****

**EE4620L/EE6620L/CEG4324L/CEG6324L**

**DIGITAL INTEGRATED CIRCUIT DESIGN LAB**

**Lab 0**

By

Yeoh\_\_\_\_\_\_\_\_\_ Alex\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

UID: U00931845

Email: yeoh.3@wright.edu

Submission date: 10/05/2024

“I have neither given nor received aid on this assignment, nor have I observed any violation of the Honor code”

Signature : Alex Yeoh

Date : 09/05/2024

**Table of Contents (include page #)**

I. Introduction (page 5)

II. Process (page 5)

III. Results (page 7)

IV. Conclusion (page 9)

V. Code (page 9)

**List of Figures**

Figure 1: K-maps for outputs F and G respectively. (Page 6)

Figure 2: Original simulation with WPD = 2ps. (Page 7)  
Figure 3: Original simulation with WPD = 4ps. (Page 7)

Figure 4: Part C simulation with WPD = 2ps. (Page 7)

Figure 5: Part C simulation with WPD = 4ps. (Page 7)

Figure 6: ZedBoard with default truth table programmed at input pattern 111. (Page 8)

Figure 7: ZedBoard with part C truth table programmed at input pattern 001. (Page 8)

**List of Tables**

Table 1: given truth table for part B. (Page 5)

Table 2: given truth table for part C. (Page 6)

I. Introduction

The objective of the lab is to become familiar with Vivado.

II. Process

We initially made a new Vivado project and imported the given source and constraint files. We then set the topmodule as the top source file and tb\_comb\_function as the top simulation file. We then ran the simulation. We then modified WPD in tb\_comb\_function.vhd to 4 to increase the worst propagation delay in the simulation and ran the simulation again. We then programmed the ZedBoard by generating bitstream, clicking on “program device”, detecting the device in the new window, clicking “program”, selecting the generated “.bit” file in the pop-up, and clicking the “program” button in the pop-up. We then flipped the switches and looked at the LEDs to confirm that the board was programmed correctly by following the given truth table as seen below in table 1, where the right most switch is the A input, the switch to its immediate left is the B input, the switch to input B’s immediate left is the C input, the LED above the switch to input C’s immediate left is S, and the LED to output S’s immediate left is C0.

A paper with numbers and lines

Description automatically generated

Table 1: given truth table for part B.

We then implemented the new given truth table seen in table 2 below, which was simplified in the k-maps seen in figure 1 below.

A number of binary code

Description automatically generated with medium confidence

Table 2: given truth table for part C.

A black background with white lines and numbers

Description automatically generated



Figure 1: K-maps for outputs F and G respectively.

After simplifying the truth table with K-maps, we made a new VHDL file that was identical to comb\_function named new\_comb\_function, but with the commented truth table and k-maps modified for this new truth table, and the relevant equations for F and G. We then modified both topmodule and tb\_comb\_function to use this new\_comb\_function. The process for simulating and and verifying the code physically is the same as the initial steps in this section.

III. Results

The results show that the propagation delay was worse than 2ns and satisfy the requirements of the lab.

1. Original truth table, 2ps WPD. There are simulation errors because the propagation delay for the circuit is 3ps, the errors manifest in the locations highlighted in figure 2 below.

A screenshot of a computer

Description automatically generated



Figure 2: Original simulation with WPD = 2ps.

1. Original truth table, 4ps WPD. There are no simulation errors, the changed output is stable for 1ps before inputs change.

A screenshot of a computer

Description automatically generated

Figure 3: Original simulation with WPD = 4ps.

1. The truth table and K-map simplifications can be seen above in table 2 and figure 1 respectively.
   1. Part C truth table, 2ps WPD. There are simulation errors because the propagation delay for the circuit is 3ps, the errors manifest in the locations highlighted in figure 4 below.

A screenshot of a computer

Description automatically generated



Figure 4: Part C simulation with WPD = 2ps.

* 1. Part C truth table, 4ps WPD. There are no simulation errors, the changed output is stable for 1ps before inputs change.

A screenshot of a computer

Description automatically generated

Figure 5: Part C simulation with WPD = 4ps.

The following two figures are images of the ZedBoard with the default and part c truth tables programmed in respectively.

A green electronic board with many different colored lights

Description automatically generated

Figure 6: ZedBoard with default truth table programmed at input pattern 111.

A green circuit board with many different components

Description automatically generated

Figure 7: ZedBoard with part C truth table programmed at input pattern 001.

IV. Conclusion

From this lab, I have learned how to use Vivado.

V. Code

|  |
| --- |
| entity TB\_COMB\_FUNC is  generic (  WPD: time:= 4 ns -- set to the worst case propagation delay  );  end ; |

Portion of tb\_comb\_function.vhd modified to change WPD.

|  |
| --- |
| -- new\_comb\_function.vhd  --------------------------------------------------------------------------  --------------------------------------------------------------------------  --------------------------------------------------------------------------  library IEEE,WORK;  use IEEE.STD\_LOGIC\_1164.ALL;  entity NEW\_COMB\_FUNC is  port( A,B,C: in std\_logic;  F,G: out std\_logic);  end;  architecture DATAFLOW\_ARCH of NEW\_COMB\_FUNC is  begin  -- example implementation of a truth-table  --  -- A B C | F G  ----------------------  -- 0 0 0 | 1 0  -- 0 0 1 | 1 1  -- 0 1 0 | 0 0  -- 0 1 1 | 0 1  -- 1 0 0 | 1 1  -- 1 0 1 | 0 1  -- 1 1 0 | 0 0  -- 1 1 1 | 1 0  --  --  -- \BC 00 01 11 10  -- A \-----------------------  -- 0 | 1 | 1 | 0 | 0 |  -- -----------------------  -- 1 | 1 | 0 | 1 | 0 |  -- -----------------------  -- F = B'C'+A'B'+ABC  --  -- \BC 00 01 11 10  -- A \-----------------------  -- 0 | 0 | 1 | 1 | 0 |  -- -----------------------  -- 1 | 1 | 1 | 0 | 0 |  -- -----------------------  -- G = AB'+A'C  --  --  F <= (not(B) and not(C)) or(not(A) and not(B)) or (A and B and C)  -- pragma synthesis\_off  after 3 ns  -- pragma synthesis\_on  ;  G <= (A and not(B)) or (not(A) and not(C))  -- pragma synthesis\_off  after 3 ns  -- pragma synthesis\_on  ;  end; |

new\_comb\_function.vhd

|  |
| --- |
| component new\_COMB\_FUNC  port(A,B,C: in std\_logic;  F,G: out std\_logic);  end component ; |

First location of tb\_comb\_function.vhd modified to use new\_comb\_function.vhd

|  |
| --- |
| CUT:new\_COMB\_FUNC -- Circuit Under Test  port map (A=>A,B=>B,C=>C,F=>F,G=>G); |

Second location of tb\_comb\_function.vhd modified to use new\_comb\_function.vhd

|  |
| --- |
| component new\_COMB\_FUNC  port( A,B,C: in std\_logic;  F,G: out std\_logic);  end component; |

First location of topmodule.vhd modified to use new\_comb\_function.vhd

|  |
| --- |
| comp\_tb: new\_COMB\_FUNC port map (A=>a, B=>b, C=>c, F=>f, G=>g); |

Second location of topmodule.vhd modified to use new\_comb\_function.vhd