LAB 2 BEHAVIORAL REPORT

Academic Integrity (more info @ https://aisc.uci.edu/): You are encouraged to discuss the labs at a high level, but the code/equations/simulations you come up with should be your own. By typing "yes" at the end of this question and filling in your name, you certify that the work you are turning in is your own work. Is the work you are turning in your own? _Yes_

If you worked on any portion of your report or vhdl code with other students (discussion at high level & debugging; if more, please describe), please list their names here: ____

Student Name: Daniel Guerra-Rojas
Student ID: 14372295
Date Completed: 4/26/2020
Time Spent: Reviewing Digital Design Material: 30 minutes
Design/Preparation Work: about 2 and a half hours
VHDL Coding & Debugging: about 4 and a half hours

BEHAVIORAL OVERVIEW

I feel like I did close to 100% of the lab. I say close to because I know my graph isn't completely accurate. By this I mean that the correct outputs will display, but just not at the time the "Assert and Report" statements ask for them. I started off with opening up the lab report and started working on the truth table first. From there I went on to making the Next_State, Permit, and ReturnChange equations. I then went on to find the minimum clock cycle. From there I moved on to the coding aspect of this lab and started on the Structural. For the Structural, I had to look at the WristWatch example to see how we were supposed to do the Structural and then back and forth between the two. Once I eventually finished the structural, I opened up Vivado to test the code with the only Test Case in the TestBench. From here, I just went through a lot of debugging trying to figure out why my Structural code wasn't working. Once I had all the bugs worked out, I copied over my Test Cases from the previous assignment and used that in my TestBench. The last thing I did was change the "Assert" timings so that it would work properly for my design and so the errors for the "Assert" timing would go away. The most difficult part about this lab was figuring out why my structural code wasn't working. This is where most of my time was spent while working on this lab. I just couldn't for the longest time figure out which of my equations was giving me incorrect outputs and messing with the result of the graph.

LAB 2 TRUTH TABLE(S)

Create a truth table showing Permit, ReturnChange, and NextState based on your FSM. You can use Word, Excel, or even attach a picture of your truth table as long as it is legible. It must be in the correct orientation & legible for full credit.

State		Input	NextState	Output	
	CurrentState			Permit	ReturnChange
Initial	0000	001	0001	0	0
		010	0011	0	0
		100	0111	0	0
Five	0001	XXX	0010	0	0
Five_wait	0010	000	0010	0	0
		001	0011	0	0
		010	0101	0	0
		100	1000	0	0
		111	1001	0	0
Ten	0011	XXX	0100	0	0
Ten_wait	0100	000	0100	0	0
		001	0101	0	0
		010	0111	0	0
		100	1000	0	0
		111	1001	0	0
Fifteen	0101	XXX	0110	0	0
Fifteen_wait	0110	000	0110	0	0
		001	0111	0	0
		010	1000	0	0
		100	1000	0	0
		111	1001	0	0
Twenty	0111	XXX	0000	1	0
Twnety_plus	1000	XXX	0000	1	1
Cancel	1001	XXX	0000	0	1

LAB 2 FINAL EQUATIONS

Permit = $CS_3'CS_2CS_1CS_0 + CS_3CS_2'CS_1'CS_0'$

ReturnChange = $CS_3CS_2'CS_1'CS_0' + CS_3CS_2'CS_1'CS_0$

$= CS_3CS_2'CS_1'$

NextState3 = $CS3'CS2'CS1CS0'(I2I1'I0') + CS3'CS2'CS1CS0'(I2I1I0) + CS3'CS2CS1'CS0'(I2I1'I0') + CS3'CS2CS1'CS0'(I2I1I0) + CS3'CS2CS1CS0'(I2'I1I0') + CS3'CS2CS1CS0'(I2I1'I0') + CS3'CS2CS1CS0'(I2I1I0)$

$= [CS_3'CS_0'((CS_2 \oplus CS_1) + CS_2CS_1)](I_2I_1I_0) + CS_3'CS_2CS_1CS_0'(I_2'I_1I_0') + CS_3'CS_1CS_0'(I_2I_1'I_0')$

NextState2 = CS3'CS2'CS1'CS0'(I2I1'I0') + CS3'CS2'CS1CS0'(I2'I1I0') + CS3'CS2'CS1CS0 + CS3'CS2CS1'CS0'(I2'I1'I0') + CS3'CS2CS1'CS0'(I2'I1'I0) + CS3'CS2CS1'CS0'(I2'I1I0') + CS3'CS2CS1'CS0 + CS3'CS2CS1CS0'(I2'I1'I0') + CS3'CS2CS1CS0'(I2'I1'I0)

= $CS_3'CS_2'CS_1'CS_0'(I_2I_1'I_0') + [CS_3'CS_0'(CS_2 \oplus CS_1)](I_2'I_1I_0') + CS_3'CS_2CS_0'(I_2'I_1') + CS_3'CS_0(CS_2 \oplus CS_1)$

