Lab # 2

Intro to Hierarchical Design

1/18/2023

For: Troy Scevers

CST 231 – DIGITAL SYSTEM DESIGN I

By Cari Blaker

Table of Contents

Abstract 2

Introduction 2

Design 2

- Physical hardware design 2
- Synthesized hardware design 4
  - Design Structure 4
  - Modules 5

Simulation and Testing 8

Problems 8

Results and Conclusion 8

Appendix – Appendix Label goes Here (i.e. A) 9

Table of Figures 14

References 15

Abstract

The second lab of CST 231 results in a seven segment display counter on the DE1-SOC board. The 16-bit counter counts upwards, gets converted to BCD, then displays in decimal on the 7-segment display, counting with the use of a clock embedded in the board. The clock can be slowed with a clock divider, and the count is enabled, or reset with programmed switches.

Introduction

The purpose of this lab is to design a 16-bit counter, with reset and enable signals, create a clock divider, and display the counter on the seven-segment displays of the DE1-SoC board. It is also to become more independent, designing the requirements using our own problem-solving skills instead of step-by-step given instructions.

Design

Physical hardware design

The DE1-SoC board has six seven-segment displays, 10 slide-switches, as well as four 50 MHz clock signals.

Slide switch 0 was selected for the reset, and slide switch 1 for the enable–both active high (Figure 1 i

Table 3-6 Pin Assignment of Slide Switches

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SW[0]	PIN_AB12	Slide Switch[0]	3.3V
SW[1]	PIN_AC12	Slide Switch[1]	3.3V
SW[2]	PIN_AF9	Slide Switch[2]	3.3V
SW[3]	PIN_AF10	Slide Switch[3]	3.3V
SW[4]	PIN_AD11	Slide Switch[4]	3.3V
SW[5]	PIN_AD12	Slide Switch[5]	3.3V
SW[6]	PIN_AE11	Slide Switch[6]	3.3V
SW[7]	PIN_AC9	Slide Switch[7]	3.3V
SW[8]	PIN_AD10	Slide Switch[8]	3.3V
SW[9]	PIN_AE12	Slide Switch[9]	3.3V

Figure 1: Slide Switch Pin Assignments from the DE1-SoC User Manual

Only five of the seven-segment displays were necessary for the 16-bit counter. HEX0 for the ones digit through HEX4 for the ten-thousands digit.

Table 3-9 Pin Assignment of 7-segment Displays

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX0[0]	PIN_AE26	Seven Segment Digit 0[0]	3.3V
HEX0[1]	PIN_AE27	Seven Segment Digit 0[1]	3.3V
HEX0[2]	PIN_AE28	Seven Segment Digit 0[2]	3.3V
HEX0[3]	PIN_AG27	Seven Segment Digit 0[3]	3.3V
HEX0[4]	PIN_AF28	Seven Segment Digit 0[4]	3.3V
HEX0[5]	PIN_AG28	Seven Segment Digit 0[5]	3.3V
HEX0[6]	PIN_AH28	Seven Segment Digit 0[6]	3.3V
HEX1[0]	PIN_AJ29	Seven Segment Digit 1[0]	3.3V
HEX1[1]	PIN_AH29	Seven Segment Digit 1[1]	3.3V
HEX1[2]	PIN_AH30	Seven Segment Digit 1[2]	3.3V
HEX1[3]	PIN_AG30	Seven Segment Digit 1[3]	3.3V
HEX1[4]	PIN_AF29	Seven Segment Digit 1[4]	3.3V
HEX1[5]	PIN_AF30	Seven Segment Digit 1[5]	3.3V
HEX1[6]	PIN_AD27	Seven Segment Digit 1[6]	3.3V
HEX2[0]	PIN_AB23	Seven Segment Digit 2[0]	3.3V
HEX2[1]	PIN_AE29	Seven Segment Digit 2[1]	3.3V
HEX2[2]	PIN_AD29	Seven Segment Digit 2[2]	3.3V
HEX2[3]	PIN_AC28	Seven Segment Digit 2[3]	3.3V
HEX2[4]	PIN_AD30	Seven Segment Digit 2[4]	3.3V
HEX2[5]	PIN_AC29	Seven Segment Digit 2[5]	3.3V
HEX2[6]	PIN_AC30	Seven Segment Digit 2[6]	3.3V
HEX3[0]	PIN_AD26	Seven Segment Digit 3[0]	3.3V
HEX3[1]	PIN_AC27	Seven Segment Digit 3[1]	3.3V
HEX3[2]	PIN_AD25	Seven Segment Digit 3[2]	3.3V
HEX3[3]	PIN_AC25	Seven Segment Digit 3[3]	3.3V
HEX3[4]	PIN_AB28	Seven Segment Digit 3[4]	3.3V
HEX3[5]	PIN_AB25	Seven Segment Digit 3[5]	3.3V
HEX3[6]	PIN_AB22	Seven Segment Digit 3[6]	3.3V
HEX4[0]	PIN_AA24	Seven Segment Digit 4[0]	3.3V
HEX4[1]	PIN_Y23	Seven Segment Digit 4[1]	3.3V
HEX4[2]	PIN_Y24	Seven Segment Digit 4[2]	3.3V
HEX4[3]	PIN_W22	Seven Segment Digit 4[3]	3.3V
HEX4[4]	PIN_W24	Seven Segment Digit 4[4]	3.3V
HEX4[5]	PIN_V23	Seven Segment Digit 4[5]	3.3V
HEX4[6]	PIN_W25	Seven Segment Digit 4[6]	3.3V
HEX5[0]	PIN_V25	Seven Segment Digit 5[0]	3.3V
HEX5[1]	PIN_AA28	Seven Segment Digit 5[1]	3.3V
HEX5[2]	PIN_Y27	Seven Segment Digit 5[2]	3.3V
HEX5[3]	PIN_AB27	Seven Segment Digit 5[3]	3.3V
HEX5[4]	PIN_AB26	Seven Segment Digit 5[4]	3.3V
HEX5[5]	PIN_AA26	Seven Segment Digit 5[5]	3.3V
HEX5[6]	PIN_AA25	Seven Segment Digit 5[6]	3.3V

Figure 2: Seven Segment Display Pin Assignment Chart from the DE1-SoC User Manual

A 50 MHz clock signal is used from the board. CLOCK_50 was chosen.

