SCHULICH
School of Engineering
UNIVERSITY OF CALGARY

Lab 2 – Multiplexer and BCD

Introduction

Lab 2 requires you to integrate an instructor provided binary to Binary Coded Decimal (BCD) converter module to your Lab 1b design, which will convert the hexadecimal values on the 7-segment displays, to decimal format. Then, you will design and integrate your own two-input multiplexer, to allow the user to select whether hexadecimal or decimal values are shown on the 7-segment displays. For further practice or design challenge, you can optionally enhance the design to show an error message, when the binary to BCD module overflows.

This lab project assumes that you have been keeping up with lecture content and developing an emerging mastery of Vivado, the Basys3 board, and SystemVerilog. The instructions in the Procedure are presented at a higher level, to reflect your growth as designers in this technology. We will continue to practice our methodical and systematic process of incremental and iterative development.

Note that you will submit your Design Record to the D2L Dropbox, prior to presenting to the TA.

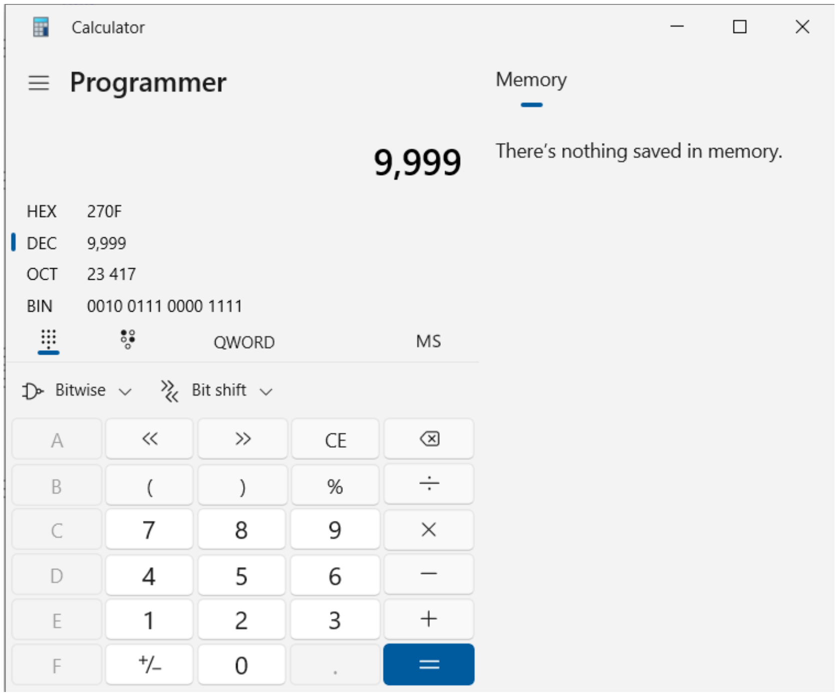
Procedure

1. Create a new Lab 2 project and bring in copies of your .SV and .XDC files from Lab 1b, and recreate it before attempting to add the *bin_to_bcd.sv* module. Download the configuration to the Basys3, to verify in hardware. **DO: copy and paste your RTL Schematic from your recreation of Lab 1b, into the Design Record.**
2. Study the *bin_to_bcd.sv* module (especially the port list), and determine where it should go in your design, to show decimal values on the 7-segment displays, based on the binary values from the switches_inputs. Integrate the *bin_to_bcd.sv* module and test it in simulation. You should be able to reuse your testbench from Lab 1b. Note that it takes the *bin_to_bcd.sv* module around 17 clock cycles to convert a new binary input value to its BCD value.

   DO: copy and paste your RTL Schematic, that shows the inclusion of the *bin_to_bcd.sv* module within your overall design, into the Design Record.

   DO: copy and paste your simulation waveforms that show the decimal outputs on the 7-segment displays, into the Design Record.

3. Download the new configuration with the *bin_to_bcd.sv* module addition, to the Basys3. Experiment with various input values and confirm where the BCD value overflows and is in error. If you are running Windows, you may find it useful to experiment with the Calculator app (see image below), to simultaneously see the hexadecimal, binary, and decimal values.



