**Oscar Alejandro Manzano**

**EE 4513 VLSI**

**Report Lab 8**

**October 26, 2019**

For this lab, I used a new program from Verilog to make a multiplier 64x64 and a counter, after that I upload files into the server to start the design. After getting all the directories with the right settings I was able to used Encounter for the actual design and successful got everything I need.

Fall 2019

# EE 4513 – Introduction to VLSI Design

**Lab Assignment # 8**

In this lab assignment,

1. You are required to design: (i) 64 x 64 Array Multiplier

(ii) 16-bit synchronous up/down (selectable) counter

1. Once you have written the Verilog code of your design and completed the behavioral simulation (for example, using Vivado), synthesize your code to generate the netlist using Cadence RTL Compiler (RC).

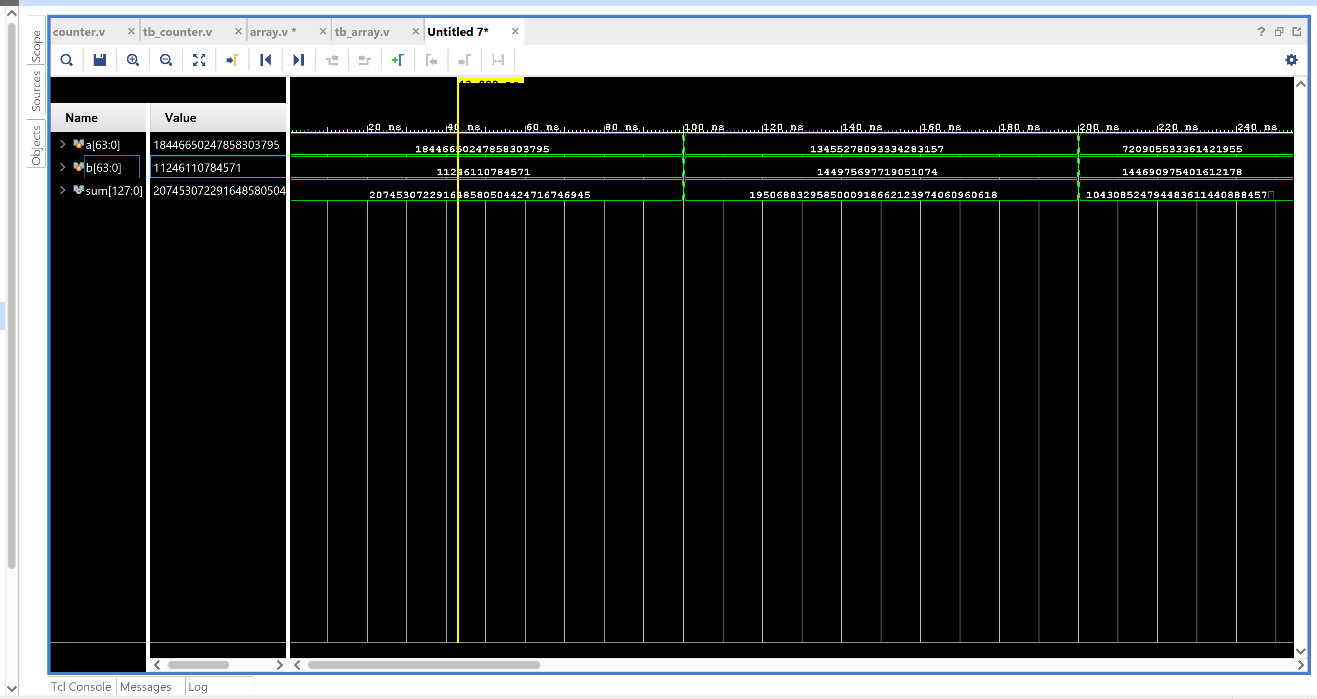
1. Then, use the Cadence Encounter platform to generate the GDS2 layout of your design. Verify your design for connectivity, and geometry violations.