Table 3-5 Pin Assignment of Clock Inputs

Signal Name	FPGA Pin No.	Description	I/O Standard
CLOCK_50	PIN_AF14	50 MHz clock input	3.3V
CLOCK2_50	PIN_AA16	50 MHz clock input	3.3V
CLOCK3_50	PIN_Y26	50 MHz clock input	3.3V
CLOCK4_50	PIN_K14	50 MHz clock input	3.3V
HPS_CLOCK1_25	PIN_D25	25 MHz clock input	3.3V
HPS_CLOCK2_25	PIN_F25	25 MHz clock input	3.3V

Figure 3: Clock Signal Pin Assignment Chart from the DE1-SoC User Manual

Synthesized hardware design

The design was synthesized using five different modules: a 16-bit counter, a clock divider, a BCD converter, a seven-segment decoder, and a top-level module to connect them all together.

Design Structure

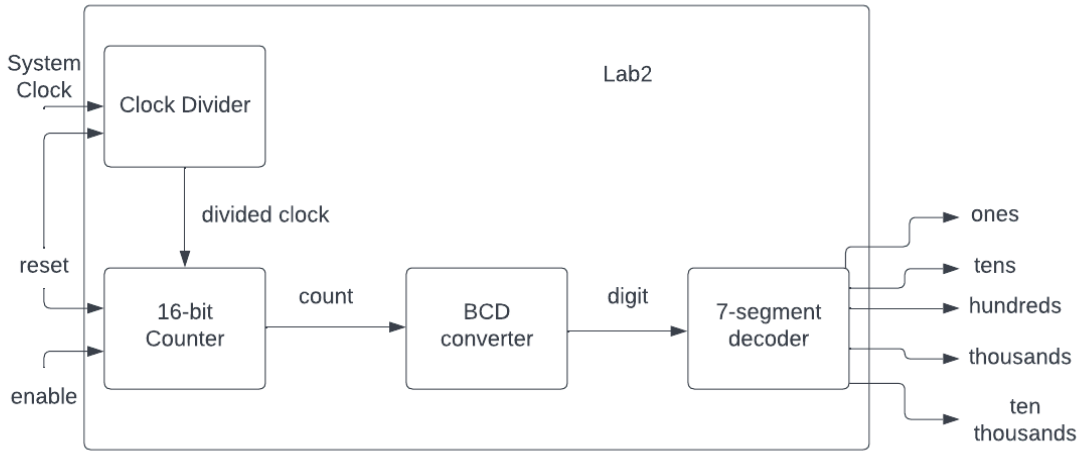

Figure 4: Design Block Diagram

The system clock signal goes directly into the clock divider, which provides the slowed clock signal for the 16-bit counter. When reset is high, it resets both the clock and the counter. The counter produces the 16 bit binary number of the count as it goes up, the BCD converter translates the binary to decimal. The seven-segment decoder then translates that number into each digit to display properly on the seven-segment.

Modules

clk_dv –the clock divider

This module takes in the system clock signal of 50 MHz and divides it to slow it so that the counter can be observed with the human mind. The default value of the divider is 25 million–making the output clock 1 Hz. The module is parameterized, however, so that any division is possible, and the signal can be slowed as much as desired. If reset becomes high, the clock signal is asynchronously set to zero, and stays there until reset is low again. Otherwise, a count keeps track of the division number and keeps the clock until the number is reached. Then, and only then, does the output clock signal change. Below is this module's RTL diagram.

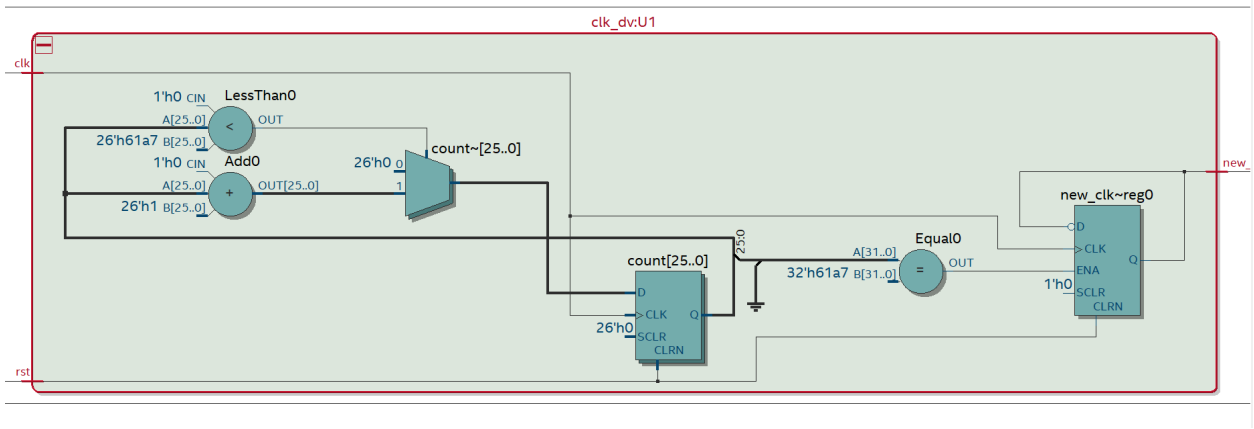

Figure 5: Clock Divider RTL Diagram

counter –the 16-bit counter

This module counts up one each clock cycle. The inputs are the clock, a reset, and an enable. An initial block starts the count at zero, and an always block adds one to the count as long as enable is high and reset is low. Reset asynchronously sets the count back to zero, and the count stays where it is if enable is low. The module is parameterized, also, so that any number counter is possible. The output is the binary number of the count.

