Verilog Test Bench Design Code

Anand Rajan

Pennsylvania State University, University Park

Section 002R

CMPEN 331

Verilog Design Code

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

// Company: Pennsylvania State University, University Park

// Engineer: Anand Rajan

//

// Create Date: 02/04/2021 10:04:14 PM

// Design Name: Restoring Division Algorithm

// Module Name: top

// Project Name: Lab 1

// Target Devices: XC7Z010--1CLG400C

// Tool Versions:

// Description: The project aims to perform unsigned binary division by implementing a restoring division algorithm.

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

`timescale 1ns/1ps

module top(input clk, input clrn, input start, input [31:0] a, input [15:0] b,

output [31:0] qout, output [15:0] r, output reg busy, output reg ready, output[5:0] count

);

reg[16:0] sub\_output = 17'b0;

reg[16:0] lhs, rhs;

reg[15:0] rmux\_output = 16'b0;

reg[15:0] reg\_r, reg\_b;

reg[31:0] reg\_q;

reg [31:0] qin = 32'b0;

reg [5:0] counter;

assign r = reg\_r;

assign qout = reg\_q;

always @(posedge clk or negedge clrn)

begin

if (clrn == 0) begin // Nothing has been activated

busy <= 0;

ready <= 0;

end else if (start == 1) begin // Division is now initialized

reg\_q <= a;

reg\_b <= b;

reg\_r <= 16'b0;

lhs <= 17'b0;

rhs <= 17'b0;

busy <= 1;

counter <= 'd0;

ready <= 0;

end else if (busy == 1) begin // Division has begun

if (count == 5'd31) begin // 32 left-shifts must be completed, where the first left-shift starts at count = 1

busy <= 0;

ready <= 1;

reg\_q <= qin;

end // Division is still ongoing

lhs = {reg\_r, reg\_q[31]};

rhs = {1'b0, reg\_b};

sub\_output = (lhs - rhs); // Output of the subtractor

// Restoring Mux Logic

if (sub\_output[16] == 0) begin // If the result was non-negative, retain subtraction and make new LSB of result 1

rmux\_output = sub\_output[15:0];

qin[0] = 1'b1;

end else begin // If result was negative, restore the r reg value and set LSB to zero

rmux\_output = lhs[15:0];

qin[0] = 1'b0;

end

qin = (qin<<1); // Left shifts the input q value to account for the incoming LSB

reg\_r = rmux\_output; // R reg value now becomes whatever the mux outputs

reg\_q = reg\_q<<1; // Q reg input is left-shifted to remove the MSB that was already used in calculations

counter = counter + 'd1;

