Lab 6 Submission

**Five-Bit Magnitude Comparator**

CPE 133 - 03

Michael Hegglin

Jonathan Skelly

**Executive Summary:**

We designed an 5-bit comparator with two 5-bit signed binary numbers. The design decided if the numbers were equal, and if so displayed their values. We implemented this circuit through Verilog, which could then be downloaded to the board.

****

**Simple Black Box Diagram**

****

**Lower Level Black Box Diagram**

**Questions:**

**1. Show a closed form formula that relates the data width the number range for an unsigned binary number. Make sure you use an accepted format for specifying number ranges.**

n = bit width

range: 0 **to** 2^(n) -1

**2. Show a closed form formula that relates the data width the number range for a signed binary number in RC format. Make sure you use an accepted format for specifying number ranges.**

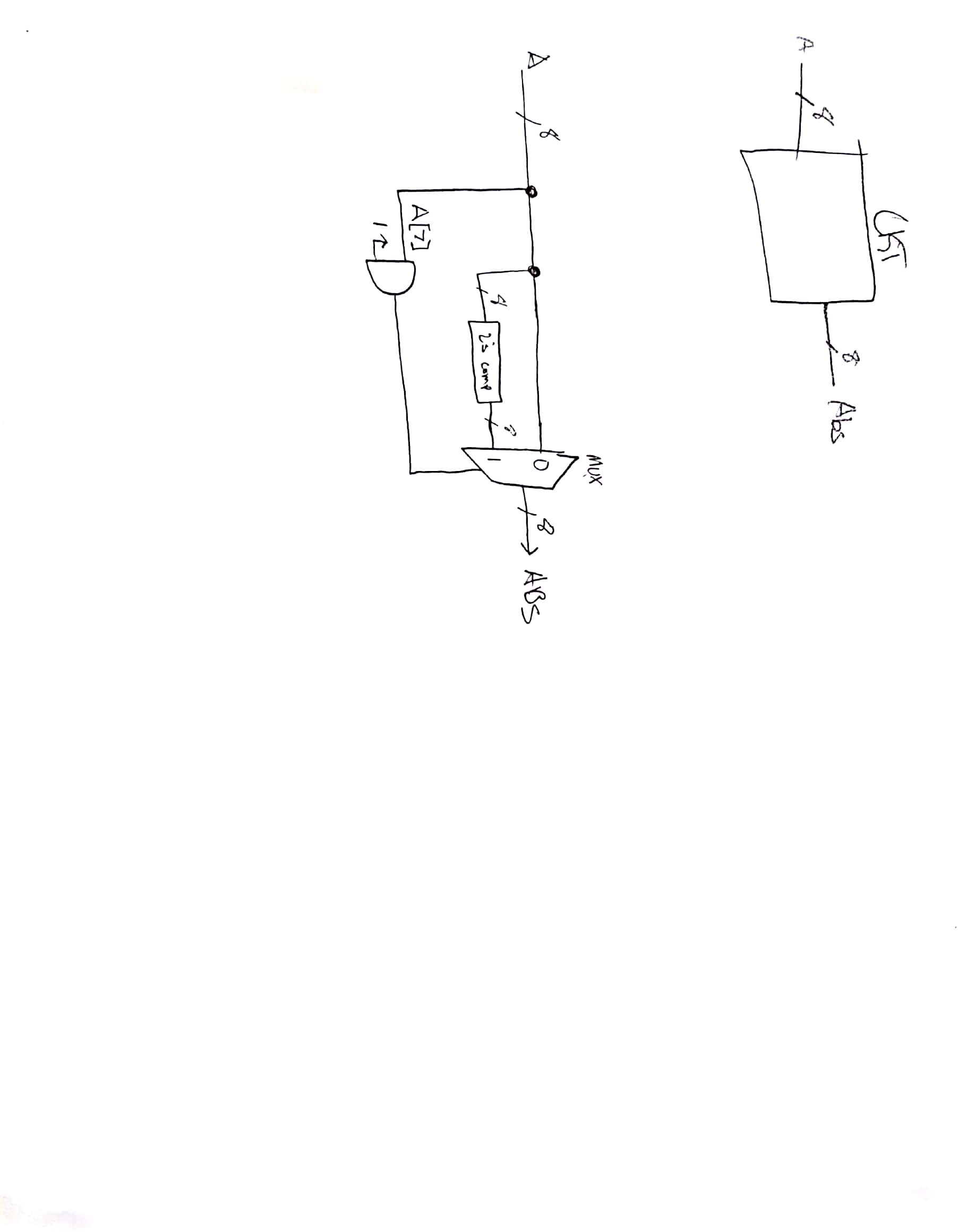
n = bit width

range: 0 - (2^(n-1) -1)  **to** 2^(n-1) -1

**3. For a given bit-width, does the number of unique numbers in an unsigned binary and signed binary number in RC format differ? Briefly explain.**

No, because you can have a certain number of bits for a number regardless if it is signed or unsigned.

