Laboratory Exercise A

Counters

TCES 330 Digital Systems Design

Spring 2014

**Authors:**

Ben Foster \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Brian Crabtree \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submission Date: May 6, 2014

**Table of Contents**

Laboratory Assignment A…………………………………………………………………………3

Requirements:

Part I\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

Part II.……………………………………………………………………………..4

Part III \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

Part IV ..…………………………………………………………………………..5

Part V \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6

Design:

Part I……………………………………………………………………………….7

Part II \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

Part III……………………………………………………………………………..9

Part IV\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10

Part V…………………………………………………………………………….11

Test Procedures

Part I \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_14

Part II…………………………………………………………………………….15

Part III\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_15

Part IV……………………………………………………………………………15

Part V \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_15

Test Results

Part I……………………………………………………………………………...16

Part II \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_16

Part III……………………………………………………………………………16

Part IV\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_16

Part V…………………………………………………………………………….16

Observations

Part I\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_17

Part II…………………………………………………………………………….17

Part III\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_17

Part IV……………………………………………………………………………17

Part V \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_17

Conclusions………………………………………………………………………………17

Appendix A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_18

Appendix B………………………………………………………………………………23

Appendix C\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_25

Appendix D………………………………………………………………………………30

Appendix E\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_32

Appendix F………………………………………………………………………………34

**Laboratory Assignment A**

The purpose of this laboratory exercise is to design a multi-bit synchronous counter which uses T-type flip-flops. We will use the switches SW[0] and KEY[0] as inputs to the circuit. We will use the 7-segment hex displays HEX0-HEX3 as the output devices.

***Requirements:***

The objectives of each part of this lab assignment are described in this section of the report.

**Part I**

Consider the circuit in Figure 1. It is a 4-bit synchronous counter which uses four T-type flip-flops. The counter increments its count on each positive edge of the clock if the Enable signal is asserted. The counter is reset to 0 by using the Reset signal. You are to implement a 16-bit counter of this type.

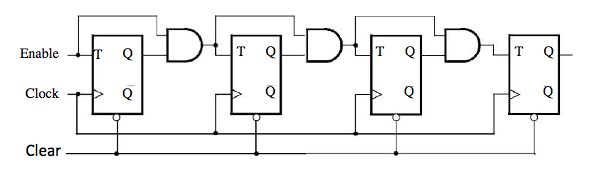


Figure 1. A 4-bit count

1. Write a Verilog file that defines a 16-bit counter by using the structure depicted in Figure 1. Your code should include a T flip-flop module that is instantiated 16 times to create the counter. Compile the circuit. How many logic elements (LEs) are used to implement your circuit? What is the maximum frequency, *Fmax*, at which your circuit can be operated?
2. Simulate your circuit to verify its correctness using ModelSim. In your report, figure out some convincing way to demonstrate correct circuit operation without including all 65,536 lines of output!
3. Augment your project with a top-level Verilog file that uses the pushbutton *KEY*0 as the *Clock* input, switch *SW*1 as an active high *Reset* and switch *SW*0 as an active high *Enable*, and 7-segment displays *HEX3-0* to display the hexadecimal count as your circuit operates. Make the necessary pin assignments needed to implement the circuit on the DE2 board, and compile the circuit.
4. Download your circuit into the FPGA chip and test its functionality by operating the implemented switches.
5. Use the Quartus II RTL Viewer to see how Quartus II software synthesized your circuit. What are the differences in comparison with Figure 1?

**Part II**

Simplify your Verilog code so that the counter specification is based on the Verilog statement

Q *<* = Q + 1;

1. Write a Verilog file that defines a 16-bit counter by using this approach. Compile the circuit. How many logic elements (LEs) are used to implement your circuit? What is the maximum frequency, *Fmax*, at which your circuit can be operated? Comment on the differences between this and Part I.
2. Simulate your circuit to verify its correctness using ModelSim.
3. Augment your project with a top-level Verilog file that uses the pushbutton *KEY*0 as the *Clock* input, switch *SW*1 as an active high *Reset* and switch *SW*0 as an active high *Enable*, and 7-segment displays *HEX3-0* to display the hexadecimal count as your circuit operates. Make the necessary pin assignments needed to implement the circuit on the DE2 board, and compile the circuit.
4. Download your circuit into the FPGA chip and test its functionality by operating the implemented switches.
5. Use the RTL Viewer to see the structure of this implementation and comment on the differences with the design from Part I.

**Part III**

Use an LPM from the Library of Parameterized modules to implement a 16-bit counter. Choose the LPM options to be consistent with the above design, i.e. with enable and synchronous clear.

1. Write a Verilog file that implements a 16-bit counter by using this approach. Compile the circuit. How many logic elements (LEs) are used to implement your circuit? What is the maximum frequency, *Fmax*, at which your circuit can be operated? Comment on the differences between this and Part I and Part II.
2. (no ModelSim for this part).
3. Augment your Verilog file to use the pushbutton *KEY*0 as the *Clock* input, switches *S W* 1 and *SW*0 as *Enable* and *Reset* inputs, and 7-segment displays *HEX3-0* to display the hexadecimal count as your circuit operates. Make the necessary pin assignments needed to implement the circuit on the DE2 board, and compile the circuit.
4. Download your circuit into the FPGA chip and test its functionality by operating the implemented switches.
5. Use the RTL Viewer to see the structure of this implementation and comment on the differences with the design from Parts I and II.

**Part IV**

Design and implement a circuit that successively flashes digits 0 through 9 on the 7-segment display *HEX*0. Each digit should be displayed for about one second. Use a counter to determine the one-second intervals. The counter should be incremented by the 50-MHz clock signal provided on the DE2 board. Do not derive any other clock signals in your design–make sure that all flip-flops in your circuit are clocked directly by the 50 MHz clock signal. This means you should have no clock warnings in your Quartus compile.

**Part V**

Design and implement a circuit that displays the word HELLO, in ticker tape fashion, on the eight 7-segment displays *H E X* 7 − 0. Make the letters move from right to left in intervals of about one second. The patterns that should be displayed in successive clock intervals are given in Table 1.

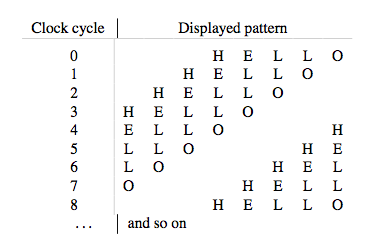


Table 1. Scrolling the word HELLO in ticker-tape fashion.

***Design***

This section of the report describes our analysis of the requirements for this laboratory exercise and resulting project design.

**Part I**

The first subtask is to write a Verilog module with the following properties which have been derived from the Requirements section:

1. Implements a T flip-flop
2. Has three one-bit inputs T, Clk, and ClrN.
3. Has 2 one-bit outputs Q and QN.

The file TFFx.v contains the module TFFx that satisfies these requirements. A listing can be found in Appendix A.

The next subtask is to write a Verilog module that contains the module TFFx and has the following properties which have been derived from the Requirements section:

1. Includes 16 instances of the TFFx module.
2. Has three inputs Clock, Enable, and Clear.
3. Has one, 16 bit, output Q.

The file CountNG.v contains the module CountNG that satisfies these requirements. A listing can be found in Appendix A.

The next subtask is to write a Verilog module with the following properties which have been derived from the Requirements section:

1. Has one 4 bit input C.
2. has one 7 bit output that goes to a particular hex display.
3. Uses the 4 bit input to determine which output.

The file Hex7seg.v contains the module Hex7seg that satisfies these requirements. A listing can be found in Appendix A.

The next subtask is to create a Quartus II project and associated verilog file that contains the CountNG and Hex7seg modules. This Verilog file has the following properties:

1. Includes the CountNG and Hex7seg modules.
2. Uses KEY0 as the clock input.
3. Uses the SW1 as an active high Clear.
4. Uses SW0 as an active high Enable.
5. Connects the output of the CountNG module to the 7 segment hex displays.

The following verilog module accomplishes this subtask:

module Part1(KEY, SW, HEX3, HEX2, HEX1, HEX0);

input [0:0] KEY;

input [1:0] SW;

output [0:6] HEX0, HEX1, HEX2, HEX3;

wire [15:0] Q;

wire Enable, Clear;

assign Enable = SW[0];

assign Clear = SW[1];

//use CountNG with N = 16

CountNG #(.N(16) U1(KEY[0], Enable, Clear, Q);

//wire up the 7-seg displays

Hex7seg H3(Q[15:12], HEX3);

Hex7seg H2(Q[11:8], HEX2);

Hex7seg H1(Q[7:4], HEX1);

Hex7seg H0(Q[3:0], HEX0);

endmodule

The file Part1.v contains the module Part1 that satisfies these requirements.

All of the above is in the Quartus II project named Part 1, contained in a folder of the same name. Note that the necessary pin assignments can be found in the file Part1\_Pins.csv, listed in Appendix F.

**Part II**

The first subtask is to write a Verilog module with the following properties which have been derived from the Requirements section:

1. Write a Verilog file that defines a 16 bit counter using the approach: Q <= Q + 1
2. The rest of the project will be the same as the one in part 1.

The file CountNG.v contains the module that satisfies these requirements. A listing can be found in Figure 1 of Appendix B.

The next subtask is to move the rest of the needed files from Part1 over to Part 2 and rename the Part1 module to Part2.

The next subtask is to use the RTL viewer to compare the circuit from this part with the one in part 1.

All of the above is in the Quartus II project named Part 2, contained in a folder of the same name. Note that the necessary pin assignments can be found in the file Part2\_Pins.csv, listed in Appendix F.

**Part III**

The first subtask is to write a Verilog module with the following properties which have been derived from the Requirements section:

1. Use an LPM from the Library of Parameterized modules to implement a 16-bit counter.

The file lpm\_counter0.v contains the module that satisfies these requirements. A listing can be found in Figure 1 of Appendix C.

