OR ew_y;
      led3_g <= ew_g OR ew_y;
```

IV. Conclusion

This lab should be considered a complete success. Students were successfully able to implement a traffic signal control script in VHDL. Students learned about the implementation of state machines in VHDL, and how clock signals can be used to synchronize a variety of circuit components. Furthermore, students learned to utilize VHDL components to create several instances of a VHDL entity. These skills will be utilized in subsequent laboratory sessions to design and implement more complex VHDL circuits.

References

[1] Wikipedia contributors. (2020, June 14). Xilinx. In *Wikipedia, The Free Encyclopedia*. Retrieved 02:24, June 16, 2020, from https://en.wikipedia.org/w/index.php?title=Xilinx&oldid=962581047

[2] FPGA Lab Manuals - ECE 742

[3] Wikipedia contributors. (2020, May 16). Synchronous circuit. In Wikipedia, The Free Encyclopedia. Retrieved 04:23, June 19, 2020, from https://en.wikipedia.org/w/index.php?title=Synchronous_circuit&oldid=957028086