Lab 7 Submission

**Signed Binary 5-Bit Adder/Subtractor with Validity Detection**

CPE 133 - 03

Michael Hegglin

Jonathan Skelly

**Executive Summary:**

We designed an 5-bit adder/subtractor with two 5-bit signed binary numbers. The design decided whether to add or subtract these numbers via a button press. We implemented this circuit through Verilog, which could then be downloaded to the board.



BBD of our 5-bit adder/subtractor.



Next-level down of our 5-bit adder/subtractor.

**Questions:**

**1. What is a really good reason that I don’t put in the effort to design a subtractor in addition to my RCA?**

* The reason is that an RCA can add a positive and a negative number to subtract instead of designing a whole new circuit.

**2. The univ\_sseg.v module you used in this lab activity seemed to do a great job of simultaneously displaying more than one number at a time on the development board’s seven-segment displays. But you were previously told that the displays could only show no more than one number at a time. Briefly explain what is going on.**

* There is a time delay that is allowing one number to stay on the display in a certain spot while the other changes. This delay is too quick to be detected by the human eye.

**3. Briefly describe the steps necessary in order to extend this design to 8-bit signed binary numbers (once again, assume your development board and provided modules are not limiting factors).**

* I would keep the design the same. I would simply assign more switches and higher bit inputs for the 8-bits.

**4. The output of this circuit arbitrarily showed dashes when the number was not valid. Is there an output that would have been more appropriate in this activity? Briefly state your thoughts on this matter.**

* I think dashes worked well in a sense that you could see that the validity of the test was false. This makes it very clear that the answer is not valid. Were our sum/subtraction result to be used elsewhere in a circuit, outputting a “0” when invalid would have been more appropriate so that a valid number was always produced (dashes arent data), but for the purpose of simply displaying information, dashes worked fine.

**5. Using the development board under the conditions stated in this lab activity, would it have been possible to design an adder/subtractor unit based on 9-bit signed binary numbers (RC format)? The problem here is that you run out of switches to support 18-bit number. Consider all possibilities and fully explain your answer.**

* As stated above, you would run out of switches for the inputs. Due to this reason it is not possible to use our board to implement this design on 9-bit signed binary numbers. Another problem would be the display. It cannot represent numbers of less than -255.

**6. Quite often in digital design, there are boundary condition issues you need to deal with. What this means to me is that 98% of the errors I make in a design are with a boundary condition. This lab activity also has a boundary condition that essentially renders the result invalid even though it passes our simple validity checker. Briefly describe this boundary condition. HINT: the notion that it is a boundary condition roughly means that is has something to do with the far end of the given number range for the RC numbers.**