The rest of the project (besides Part3) will be the same as the one in parts 1 and 2.

The next subtask is to move the rest of the needed files from Part 1 over to Part 3.

The next subtask is to create a Quartus II project and associated verilog file that contains the lpm\_counter0 and Hex7seg modules. This Verilog file has the following properties:

1. Includes the lpm\_counter0 and Hex7seg modules.
2. Uses KEY0 as the clock input.
3. Uses the SW1 as an active high Clear.
4. Uses SW0 as an active high Enable.
5. Connects the output of the lpm\_counter0 module to the 7 segment hex displays.

The following verilog module accomplishes this subtask:

module Part3( KEY, SW, HEX3, HEX2, HEX1, HEX0 );

input [0:0]KEY;

input [1:0]SW;

output [0:6]HEX0, HEX1, HEX2, HEX3; // 7-segment displays

parameter lpm\_width = 16; // counter width

wire [15:0]Q;

wire Enable, Clear;

assign Enable = SW[0];

assign Clear = SW[1];

// use CountNG with N=16

lpm\_counter #(.LPM\_WIDTH(lpm\_width)) U1( .clock(KEY[0]), .clk\_en(Enable), .sclr(Clear), .q(Q) );

// wire up the 7-seg displays

Hex7seg H3( Q[15:12], HEX3 );

Hex7seg H2( Q[11:8], HEX2 );

Hex7seg H1( Q[7:4], HEX1 );

Hex7seg H0( Q[3:0], HEX0 );

endmodule

The file Part3.v contains the module Part3 that satisfies these requirements.

The next subtask is to use the RTL viewer to compare the circuit from this part with the one in parts 1 and 2.

All of the above is in the Quartus II project named Part 3, contained in a folder of the same name. Note that the necessary pin assignments can be found in the file Part3\_Pins.csv, listed in Appendix F.

**Part IV**

The first subtask is to create a Quartus II project and associated verilog file that contains the lpm\_counter0 and Hex7seg modules. This Verilog file has the following properties:

1. Includes the lpm\_counter0 and Hex7seg modules.
2. Connects the output of the lpm\_counter0 module to the 7 segment hex displays.
3. Flashes through the digits 0-9 and stays on each for one second.

The following verilog module accomplishes this subtask.

// TCES 330

// Brian Crabtree and Ben Foster

// Lab 4, Part 4

// Depends: Hex7seg, lpm\_counter

// This module flashes the digits 0 through 9 on display HEX0

// for one second each

module Part4( HEX0, CLOCK\_50);

input CLOCK\_50;

output [0:6]HEX0; // 7-segment displays

reg ACLR; //reset for counter

reg [3:0]Q; //display

reg [25:0]X; //counter

always @(posedge CLOCK\_50)

begin

if (X == 26’d50000000) //counter has gone through one cycle.

begin

Q = (Q + 4'b0001) % 4'd10; //increment Q, and when Q > 9 reset to 0.

ACLR = 1’b1; //reset timer

end

else begin

ACLR = 1’b0; //otherwise don’t reset timer

end

// count 1 second worth of clock

lpm\_counter #(.LPM\_WIDTH(26)) U1( .aclr(ACLR), .clock(CLOCK\_50), .q(X) ); //mod 50,000,000 counter at 50MHz

// wire up the 7-seg displays

Hex7seg H0( Q, HEX0 );

endmodule

The file Part4.v contains the module Part4 that satisfies these requirements.

All of the above is in the Quartus II project named Part 4, contained in a folder of the same name. Note that the necessary pin assignments can be found in the file Part4\_Pins.csv, listed in Appendix F.

**Part V**

The first subtask is to write a Verilog module with the following properties which have been derived from the Requirements section:

1. Has one 4 bit input C.
2. Has one 7 bit output Display.
3. Uses C to determine what gets displayed on the 7 segment displays.

The file Ascii7seg.v contains the module that satisfies these requirements. A listing can be found in Figure 1 of Appendix E.

The next subtask is to create a Quartus II project and associated verilog file that contains the lpm\_counter0 and Ascii7seg modules. This Verilog file has the following properties:

The following verilog module accomplishes this task

// TCES 330

// Brian Crabtree and Ben Foster

// Lab 4, Part 4

// Depends: Ascii7seg, lpm\_counter

// This module flashes the digits 0 through 9 on display HEX0

// for one second each

module Part5( HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, HEX6, HEX7, CLOCK\_50);

input CLOCK\_50;

output [0:6]HEX0, HEX1, HEX2, HEX3, HEX4, HEX5, HEX6, HEX7; // 7-segment displays

reg [3:0] Q, A0, A1, A2, A3, A4, A5, A6, A7;

reg [25:0]X;

reg ACLR;

always @(posedge CLOCK\_50)

begin

if (X == 26’d50000000)

begin

Q = (Q + 4'b0001) % 4'd8;

A0 = Q;

A1 = (Q + 4'd1) % 4’d8;

A2 = (Q + 4'd2) % 4’d8;

A3 = (Q + 4'd3) % 4’d8;

A4 = (Q + 4'd4) % 4’d8;

A5 = (Q + 4'd5) % 4’d8;

A6 = (Q + 4'd6) % 4’d8;

A7 = (Q + 4'd7) % 4’d8;

ACLR = 1’d1; //reset counter

end

else begin

ACLR = 1’d0; //otherwise don’t reset counter (this is important)

end

end

// count 1 second worth of clock

lpm\_counter #(.LPM\_WIDTH(26)) U1( .aclr(ACLR), .clock(CLOCK\_50), .q(X) );

// wire up the 7-seg displays

Ascii7seg H0( A7, HEX0 );

Ascii7seg H1( A6, HEX1 );

Ascii7seg H2( A5, HEX2 );

Ascii7seg H3( A4, HEX3 );

Ascii7seg H4( A3, HEX4 );

Ascii7seg H5( A2, HEX5 );

Ascii7seg H6( A1, HEX6 );

Ascii7seg H7( A0, HEX7 );

endmodule

The file Part5.v contains the module Part5 that satisfies these requirements.

All of the above is in the Quartus II project named Part 5, contained in a folder of the same name. Note that the necessary pin assignments can be found in the file Part5\_Pins.csv, listed in Appendix F.

***Test Procedures***

The following test procedures will be used to verify that each part of this laboratory exercise satisfies the requirements given in the Requirements section above.

**Part I**

The following test procedure will be used to verify that the Quartus II project Part1 satisfies the requirements for this part.

1. The counter circuit will be tested using ModelSim and the testbench shown in Figure 4, Appendix A. This testbench generates the test vectors shown in Table 1 and outputs the multiplexer output X. Simulations will be run in order to verify the behavior shown in Table 1.
2. Open the project and verify that compilation produces no errors or unallowed warnings.
3. Verify that the RTL Viewer shows the correct circuit.
4. Load the project onto the DE2 board without errors.
5. Verify that the HEX3-0 displays are showing the correct output as the KEY0 button is pushed.
6. Verify that the output is being cleared when SW1 is switched ON and the KEY0 button is pushed with the clock cycle.
7. Generate the test vectors shown in Table 1 and verify the corresponding outputs, X.

Table 1. Part 1 Test Vectors

|  |  |  |  |
| --- | --- | --- | --- |
| **Enable = SW0** | **Clear = SW1** | **stime** | **X** |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 10 | 1 |
| 1 | 0 | 30 | 2 |
| 1 | 0 | 50 | 3 |
| 1 | 0 | 70 | 4 |
| ... | … | ... | ... |
| 1 | 0 | 290 | 15 |

**Part II**

The test procedure for Part 2 is exactly the same as Part 1.

**Part III**

The following test procedure will be used to verify that the Quartus II project Part1 satisfies the requirements for this part.

1. Open the project and verify that compilation produces no errors or unallowed warnings.
2. Verify that the RTL Viewer shows the correct circuit.
3. Load the project onto the DE2 board without errors.
4. Verify that the HEX3-0 displays are showing the correct output as the KEY0 button is pushed.
5. Verify that the output is being cleared when SW1 is switched ON and the KEY0 button is pushed with the clock cycle.

**Part IV**

The following test procedure will be used to verify that the Quartus II project Part1 satisfies the requirements for this part.

1. Open the project and verify that compilation produces no errors or unallowed warnings.
2. Load the project onto the DE2 board without errors.
3. Verify that the HEX0 display is showing the correct output.

**Part V**

The following test procedure will be used to verify that the Quartus II project Part1 satisfies the requirements for this part.

1. Open the project and verify that compilation produces no errors or unallowed warnings.
2. Load the project onto the DE2 board without errors.
3. Verify that the HEX7-0 display is showing the correct output.

***Test Results***

**Part I**

The following observations correspond to the numbers in the Test Procedures section for this part:

1. The simulation using ModelSim produced the output shown in Table 1 as expected.
2. Compilation was successful, with the Flow Summary shown in Figure 5 in Appendix A.
3. The RTL View produced the result shown in Figure 6. This is a reasonable result as it depicts a CountNG module leading into 4 Hex7seg modules that produce the output.
4. The project was downloaded to the DE2 board without errors.
5. The displays HEX3-0 displayed the hex digits 1-F and counted up as the KEY0 button was pressed with the clock as described in the test procedures.
6. All test vectors as specified in Table 1 produced the corresponding outputs X.

**Part II**

The test results for part 2 were similar to the test results for part 1. The only difference between the two was that the Flow Summary produced for this part is in Figure 2 of Appendix B and the RTL View produced for this part is in Figure 3 of Appendix B.

**Part III**

The test results for Part 3 were similar to the test results for part 1. The only difference between them was that the Flow Summary produced for this part is in Figure 2 of Appendix C and the RTL View produced for this part is in Figure 3 of Appendix C.