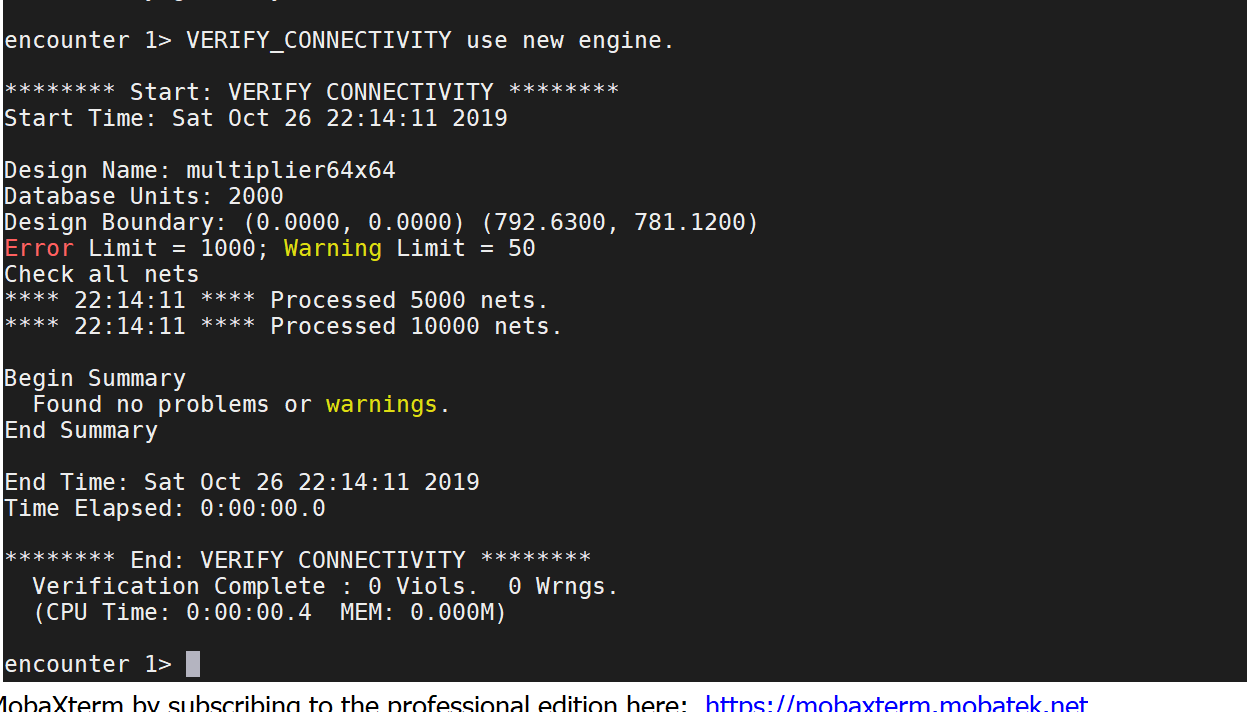
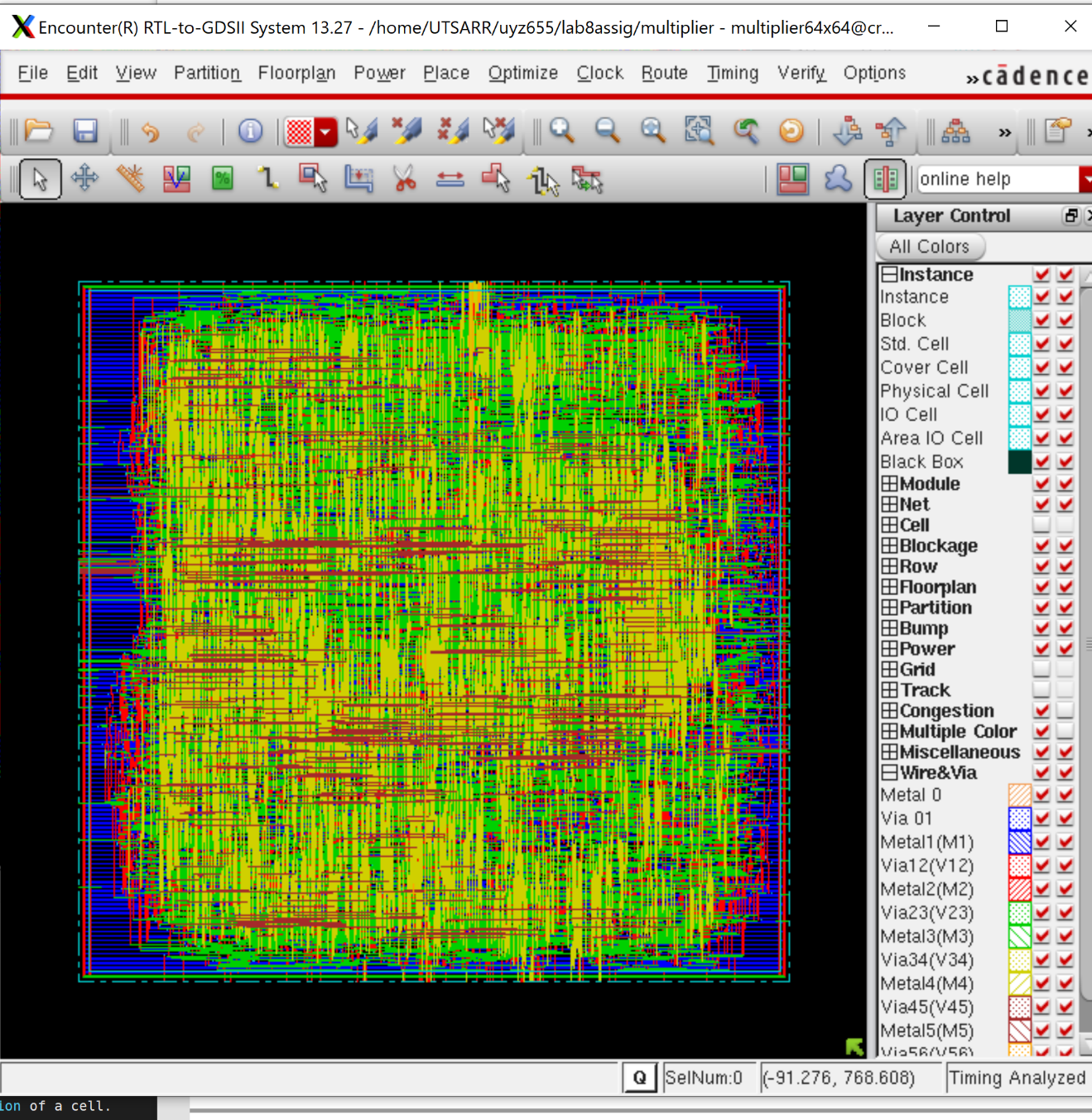
Turn in report, which includes the following: -