NextState1 = CS3'CS2'CS1'CS0'(I2'I1I0') + CS3'CS2'CS1'CS0'(I2I1'I0') + CS3'CS2'CS1'CS0 + CS3'CS2'CS1CS0'(I2'I1'I0') + CS3'CS2'CS1CS0'(I2'I1'I0) + CS3'CS2CS1'CS0'(I2'I1I0') + CS3'CS2CS1'CS0 + CS3'CS2CS1CS0'(I2'I1'I0') + CS3'CS2CS1CS0'(I2'I1'I0)

= $CS_3'CS_1'CS_0'(I_2'I_1I_0') + CS_3'CS_2'CS_1'CS_0'(I_2I_1'I_0') + CS_3'CS_1CS_0'(I_2'I_1') + CS_3'CS_2'CS_0$

NextState0 = CS3'CS2'CS1'CS0'(I2'I1'I0) + CS3'CS2'CS1'CS0'(I2'I1I0') + CS3'CS2'CS1'CS0'(I2I1'I0') + CS3'CS2'CS1CS0'(I2'I1'I0) + CS3'CS2'CS1CS0'(I2'I1I0') + CS3'CS2'CS1CS0'(I2I1I0) + CS3'CS2CS1'CS0'(I2'I1'I0) + CS3'CS2CS1'CS0'(I2'I1I0') + CS3'CS2CS1'CS0'(I2I1I0) + CS3'CS2CS1CS0'(I2'I1'I0) + CS3'CS2CS1CS0'(I2I1I0)

= $CS_3'CS_0'(I_2'I_1'I_0) + [CS_3'CS_0'(CS_2'CS_1' + (CS_2 \oplus CS_1))](I_2'I_1I_0') + CS_3'CS_2'CS_1'CS_0'(I_2I_1'I_0') + [CS_3'CS_0'(CS_2 + CS_1)](I_2I_1I_0)$

Explain any optimizations you may have made to your equations here:

> I was able to minimize the Boolean Equations above by using the Laws of Boolean Algebra, such as Complement and Distributive Laws, and noticing patterns where (AB' + A'B) is the equivalent of $A \oplus B$.

LAB 2 MINIMUM CLOCK CYCLE

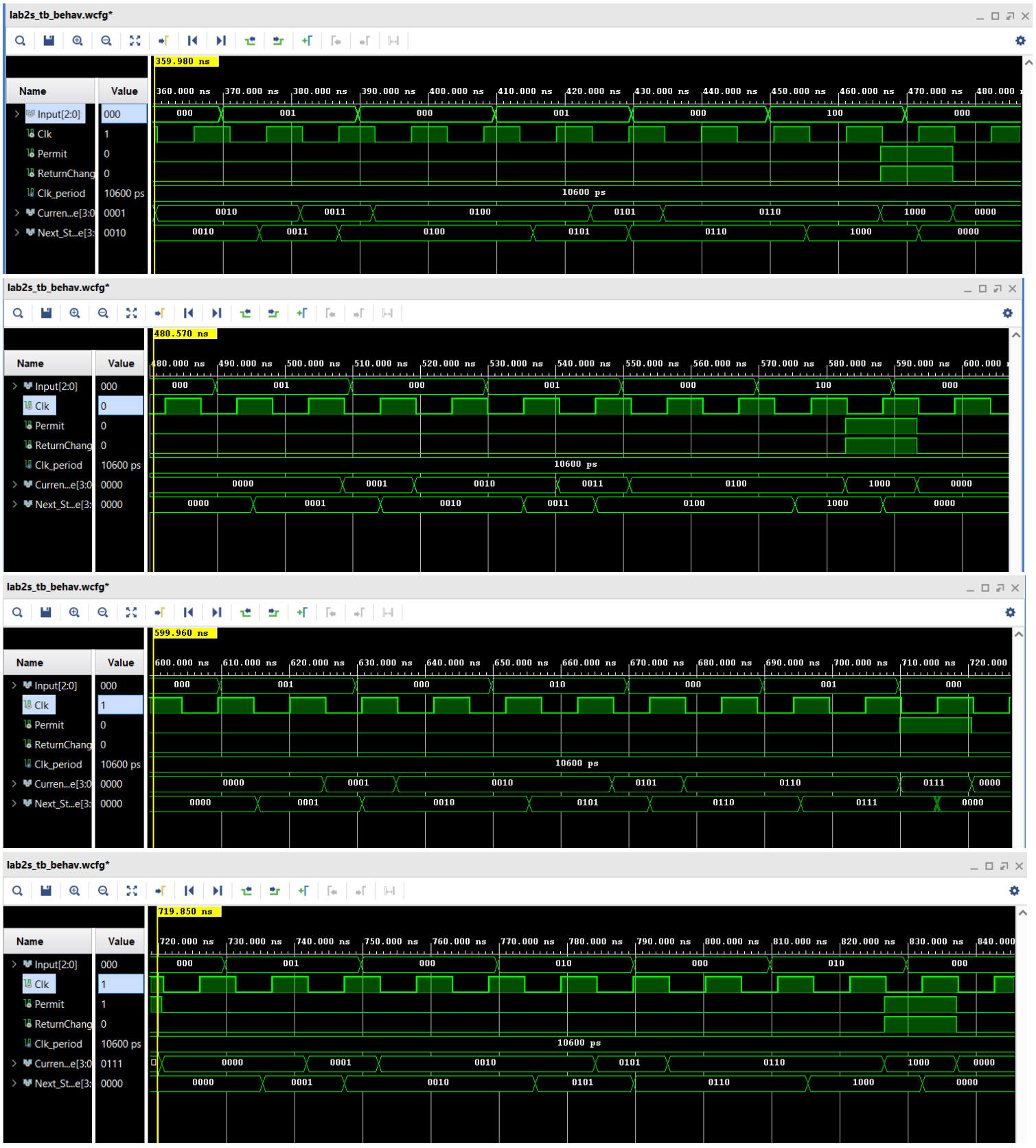
Minimum clock cycle: the minimum clock cycle is about 10.6 ns