**Part IV**

The test results for Part 4 were all expected. No warnings were generated and no errors came up. The HEX0 display flashed through the digits 0-9 staying on each for one second, just as planned.

The Flow Summary produced for this part is in Figure 1 of Appendix D. The RTL View produced for this part is in Figure 2 of Appendix D.

**Part V**

The test results for Part 5 were all expected. No warnings were generated and no errors came up. The HEX7-0 displays showed HELLO scrolling across the displays every second, just as planned. The Flow Summary produced for this part is in Figure 2 of Appendix E. The RTL View produced for this part is in Figure 3 of Appendix E.

***Observations***

**Part I**

Part 1 project was written for us by Robert Gutmann. The testCountNG module does not specifically test the Enable and Clear switches very well since setting the Enable to 0 or the clear to 1 would cause an infinite loop that will not terminate. This is okay since there are only 4 different outcomes from different switch positions and all can be tested by hand on the board. All tests were successful as defined by the Test Procedures section.

**Part II**

The only thing that was changed in this part was the CountNG module. Instead of instantiating the TFFx module 16 times, a counter was used. This changed a part of the RTL viewer. In the RTL viewer, when you double click on the CountNG in Part 1, 16 instances of TFFx show up, making it difficult to put that specific image on this lab report while when you double click on the CountNG in Part 2, what shows up is an adder, a register and two muxes that represent the counter. All tests were successful as defined by the Test Procedures section.

**Part III**

The only thing that was changed in this part was that the CountNG module is no longer in the project. Instead, the lpm\_counter0.v module is used in place of a counter. In the RTL viewer, the CountNG module was replaced with the lpm\_counter:U1 module. All tests were successful as defined by the Test Procedures section.

**Part IV**

There were no problems with writing the verilog code for Part 4. All tests were successful as defined by the Test Procedures section.

**Part V**

There was slight trouble at first getting the specific clock defined warning to go away, but it was resolved in the end. All tests were successful as defined by the Test Procedures section.

**Conclusion**

In conclusion, we learned that there are many different ways to code a project that does the exact same thing. We also learned how to use the Library of Parameterized modules without creating an entirely separate module. All projects were compiled and completed without errors.

***Appendix A***

This appendix contains listings of the files used in Part 1.

module TFFx(T, Clk, ClrN, Q, QN);

input T, Clk, ClrN;

output reg Q = 0; // the output

output QN;

assign QN = ~Q; //the inverted output

always @(posedge Clk) begin

if(~ClrN) begin

Q <= 0;

end //not enabled

else begin //enabled

if(T)

Q <= ~Q;

else

Q <= Q;

end //enabled

end //always

endmodule

Figure 1. Listing of TFFx.v.

module CountNG(Clock, Enable, Clear, Q);

parameter N = 16; //default size of counter (bits)

input Clock; //System clock

input Enable; //Enable

input Clear; //Clear

output [N-1:0] Q; //counter output

genvar i; //used in generate block

wire T[N-1:0]; //all the T’s wire

wire Qn[N-1:0]; //

//reference: module TFFx(T, Clk, ClrN, Q, QN);

generate

for(i = 0; i < N; i = i + 1) begin: Tffg

assign T[i] = i ? T[i-1] & Q[i-1] : Enable; //do something special for I = 0

TFFx U( .Clk(Clock), .T(T[i]), .ClrN(~Clear), .Q(Q[i]), .QN(Qn[i]) );

end //for loop

endgenerate

endmodule

Figure 2. Listing of CountNG.v

// TCES 330

// R. Gutmann

// Generic 7-Segment Hex Decoder, long version

// This module implements a 7-segment decoder '0' through 'F'

//

module Hex7seg( C, Display );

input [3:0]C; // input code

output [0:6]Display; // seven-segment display output

// implement the truth table (active low segments)

assign Display[0] = (~C[3]&~C[2]&~C[1]&C[0]) + (~C[3]&C[2]&~C[1]&~C[0]) + (C[3]&~C[2]&C[1]&C[0]) + (C[3]&C[2]&~C[1]&C[0]);

assign Display[1] = (~C[3]&C[2]&~C[1]&C[0]) + (~C[3]&C[2]&C[1]&~C[0]) + (C[3]&~C[2]&C[1]&C[0]) + (C[3]&C[2]&~C[1]&~C[0]) + (C[3]&C[2]&C[1]&~C[0]) + (C[3]&C[2]&C[1]&C[0]);

assign Display[2] = (~C[3]&~C[2]&C[1]&~C[0]) + (C[3]&C[2]&~C[1]&~C[0]) + (C[3]&C[2]&C[1]&~C[0]) + (C[3]&C[2]&C[1]&C[0]);

assign Display[3] = (~C[3]&~C[2]&~C[1]&C[0]) + (~C[3]&C[2]&~C[1]&~C[0]) + (~C[3]&C[2]&C[1]&C[0]) + (C[3]&~C[2]&C[1]&~C[0]) + (C[3]&C[2]&C[1]&C[0]);

assign Display[4] = (~C[3]&~C[2]&~C[1]&C[0]) + (~C[3]&~C[2]&C[1]&C[0]) + (~C[3]&C[2]&~C[1]&~C[0]) +(~C[3]&C[2]&~C[1]&C[0]) + (~C[3]&C[2]&C[1]&C[0]) + (C[3]&~C[2]&~C[1]&C[0]);

assign Display[5] = (~C[3]&~C[2]&~C[1]&C[0]) + (~C[3]&~C[2]&C[1]&~C[0]) + (~C[3]&~C[2]&C[1]&C[0]) + (~C[3]&C[2]&C[1]&C[0]) + (C[3]&C[2]&~C[1]&C[0]);

assign Display[6] = (~C[3]&~C[2]&~C[1]&~C[0]) + (~C[3]&~C[2]&~C[1]&C[0]) + (~C[3]&C[2]&C[1]&C[0]) + (C[3]&C[2]&~C[1]&~C[0]);

endmodule

Figure 3. Listing of Hex7seg.v

// TCES 330

// 4/10/12

// R. Gutmann

// This module is a testbench for the module

// ThreeOnes. Use ModelSim to perform the simulation

`timescale 1ns/1ns

module testCountNG; // note: no I/O ports

localparam M=4;

reg Clk; // our system clock

reg Enable; // test module enable bit

reg Clear; // test module clear bit

wire [M-1:0] X; // test module output

CountNG #(.N(M)) DUT( Clk, Enable, Clear, X );

// generate the clock signal

always begin

Clk <= 0;

#10;

Clk <= 1;

#10;

end

// when to stop

always @( posedge Clk )

if ( X == {M{1'b1}} ) // X = all 1s

$stop;

// generate the test stimulus

// note this really doesn't test Enable and Clear

// very well

initial

begin

Enable = 1'b1;

Clear = 1'b0;

end

// print out something

initial

$monitor( $stime,, X );

endmodule

Figure 4. Listing of testCountNG.v

Flow Status Successful - Mon May 5 19:30:54 2014

Quartus II 32-bit Version 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition

Revision Name Part1

Top-level Entity Name Part1

Family Cyclone II

Device EP2C35F672C6

Timing Models Final

Total logic elements 50 / 33.216 ( < 1 % )

Total combinational functions 50 / 33.216 ( < 1 % )

Dedicated logic registers 16 / 33.216 ( < 1 % )

Total registers 16

Total pins 31 / 475 ( 7 % )

Total vitrual pins 0

Total memory bits 0 / 483.840 ( 0 % )

Embedded Multiplier 9-bit elements 0 / 70 ( 0 % )

Total PLLs 0 / 4 ( 0 % )

Figure 5. Flow Summary for Part1 of lab A.

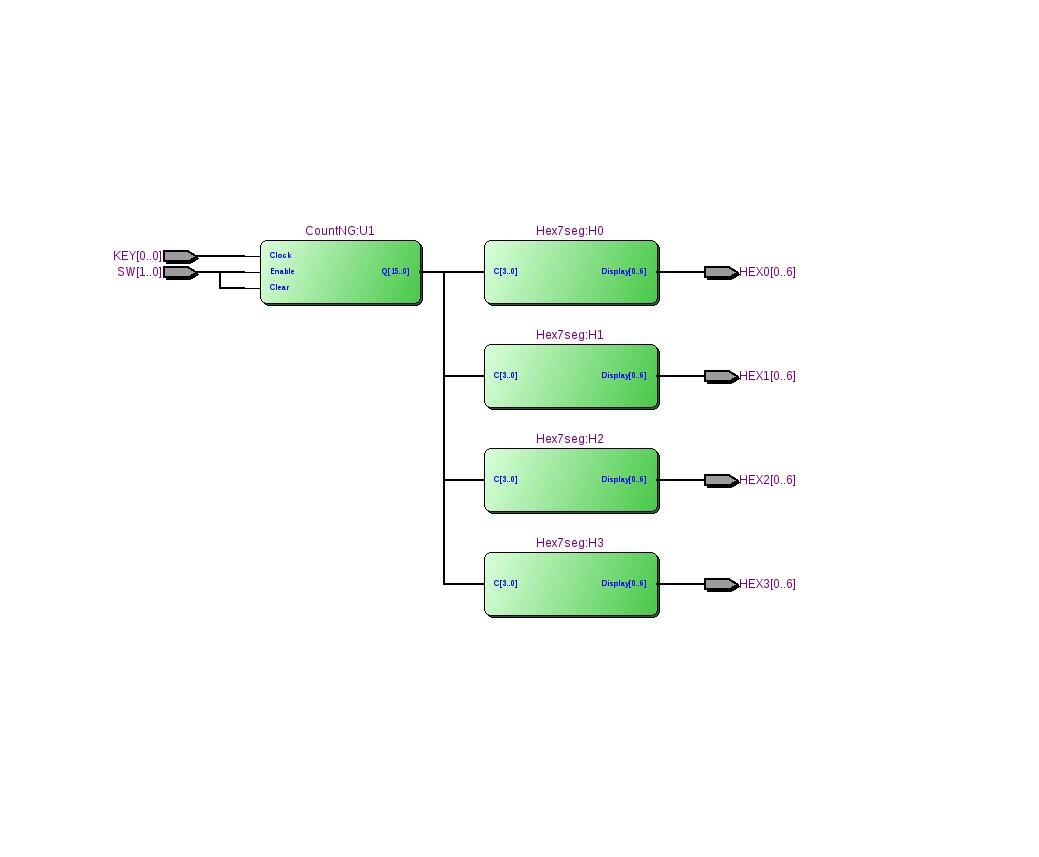


Figure 6. RTL for Part 1

***Appendix B***

This appendix contains listings of the files used in Part 2.