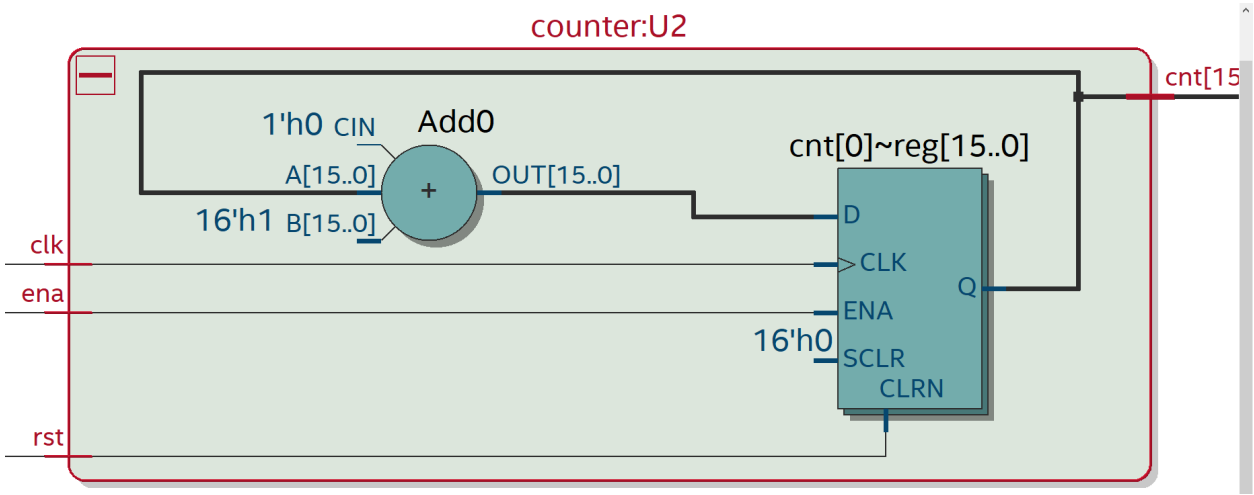

Figure 6: Counter RTL Diagram

bin2bcd–Binary to BCD converter

This module was copied from a Wikipedia page ( ). Its input is a parameterized input length, and its output register is the binary coded decimal for each digit, which adjusts based on the input length. The output provides the ones in digits [3:0], tens in [7:4], and so on. The module uses two integers, and one always block whose sensitivity list is the binary input number. It uses three for-loops to accurately gather the four digit binary of each decimal digit as output. This method is called the "double dabble" method, and a pictorial representation of it can be found below.

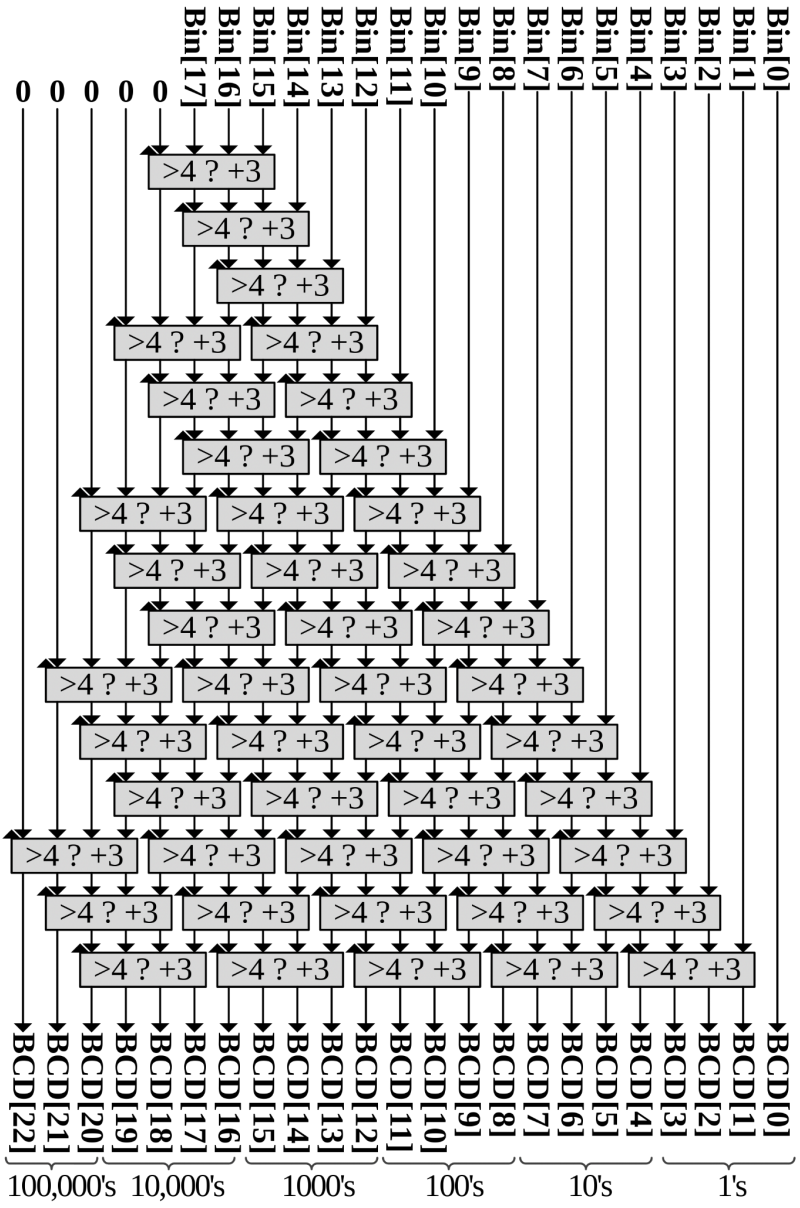

Figure 7: Double Dabble representation from Wikipedia

seg_7–Seven-Segment Display Converter

This module takes in four bits of binary coded decimal, and outputs a seven-bit register, each bit representing one line light on a seven-segment display. It uses one always block whose sensitivity list is the input binary number, and one case statement on that same number to assign the proper seven-segment display bits for numbers zero through nine.