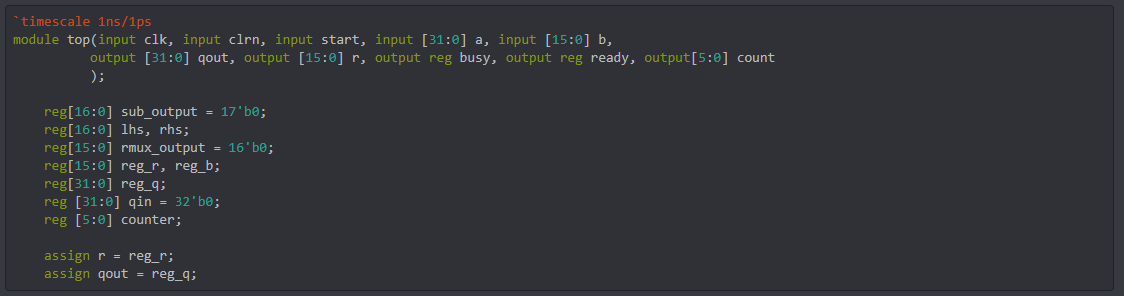
end

end

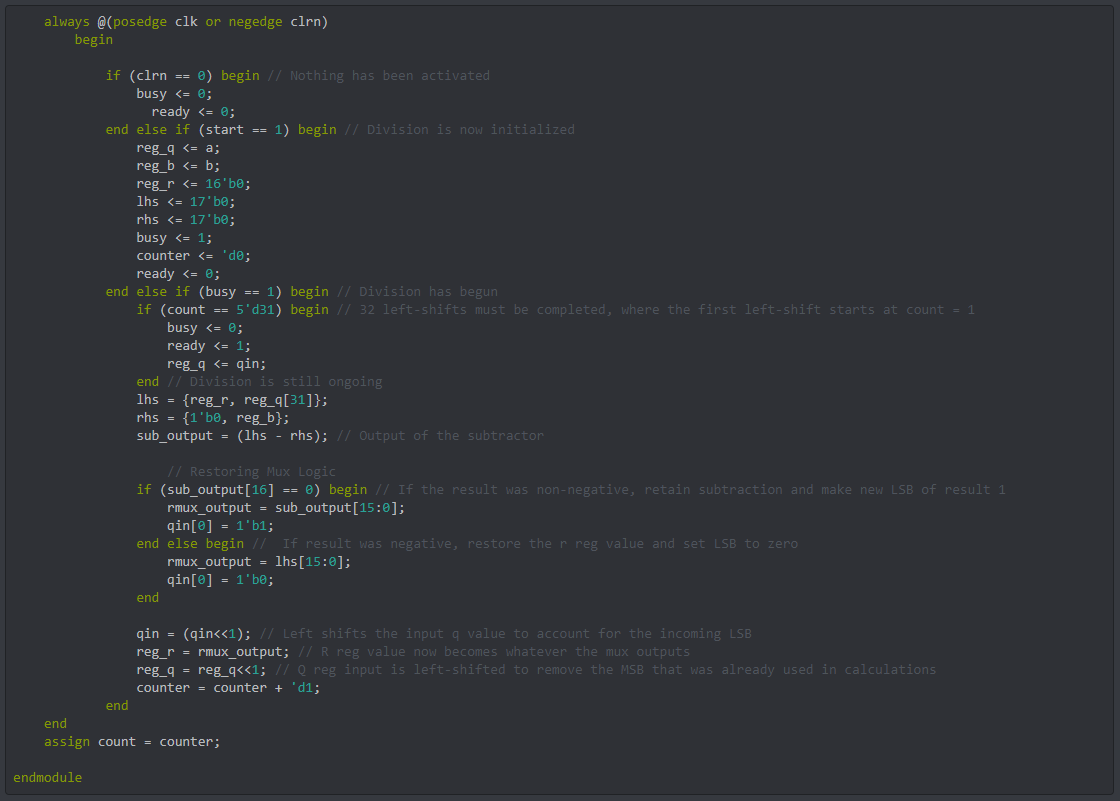
assign count = counter;

endmodule

Note that the device used, as listed in the prefacing comments, is XC7Z010-1CLG400C, as prescribed by the laboratory instructions. For readability purposes, screenshots of the same code is provided with keyword color highlighting as well as indentation displayed below (with the omission of the prefacing comments).



*Image 1. Screenshot containing first half of Verilog design code – included for display, readability, and aesthetic purposes.*



*Image 2. Screenshot containing second and majority half of Verilog design code – included for display, readability, and aesthetic purposes.*

Verilog Test Bench Design Code

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

// Company: Pennsylvania State University, University Park

// Engineer: Anand Rajan

//

// Create Date: 02/04/2021 10:04:14 PM

// Design Name: Restoring Division Algorithm

// Module Name: top

// Project Name: Lab 1

// Target Devices: XC7Z010--1CLG400C

// Tool Versions:

// Description: The project aims to perform unsigned binary division by implementing a restoring division algorithm.