module CountNG(Clock, Enable, Clear, Q);

parameter N = 16; //default size of counter (bits)

input Clock; //System clock

input Enable; //Enable

input Clear; //Clear

output [N-1:0] Q; //counter output

wire T[N-1:0]; //all the T’s wire

wire Qn[N-1:0]; //

always @(posedge Clock) begin

if(Enable) begin

Q <= Q + 1’b1;

end

if(Clear) begin

Q <= 1’b0;

end

end

endmodule

Figure 1. Listing of CountNG.v

Flow Status Successful - Mon May 5 19:47:35 2014

Quartus II 32-bit Version 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition

Revision Name Part2

Top-level Entity Name Part2

Family Cyclone II

Device EP2C35F672C6

Timing Models Final

Total logic elements 45 / 33.216 ( < 1 % )

Total combinational functions 45 / 33.216 ( < 1 % )

Dedicated logic registers 16 / 33.216 ( < 1 % )

Total registers 16

Total pins 31 / 475 ( 7 % )

Total vitrual pins 0

Total memory bits 0 / 483.840 ( 0 % )

Embedded Multiplier 9-bit elements 0 / 70 ( 0 % )

Total PLLs 0 / 4 ( 0 % )

Figure 2. Flow Summary for Part 2 of Lab A

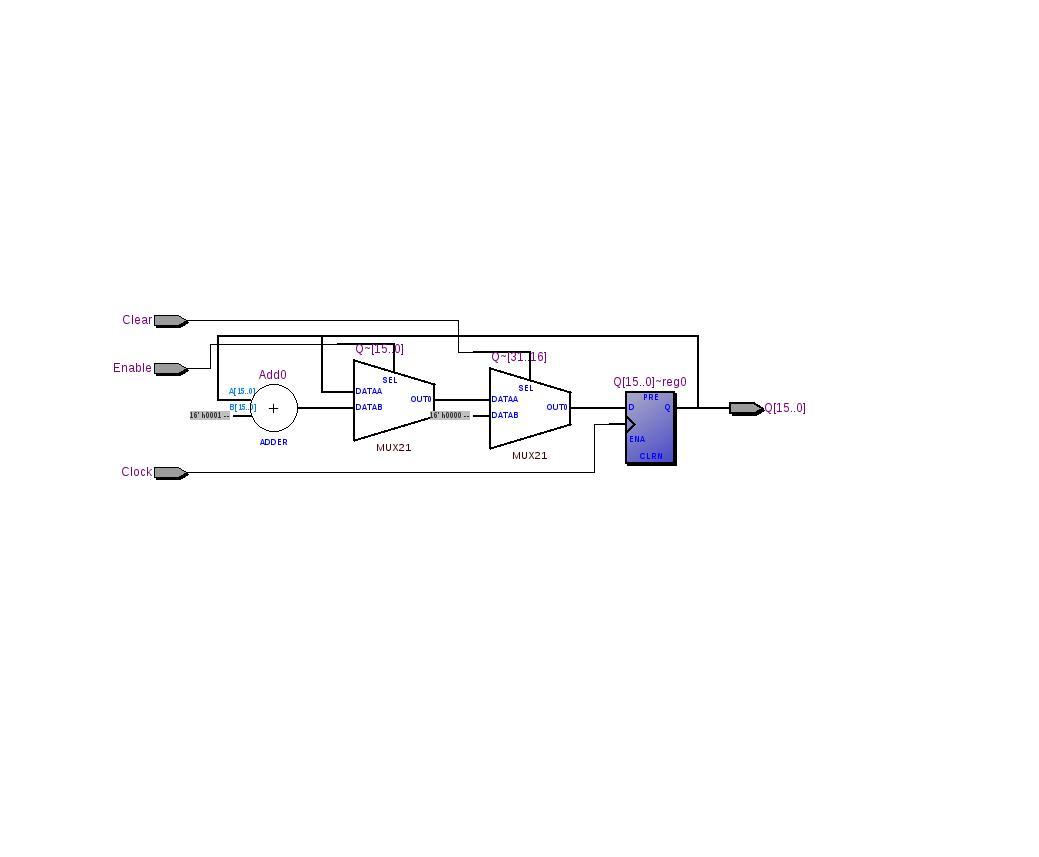


Figure 3. RTL for Part 2

***Appendix C***

This appendix contains listings of the files used in Part 3.

// megafunction wizard: %LPM\_COUNTER%

// GENERATION: STANDARD

// VERSION: WM1.0

// MODULE: LPM\_COUNTER

// ============================================================

// File Name: lpm\_counter0.v

// Megafunction Name(s):

// LPM\_COUNTER

//

// Simulation Library Files(s):

// lpm

// ============================================================

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS IS A WIZARD-GENERATED FILE. DO NOT EDIT THIS FILE!

//

// 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Copyright (C) 1991-2013 Altera Corporation

//Your use of Altera Corporation's design tools, logic functions

//and other software and tools, and its AMPP partner logic

//functions, and any output files from any of the foregoing

//(including device programming or simulation files), and any

//associated documentation or information are expressly subject

//to the terms and conditions of the Altera Program License

//Subscription Agreement, Altera MegaCore Function License

//Agreement, or other applicable license agreement, including,

//without limitation, that your use is for the sole purpose of

//programming logic devices manufactured by Altera and sold by

//Altera or its authorized distributors. Please refer to the

//applicable agreement for further details.

// synopsys translate\_off

`timescale 1 ps / 1 ps

// synopsys translate\_on

module lpm\_counter0 (

clk\_en,

clock,

sclr,

q);

input clk\_en;

input clock;

input sclr;

output [15:0] q;

wire [15:0] sub\_wire0;

wire [15:0] q = sub\_wire0[15:0];

lpm\_counter LPM\_COUNTER\_component (

.clk\_en (clk\_en),

.clock (clock),

.sclr (sclr),

.q (sub\_wire0),

.aclr (1'b0),

.aload (1'b0),

.aset (1'b0),

.cin (1'b1),

.cnt\_en (1'b1),

.cout (),

.data ({16{1'b0}}),

.eq (),

.sload (1'b0),

.sset (1'b0),

.updown (1'b1));

defparam

LPM\_COUNTER\_component.lpm\_direction = "UP",

LPM\_COUNTER\_component.lpm\_port\_updown = "PORT\_UNUSED",

LPM\_COUNTER\_component.lpm\_type = "LPM\_COUNTER",

LPM\_COUNTER\_component.lpm\_width = 16;

endmodule

// ============================================================

// CNX file retrieval info

// ============================================================

// Retrieval info: PRIVATE: ACLR NUMERIC "0"

// Retrieval info: PRIVATE: ALOAD NUMERIC "0"

// Retrieval info: PRIVATE: ASET NUMERIC "0"

// Retrieval info: PRIVATE: ASET\_ALL1 NUMERIC "1"

// Retrieval info: PRIVATE: CLK\_EN NUMERIC "1"

// Retrieval info: PRIVATE: CNT\_EN NUMERIC "0"

// Retrieval info: PRIVATE: CarryIn NUMERIC "0"

// Retrieval info: PRIVATE: CarryOut NUMERIC "0"

// Retrieval info: PRIVATE: Direction NUMERIC "0"

// Retrieval info: PRIVATE: INTENDED\_DEVICE\_FAMILY STRING "Cyclone II"

// Retrieval info: PRIVATE: ModulusCounter NUMERIC "0"

// Retrieval info: PRIVATE: ModulusValue NUMERIC "0"

// Retrieval info: PRIVATE: SCLR NUMERIC "1"

// Retrieval info: PRIVATE: SLOAD NUMERIC "0"

// Retrieval info: PRIVATE: SSET NUMERIC "0"

// Retrieval info: PRIVATE: SSET\_ALL1 NUMERIC "1"

// Retrieval info: PRIVATE: SYNTH\_WRAPPER\_GEN\_POSTFIX STRING "0"

// Retrieval info: PRIVATE: nBit NUMERIC "16"

// Retrieval info: PRIVATE: new\_diagram STRING "1"

// Retrieval info: LIBRARY: lpm lpm.lpm\_components.all

// Retrieval info: CONSTANT: LPM\_DIRECTION STRING "UP"

// Retrieval info: CONSTANT: LPM\_PORT\_UPDOWN STRING "PORT\_UNUSED"

// Retrieval info: CONSTANT: LPM\_TYPE STRING "LPM\_COUNTER"

// Retrieval info: CONSTANT: LPM\_WIDTH NUMERIC "16"

// Retrieval info: USED\_PORT: clk\_en 0 0 0 0 INPUT NODEFVAL "clk\_en"

// Retrieval info: USED\_PORT: clock 0 0 0 0 INPUT NODEFVAL "clock"

// Retrieval info: USED\_PORT: q 0 0 16 0 OUTPUT NODEFVAL "q[15..0]"