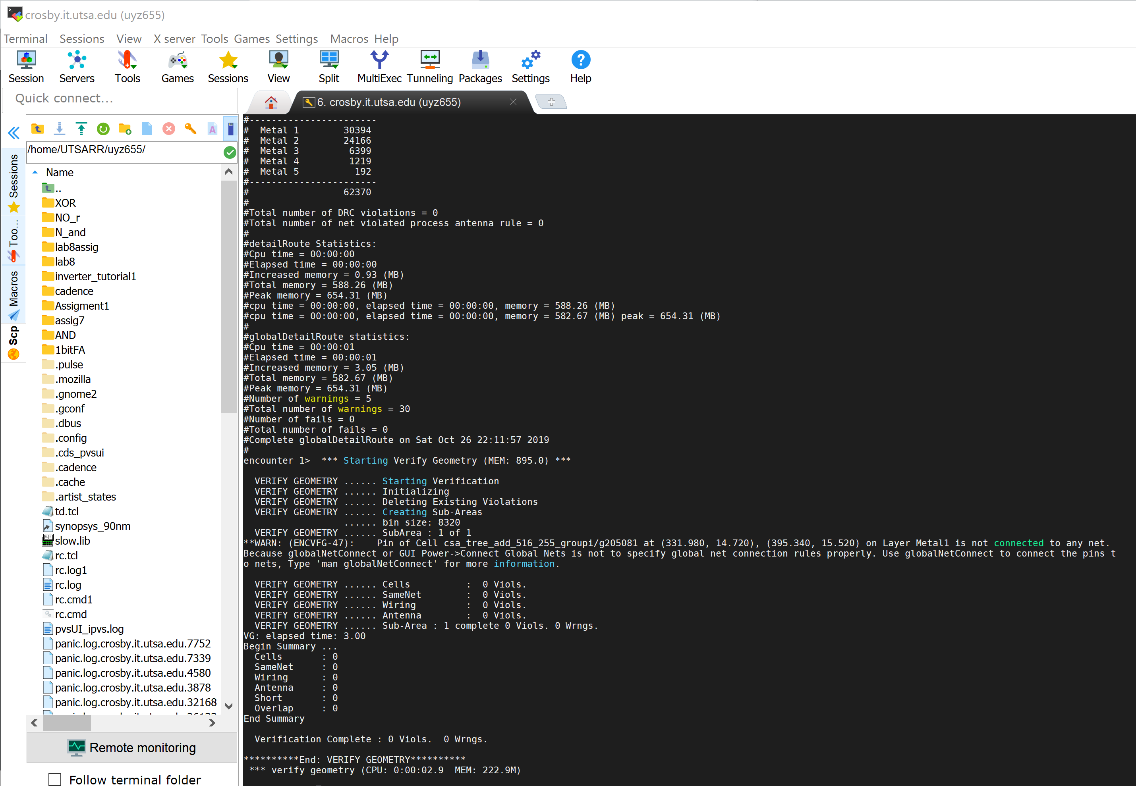
* 1. The directory path (in Linux) where you worked on and created the multiplier and counter. Work on the two designs in separate directory.
  2. Simulation waveforms for your multiplier and counter
  3. Physical Layout snapshot of your design
  4. Snapshots of portions of the connectivity and geometry verification reports, showing any violations, if present.
  5. Schematic of your design (can be seen by clicking Tools -> Schematic Viewer)
  6. Report Your Timing Analysis (Slew, Delay, Arrival...etc) and Total Area from your RTL compiler for your design.
  7. Snapshot of the message showing successful generation of GDS2 layout
  8. Try to optimize your design in the Verilog (efficient /structured code) and post a conclusion for your design.