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

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

`timescale 1ns/1ps

module testbench();

reg clrn\_tb;

reg clk\_tb;

reg start\_tb; // Hardcoded signals

wire busy\_tb;

wire ready\_tb;

wire[5:0] count\_tb; // Derived signals

reg [31:0] a\_tb;

reg [15:0] b\_tb;

wire [31:0] qout\_tb;

wire [15:0] r\_tb; // All input/outputs

top dut(clk\_tb, clrn\_tb, start\_tb, a\_tb, b\_tb, qout\_tb, r\_tb, busy\_tb, ready\_tb, count\_tb); // Initializing an instance of top()

initial begin

clrn\_tb = 0;

start\_tb = 0;

clk\_tb = 1;

// Fix the values of the inputs

a\_tb <= 32'b01001100011111110010001010001010;

b\_tb <= 16'b0110101000001110;

// Signals take value

#5 clrn\_tb = 1;

#0 start\_tb = 1;

#10 start\_tb = 0;

// CLRN shuts off to reset circuit

#335 clrn\_tb = 0;

#0 start\_tb = 0;

// New values for inputs are provided while CLRN is disabled

a\_tb <= 32'b00000000111111111111111100000000;

b\_tb <= 16'b0000000000000100;

// CLRN starts up again, and division can recontinue - note that CLRN and start must begin before clk posedge

#5 clrn\_tb = 1;

#0 start\_tb = 1;

#10 start\_tb = 0;

end

always begin

#5;