Explain how you derived your minimum clock cycle (as discussed in EECS 31) here:

> In the Wristwatch example structure video, the instructor said that we just add up the Combinational Logic timing, the State Register timing, and the setup time together and that gives us our total delay. Because these values were given to us in the assignment, I just added up 5.6ns (the Combinational Logic timing), 4ns (State Register timing), and 1ns (the setup time) to get a minimum clock cycle of 10.6ns.

LAB 2 STRUCTURAL SIMULATION GRAPH

Show a screenshot of your final graph here. You should crop it to the appropriate size so that it is legible.


lab2s_tb_behav.wcfg*
359.980 ns
Name
Value
Input[2:0]
000
Clk
1
Permit
0
ReturnChang
0
Clk_period
10600 ps
Curren...e[3:0
0001
Next_St...e[3:
0010
360.000 ns
370.000 ns
380.000 ns
390.000 ns
400.000 ns
410.000 ns
420.000 ns
430.000 ns
440.000 ns
450.000 ns
460.000 ns
470.000 ns
480.000 ns
000
001
000
001
000
100
000
10600 ps
0010
0011
0100
0101
0110
1000
0000
lab2s_tb_behav.wcfg*
480.570 ns
Name
Value
Input[2:0]
000
Clk
0
Permit
0
ReturnChang
0
Clk_period
10600 ps
Curren...e[3:0
0000
Next_St...e[3:
0000
480.000 ns
490.000 ns
500.000 ns
510.000 ns
520.000 ns
530.000 ns
540.000 ns
550.000 ns
560.000 ns
570.000 ns
580.000 ns
590.000 ns
600.000 ns
000
001
000
001
000
100
000
10600 ps
0000
0001
0010
0011
0100
1000
0000
lab2s_tb_behav.wcfg*
599.960 ns
Name
Value
Input[2:0]
000
Clk
1
Permit
0
ReturnChang
0
Clk_period
10600 ps
Curren...e[3:0
0000
Next_St...e[3:
0000
600.000 ns
610.000 ns
620.000 ns
630.000 ns
640.000 ns
650.000 ns
660.000 ns
670.000 ns
680.000 ns
690.000 ns
700.000 ns
710.000 ns
720.000
000
001
000
010
000
001
000
10600 ps
0000
0001
0010
0101
0110
0111
0000
lab2s_tb_behav.wcfg*
719.850 ns
Name
Value
Input[2:0]
000
Clk
1
Permit
1
ReturnChang
0
Clk_period
10600 ps
Curren...e[3:0
0111
Next_St...e[3:
0000
720.000 ns
730.000 ns
740.000 ns
750.000 ns
760.000 ns
770.000 ns
780.000 ns
790.000 ns
800.000 ns
810.000 ns
820.000 ns
830.000 ns
840.000
000
001
000
010
000
010
000
10600 ps
0000
0001
0010
0101
0110
1000
0000

LAB 2 STRUCTURAL AND BEHAVIORAL SIMULATION GRAPH COMPARISONS

Compare your behavioral & structural graphs here. If there are any differences (delays, outputs, etc.), be sure to explain them here.

The graphs are fairly similar in general. Specifically, the graphs for the structural have different layout, as in I made the inputs display in binary so that it would be easier to understand what is being inputted into the code. I also added Current_State and Next_State to the structural graphs to make it easier to debug the code when the graph doesn't match with what I expected it to look like. Other than these small differences, I would like to say that everything else is the same.