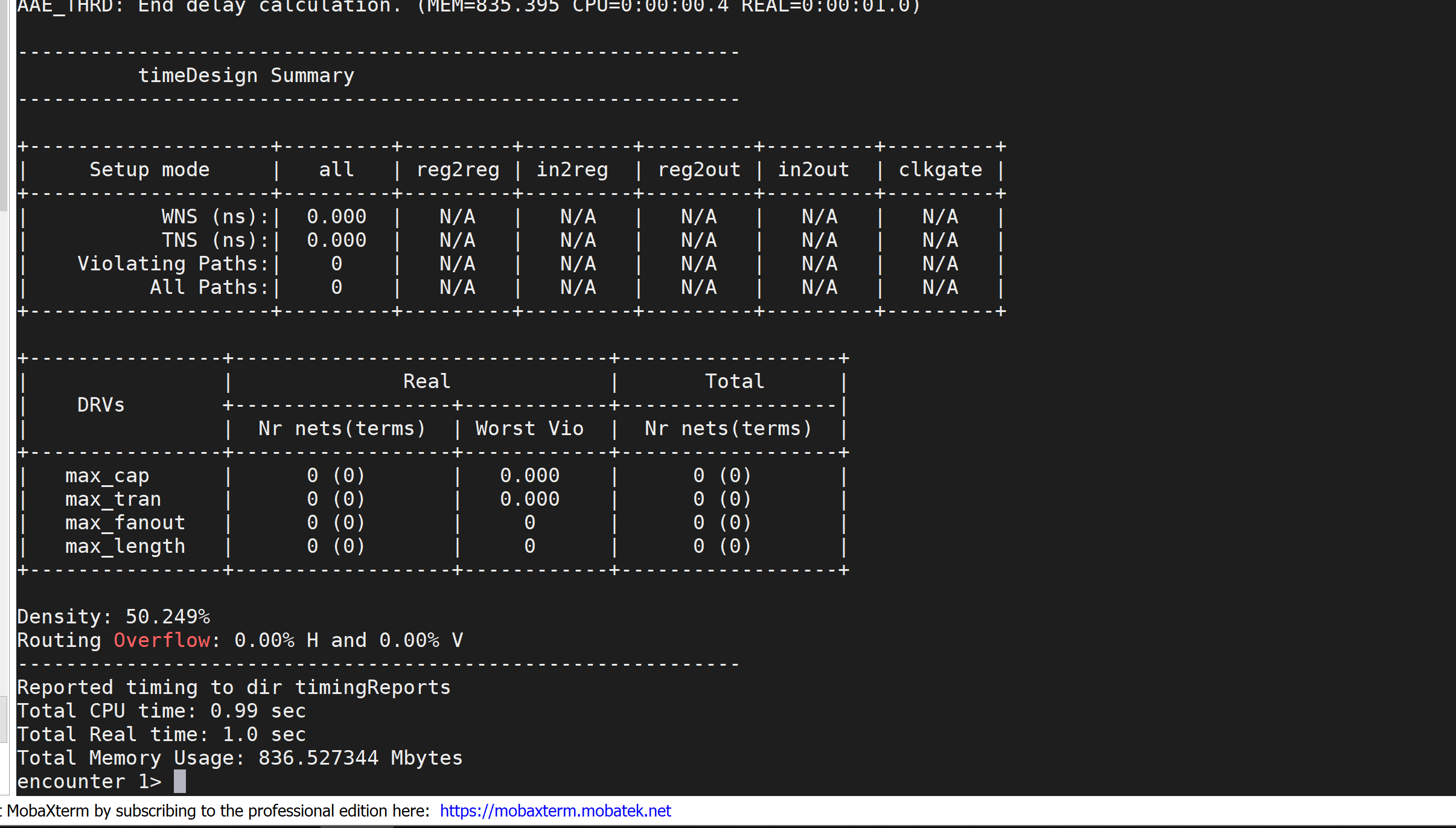
**Note**: -

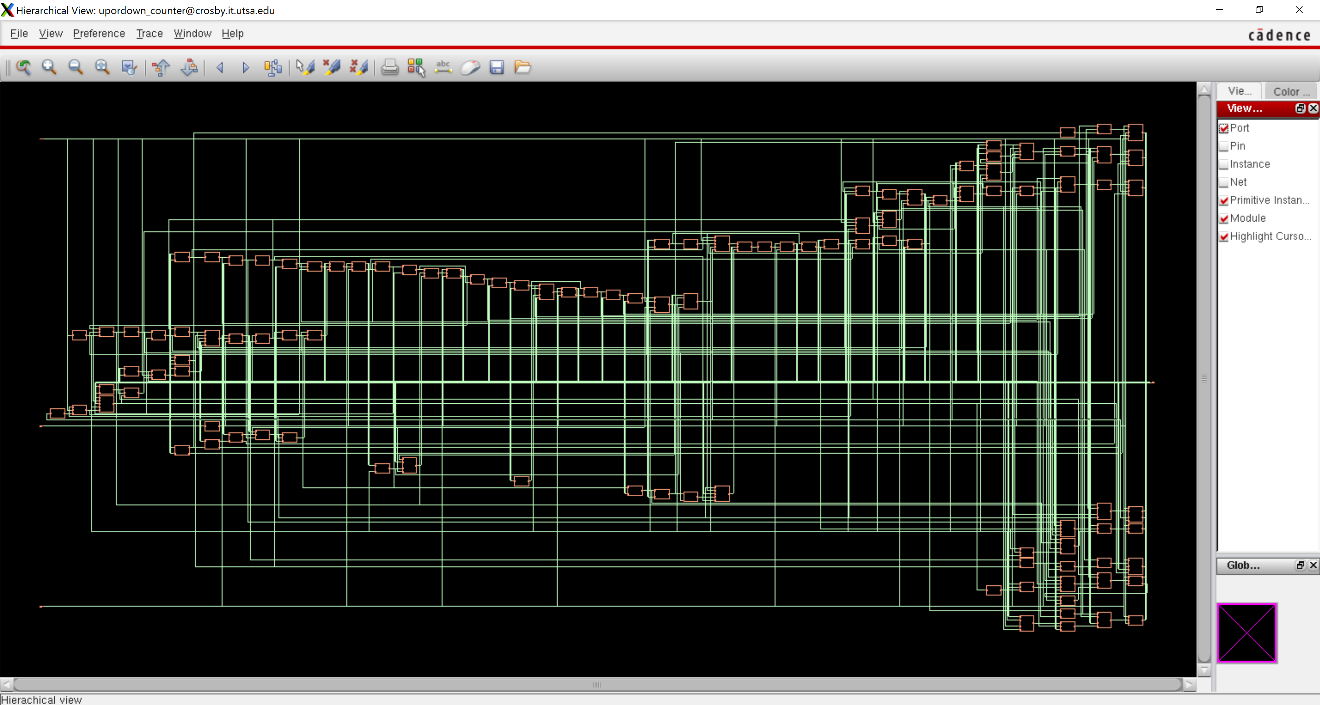
1. Once you have logged in, you can make a new directory called **lab8hw** using the **mkdir** command (**mkdir lab8hw**) in the terminal. Then change over to the created directory using **cd lab8hw.** Inside this directory, make two new directories called “**multiplier**” and “**counter**” using the **mkdir** command. Use these two directories to create the layout of the respective Verilog modules that you will be making in this assignment. Provide the path of these two directories in the report.
2. Multiplier