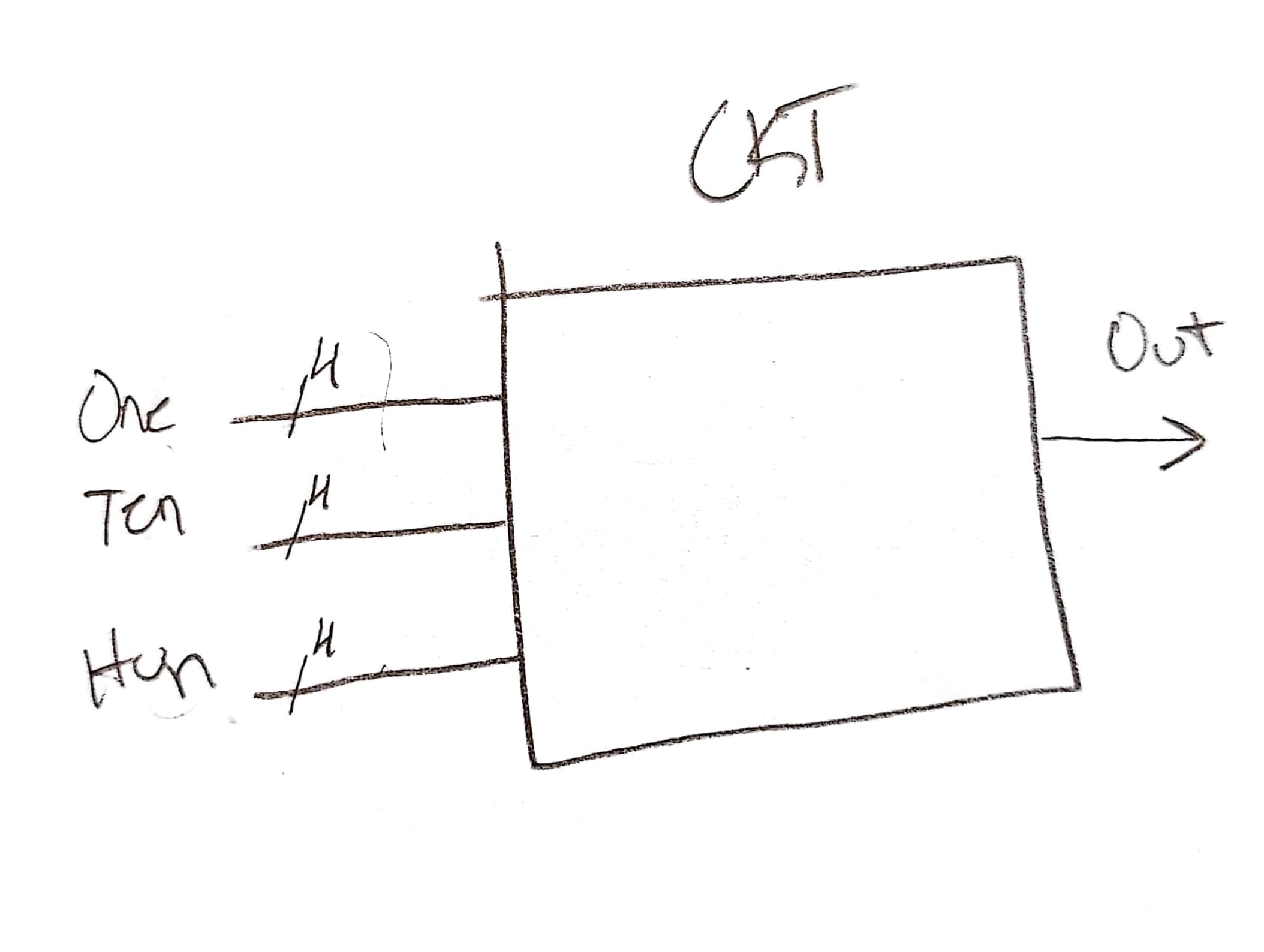
* Yes, the boundary value that displayed the wrong output was -16 (4’b 10000). The function 0 - (-16) showed a sum of -16, when it should have been +16. This is because the 2s complement of 10000 is still 10000.

**7. Describe the modifications you would need to make to the circuit in this lab activity is you needed a 2\*A or A-B circuit instead of the A+B or A-B. Do not use a shift register in your solution.**