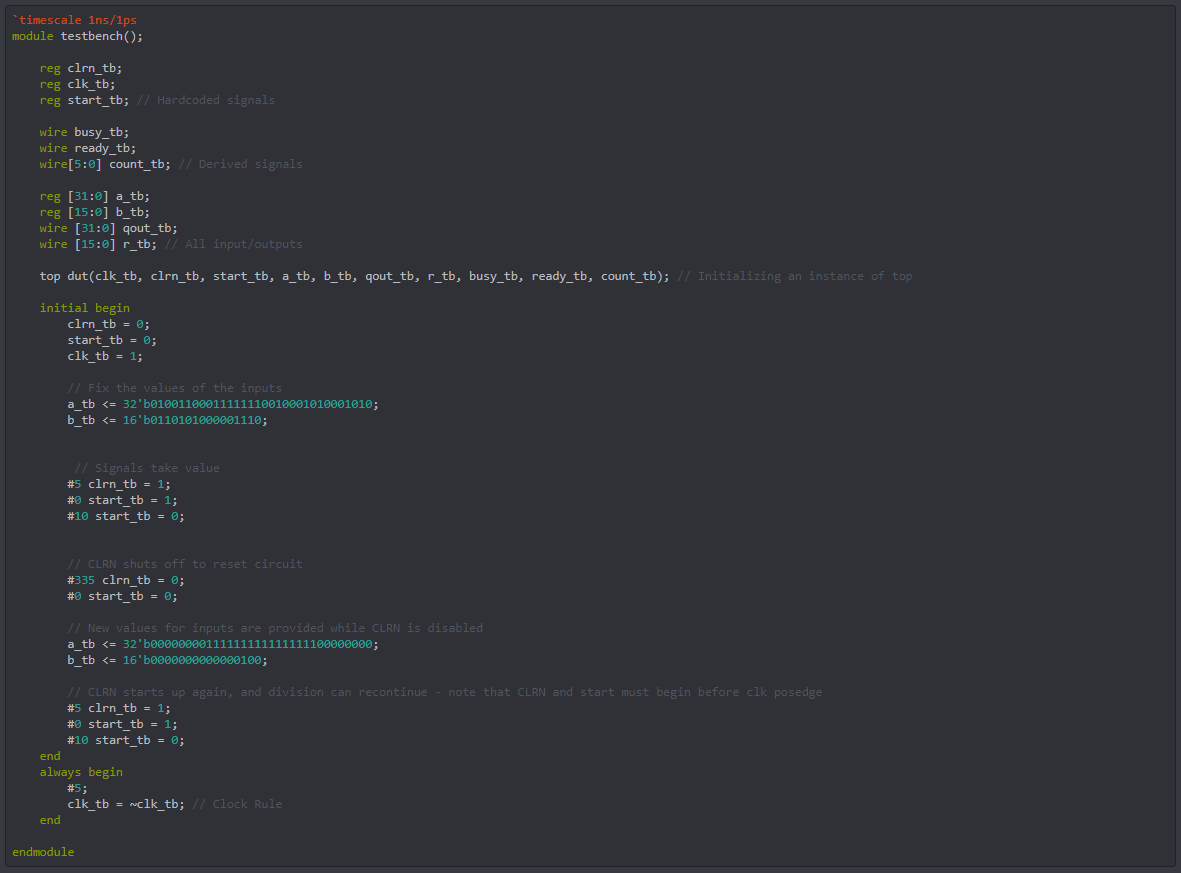
clk\_tb = ~clk\_tb; // Clock Rule

end

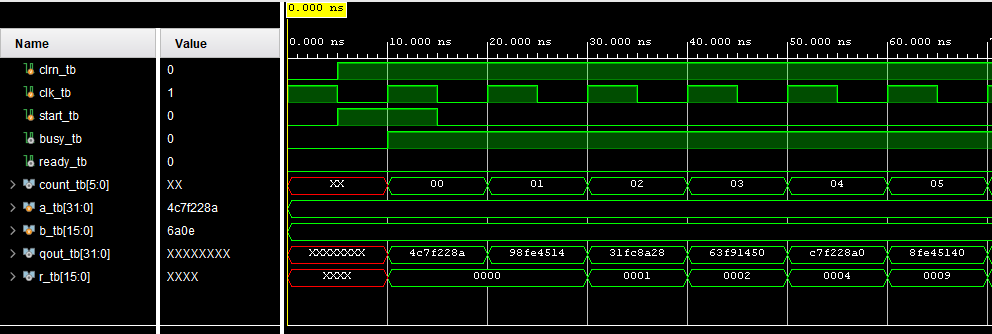
endmodule

Note that in this testbench, while the instructions ask for the bench code to perform specifically the division of (), the provided code accounts for both this division as well as the division of (), which is asked for later on in the instructions. Once again, for readability purposes, screenshots of the same code is provided with keyword color highlighting as well as indentation displayed below (with the omission of the prefacing comments).

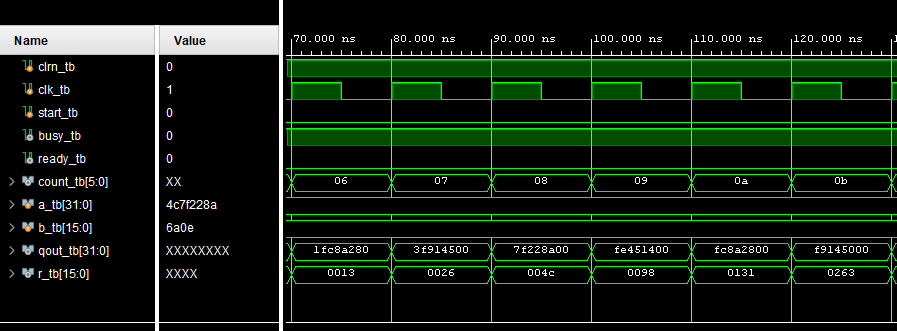
The corresponding waveforms for the simulation of this design and testbench design code is also provided below the code screenshot. **A length of 680 ns was not achievable for display in one screenshot, so this signal display is broken down into multiple screenshots (in order to make the screenshots even remotely legible, each one displays a waveform of length 60 ns). Additionally, the values on each signal display the value of the marker that wasn’t moved from the start of the waveform. The accurate values are displayed on each signal.** At 330 ns, the ready signal changes value, indicating that the first division is complete. CLRN shuts off at 350 ns, and turns back on at 355 ns, along with the start signal. The new division begins at 360 ns, when the busy signal changes value to 1 again. This second division completes when the ready signal changes back to 1 at 680 ns. The values displayed for qout\_tb and r\_tb are the final quotient and remainder of the divisions at these specific times (when ready has a value of 1).