// Retrieval info: USED\_PORT: sclr 0 0 0 0 INPUT NODEFVAL "sclr"

// Retrieval info: CONNECT: @clk\_en 0 0 0 0 clk\_en 0 0 0 0

// Retrieval info: CONNECT: @clock 0 0 0 0 clock 0 0 0 0

// Retrieval info: CONNECT: @sclr 0 0 0 0 sclr 0 0 0 0

// Retrieval info: CONNECT: q 0 0 16 0 @q 0 0 16 0

// Retrieval info: GEN\_FILE: TYPE\_NORMAL lpm\_counter0.v TRUE

// Retrieval info: GEN\_FILE: TYPE\_NORMAL lpm\_counter0.inc FALSE

// Retrieval info: GEN\_FILE: TYPE\_NORMAL lpm\_counter0.cmp FALSE

// Retrieval info: GEN\_FILE: TYPE\_NORMAL lpm\_counter0.bsf TRUE FALSE

// Retrieval info: GEN\_FILE: TYPE\_NORMAL lpm\_counter0\_inst.v FALSE

// Retrieval info: GEN\_FILE: TYPE\_NORMAL lpm\_counter0\_bb.v TRUE

// Retrieval info: LIB\_FILE: lpm

Figure 1. Listing of lpm\_counter0.v

Flow Status Successful - Mon May 5 19:51:46 2014

Quartus II 32-bit Version 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition

Revision Name Part3

Top-level Entity Name Part3

Family Cyclone II

Device EP2C35F672C6

Timing Models Final

Total logic elements 45 / 33.216 ( < 1 % )

Total combinational functions 45 / 33.216 ( < 1 % )

Dedicated logic registers 16 / 33.216 ( < 1 % )

Total registers 16

Total pins 31 / 475 ( 7 % )

Total vitrual pins 0

Total memory bits 0 / 483.840 ( 0 % )

Embedded Multiplier 9-bit elements 0 / 70 ( 0 % )

Total PLLs 0 / 4 ( 0 % )

Figure 2. Flow Summary for Part 3 of Lab A

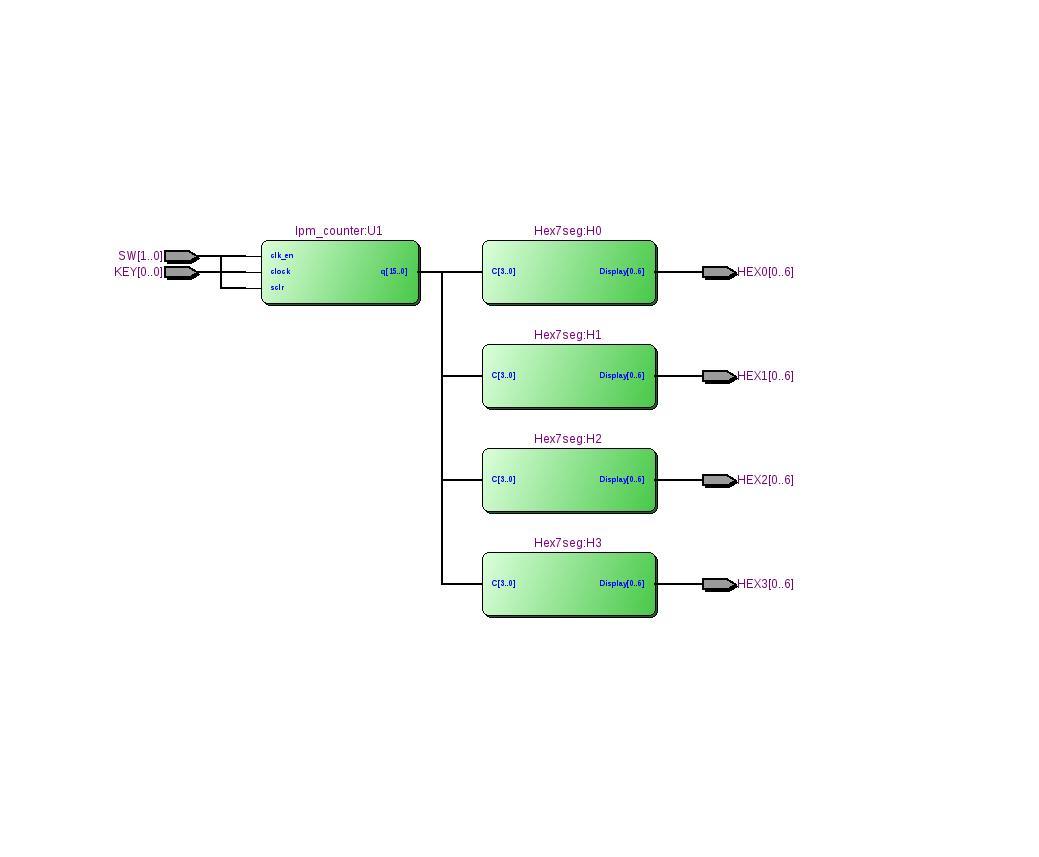


Figure 3. RTL view for Part 3

***Appendix D***

This appendix contains listings of the files used in Part 4.

Flow Status Successful - Mon May 5 20:02:16 2014

Quartus II 32-bit Version 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition

Revision Name Part4

Top-level Entity Name Part4

Family Cyclone II

Device EP2C35F672C6

Timing Models Final

Total logic elements 53 / 33.216 ( < 1 % )

Total combinational functions 53 / 33.216 ( < 1 % )

Dedicated logic registers 34 / 33.216 ( < 1 % )

Total registers 34

Total pins 8 / 475 ( 7 % )

Total vitrual pins 0

Total memory bits 0 / 483.840 ( 0 % )

Embedded Multiplier 9-bit elements 0 / 70 ( 0 % )

Total PLLs 0 / 4 ( 0 % )

Figure 1. The Flow Summary for Part 4 of Lab A

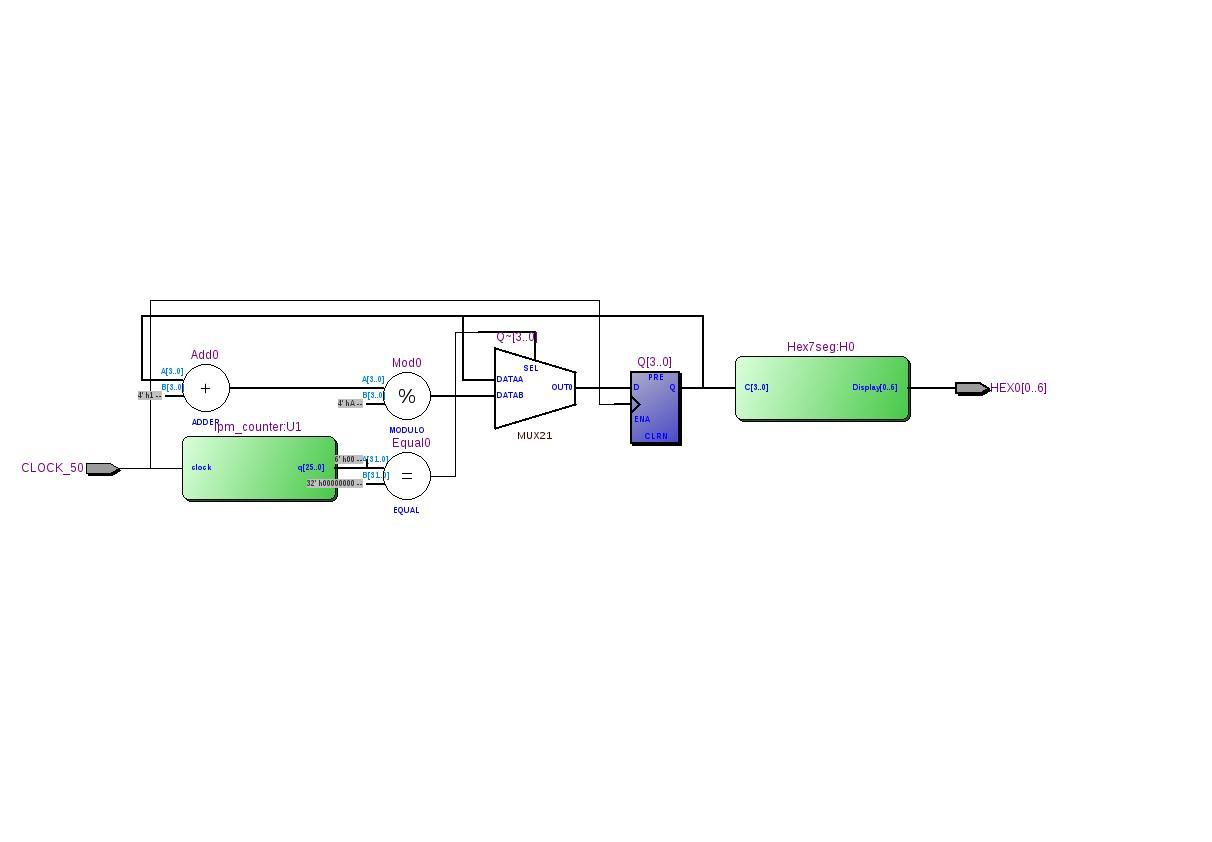


Figure 2. RTL View for Part 4

***Appendix E***

This appendix contains listings of the files used in Part 5.