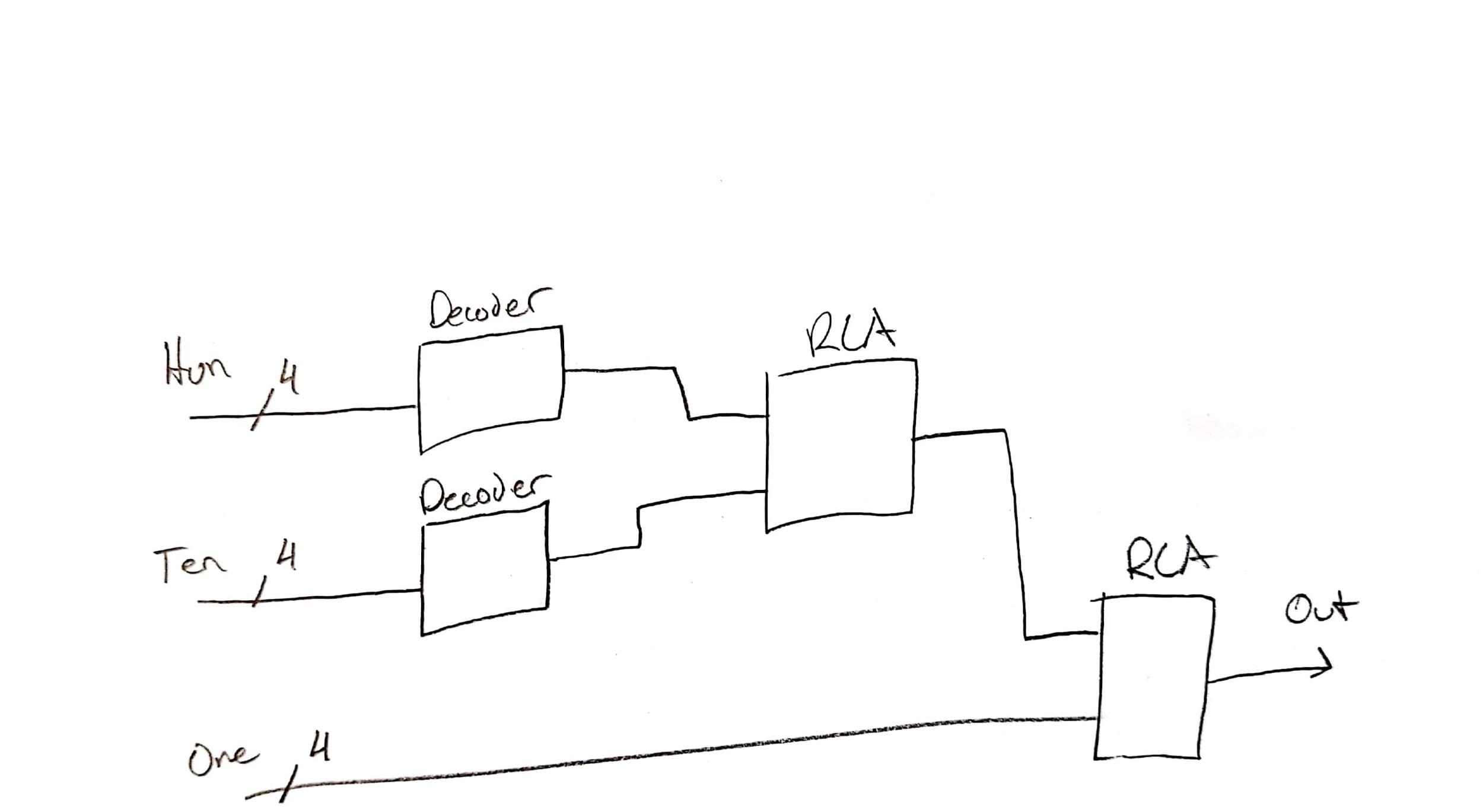
* You would have to input A into both inputs for the RCA. This is due to the fact that 2 \* A = A + A.

**Design Problems:**

**1. Design a circuit that converts a three-digit decimal number to an 8-bit unsigned binary number. This circuit has three BCD inputs, which means four bits for the 100’s, 10’s, and 1’s digit. The 8- bit output will always be sufficient to encode the three digital input value. State how the circuit is controlled.**



BBD of 3-digit BCD-to-Binary decoder.

Next-Level-down of our 3-digit BCD-to-Binary converter. This circuit features no controls, always giving a function of the input.