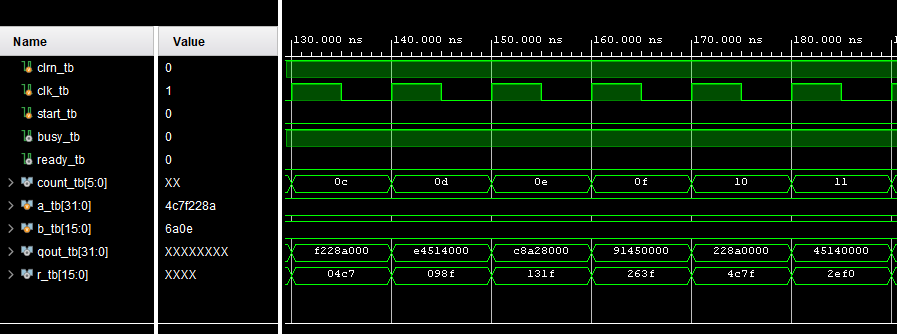
*Image 3. Screenshot of Verilog Test Bench Design Code – accounts for both provided division values in the instruction sheet (item 5)*



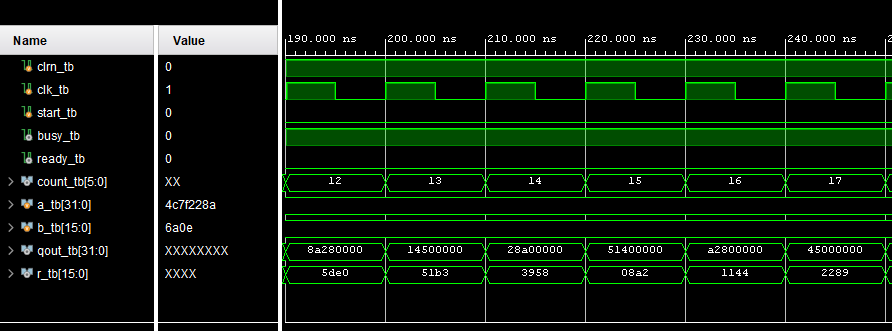
*Image 4. Screenshot 1 of Final Waveform*



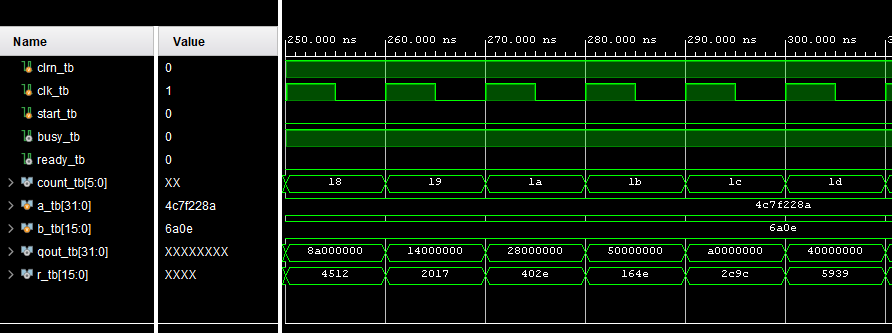
*Image 5. Screenshot 2 of Final Waveform*



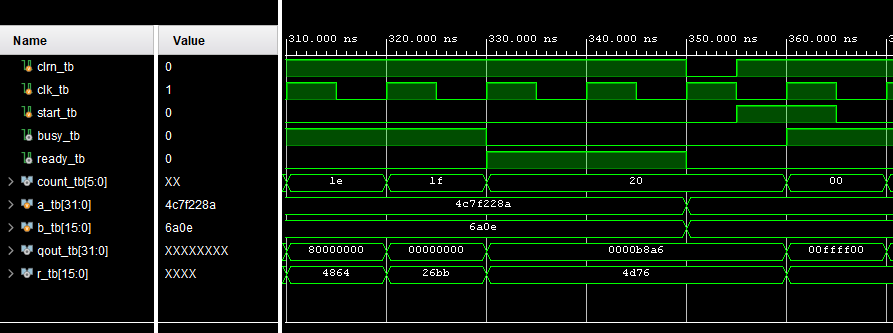
*Image 5. Screenshot 3 of Final Waveform*