**4. Show a schematic diagram of an 8-bit absolute value circuit that works with signed binary numbers in RC format. This circuit has an 8-bit input and an 8-bit output.**

**8-bit RC Absolute Value circuit BBD and Next-level down**

**5. Can an 8-bit binary number in unsigned format ever be an odd number but have even parity? Briefly explain.**

Yes, for example the number 3(0011). It has an even parity, but is an odd number.

**6. Can signed binary numbers in RC format have the notion of parity associated with them? Briefly explain your answer.**

Yes, parity is impartial to the bit signature. It looks at the bits the same way regardless of being signed or unsigned.

**7. In computer programming, briefly describe why it is the best idea to use an unsigned integer type when it is known that the value will never be negative.**

It is better to use an unsigned number because it is easier to do calculations.

**8. In computer programming, briefly describe what happens when a mathematical operation exceeds the ranges for the data types associated with that operation? Does the programmer typically know the range has been exceeded?**

The programmer gets an error. The type must be changed in order for the function to work.

**8. Often we refer to a design as a “flat” design. In terms of HDL modeling, briefly but fully describe what that term refers to.**

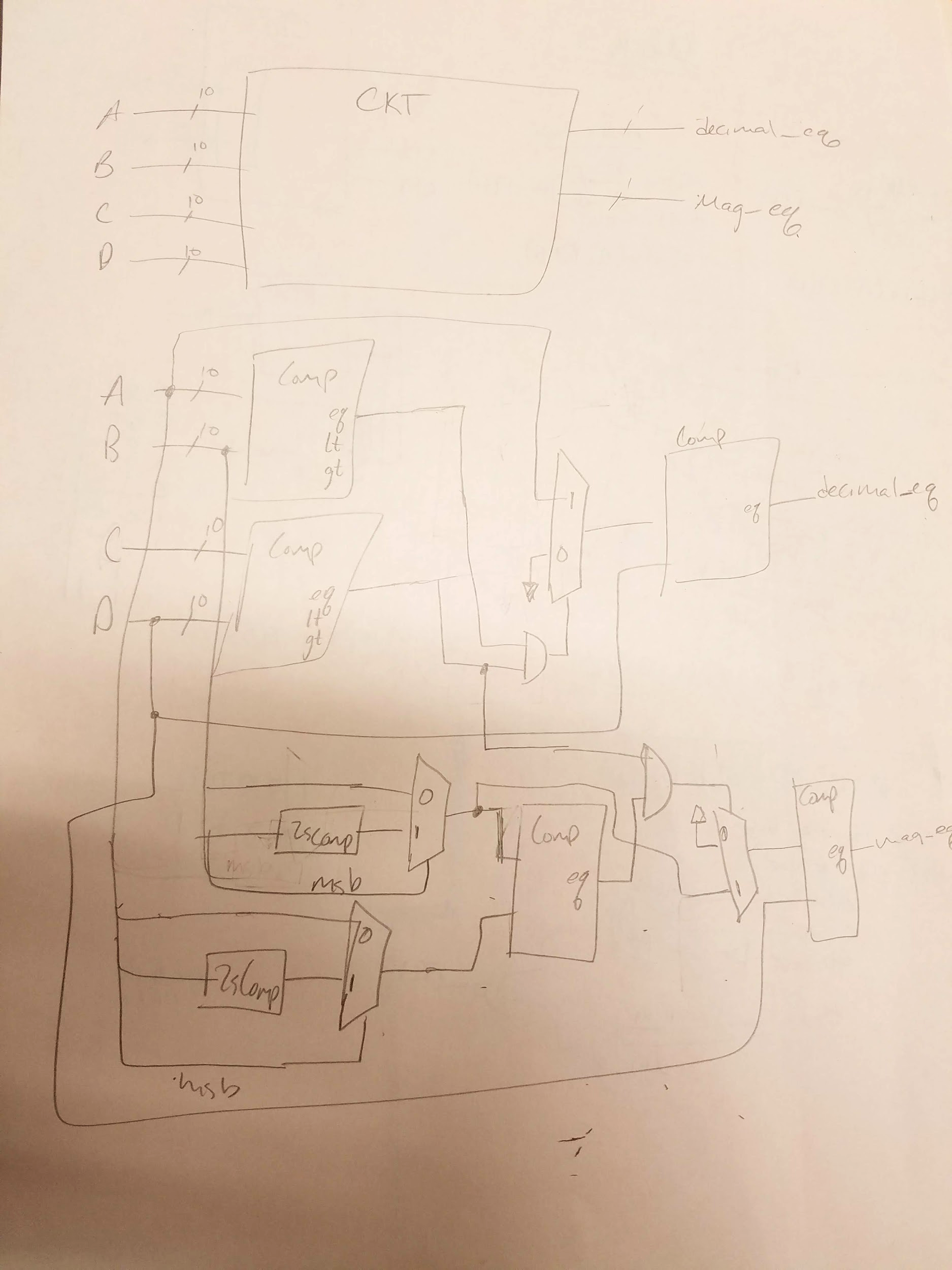
It means it does not use structural modeling. For example trying to do something in one line of code.