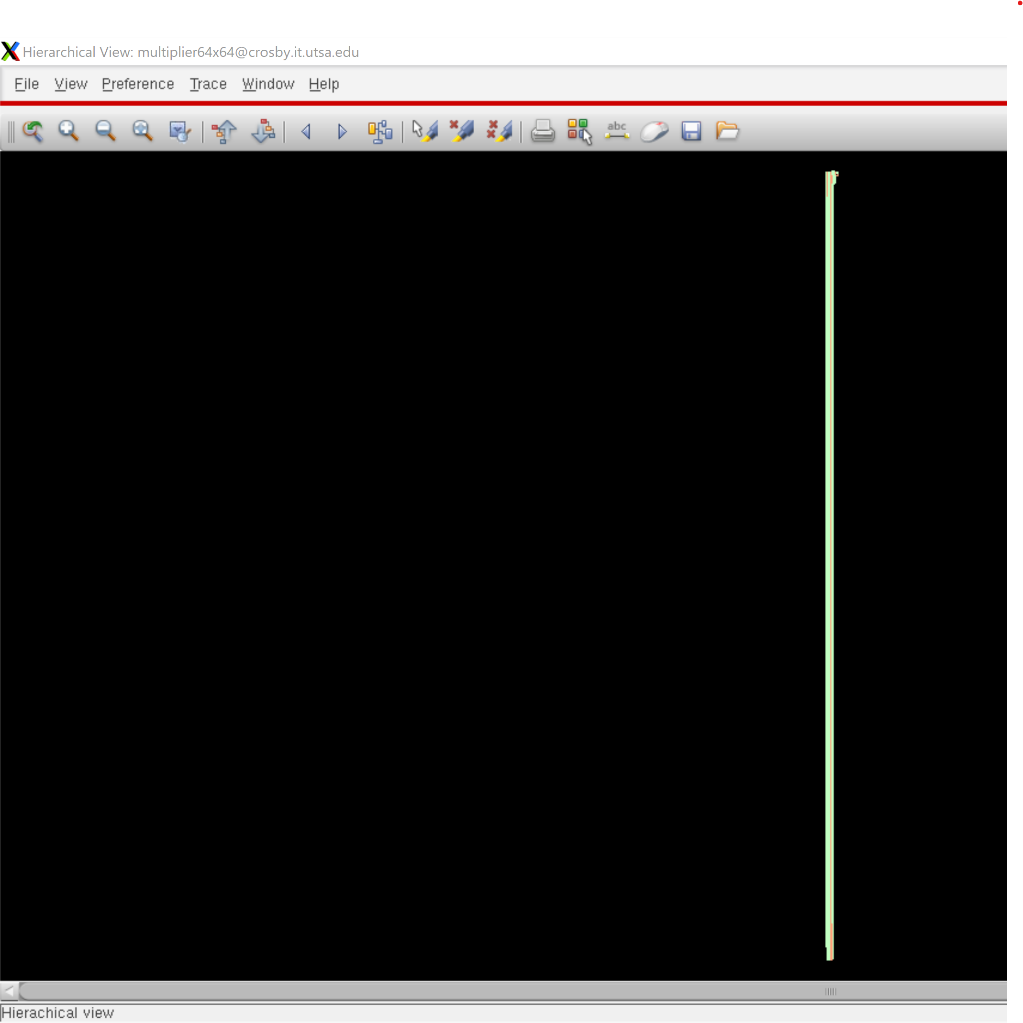


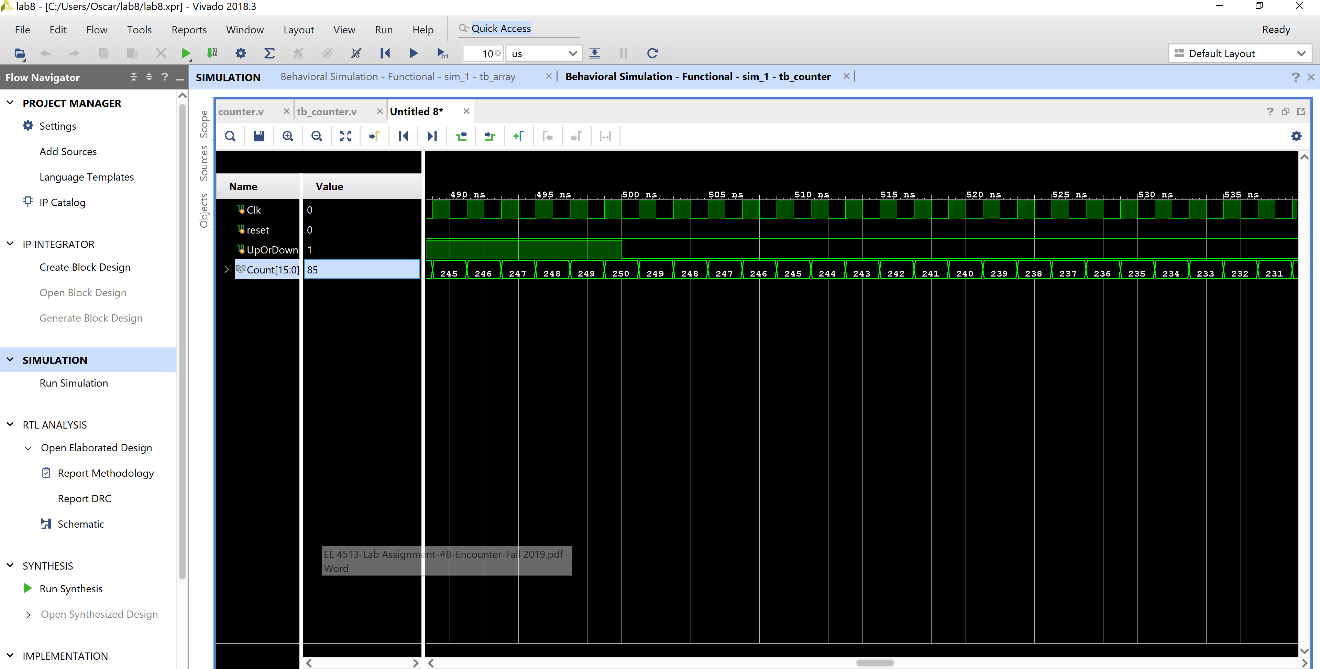


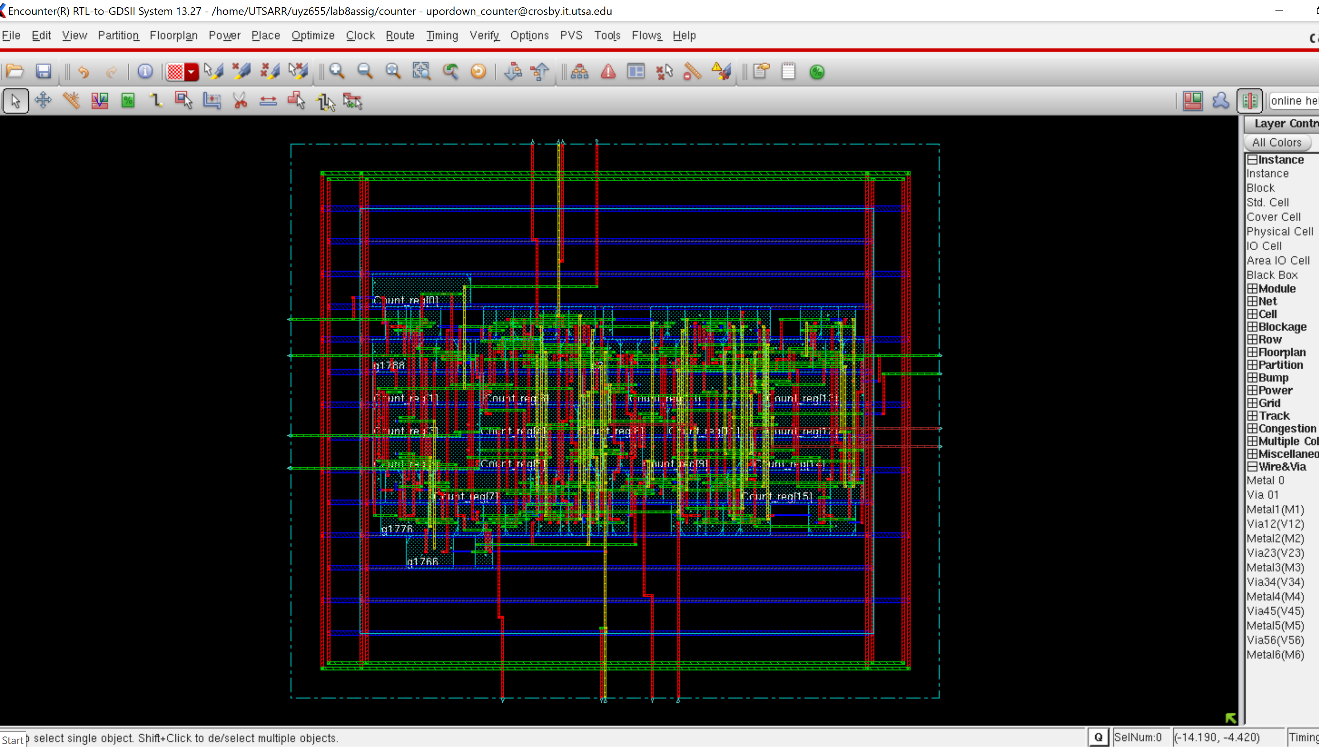