**Source Code:**

**`timescale 1ns / 1ps**

**//////////////////////////////////////////////////////////////////////////////////**

**// Company: DogsWithJobs**

**// Engineer: Skelly, Heddlin**

**//**

**// Create Date: 10/17/2018 12:04:42 PM**

**// Design Name: AddSub**

**// Module Name: MainLogic**

**// Project Name: 5-Bit Adder/Subtractor**

**// Target Devices: Digilent Board**

**// Tool Versions:**

**// Description: Add two 5-bit RC numbers, or subtract two 5-bit RC**

**// numbers when a button is pressed.**

**//**

**// Dependencies:**

**//**

**// Revision:**

**// Revision 0.01 - File Created**

**// Additional Comments:**

**//**

**//////////////////////////////////////////////////////////////////////////////////**

**module MainLogic(a, b, clk, seg, an, button);**

**input [4:0] a, b;**

**input clk, button;**

**output [7:0] seg;**

**output [3:0] an;**

**wire [4:0] sum;**

**wire co;**

**reg [4:0] a\_mag, b\_mag;**

**reg valid;**

**always @ (button, a, b)**

**begin**

**if (button == 1)**

**begin**

**a\_mag = a;**

**b\_mag = -b;**

**//do a - b here**

**end**

**else**

**begin**

**//do a + b here**

**a\_mag = a;**

**b\_mag = b;**

**end**

**end**

**rca\_nb #(.n(5)) rca1(**

**.a (a\_mag),**

**.b (b\_mag),**

**.cin (0),**

**.sum (sum),**

**.co (co)**

**);**

**always @ (a\_mag, b\_mag, sum, valid)**

**begin**

**if((b\_mag[4] == a\_mag[4]) & (a\_mag[4] != sum[4]))**

**begin**

**valid = 0;**

**end**

**else**

**begin**

**valid = 1;**

**end**

**end**

**univ\_sseg u1 (**

**.cnt1 (sum),**

**.cnt2 (0),**

**.valid (valid),**

**.dp\_en (0),**

**.dp\_sel (0),**

**.mod\_sel (0),**

**.sign (sum[4]),**