// TCES 330

// Brian Crabtree and Ben Foster

// 7 segment decoder for HELLO string

// This module uses

module Ascii7seg( C, Display );

input [3:0]C; // input code

output [0:6]Display; // seven-segment display output

assign Display = D;

reg [6:0]D;

always@ (C)

case (C)

3 : D = 7'b1001000; // H

4 : D = 7'b0110000; // E

5 : D = 7'b1110001; // L

6 : D = 7'b1110001; // L

7 : D = 7'b0000001; // O

default : D = 7'b1111111; // blank

endcase

endmodule

Figure 1. Listing of Ascii7seg.v

Flow Status Successful - Mon May 5 20:10:43 2014

Quartus II 32-bit Version 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition

Revision Name Part5

Top-level Entity Name Part5

Family Cyclone II

Device EP2C35F672C6

Timing Models Final

Total logic elements 105 / 33.216 ( < 1 % )

Total combinational functions 105 / 33.216 ( < 1 % )

Dedicated logic registers 41 / 33.216 ( < 1 % )

Total registers 41

Total pins 57 / 475 ( 7 % )

Total vitrual pins 0

Total memory bits 0 / 483.840 ( 0 % )

Embedded Multiplier 9-bit elements 0 / 70 ( 0 % )

Total PLLs 0 / 4 ( 0 % )

Figure 2. Flow Summary for Part 5 of Lab A

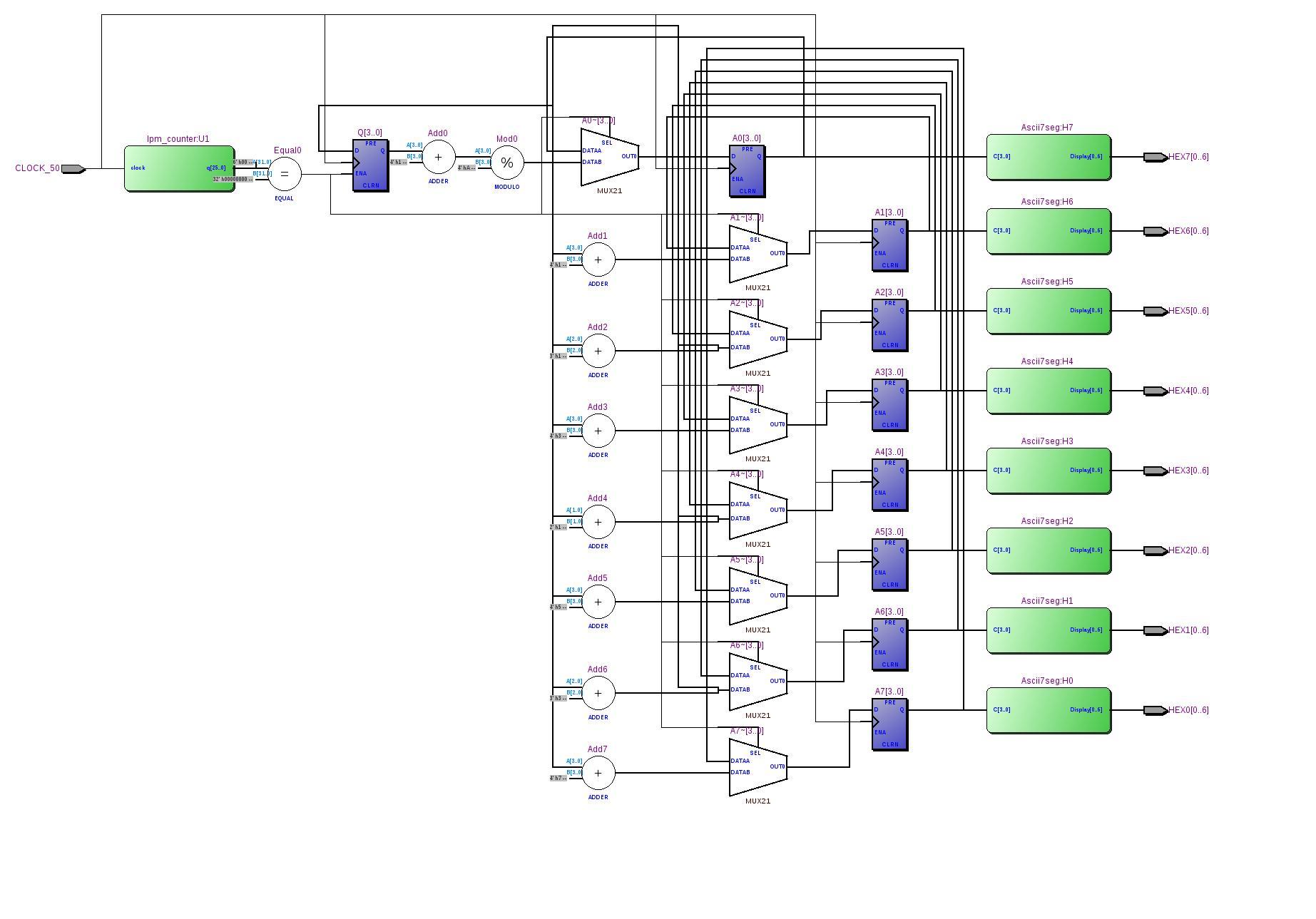


Figure 3. RTL View of Part 5

***Appendix F***

This appendix contains listings of the pin assignments used throughout the lab.

set\_location\_assignment PIN\_N25 -to SW[0]

set\_location\_assignment PIN\_N26 -to SW[1]

set\_location\_assignment PIN\_AF10 -to HEX0[0]

set\_location\_assignment PIN\_AB12 -to HEX0[1]

set\_location\_assignment PIN\_AC12 -to HEX0[2]

set\_location\_assignment PIN\_AD11 -to HEX0[3]

set\_location\_assignment PIN\_AE11 -to HEX0[4]

set\_location\_assignment PIN\_V14 -to HEX0[5]

set\_location\_assignment PIN\_V13 -to HEX0[6]

set\_location\_assignment PIN\_V20 -to HEX1[0]

set\_location\_assignment PIN\_V21 -to HEX1[1]

set\_location\_assignment PIN\_W21 -to HEX1[2]

set\_location\_assignment PIN\_Y22 -to HEX1[3]

set\_location\_assignment PIN\_AA24 -to HEX1[4]

set\_location\_assignment PIN\_AA23 -to HEX1[5]

set\_location\_assignment PIN\_AB24 -to HEX1[6]

set\_location\_assignment PIN\_AB23 -to HEX2[0]

set\_location\_assignment PIN\_V22 -to HEX2[1]

set\_location\_assignment PIN\_AC25 -to HEX2[2]

set\_location\_assignment PIN\_AC26 -to HEX2[3]

set\_location\_assignment PIN\_AB26 -to HEX2[4]

set\_location\_assignment PIN\_AB25 -to HEX2[5]

set\_location\_assignment PIN\_Y24 -to HEX2[6]

set\_location\_assignment PIN\_Y23 -to HEX3[0]

set\_location\_assignment PIN\_AA25 -to HEX3[1]

set\_location\_assignment PIN\_AA26 -to HEX3[2]

set\_location\_assignment PIN\_Y26 -to HEX3[3]

set\_location\_assignment PIN\_Y25 -to HEX3[4]

set\_location\_assignment PIN\_U22 -to HEX3[5]

set\_location\_assignment PIN\_W24 -to HEX3[6]

set\_location\_assignment PIN\_G26 -to KEY[0]

set\_instance\_assignment -name PARTITION\_HIERARCHY root\_partition -to | -section\_id Top

set\_global\_assignment -name FAMILY "Cyclone II"

set\_global\_assignment -name DEVICE EP2C35F672C6

set\_global\_assignment -name TOP\_LEVEL\_ENTITY Part1

set\_global\_assignment -name ORIGINAL\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name PROJECT\_CREATION\_TIME\_DATE "16:47:01 APRIL 29, 2014"

set\_global\_assignment -name LAST\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name VERILOG\_FILE TFFx.v

set\_global\_assignment -name VERILOG\_FILE testCountNG.v

set\_global\_assignment -name VERILOG\_FILE Hex7seg.v

set\_global\_assignment -name VERILOG\_FILE CountNG.v

set\_global\_assignment -name VERILOG\_FILE Part1.v

set\_global\_assignment -name PROJECT\_OUTPUT\_DIRECTORY output\_files

set\_global\_assignment -name MIN\_CORE\_JUNCTION\_TEMP 0

set\_global\_assignment -name MAX\_CORE\_JUNCTION\_TEMP 85

set\_global\_assignment -name DEVICE\_FILTER\_PIN\_COUNT 672

set\_global\_assignment -name ERROR\_CHECK\_FREQUENCY\_DIVISOR 1

set\_global\_assignment -name EDA\_SIMULATION\_TOOL "ModelSim-Altera (VHDL)"

set\_global\_assignment -name EDA\_OUTPUT\_DATA\_FORMAT VHDL -section\_id eda\_simulation

set\_global\_assignment -name PARTITION\_NETLIST\_TYPE SOURCE -section\_id Top

set\_global\_assignment -name PARTITION\_FITTER\_PRESERVATION\_LEVEL PLACEMENT\_AND\_ROUTING -section\_id Top

set\_global\_assignment -name PARTITION\_COLOR 16764057 -section\_id Top

set\_global\_assignment -name USE\_CONFIGURATION\_DEVICE ON

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS "AS INPUT TRI-STATED"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 2 -section\_id "3.3-V LVTTL"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI-X"

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS\_NO\_OUTPUT\_GND "AS INPUT TRI-STATED"