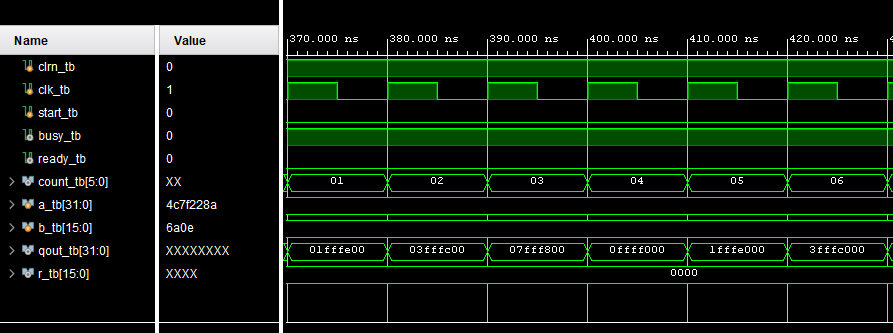
*Image 6. Screenshot 4 of Final Waveform*



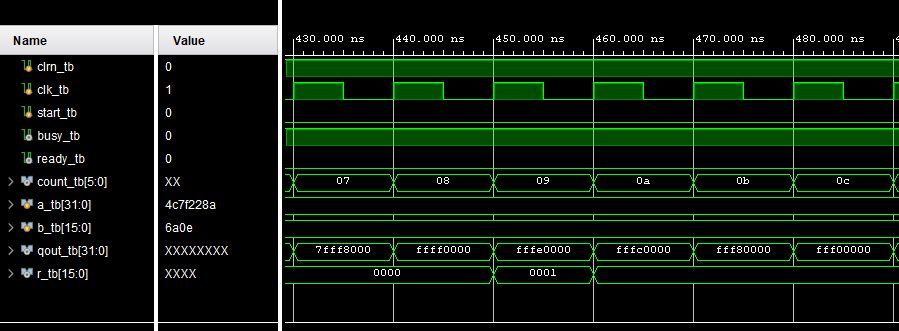
*Image 7. Screenshot 5 of Final Waveform*



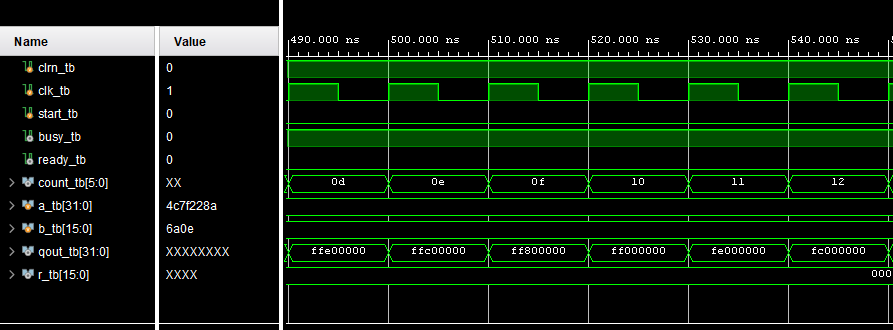
*Image 8. Screenshot 6 of Final Waveform*



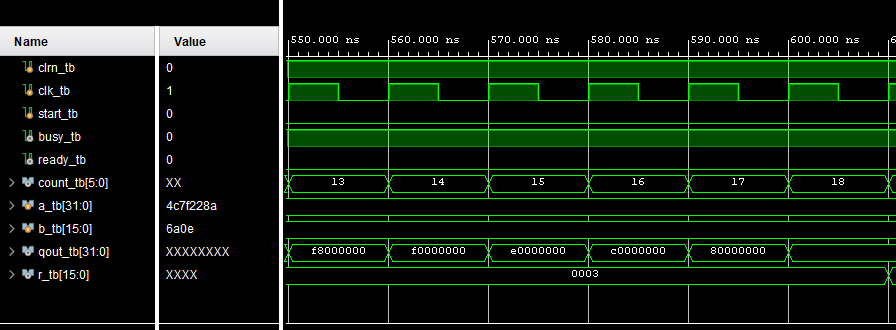
*Image 9. Screenshot 7 of Final Waveform*