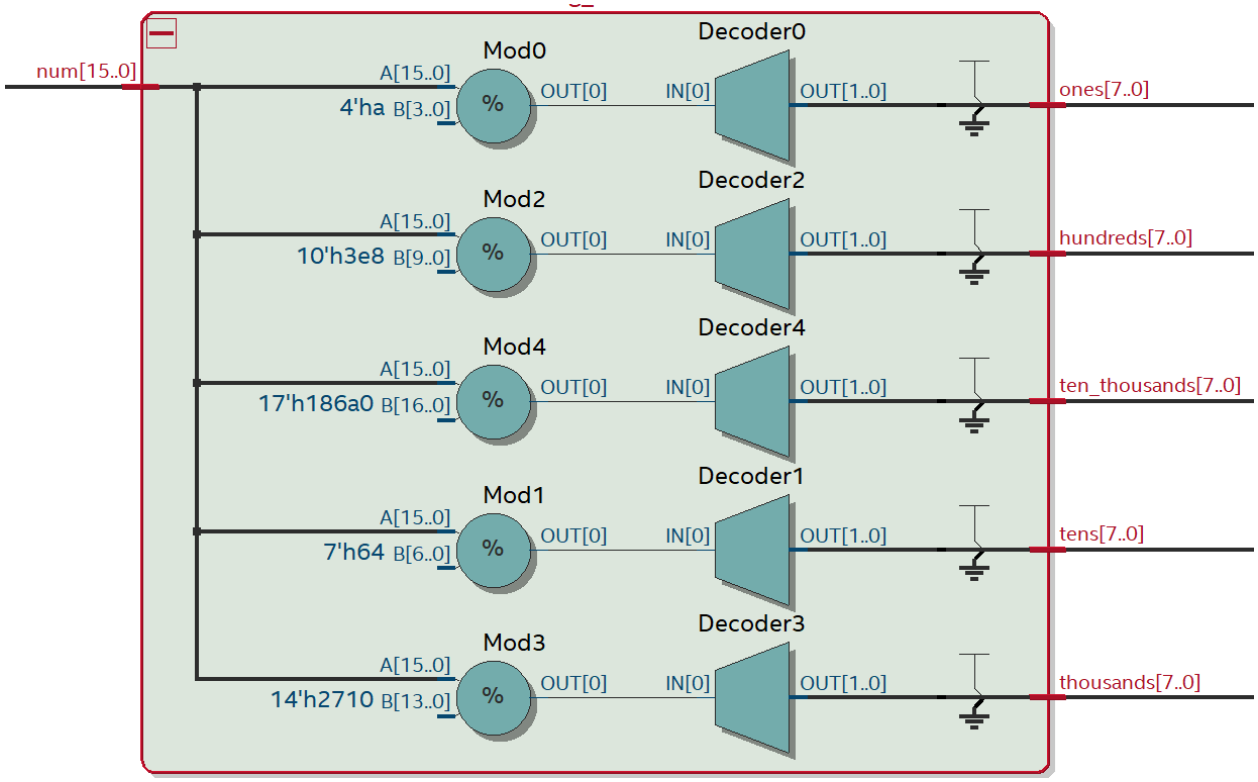

Figure 8: Seven-Segment RTL Diagram

Lab2–Top-Level Design

This module links all of the previous modules together to interact and ultimately create the desired counter functionality on the DE1-SoC board. Its inputs are a clock, reset, and an enable. Its outputs are seven bits for the ones, tens, hundreds, thousands, and ten-thousands digits being sent to the seven segment display. Three wires are declared, one for the output clock of the clock divider, a 16 bit wire for the output of the counter, and a 20 bit wire for the output of the binary to BCD converter. The first module instantiated is the clock divider, which, again, can be parameterized for any slower clock speed than 50 MHz, down to 1 Hz. The output of the clock divider is the input of the next instantiated module–the counter, which receives the same reset as the clock and the Lab2 module, and the same enable as the Lab2 module. Its output is then sent to the next instantiated module, the binary to BCD converter. The binary to BCD converter is parameterized with 16 bits, and its output is then sent four bits at a time to five seven-segment decoder instantiations, each of which are the respective outputs of this Lab2, top level design module.

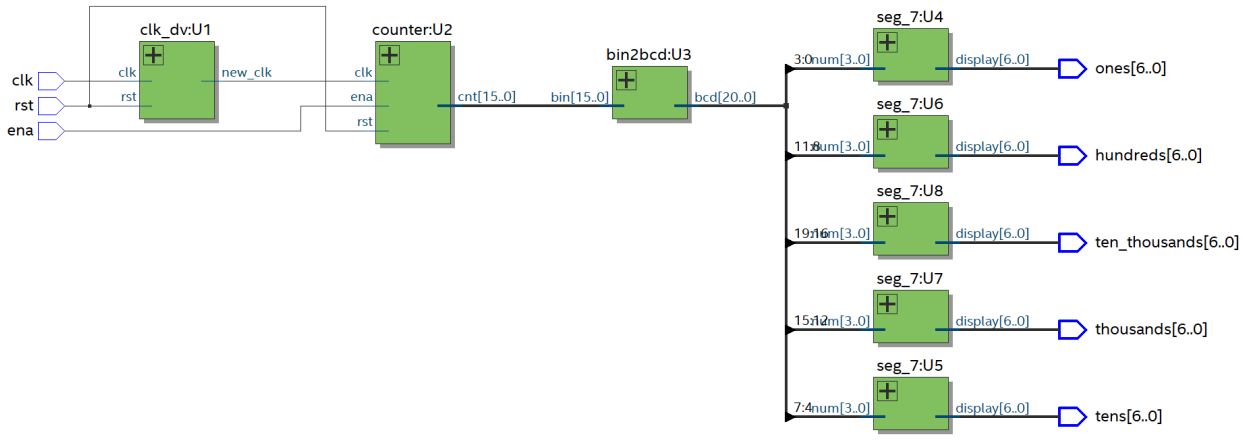

Figure 9: Top-Level Design RTL Diagram

Simulation and Testing

Set each module as the "top-level design file," and simulate it at its completion to confirm it is in working order. For the clock divider, set its input as a clock in ModelSim, and expect its output to divide the clock down by a factor of the parameter designated in the module. To test the counter, set the clock in ModelSim to clock (a quick one to get up to five digits), force the reset to be zero, and the enable to be one. Be sure to see it all the way through to the highest all-ones binary number, and flip the enable and reset to make sure they're working properly as well. The binary to BCD converter has one input and one output, so just clock each bit of the input, doubling the clock for each bit, and make sure the output corresponds correctly. The seven segment decoder can be tested in the same way. Lastly, the top level design module should mimic all of the above when it's tested in ModelSim. Plugging in the board and programming it, with the enable on and the reset off, the seven-segment display should have a five-digit decimal number counting up. The enable going off should hold the count, and the reset flipping on should reset the number. Count resumes only when the reset is off and enable is on again.