Figure 1. Listing of Part1\_Pins.csv

set\_location\_assignment PIN\_N25 -to SW[0]

set\_location\_assignment PIN\_N26 -to SW[1]

set\_location\_assignment PIN\_AF10 -to HEX0[0]

set\_location\_assignment PIN\_AB12 -to HEX0[1]

set\_location\_assignment PIN\_AC12 -to HEX0[2]

set\_location\_assignment PIN\_AD11 -to HEX0[3]

set\_location\_assignment PIN\_AE11 -to HEX0[4]

set\_location\_assignment PIN\_V14 -to HEX0[5]

set\_location\_assignment PIN\_V13 -to HEX0[6]

set\_location\_assignment PIN\_V20 -to HEX1[0]

set\_location\_assignment PIN\_V21 -to HEX1[1]

set\_location\_assignment PIN\_W21 -to HEX1[2]

set\_location\_assignment PIN\_Y22 -to HEX1[3]

set\_location\_assignment PIN\_AA24 -to HEX1[4]

set\_location\_assignment PIN\_AA23 -to HEX1[5]

set\_location\_assignment PIN\_AB24 -to HEX1[6]

set\_location\_assignment PIN\_AB23 -to HEX2[0]

set\_location\_assignment PIN\_V22 -to HEX2[1]

set\_location\_assignment PIN\_AC25 -to HEX2[2]

set\_location\_assignment PIN\_AC26 -to HEX2[3]

set\_location\_assignment PIN\_AB26 -to HEX2[4]

set\_location\_assignment PIN\_AB25 -to HEX2[5]

set\_location\_assignment PIN\_Y24 -to HEX2[6]

set\_location\_assignment PIN\_Y23 -to HEX3[0]

set\_location\_assignment PIN\_AA25 -to HEX3[1]

set\_location\_assignment PIN\_AA26 -to HEX3[2]

set\_location\_assignment PIN\_Y26 -to HEX3[3]

set\_location\_assignment PIN\_Y25 -to HEX3[4]

set\_location\_assignment PIN\_U22 -to HEX3[5]

set\_location\_assignment PIN\_W24 -to HEX3[6]

set\_location\_assignment PIN\_G26 -to KEY[0]

set\_instance\_assignment -name PARTITION\_HIERARCHY root\_partition -to | -section\_id Top

set\_global\_assignment -name FAMILY "Cyclone II"

set\_global\_assignment -name DEVICE EP2C35F672C6

set\_global\_assignment -name TOP\_LEVEL\_ENTITY Part2

set\_global\_assignment -name ORIGINAL\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name PROJECT\_CREATION\_TIME\_DATE "20:16:29 MAY 05, 2014"

set\_global\_assignment -name LAST\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name VERILOG\_FILE testCountNG.v

set\_global\_assignment -name VERILOG\_FILE Part2.v

set\_global\_assignment -name VERILOG\_FILE Hex7seg.v

set\_global\_assignment -name VERILOG\_FILE CountNG.v

set\_global\_assignment -name PROJECT\_OUTPUT\_DIRECTORY output\_files

set\_global\_assignment -name MIN\_CORE\_JUNCTION\_TEMP 0

set\_global\_assignment -name MAX\_CORE\_JUNCTION\_TEMP 85

set\_global\_assignment -name DEVICE\_FILTER\_PIN\_COUNT 672

set\_global\_assignment -name ERROR\_CHECK\_FREQUENCY\_DIVISOR 1

set\_global\_assignment -name EDA\_SIMULATION\_TOOL "ModelSim-Altera (VHDL)"

set\_global\_assignment -name EDA\_OUTPUT\_DATA\_FORMAT VHDL -section\_id eda\_simulation

set\_global\_assignment -name USE\_CONFIGURATION\_DEVICE ON

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS "AS INPUT TRI-STATED"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 2 -section\_id "3.3-V LVTTL"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI-X"

set\_global\_assignment -name PARTITION\_NETLIST\_TYPE SOURCE -section\_id Top

set\_global\_assignment -name PARTITION\_FITTER\_PRESERVATION\_LEVEL PLACEMENT\_AND\_ROUTING -section\_id Top

set\_global\_assignment -name PARTITION\_COLOR 16764057 -section\_id Top

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS\_NO\_OUTPUT\_GND "AS INPUT TRI-STATED"

Figure 2. Listing of Part2\_Pins.csv

set\_location\_assignment PIN\_N25 -to SW[0]

set\_location\_assignment PIN\_N26 -to SW[1]

set\_location\_assignment PIN\_AF10 -to HEX0[0]

set\_location\_assignment PIN\_AB12 -to HEX0[1]

set\_location\_assignment PIN\_AC12 -to HEX0[2]

set\_location\_assignment PIN\_AD11 -to HEX0[3]

set\_location\_assignment PIN\_AE11 -to HEX0[4]

set\_location\_assignment PIN\_V14 -to HEX0[5]

set\_location\_assignment PIN\_V13 -to HEX0[6]

set\_location\_assignment PIN\_V20 -to HEX1[0]

set\_location\_assignment PIN\_V21 -to HEX1[1]

set\_location\_assignment PIN\_W21 -to HEX1[2]

set\_location\_assignment PIN\_Y22 -to HEX1[3]

set\_location\_assignment PIN\_AA24 -to HEX1[4]

set\_location\_assignment PIN\_AA23 -to HEX1[5]

set\_location\_assignment PIN\_AB24 -to HEX1[6]

set\_location\_assignment PIN\_AB23 -to HEX2[0]

set\_location\_assignment PIN\_V22 -to HEX2[1]

set\_location\_assignment PIN\_AC25 -to HEX2[2]

set\_location\_assignment PIN\_AC26 -to HEX2[3]

set\_location\_assignment PIN\_AB26 -to HEX2[4]

set\_location\_assignment PIN\_AB25 -to HEX2[5]

set\_location\_assignment PIN\_Y24 -to HEX2[6]

set\_location\_assignment PIN\_Y23 -to HEX3[0]

set\_location\_assignment PIN\_AA25 -to HEX3[1]

set\_location\_assignment PIN\_AA26 -to HEX3[2]

set\_location\_assignment PIN\_Y26 -to HEX3[3]

set\_location\_assignment PIN\_Y25 -to HEX3[4]

set\_location\_assignment PIN\_U22 -to HEX3[5]

set\_location\_assignment PIN\_W24 -to HEX3[6]

set\_location\_assignment PIN\_G26 -to KEY[0]

set\_instance\_assignment -name PARTITION\_HIERARCHY root\_partition -to | -section\_id Top

set\_global\_assignment -name FAMILY "Cyclone II"

set\_global\_assignment -name DEVICE EP2C35F672C6

set\_global\_assignment -name TOP\_LEVEL\_ENTITY Part3

set\_global\_assignment -name ORIGINAL\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name PROJECT\_CREATION\_TIME\_DATE "19:41:06 MAY 05, 2014"

set\_global\_assignment -name LAST\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name VERILOG\_FILE Part3.v

set\_global\_assignment -name VERILOG\_FILE lpm\_counter0.v

set\_global\_assignment -name VERILOG\_FILE Hex7seg.v

set\_global\_assignment -name BDF\_FILE counter.bdf

set\_global\_assignment -name PROJECT\_OUTPUT\_DIRECTORY output\_files

set\_global\_assignment -name DEVICE\_FILTER\_PIN\_COUNT 672

set\_global\_assignment -name EDA\_SIMULATION\_TOOL "ModelSim-Altera (VHDL)"

set\_global\_assignment -name EDA\_OUTPUT\_DATA\_FORMAT VHDL -section\_id eda\_simulation

set\_global\_assignment -name USE\_CONFIGURATION\_DEVICE ON

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS "AS INPUT TRI-STATED"

set\_global\_assignment -name PARTITION\_NETLIST\_TYPE SOURCE -section\_id Top

set\_global\_assignment -name PARTITION\_FITTER\_PRESERVATION\_LEVEL PLACEMENT\_AND\_ROUTING -section\_id Top

set\_global\_assignment -name PARTITION\_COLOR 16764057 -section\_id Top

set\_global\_assignment -name MIN\_CORE\_JUNCTION\_TEMP 0

set\_global\_assignment -name MAX\_CORE\_JUNCTION\_TEMP 85

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 2 -section\_id "3.3-V LVTTL"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI-X"

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS\_NO\_OUTPUT\_GND "AS INPUT TRI-STATED"

Figure 3. Listing of Part3\_Pins.csv

set\_location\_assignment PIN\_AF10 -to HEX0[0]

set\_location\_assignment PIN\_AB12 -to HEX0[1]

set\_location\_assignment PIN\_AC12 -to HEX0[2]

set\_location\_assignment PIN\_AD11 -to HEX0[3]

set\_location\_assignment PIN\_AE11 -to HEX0[4]

set\_location\_assignment PIN\_V14 -to HEX0[5]

set\_location\_assignment PIN\_V13 -to HEX0[6]

set\_location\_assignment PIN\_N2 -to CLOCK\_50

set\_instance\_assignment -name PARTITION\_HIERARCHY root\_partition -to | -section\_id Top

set\_global\_assignment -name FAMILY "Cyclone II"

set\_global\_assignment -name DEVICE EP2C35F672C6

set\_global\_assignment -name TOP\_LEVEL\_ENTITY Part4

set\_global\_assignment -name ORIGINAL\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name PROJECT\_CREATION\_TIME\_DATE "09:21:00 MAY 05, 2014"

set\_global\_assignment -name LAST\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name PROJECT\_OUTPUT\_DIRECTORY output\_files

set\_global\_assignment -name MIN\_CORE\_JUNCTION\_TEMP 0

set\_global\_assignment -name MAX\_CORE\_JUNCTION\_TEMP 85

set\_global\_assignment -name DEVICE\_FILTER\_PIN\_COUNT 672

set\_global\_assignment -name ERROR\_CHECK\_FREQUENCY\_DIVISOR 1

set\_global\_assignment -name EDA\_SIMULATION\_TOOL "ModelSim-Altera (VHDL)"

set\_global\_assignment -name EDA\_OUTPUT\_DATA\_FORMAT VHDL -section\_id eda\_simulation