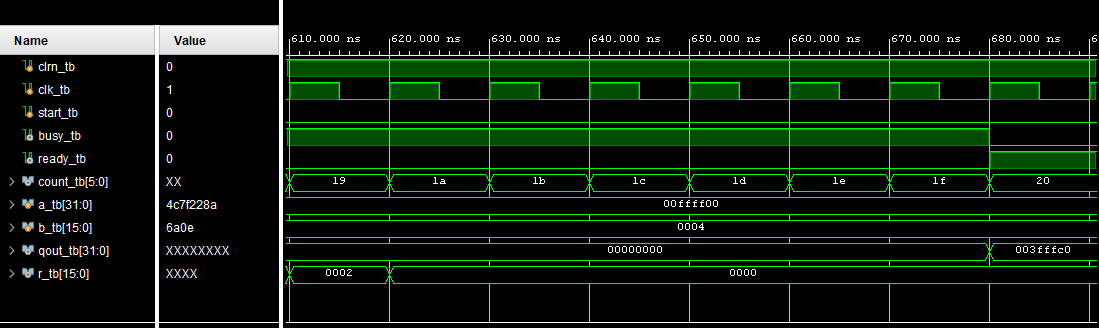
*Image 10. Screenshot 8 of Final Waveform*



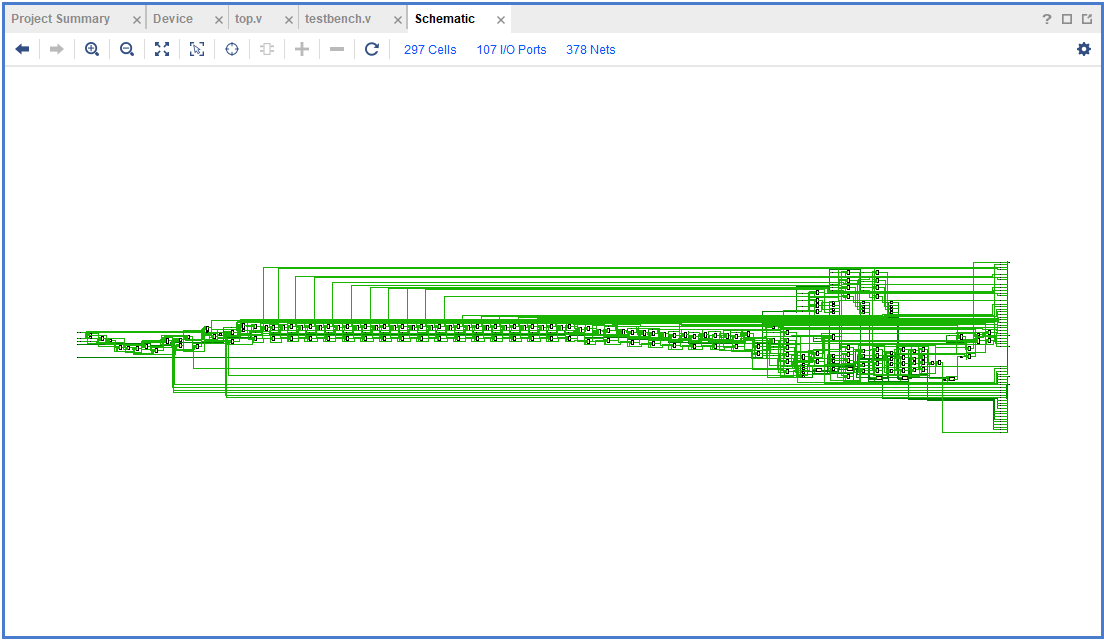
*Image 11. Screenshot 9 of Final Waveform*