9,999

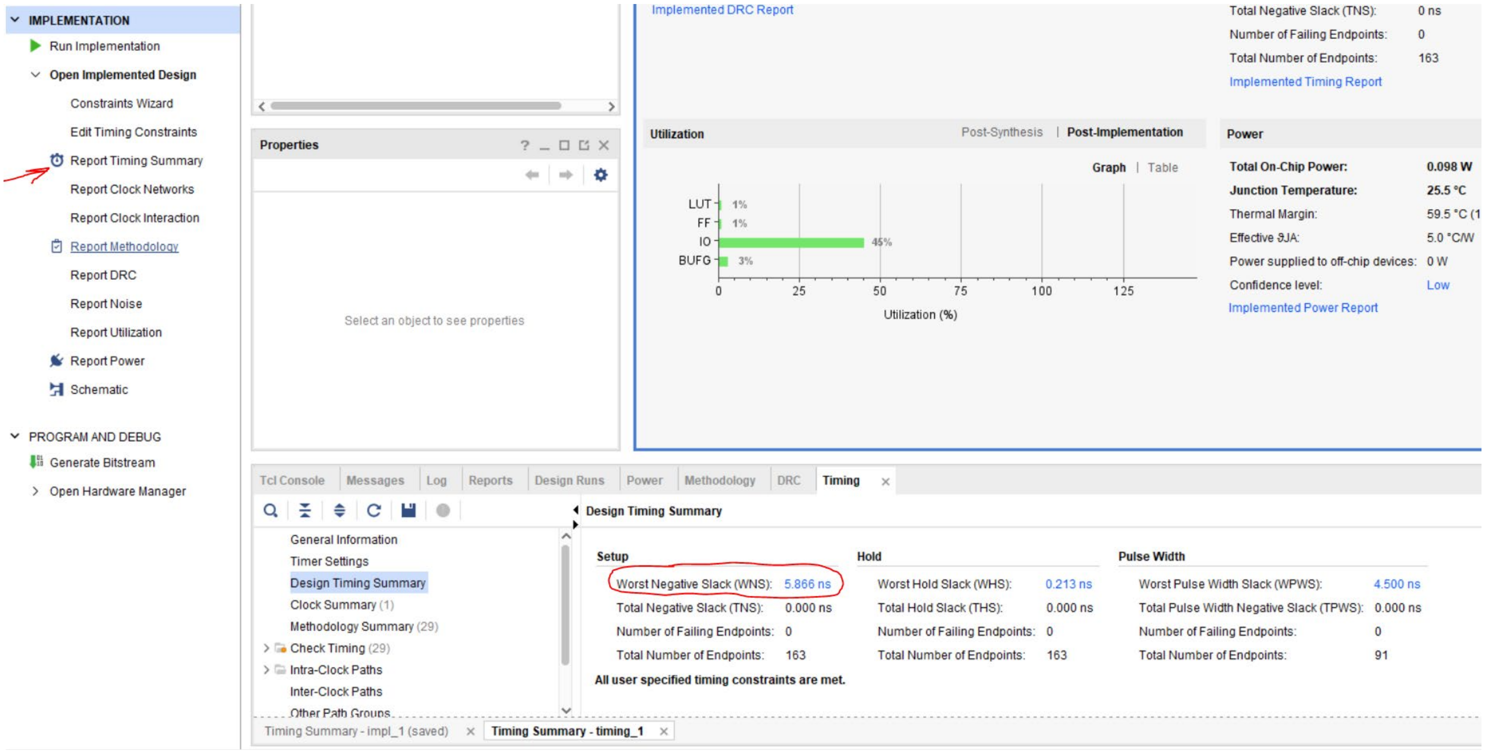
HEX 270F
DEC 9,999
OCT 23 417
BIN 0010 0111 0000 1111

There's nothing saved in memory.

DO: Write in the Design Record at what values the BCD module overflows and explain the relationship between BCD overflow values and the correct BCD values.

4. Now, we want to be able to switch between decimal and hexadecimal values on the 7-segment displays. You will first design and test a multiplexer that will switch between those two values, under the control of the top pushbutton.
5. Write the RTL SystemVerilog code for your multiplexer and its testbench (recall, this testbench is for "unit testing"). Ensure that the testbench tests the complete functionality of the multiplexer, with changing inputs and a changing select signal. Use the hexadecimal A5 pattern (alternating binary ones and zeros) for the inputs. Look up the A5 pattern for testing memory and apply that concept to the multiplexer test. If you have the interest, consider making your multiplexer testbench to be "self-checking," as shown in the lectures covering testbenches. Don't forget to set your "top" in Vivado, to the multiplexer (i.e. the three little dots). Then set "top" back to your top level, later.
    - **DO: Copy and paste your multiplexer RTL code and testbench code into your Design Record.**
    - **DO: copy and paste your multiplexer testbench simulation waveforms, into the Design Record. Ensure that the waveforms clearly demonstrate the multiplexer functionality. Note, this is just for the multiplexer testbench, nothing to do with your top level.**
6. Incorporate your multiplexer into your design. Be sure to add the pushbutton control signal for the multiplexer select input to your top level, to your top level testbench, and to your constraints file. Use the top pushbutton for the multiplexer control (set this in your .XDC file). **DO: copy and paste your top level simulation waveforms, into the Design Record. Ensure that the waveforms clearly demonstrate the multiplexer functionality, within the top level.**
7. **Do: copy and paste the RTL schematic into the Design Record.**
8. Download the configuration to the Basys3 and test the functionality in hardware.