Problems

The biggest problem when testing this design had to do with the original version of the seven-segment decoder. The first design took the number straight from the counter, and included multiple modulus and division to find the BCD representations of the count, and translated them to seven-segment display code, all in one module. The design simulated exactly as expected–however, implementing the program on the board, it did not. As it turns out, each module needs to perform one task, and one task alone–and also, Wikipedia has better code for writing binary to BCD conversion, and it works perfectly when implemented in a separate module.

Results and Conclusion

This lab made it abundantly clear how important it is to define modules that perform only one task at a time. It's also good to know that code can be sourced from outside and used as long as it is cited. The importance of data and control paths being separated in different always blocks was also impressed during this lab.

Appendix – Appendix Label goes Here (i.e. A)

The appendix is used to organize large examples, pin tables, function tables, schematic diagrams, RTL diagrams for the entire hierarchical, code, top level simulation and state diagrams etc. This section should be referenced in the body of the lab report.

You must include the following:

- Pin mappings in table format for your board
- RTL diagrams for different modules
- State machine diagrams (If used)
- Pinout diagrams from datasheets for button, 7-segment, RGB led, keypad, and other possible added components.

The diagrams, figures, and tables must be properly referenced. (Figure 1, Figure 2, Table 5, etc.)

Use KiCAD, Visio, Dia, or another Schematic Capture program to draw diagrams and schematics.

Table 3-6 Pin Assignment of Slide Switches

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SW[0]	PIN_AB12	Slide Switch[0]	3.3V
SW[1]	PIN_AC12	Slide Switch[1]	3.3V
SW[2]	PIN_AF9	Slide Switch[2]	3.3V
SW[3]	PIN_AF10	Slide Switch[3]	3.3V
SW[4]	PIN_AD11	Slide Switch[4]	3.3V
SW[5]	PIN_AD12	Slide Switch[5]	3.3V
SW[6]	PIN_AE11	Slide Switch[6]	3.3V
SW[7]	PIN_AC9	Slide Switch[7]	3.3V
SW[8]	PIN_AD10	Slide Switch[8]	3.3V
SW[9]	PIN_AE12	Slide Switch[9]	3.3V

Figure 1: Slide Switch Pin Assignments from the DE1-SoC User Manual

Table 3-9 Pin Assignment of 7-segment Displays

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX0[0]	PIN_AE26	Seven Segment Digit 0[0]	3.3V
HEX0[1]	PIN_AE27	Seven Segment Digit 0[1]	3.3V
HEX0[2]	PIN_AE28	Seven Segment Digit 0[2]	3.3V
HEX0[3]	PIN_AG27	Seven Segment Digit 0[3]	3.3V
HEX0[4]	PIN_AF28	Seven Segment Digit 0[4]	3.3V
HEX0[5]	PIN_AG28	Seven Segment Digit 0[5]	3.3V
HEX0[6]	PIN_AH28	Seven Segment Digit 0[6]	3.3V
HEX1[0]	PIN_AJ29	Seven Segment Digit 1[0]	3.3V
HEX1[1]	PIN_AH29	Seven Segment Digit 1[1]	3.3V
HEX1[2]	PIN_AH30	Seven Segment Digit 1[2]	3.3V
HEX1[3]	PIN_AG30	Seven Segment Digit 1[3]	3.3V
HEX1[4]	PIN_AF29	Seven Segment Digit 1[4]	3.3V
HEX1[5]	PIN_AF30	Seven Segment Digit 1[5]	3.3V
HEX1[6]	PIN_AD27	Seven Segment Digit 1[6]	3.3V
HEX2[0]	PIN_AB23	Seven Segment Digit 2[0]	3.3V
HEX2[1]	PIN_AE29	Seven Segment Digit 2[1]	3.3V
HEX2[2]	PIN_AD29	Seven Segment Digit 2[2]	3.3V
HEX2[3]	PIN_AC28	Seven Segment Digit 2[3]	3.3V
HEX2[4]	PIN_AD30	Seven Segment Digit 2[4]	3.3V
HEX2[5]	PIN_AC29	Seven Segment Digit 2[5]	3.3V
HEX2[6]	PIN_AC30	Seven Segment Digit 2[6]	3.3V
HEX3[0]	PIN_AD26	Seven Segment Digit 3[0]	3.3V
HEX3[1]	PIN_AC27	Seven Segment Digit 3[1]	3.3V
HEX3[2]	PIN_AD25	Seven Segment Digit 3[2]	3.3V
HEX3[3]	PIN_AC25	Seven Segment Digit 3[3]	3.3V
HEX3[4]	PIN_AB28	Seven Segment Digit 3[4]	3.3V
HEX3[5]	PIN_AB25	Seven Segment Digit 3[5]	3.3V
HEX3[6]	PIN_AB22	Seven Segment Digit 3[6]	3.3V
HEX4[0]	PIN_AA24	Seven Segment Digit 4[0]	3.3V
HEX4[1]	PIN_Y23	Seven Segment Digit 4[1]	3.3V
HEX4[2]	PIN_Y24	Seven Segment Digit 4[2]	3.3V
HEX4[3]	PIN_W22	Seven Segment Digit 4[3]	3.3V
HEX4[4]	PIN_W24	Seven Segment Digit 4[4]	3.3V
HEX4[5]	PIN_V23	Seven Segment Digit 4[5]	3.3V
HEX4[6]	PIN_W25	Seven Segment Digit 4[6]	3.3V
HEX5[0]	PIN_V25	Seven Segment Digit 5[0]	3.3V
HEX5[1]	PIN_AA28	Seven Segment Digit 5[1]	3.3V
HEX5[2]	PIN_Y27	Seven Segment Digit 5[2]	3.3V
HEX5[3]	PIN_AB27	Seven Segment Digit 5[3]	3.3V
HEX5[4]	PIN_AB26	Seven Segment Digit 5[4]	3.3V
HEX5[5]	PIN_AA26	Seven Segment Digit 5[5]	3.3V
HEX5[6]	PIN_AA25	Seven Segment Digit 5[6]	3.3V

Figure 2: Seven Segment Display Pin Assignment Chart from the DE1-SoC User Manual

