**CMPEN 371: Advanced Digital Design**

**Fall 2015**

**Lab 4: Sequential Components**

**Due: 30 September 2015**

**Morayo Ogunsina, mao5270**

**ACKNOWLEDGEMENT**

This work is entirely our own and I did not provide any assistance except as noted. The approximate contribution of each team member is as follows:

(100%) Morayo Ogunsina \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Grading Rubric**

|  |  |
| --- | --- |
| **Criteria** | **Grade** |
| Design (Sound algorithm, good block diagrams / state diagrams / state tables, efficient design, etc.)  Excellent: 12; Good: 8; Satisfactory: 4; Unsatisfactory: 0; Failure: -4 or worse | / 12 |
| Translate to VHDL (VHDL matches design docs, good comments / whitespace, follows guidelines, efficient coding, etc.)  Excellent: 12; Good: 8; Satisfactory: 4; Unsatisfactory: 0; Failure: -4 or worse | / 12 |
| Test in Simulator (test bench or script for each component and FSM, good coverage, components / FSMs verified, etc.)  Excellent: 6; Good: 4; Satisfactory: 2; Unsatisfactory: 0; Failure: -2 or worse | / 6 |
| Test in Hardware (demo in lab)   * Registers work as specified * 8-digit 7-segment displays cleanly (no flicker or ghosting) shows proper values * LEDs show the value of register 0 * Other | / 70 |
| Bonus (optional challenge, etc.) |  |
| Penalty (late demo, late submission on ANGEL, team member absent from demo, etc.)  Demo late in lab: -4; Demo up to 5 days late: -10; Demo more than 5 days late: -20  Submitted late or incompletely on ANGEL: -10; Not submitted on ANGEL: no grade for lab |  |
| TOTAL | / 100 |

**DESIGN**

The following design documents are attached:

Filename Description

Lab4\_BlockDiagram Block diagram includes Top Level BD and Components BD.

Lab4ComponentsBlkDiag1 Component Block Diagram.

Lab4ComponentsBlkDiag 2 Components Block Diagram.

**TRANSLATE TO VHDL**

The following HDL models are attached:

Filename Description

Lab04\_mao5270 Top Level: Implements a sequential circuits.

OneShot Component: Produces a one cycle pulse when input goes from low to high.

Debouncer Componentl: Removes rapid transitions, so that mechanical transition results

in one signal transitions.

Register\_nbit Component: An edge sensitive bit storage device that is enabled

synchronously.

Counter\_nbit Component: A counter that counts up per button press.

PulseGenerator Component: Produces a one -cycle pulse every (maxCount +1 )

cycles.

Dff\_CE Component: A D Flip-Flop device that is enabled synchronously.

Mux4to1 Component: Modified to a Generic Mux8to1

Nexys4DDR\_Master.ucf UCF file

The following HDL models and other files are not attached but are submitted electronically:

Filename Description

HexToSevenSeg Component: Display equivalent hex digit

Decoder3To8 Component: Controls which 7-Seg display is used using the 3 LSB of A

HexToSevenSeg\_tb Test bench: Display hex digit.

**TEST IN SIMULATOR**

The following files were used for testing to verify the design:

Filename Description

HexToSevenSeg\_tb Test bench: Display hex digit.

The only form of tests carried out was in the iSIM .No test benches or tcl files were used to test the components. Each component created was tested before using them to create bigger components.

**TEST IN HARDWARE**

The project implements all the functionality specified except to correctly display the layout of the digits

**PERFORMANCE**

The following cost and performance metrics were obtained:

Area (resources used)

Number of Slice Registers: 205 out of 126.

Number of Slice LUTs: 249 out of 63

Delay

Minimum Period: 4.609ns.

Maximum Frequency 216.967MHz.

The critical path goes from P16M, through Counter\_nbit, Debouncer, and OneShot, to ONESHT\_down.

**QUESTIONS**

1. (Design) Briefly describe how you made register 2 increment 50,000,000 times per second with a 100 MHz clock.

205 out of 126

ANS – A 100MHZ clock results in a timer clock period of 10ns. To slow down the 100MHz clock to 50MHz with timer clock period of 5ns, the 100MHz clock must increment 2 times slower than it was before.

So, to achieve this, the formula 2n = C where n is the number of bits and C is the number of times the clock counts.

Using this formula and plugging in values, 2n = 2, n = 1, but bit mapping occurs in multiples of 2, so n = 2 is used instead.