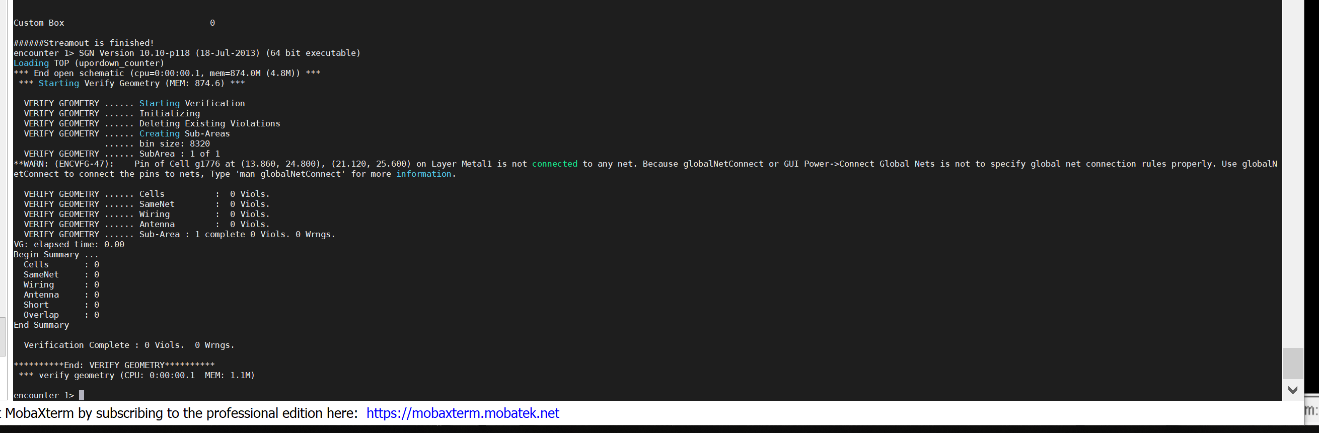


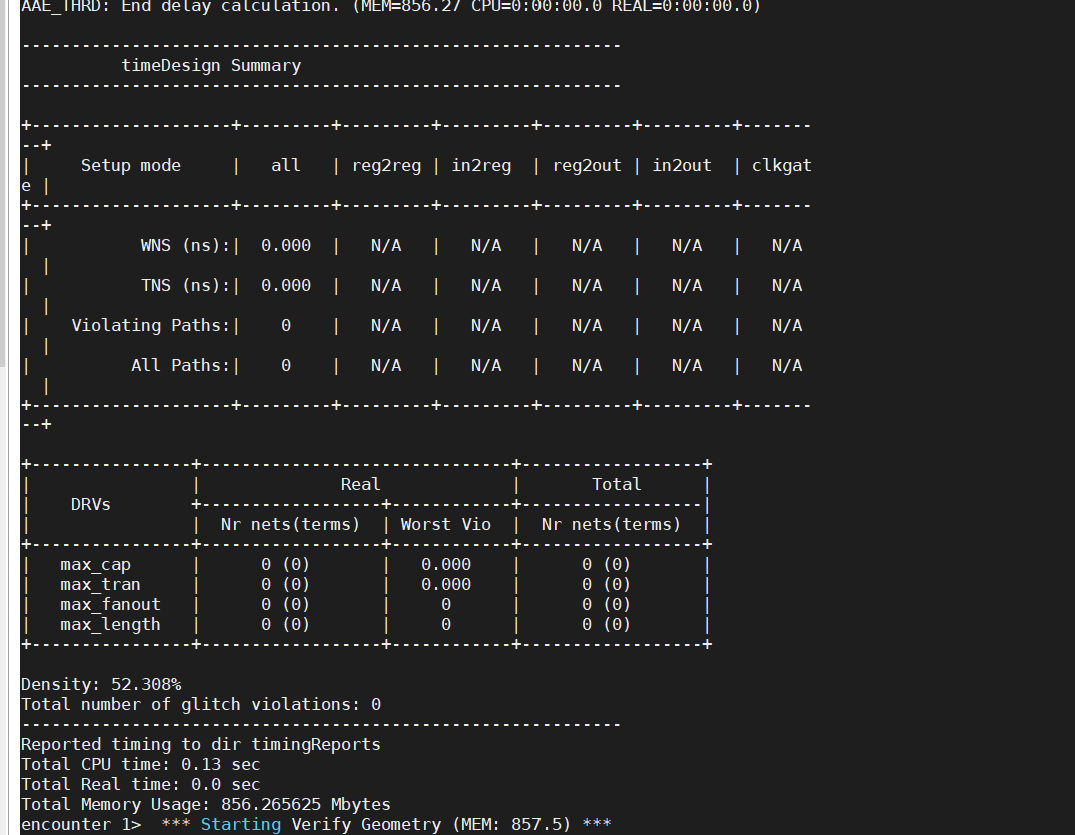




2.Count









Code:

Multiplier

`timescale 1ns / 1ps

module multiplier64x64(input [63:0]a,b, output [127:0]sum);

reg[63:0] p1;

reg[64:0] p2;

reg[65:0] p3;

reg[66:0] p4;

reg[67:0] p5;

reg[68:0] p6;

reg[69:0] p7;

reg[70:0] p8;

reg[71:0] p9;

reg[72:0] p10;

reg[73:0] p11;

reg[74:0] p12;

reg[75:0] p13;

reg[76:0] p14;

reg[77:0] p15;

reg[78:0] p16;

reg[79:0] p17;

reg[80:0] p18;

reg[81:0] p19;

reg[82:0] p20;

reg[83:0] p21;

reg[84:0] p22;

reg[85:0] p23;

reg[86:0] p24;

reg[87:0] p25;

reg[88:0] p26;

reg[89:0] p27;

reg[90:0] p28;

reg[91:0] p29;

reg[92:0] p30;

reg[93:0] p31;

reg[94:0] p32;

reg[95:0] p33;

reg[96:0] p34;

reg[97:0] p35;

reg[98:0] p36;

reg[99:0] p37;

reg[100:0] p38;

reg[101:0] p39;

reg[102:0] p40;

reg[103:0] p41;

reg[104:0] p42;

reg[105:0] p43;

reg[106:0] p44;

reg[107:0] p45;

reg[108:0] p46;

reg[109:0] p47;

reg[110:0] p48;

reg[111:0] p49;

reg[112:0] p50;

reg[113:0] p51;

reg[114:0] p52;

reg[115:0] p53;

reg[116:0] p54;

reg[117:0] p55;

reg[118:0] p56;

reg[119:0] p57;

reg[120:0] p58;

reg[121:0] p59;

reg[122:0] p60;

reg[123:0] p61;

reg[124:0] p62;

reg[125:0] p63;

reg[126:0] p64;

always @(a or b) begin

if(b[0]==1'b0)

p1[63:0] <= 0;

else

p1[63:0] <= a;

if(b[1]==1'b0)

p2[64:0] <= 0;

else begin

p2[0] <= 0;

p2[64:1] <= a;

end

if(b[2]==1'b0)

p3[65:0] <= 0;

else begin

p3[1:0] <= 0;

p3[65:2] <= a;

end

if(b[3]==1'b0)

p4[66:0] <= 0;

else begin

p4[2:0] <= 0;

p4[66:3] <= a;

end

if(b[4]==1'b0)

p5[67:0] <= 0;

else begin

p5[3:0] <= 0;

p5[67:4] <= a;

end

if(b[5]==1'b0)

p6[68:0] <= 0;

else begin

p6[4:0] <= 0;

p6[68:5] <= a;

end

if(b[6]==1'b0)

p7[69:0] <= 0;

else begin

p7[5:0] <= 0;

p7[69:6] <= a;

end

if(b[7]==1'b0)

p8[70:0] <= 0;

else begin

p8[6:0] <= 0;

p8[70:7] <= a;

end

if(b[8]==1'b0)

p9[71:0] <= 0;

else begin

p9[7:0] <= 0;

p9[71:8] <= a;

end

if(b[9]==1'b0)

p10[72:0] <= 0;

else begin

p10[8:0] <= 0;

p10[72:9] <= a;

end

if(b[10]==1'b0)

p11[73:0] <= 0;

else begin

p11[9:0] <= 0;

p11[73:10] <= a;

end

if(b[11]==1'b0)

p12[74:0] <= 0;

else begin

p12[10:0] <= 0;

p12[74:11] <= a;

end

if(b[12]==1'b0)

p13[75:0] <= 0;

else begin

p13[11:0] <= 0;

p13[75:12] <= a;

end

if(b[13]==1'b0)

p14[76:0] <= 0;

else begin

p14[12:0] <= 0;

p14[76:13] <= a;

end

if(b[14]==1'b0)

p15[77:0] <= 0;

else begin

p15[13:0] <= 0;

p15[77:14] <= a;

end

if(b[15]==1'b0)

p16[78:0] <= 0;

else begin

p16[14:0] <= 0;

p16[78:15] <= a;

end

if(b[16]==1'b0)

p17[79:0] <= 0;

else begin

p17[15:0] <= 0;

p17[79:16] <= a;

end

if(b[17]==1'b0)

p18[80:0] <= 0;

else begin

p18[16:0] <= 0;

p18[80:17] <= a;

end

if(b[18]==1'b0)

p19[81:0] <= 0;

else begin

p19[17:0] <= 0;

p19[81:18] <= a;

end

if(b[19]==1'b0)

p20[82:0] <= 0;

else begin

p20[18:0] <= 0;

p20[82:19] <= a;

end

if(b[20]==1'b0)

p21[83:0] <= 0;

else begin

p21[19:0] <= 0;

p21[83:20] <= a;

end

if(b[21]==1'b0)

p22[84:0] <= 0;

else begin

p22[20:0] <= 0;

p22[84:21] <= a;

end

if(b[22]==1'b0)

p23[85:0] <= 0;

else begin

p23[21:0] <= 0;

p23[85:22] <= a;

end

if(b[23]==1'b0)

p24[86:0] <= 0;

else begin

p24[22:0] <= 0;

p24[86:23] <= a;

end

if(b[24]==1'b0)

p25[87:0] <= 0;

else begin

p25[23:0] <= 0;

p25[87:24] <= a;

end

if(b[25]==1'b0)

p26[88:0] <= 0;

else begin

p26[24:0] <= 0;

p26[88:25] <= a;

end

if(b[26]==1'b0)

p27[89:0] <= 0;

else begin

p27[25:0] <= 0;

p27[89:26] <= a;

end

if(b[27]==1'b0)

p28[90:0] <= 0;

else begin

p28[26:0] <= 0;

p28[90:27] <= a;

end

if(b[28]==1'b0)

p29[91:0] <= 0;

else begin

p29[27:0] <= 0;

p29[91:28] <= a;

end

if(b[29]==1'b0)

p30[92:0] <= 0;

else begin

p30[28:0] <= 0;

p30[92:29] <= a;