Table 3-5 Pin Assignment of Clock Inputs

Signal Name	FPGA Pin No.	Description	I/O Standard
CLOCK_50	PIN_AF14	50 MHz clock input	3.3V
CLOCK2_50	PIN_AA16	50 MHz clock input	3.3V
CLOCK3_50	PIN_Y26	50 MHz clock input	3.3V
CLOCK4_50	PIN_K14	50 MHz clock input	3.3V
HPS_CLOCK1_25	PIN_D25	25 MHz clock input	3.3V
HPS_CLOCK2_25	PIN_F25	25 MHz clock input	3.3V

Figure 3: Clock Signal Pin Assignment Chart from the DE1-SoC User Manual

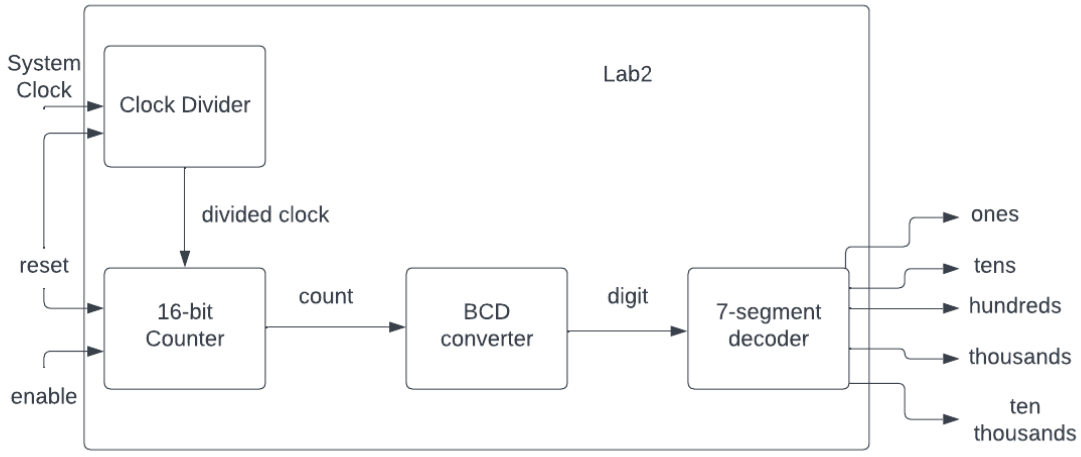

Figure 4: Design Block Diagram

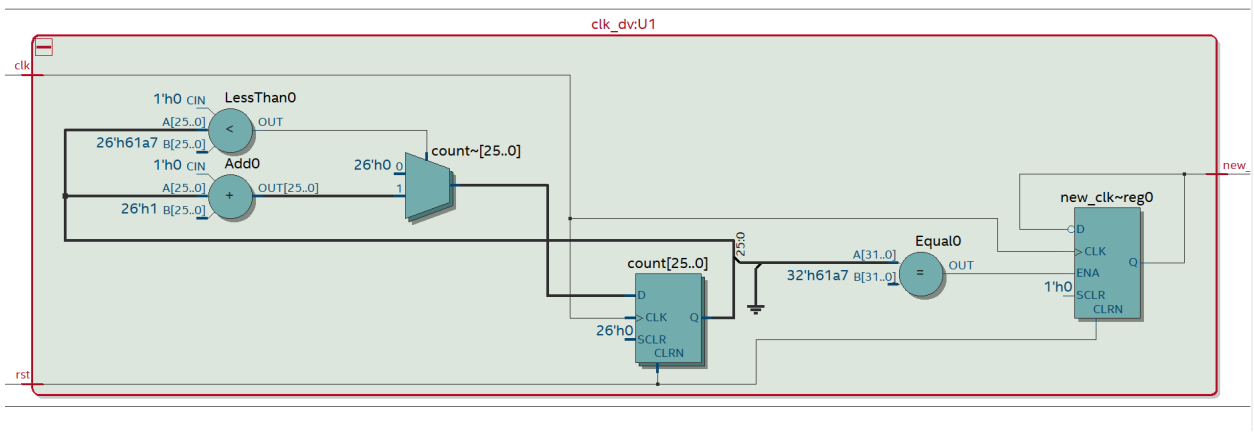

Figure 5: Design Block Diagram

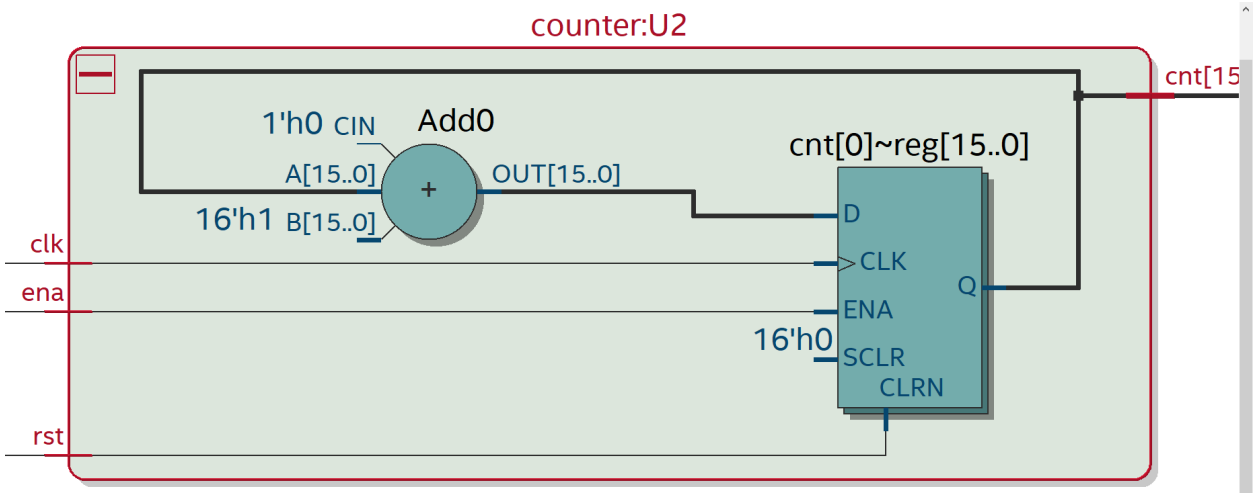

Figure 6: Counter RTL Diagram

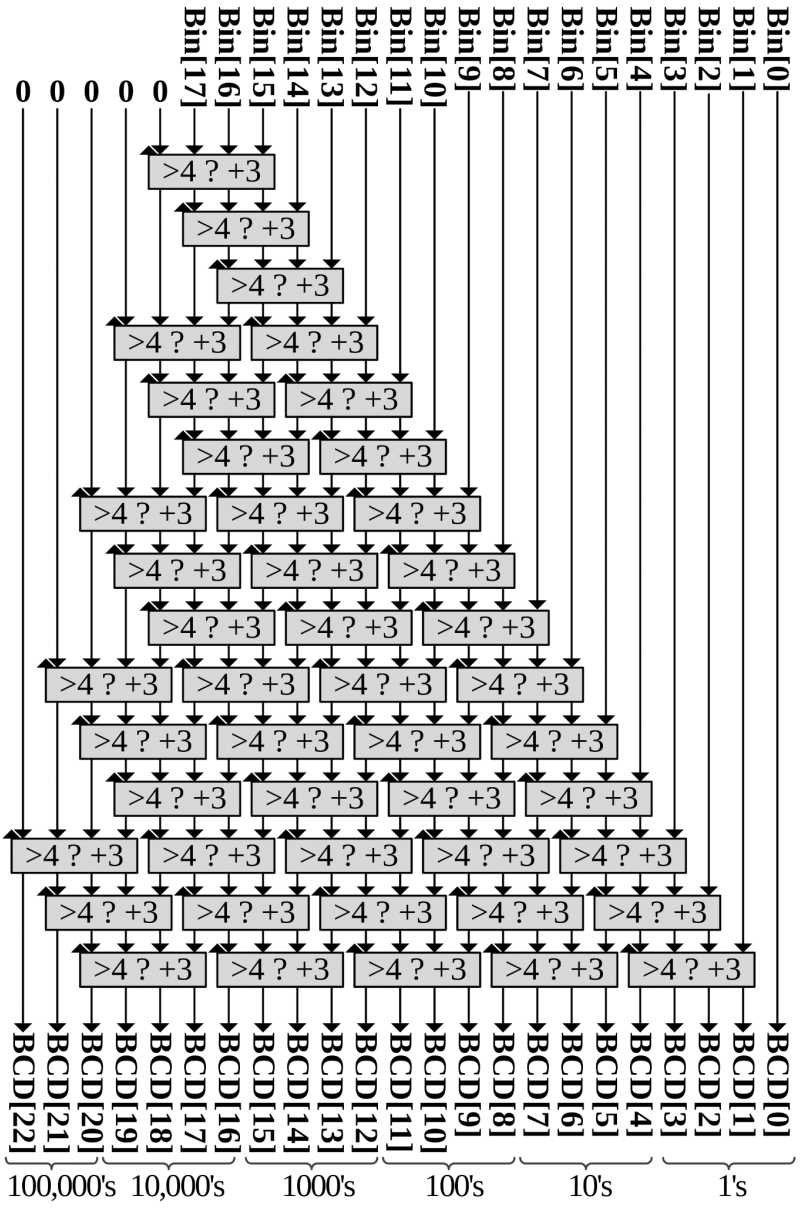

Figure 7: Double Dabble representation from Wikipedia

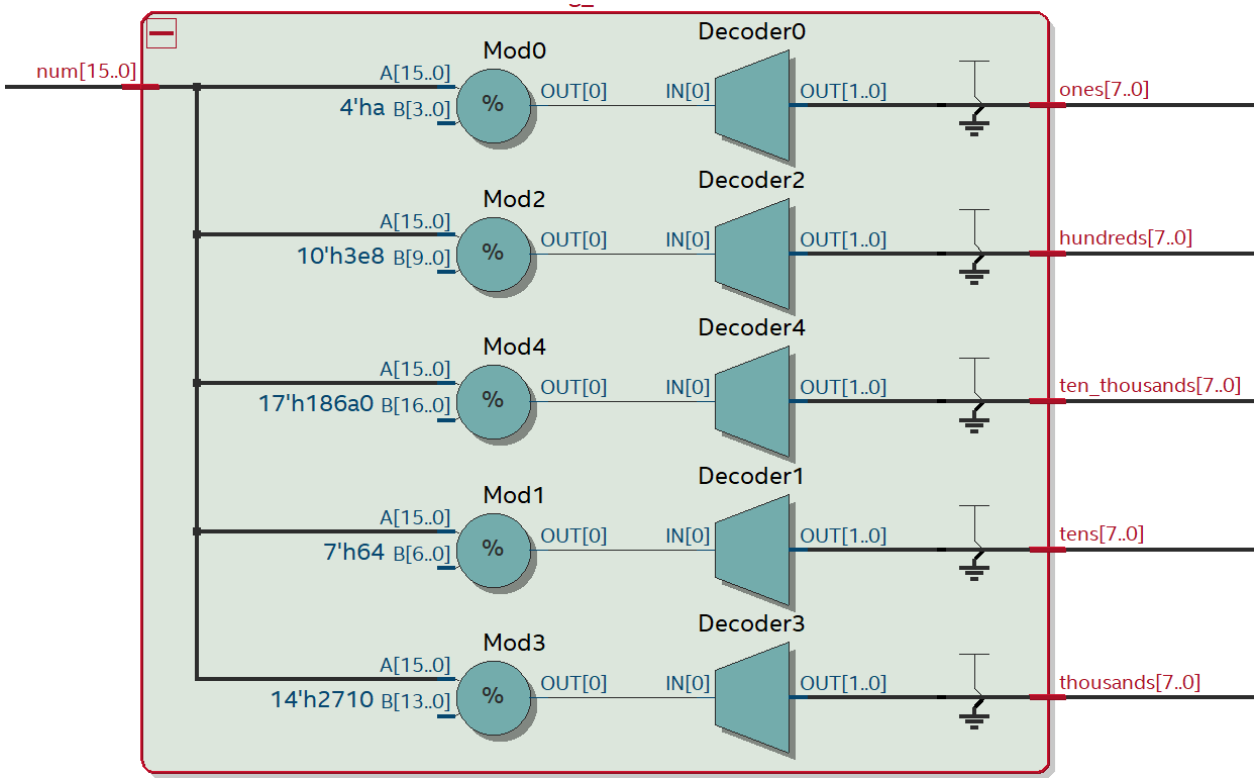

Figure 8: Seven-Segment RTL Diagram

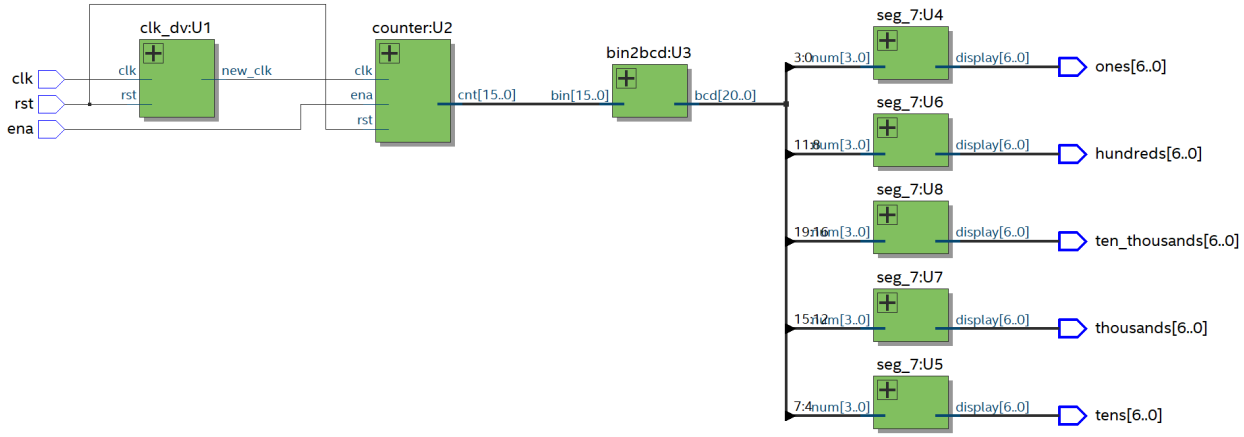

Figure 9: Top-Level Design RTL Diagram

Table of Figures

Figure 1: Slide Switch Pin Assignments 2, 9
Figure 2: Seven Segment Pin Assignments 3, 10