set\_global\_assignment -name VERILOG\_FILE Part4.v

set\_global\_assignment -name VERILOG\_FILE Hex7seg.v

set\_global\_assignment -name QIP\_FILE lpm\_counter0.qip

set\_global\_assignment -name IP\_TOOL\_NAME LPM\_COUNTER -qip lpm\_counter0.qip

set\_global\_assignment -name IP\_TOOL\_VERSION 13.0 -qip lpm\_counter0.qip

set\_global\_assignment -name VERILOG\_FILE lpm\_counter0.v -qip lpm\_counter0.qip

set\_global\_assignment -name MISC\_FILE lpm\_counter0.bsf -qip lpm\_counter0.qip

set\_global\_assignment -name MISC\_FILE lpm\_counter0\_inst.v -qip lpm\_counter0.qip

set\_global\_assignment -name MISC\_FILE lpm\_counter0\_bb.v -qip lpm\_counter0.qip

set\_global\_assignment -name MISC\_FILE lpm\_counter0.inc -qip lpm\_counter0.qip

set\_global\_assignment -name MISC\_FILE lpm\_counter0.cmp -qip lpm\_counter0.qip

set\_global\_assignment -name USE\_CONFIGURATION\_DEVICE ON

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS "AS INPUT TRI-STATED"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 2 -section\_id "3.3-V LVTTL"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI-X"

set\_global\_assignment -name BDF\_FILE counter.bdf

set\_global\_assignment -name PARTITION\_NETLIST\_TYPE SOURCE -section\_id Top

set\_global\_assignment -name PARTITION\_FITTER\_PRESERVATION\_LEVEL PLACEMENT\_AND\_ROUTING -section\_id Top

set\_global\_assignment -name PARTITION\_COLOR 16764057 -section\_id Top

set\_global\_assignment -name SMART\_RECOMPILE ON

set\_global\_assignment -name ENABLE\_DRC\_SETTINGS ON

set\_global\_assignment -name CYCLONEII\_OPTIMIZATION\_TECHNIQUE SPEED

set\_global\_assignment -name PHYSICAL\_SYNTHESIS\_COMBO\_LOGIC ON

set\_global\_assignment -name PHYSICAL\_SYNTHESIS\_REGISTER\_DUPLICATION ON

set\_global\_assignment -name PHYSICAL\_SYNTHESIS\_REGISTER\_RETIMING ON

set\_global\_assignment -name ROUTER\_TIMING\_OPTIMIZATION\_LEVEL MAXIMUM

set\_global\_assignment -name AUTO\_PACKED\_REGISTERS\_STRATIXII NORMAL

set\_global\_assignment -name FITTER\_EFFORT "STANDARD FIT"

set\_global\_assignment -name SYNTH\_TIMING\_DRIVEN\_SYNTHESIS ON

set\_global\_assignment -name ADV\_NETLIST\_OPT\_SYNTH\_WYSIWYG\_REMAP ON

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS\_NO\_OUTPUT\_GND "AS INPUT TRI-STATED"

set\_global\_assignment -name MUX\_RESTRUCTURE OFF

set\_global\_assignment -name OPTIMIZE\_HOLD\_TIMING "IO PATHS AND MINIMUM TPD PATHS"

set\_global\_assignment -name OPTIMIZE\_MULTI\_CORNER\_TIMING ON

set\_global\_assignment -name POWER\_PRESET\_COOLING\_SOLUTION "23 MM HEAT SINK WITH 200 LFPM AIRFLOW"

set\_global\_assignment -name POWER\_BOARD\_THERMAL\_MODEL "NONE (CONSERVATIVE)"

Figure 4. Listing of Part4\_Pins.csv

set\_location\_assignment PIN\_AF10 -to HEX0[0]

set\_location\_assignment PIN\_AB12 -to HEX0[1]

set\_location\_assignment PIN\_AC12 -to HEX0[2]

set\_location\_assignment PIN\_AD11 -to HEX0[3]

set\_location\_assignment PIN\_AE11 -to HEX0[4]

set\_location\_assignment PIN\_V14 -to HEX0[5]

set\_location\_assignment PIN\_V13 -to HEX0[6]

set\_location\_assignment PIN\_V20 -to HEX1[0]

set\_location\_assignment PIN\_V21 -to HEX1[1]

set\_location\_assignment PIN\_W21 -to HEX1[2]

set\_location\_assignment PIN\_Y22 -to HEX1[3]

set\_location\_assignment PIN\_AA24 -to HEX1[4]

set\_location\_assignment PIN\_AA23 -to HEX1[5]

set\_location\_assignment PIN\_AB24 -to HEX1[6]

set\_location\_assignment PIN\_AB23 -to HEX2[0]

set\_location\_assignment PIN\_V22 -to HEX2[1]

set\_location\_assignment PIN\_AC25 -to HEX2[2]

set\_location\_assignment PIN\_AC26 -to HEX2[3]

set\_location\_assignment PIN\_AB26 -to HEX2[4]

set\_location\_assignment PIN\_AB25 -to HEX2[5]

set\_location\_assignment PIN\_Y24 -to HEX2[6]

set\_location\_assignment PIN\_Y23 -to HEX3[0]

set\_location\_assignment PIN\_AA25 -to HEX3[1]

set\_location\_assignment PIN\_AA26 -to HEX3[2]

set\_location\_assignment PIN\_Y26 -to HEX3[3]

set\_location\_assignment PIN\_Y25 -to HEX3[4]

set\_location\_assignment PIN\_U22 -to HEX3[5]

set\_location\_assignment PIN\_W24 -to HEX3[6]

set\_location\_assignment PIN\_U9 -to HEX4[0]

set\_location\_assignment PIN\_U1 -to HEX4[1]

set\_location\_assignment PIN\_U2 -to HEX4[2]

set\_location\_assignment PIN\_T4 -to HEX4[3]

set\_location\_assignment PIN\_R7 -to HEX4[4]

set\_location\_assignment PIN\_R6 -to HEX4[5]

set\_location\_assignment PIN\_T3 -to HEX4[6]

set\_location\_assignment PIN\_T2 -to HEX5[0]

set\_location\_assignment PIN\_P6 -to HEX5[1]

set\_location\_assignment PIN\_P7 -to HEX5[2]

set\_location\_assignment PIN\_T9 -to HEX5[3]

set\_location\_assignment PIN\_R5 -to HEX5[4]

set\_location\_assignment PIN\_R4 -to HEX5[5]

set\_location\_assignment PIN\_R3 -to HEX5[6]

set\_location\_assignment PIN\_R2 -to HEX6[0]

set\_location\_assignment PIN\_P4 -to HEX6[1]

set\_location\_assignment PIN\_P3 -to HEX6[2]

set\_location\_assignment PIN\_M2 -to HEX6[3]

set\_location\_assignment PIN\_M3 -to HEX6[4]

set\_location\_assignment PIN\_M5 -to HEX6[5]

set\_location\_assignment PIN\_M4 -to HEX6[6]

set\_location\_assignment PIN\_L3 -to HEX7[0]

set\_location\_assignment PIN\_L2 -to HEX7[1]

set\_location\_assignment PIN\_L9 -to HEX7[2]

set\_location\_assignment PIN\_L6 -to HEX7[3]

set\_location\_assignment PIN\_L7 -to HEX7[4]

set\_location\_assignment PIN\_P9 -to HEX7[5]

set\_location\_assignment PIN\_N9 -to HEX7[6]

set\_location\_assignment PIN\_N2 -to CLOCK\_50

set\_instance\_assignment -name PARTITION\_HIERARCHY root\_partition -to | -section\_id Top

set\_global\_assignment -name FAMILY "Cyclone II"

set\_global\_assignment -name DEVICE EP2C35F672C6

set\_global\_assignment -name TOP\_LEVEL\_ENTITY Part5

set\_global\_assignment -name ORIGINAL\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name PROJECT\_CREATION\_TIME\_DATE "15:57:21 MAY 05, 2014"

set\_global\_assignment -name LAST\_QUARTUS\_VERSION "13.0 SP1"

set\_global\_assignment -name PROJECT\_OUTPUT\_DIRECTORY output\_files

set\_global\_assignment -name MIN\_CORE\_JUNCTION\_TEMP 0

set\_global\_assignment -name MAX\_CORE\_JUNCTION\_TEMP 85

set\_global\_assignment -name DEVICE\_FILTER\_PIN\_COUNT 672

set\_global\_assignment -name ERROR\_CHECK\_FREQUENCY\_DIVISOR 1

set\_global\_assignment -name EDA\_SIMULATION\_TOOL "ModelSim-Altera (VHDL)"

set\_global\_assignment -name EDA\_OUTPUT\_DATA\_FORMAT VHDL -section\_id eda\_simulation

set\_global\_assignment -name VERILOG\_FILE Part5.v

set\_global\_assignment -name VERILOG\_FILE lpm\_counter0.v

set\_global\_assignment -name PARTITION\_NETLIST\_TYPE SOURCE -section\_id Top

set\_global\_assignment -name PARTITION\_FITTER\_PRESERVATION\_LEVEL PLACEMENT\_AND\_ROUTING -section\_id Top

set\_global\_assignment -name PARTITION\_COLOR 16764057 -section\_id Top

set\_global\_assignment -name USE\_CONFIGURATION\_DEVICE ON

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS "AS INPUT TRI-STATED"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 2 -section\_id "3.3-V LVTTL"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI"

set\_global\_assignment -name OUTPUT\_PIN\_LOAD 10 -section\_id "3.3-V PCI-X"

set\_global\_assignment -name STRATIX\_DEVICE\_IO\_STANDARD "3.3-V LVTTL"

set\_global\_assignment -name VERILOG\_FILE Ascii7seg.v

set\_global\_assignment -name CDF\_FILE output\_files/Chain3.cdf

set\_global\_assignment -name RESERVE\_ALL\_UNUSED\_PINS\_NO\_OUTPUT\_GND "AS INPUT TRI-STATED"

Figure 5. Listing of Part5\_Pins.csv