**9. In general, does the number of levels of a particular design affect the resources required to implement that design using an HDL and implementing that circuit on an FPGA? Briefly but fully explain.**

Yes, more levels means more resources are used. For example, if your using every electrical component on the FPGA more resources will be required then just using the switches.

**Design Problem:**

**1. Design a circuit that four 10-bit inputs; two of the inputs are in RC format while the other two inputs are in unsigned binary format. The circuit has two outputs; one output indicates when all four inputs have equivalent decimal equivalents. The other output indicates when all four values have equivalent magnitudes. Use foundation modules when possible; minimize your use of hardware. State how the circuit is controlled.**

****

**Two 10-bit unsigned and two 10-bit signed magnitude and decimal equivalent comparator circuit BBD and Next-level down.**

**MainCode:**

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company: DogsWithJobs

// Engineer: Hegglin/Skelly

//

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

// Design Name: LED Mag

// Module Name: Lab\_6\_source

// Project Name: 5\_bit Magnitude Comparator

// Target Devices: Digilent Board

// Tool Versions: Verilog

// Description: Output magnitudes of equivalent binary numbers to LED // board

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module MainLogic(a, b, clk, segs, anodes);

input [4:0] a, b;

input clk;

output [7:0] segs;

output [3:0] anodes;

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

wire eq, lt, gt, temp, final\_a;

// assign twos\_comp = ~a\_i + 1'b1;

// assign also\_twos\_comp = -a\_i;

always @ (a, b)

begin

if(a[4] == 1)

begin

a\_mag = -a;

end

else

begin

a\_mag = a;

end

if (b[4] == 1)

begin

b\_mag = -b;

end

else

begin

b\_mag = b;

end

end

comp\_nb #(.n(5)) comp1 (

.a (a\_mag),

.b (b\_mag),

.eq (eq),

.lt (lt),

.gt (gt)

);

assign temp = (a[4] & b[4]);

univ\_sseg u1 (

.cnt1 (a\_mag),

.cnt2 (0),

.valid (eq),

.dp\_en (0),

.dp\_sel (0),

.mod\_sel (0),

.sign (temp),

.clk (clk),

.ssegs (segs),

.disp\_en (anodes)

);

endmodule