Figure 3: Clock Signal Pin Assignments 3, 10
Figure 4: Design Block Diagram 4, 11
Figure 5: Clock Divider RTL Diagram 5,11
Figure 6: Counter RTL Diagram 5, 11
Figure 7: Double Dabble representation 6, 12
Figure 8: Seven-Segment RTL Diagram 7, 13
Figure 9: Top-Level Design RTL Diagram 8, 13

References

[1] “Double Dabble.” Wikipedia. https://en.wikipedia.org/wiki/Double_dabble (accessed January 18, 2023)

[2] *DE1-SoC User Manual* (2014). Accessed: January 16, 2023. [Online]. Available: https://inst-fs-iad-prod.inscloudgate.net/files/3607afc4-522a-4f44-a5c1-1580f22dd310/DE1-SoC_User_manual.pdf?download=1&token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE2NzQxOTc2NTcsInVzZXJfaWQiOiIxMjgzOTAwMDAwMDAwMzc5MjYiLCJyZXNvdXJjZSI6Ii9maWxlcy8zNjA3YWZjNC01MjJhLTRmNDQtYTVjMS0xNTgwZjIyZGQzMTAvREUxLVNvQ19Vc2VyX21hbnVhbC5wZGYiLCJqdGkiOiI4YzdhYjE1ZC03NTY5LTQ1YmUtYTQ4NC0xNzc4YjZmZjI1N2MiLCJob3N0Ijoib2l0Lmluc3RydWN0dXJlLmNvbSIsIm9yaWdpbmFsX3VybCI6Imh0dHBzOi8vYTEyODM5LTMwMjY0ODguY2x1c3RlcjkxLmNhbnZhcy11c2VyLWNvbnRlbnQuY29tL2NvdXJzZXMvMTI4Mzl-MTc2OTUvZmlsZXMvMTI4Mzl-MzAyNjQ4OC9jb3Vyc2UlMjBmaWxlcy9ERTEtU29DX1VzZXJfbWFudWFsLnBkZj9kb3dubG9hZF9mcmQ9MVx1MDAyNm5vX2NhY2hlPXRydWVcdTAwMjZyZWRpcmVjdD10cnVlXHUwMDI2c2ZfdmVyaWZpZXI9ZXlKMGVYQWlPaUpLVjFRaUxDSmhiR2NpT2lKSVV6VXhNaUo5LmV5SjFjMlZ5WDJsa0lqb2lNVEk0TXprd01EQXdNREF3TURNM09USTJJaXdpY205dmRGOWhZMk52ZFc1MFgybGtJam9pTVRJNE16a3dNREF3TURBd01EQXdNREF4SWl3aWIyRjFkR2hmYUc5emRDSTZJbTlwZEM1cGJuTjBjblZqZEhWeVpTNWpiMjBpTENKeVpYUjFjbTVmZFhKc0lqcHVkV3hzTENKbVlXeHNZbUZqYTE5MWNtd2lPaUpvZEhSd2N6b3ZMMjlwZEM1cGJuTjBjblZqZEhWeVpTNWpiMjB2WTI5MWNuTmxjeTh4TnpZNU5TOW1hV3hsY3k4ek1ESTJORGc0TDJSdmQyNXNiMkZrZUDJSdmQyNXNiMkZrZWDJaeVpEMHhYSFV3TURJMlptRnNiR0poWTJ0ZmRITTlNVFkzTkRJd09UWTROeUlzSW1WNGNDSTZNVFkzTkRJd09UazROMzAuTWtxN0JiQ3dJWnhyTlp0SVNGQk9GNElCOFRVVEZGUVdnR3lwMk5QV21ITUFwdjRPVjRGMlkyYlViVjdhWnlaWF9uMExuWEJBRUJObXVvUVpRZTEzdVEiLCJleHAiOjE2NzQyODQwNTd9.rAUBUNddvg1bZIEEdqQl46VLR4ma64u9MRfpNPFygfuL28WGrF8huwyLI-6aA0d8GSLMLjCmXed53_VN1RCaog