**.clk (clk),**

**.ssegs (seg),**

**.disp\_en (an)**

**);**

**endmodule**

**Test Bench:**

**`timescale 1ns / 1ps**

**//////////////////////////////////////////////////////////////////////////////////**

**// Company:**

**// Engineer:**

**//**

**// Create Date: 10/15/2018 12:43:46 PM**

**// Design Name: Test Bench**

**// Module Name: testbench**

**// Project Name:**

**// Target Devices:**

**// Tool Versions:**

**// Description:**

**//**

**// Dependencies:**

**//**

**// Revision:**

**// Revision 0.01 - File Created**

**// Additional Comments:**

**//**

**//////////////////////////////////////////////////////////////////////////////////**

**module adder( );**

**reg [4:0] a, b; //- stimulus outputs**

**wire [7:0] seg;**

**wire [3:0] an;**

**wire eq, lt, gt;**

**reg clk, button;**

**MainLogic AddSub (**

**.a (a),**

**.b (b),**

**.clk (clk),**

**.seg (seg),**

**.an (an),**

**.button (button)**

**);**

**initial**

**begin**

**clk = 0; //- init signalinitial**

**forever #10 clk = ~clk;**

**end**

**initial**

**begin**

**//- initial values of a & b**

**a = 5'b00000;**

**b = 5'b00000;**

**button = 0;**

**clk =0;**

**//- a & b values 20 time units later**

**#20**

**a = 5'b00001;**

**b = 5'b00001;**

**button = 1;**

**//- a & b values 20 time units later**

**#20**

**a = 5'b00001;**

**b = 5'b11110;**

**button = 0;**

**//- a & b values 20 time units later**

**#20**

**a = 5'b00100;**

**b = 5'b11100;**

**button = 1;**

**#20**

**a = 8'b11111;**

**b = 8'b11111;**

**button = 0;**

**#20**

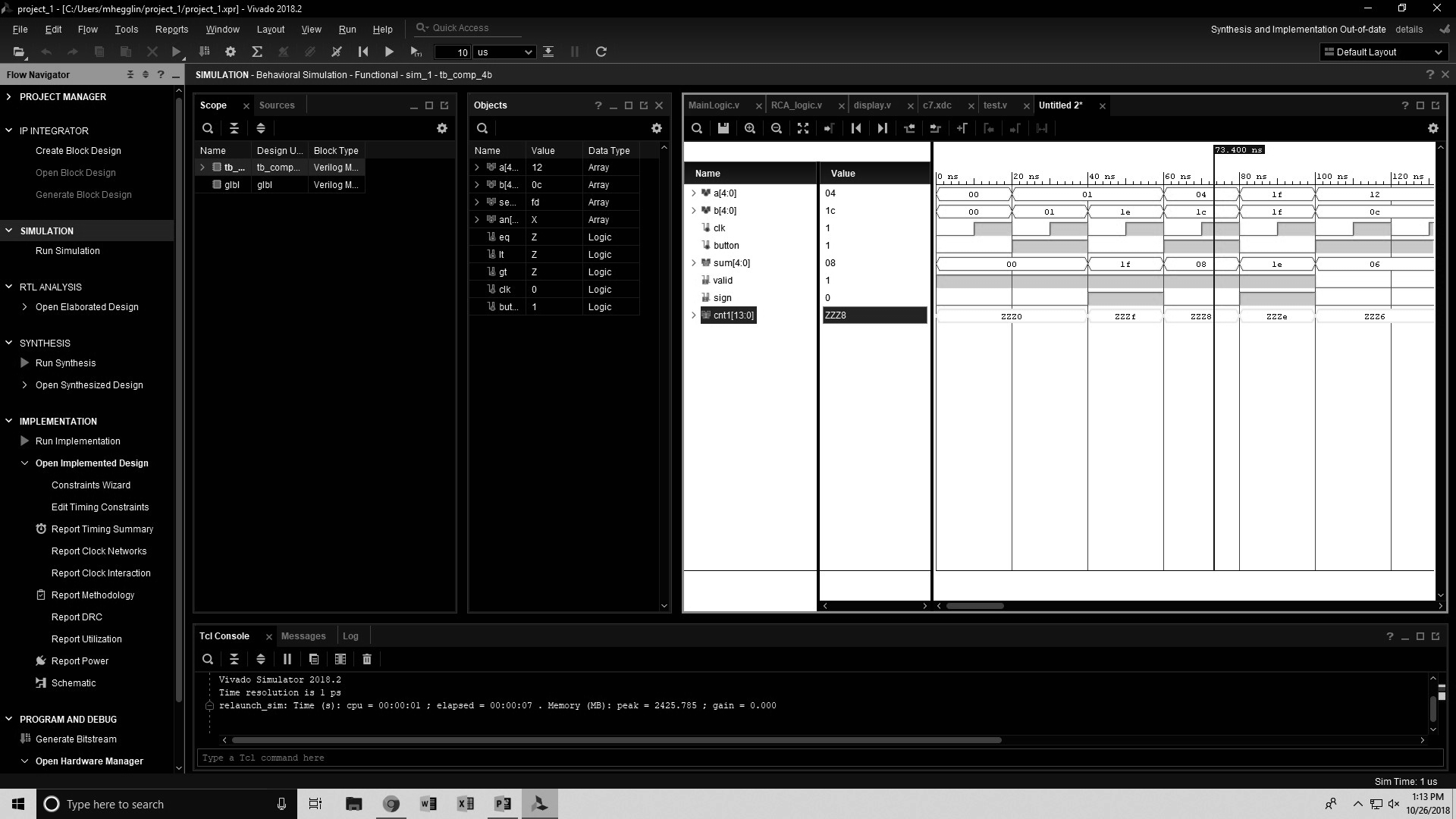
**a = 8'b10010;**

**b = 8'b01100;**

**button = 1;**

**end**

**endmodule**

****