end

if(b[30]==1'b0)

p31[93:0] <= 0;

else begin

p31[29:0] <= 0;

p31[93:30] <= a;

end

if(b[31]==1'b0)

p32[94:0] <= 0;

else begin

p32[30:0] <= 0;

p32[94:31] <= a;

end

if(b[32]==1'b0)

p33[95:0] <= 0;

else begin

p33[31:0] <= 0;

p33[95:32] <= a;

end

if(b[33]==1'b0)

p34[96:0] <= 0;

else begin

p34[32:0] <= 0;

p34[96:33] <= a;

end

if(b[34]==1'b0)

p35[97:0] <= 0;

else begin

p35[33:0] <= 0;

p35[97:34] <= a;

end

if(b[35]==1'b0)

p36[98:0] <= 0;

else begin

p36[34:0] <= 0;

p36[98:35] <= a;

end

if(b[36]==1'b0)

p37[99:0] <= 0;

else begin

p37[35:0] <= 0;

p37[99:36] <= a;

end

if(b[37]==1'b0)

p38[100:0] <= 0;

else begin

p38[36:0] <= 0;

p38[100:37] <= a;

end

if(b[38]==1'b0)

p39[101:0] <= 0;

else begin

p39[37:0] <= 0;

p39[101:38] <= a;

end

if(b[39]==1'b0)

p40[102:0] <= 0;

else begin

p40[38:0] <= 0;

p40[102:39] <= a;

end

if(b[40]==1'b0)

p41[103:0] <= 0;

else begin

p41[39:0] <= 0;

p41[103:40] <= a;

end

if(b[41]==1'b0)

p42[104:0] <= 0;

else begin

p42[40:0] <= 0;

p42[104:41] <= a;

end

if(b[42]==1'b0)

p43[105:0] <= 0;

else begin

p43[41:0] <= 0;

p43[105:42] <= a;

end

if(b[43]==1'b0)

p44[106:0] <= 0;

else begin

p44[42:0] <= 0;

p44[106:43] <= a;

end

if(b[44]==1'b0)

p45[107:0] <= 0;

else begin

p45[43:0] <= 0;

p45[107:44] <= a;

end

if(b[45]==1'b0)

p46[108:0] <= 0;

else begin

p46[44:0] <= 0;

p46[108:45] <= a;

end

if(b[46]==1'b0)

p47[109:0] <= 0;

else begin

p47[45:0] <= 0;

p47[109:46] <= a;

end

if(b[47]==1'b0)

p48[110:0] <= 0;

else begin

p48[46:0] <= 0;

p48[110:47] <= a;

end

if(b[48]==1'b0)

p49[111:0] <= 0;

else begin

p49[47:0] <= 0;

p49[111:48] <= a;

end

if(b[49]==1'b0)

p50[112:0] <= 0;

else begin

p50[48:0] <= 0;

p50[112:49] <= a;

end

if(b[50]==1'b0)

p51[113:0] <= 0;

else begin

p51[49:0] <= 0;

p51[113:50] <= a;

end

if(b[51]==1'b0)

p52[114:0] <= 0;

else begin

p52[50:0] <= 0;

p52[114:51] <= a;

end

if(b[52]==1'b0)

p53[115:0] <= 0;

else begin

p53[51:0] <= 0;

p53[115:52] <= a;

end

if(b[53]==1'b0)

p54[116:0] <= 0;

else begin

p54[52:0] <= 0;

p54[116:53] <= a;

end

if(b[54]==1'b0)

p55[117:0] <= 0;

else begin

p55[53:0] <= 0;

p55[117:54] <= a;

end

if(b[55]==1'b0)

p56[118:0] <= 0;

else begin

p56[54:0] <= 0;

p56[118:55] <= a;

end

if(b[56]==1'b0)

p57[119:0] <= 0;

else begin

p57[55:0] <= 0;

p57[119:56] <= a;

end

if(b[57]==1'b0)

p58[120:0] <= 0;

else begin

p58[56:0] <= 0;

p58[120:57] <= a;

end

if(b[58]==1'b0)

p59[121:0] <= 0;

else begin

p59[57:0] <= 0;

p59[121:58] <= a;

end

if(b[59]==1'b0)

p60[122:0] <= 0;

else begin

p60[58:0] <= 0;

p60[122:59] <= a;

end

if(b[60]==1'b0)

p61[123:0] <= 0;

else begin

p61[59:0] <= 0;

p61[123:60] <= a;

end

if(b[61]==1'b0)

p62[124:0] <= 0;

else begin

p62[60:0] <= 0;

p62[124:61] <= a;

end

if(b[62]==1'b0)

p63[125:0] <= 0;

else begin

p63[61:0] <= 0;

p63[125:62] <= a;

end

if(b[63]==1'b0)

p64[126:0] <= 0;

else begin

p64[62:0] <= 0;

p64[126:63] <= a;

end

end

assign sum=p1+p2+p3+p4+p5+p6+p7+p8+p9+p10+p11+p12+p13+p14+p15+p16+p17+p18+p19+p20+p21+p22+p23+p24+p25+p26+p27+p28+p29+p30+p31+p32+p33+p34+p35+p36+p37+p38+p39+p40+p41+p42+p43+p44+p45+p46+p47+p48+p49+p50+p51+p52+p53+p54+p55+p56+p57+p58+p59+p60+p61+p62+p63+p64;

endmodule

counter:

`timescale 1ns / 1ps

module upordown\_counter(

Clk,

reset,

UpOrDown, //high for UP counter and low for Down counter

Count

);

//input ports and their sizes

input Clk,reset,UpOrDown;

//output ports and their size

output [15 : 0] Count;

//Internal variables

reg [15 : 0] Count = 0;

always @(posedge(Clk) or posedge(reset))

begin

if(reset == 1)

Count <= 0;

else

if(UpOrDown == 1) //Up mode selected

if(Count == 65536)

Count <= 0;

else

Count <= Count + 1; //Incremend Counter

else //Down mode selected

if(Count == 0)

Count <= 65536;

else

Count <= Count - 1; //Decrement counter

end

endmodule