9. Calculate the maximum clock frequency your design can run, using the Worst Negative Slack (WNS). This is one of the most important features of your project, whether it can meet timing, which in our case is a 100 MHz clock frequency. In Flow Navigator (left-hand panel in Vivado), click IMPLEMENTATION > Open Implemented Design > Report Timing Summary (see red arrow in image below). The WNS (red circle in the image) for your design will be reported in the Timing window. **DO: take a snip of the Timing window and paste it into your Design Record.**



Total Negative Slack (TNS): 0 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 163

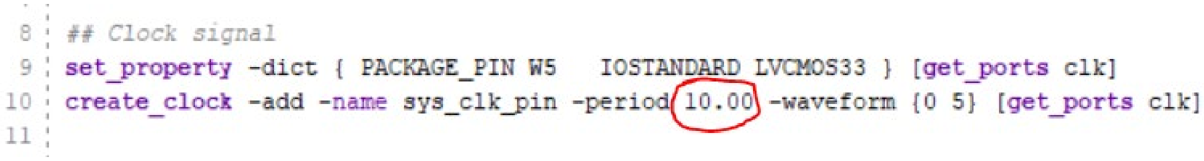
Total On-Chip Power: 0.098 W
Junction Temperature: 25.5 °C
Thermal Margin: 59.5 °C (1
Effective ϑJA: 5.0 °C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.866 ns	Worst Hold Slack (WHS):	0.213 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	163	Total Number of Endpoints:	163	Total Number of Endpoints:	91

All user specified timing constraints are met.

Calculate the maximum frequency that your project can run on the Basys3, by using the user desired clock period from the .XDC file, shown in nanoseconds in the red circle below. It should be 10.00 (for 10 ns), and let's call this *UserClockPeriod*.



```
## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
```

For our example, the calculation is:

$$MaxClockFrequency\ (MHz) = \frac{1}{UserClockPeriod - WNS} = \frac{1}{10\ ns - \ 5.866\ ns} = 241.9\ MHz$$

The result of 241.9 MHz means that we meet our minimum required clock frequency of 100 MHz, so our design should run fine if our functionality is correct. Note, we want our WNS to be a positive number. If the WNS is negative, our calculation will result in maximum clock frequency that is lower than which we specified in our .XDC file (based on the *UserClockPeriod*).

DO: present your calculations (show your work) and your result for the maximum clock frequency in the Design Record, and also state whether your design meets timing.

10. **Optional:** Since the BCD overflow is an undesirable condition, modify the design to display an error message on the 7-segment displays, to indicate an overflow. There are a number of ways to do this, it is an open-ended design problem. Mention and demonstrate your solution to the instructor. **DO: if you complete this optional activity, copy and paste the code that you added/modified, into your Design Record. Also include a picture of your RTL Schematic.**

Deliverables

By the end of the lab period **or** the beginning of the next lab period, demonstrate to the TA:

1. Your Basys3 board running with the latest iteration of the lab project, be prepared to explain the design and any part of the Vivado design and programming flow.
2. Your latest simulation and be prepared to explain the signals and their meaning.
3. Your Design Record document. Ensure that the Design Record is uploaded to the D2L Dropbox, before seeing the TA.

Your TA may ask any team members at random to answer any questions. Work together to ensure that all team members fully understand the lab project and its deliverables. You may also be asked to demonstrate your ability to proceed through the entire design flow, including:

1. Create and start a new project.
2. Synthesize and download your design to the Basys3.
3. Simulate your design and setup the waveforms.

Rubric

As the term progresses, the expectations will increase as your skills develop. All team members are required to participate in the entire lab period and to present their project to the TA. Missing team members will not receive a mark for the project. The three strikes policy outlined in the Course Outline, will be in effect for all labs.

Points	Criteria
4	Fully complete and high quality, questions answered well.
3	Fully complete and high quality, some questions not answered well or had to have other team members answer.
2	Mostly complete or answers are weakly answered by team members
0	Below acceptable for credit.

There is no 1 point given. Fractional points, e.g. 2.5, are not given.