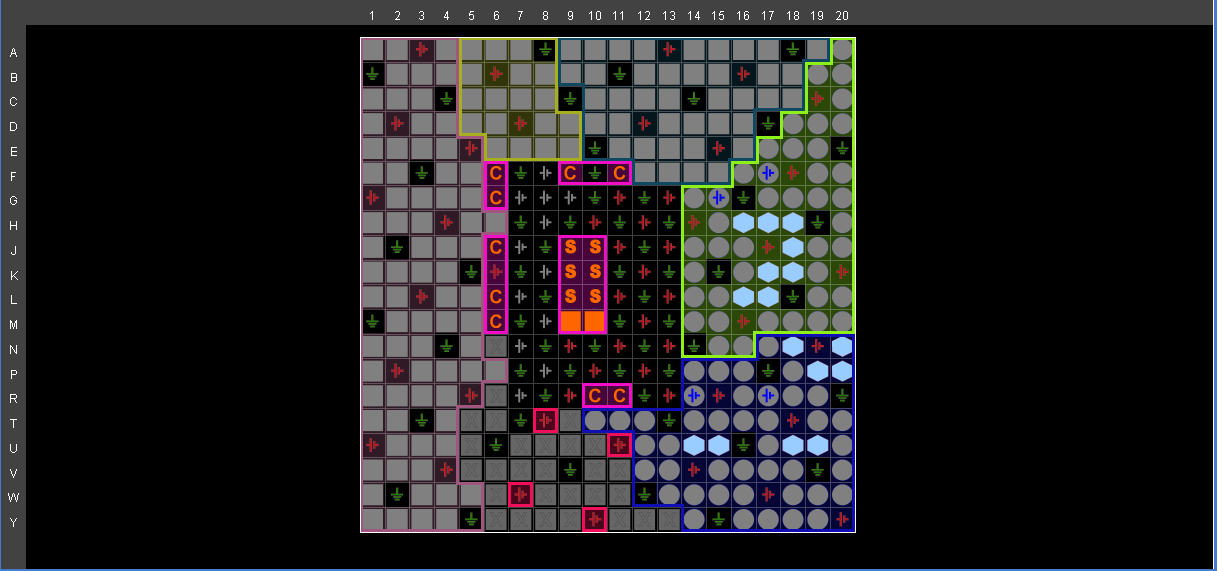
*Image 12. Screenshot 10 of Final Waveform*



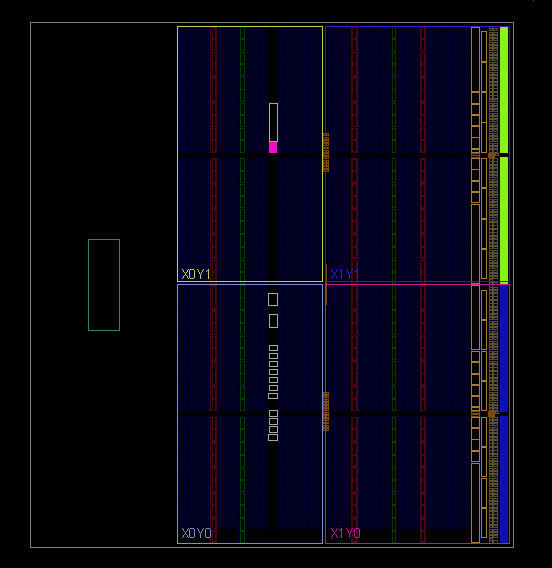
*Image 13. (Final) Screenshot 11 of Final Waveform*



*Image 14. Design Schematics from the Synthesis of the Project (****SYNTHESIZED DESIGN –*** *xc7z010clg400-3)*



*Image 15. Snapshot of the I/O Planning*



*Image 16. Snapshot of the Floor Planning*

Images 14 through 16 provide the design schematic of the synthesized design and snapshots of the I/O and Floor planning of the same. Note that the schematic provided could not be zoomed or resized in any way to make it clearer or larger. The information of the schematic is provided just above the design, in the same bar where the number of cells and such are mentioned.

With this, the requirements specified by the lab’s instructions are concluded. To recap, the specified device was used in this lab, which was done in Vivado (after initial testing codes were run on EDAPlayground). The Verilog design code as well as test bench code were both provided (albeit not with the 2x line spacing condition, but with 1x in order to fit the code into as compact a space as possible). The code accounted for the division specified, as well as the other division mentioned in item 5 of the instructions. Snapshots of the same were also provided in order to supplement readability and aesthetics such as indentations and comments. Waveforms resulting from the verification of my design using the simulation software were also provided, with all the specified signals shown (but not in the same order). These consist of 10 consecutive images pasted earlier in my report. The waveforms cover a timing period from 0 till 680 ns. Additionally, the design was synthesized and the design schematics of this as well as the I/O and Floor planning designs were provided in snapshots (images 14 through 16). The cover page requirements were met to the best of my ability, and this project will be uploaded as word file (with no use of LaTex).