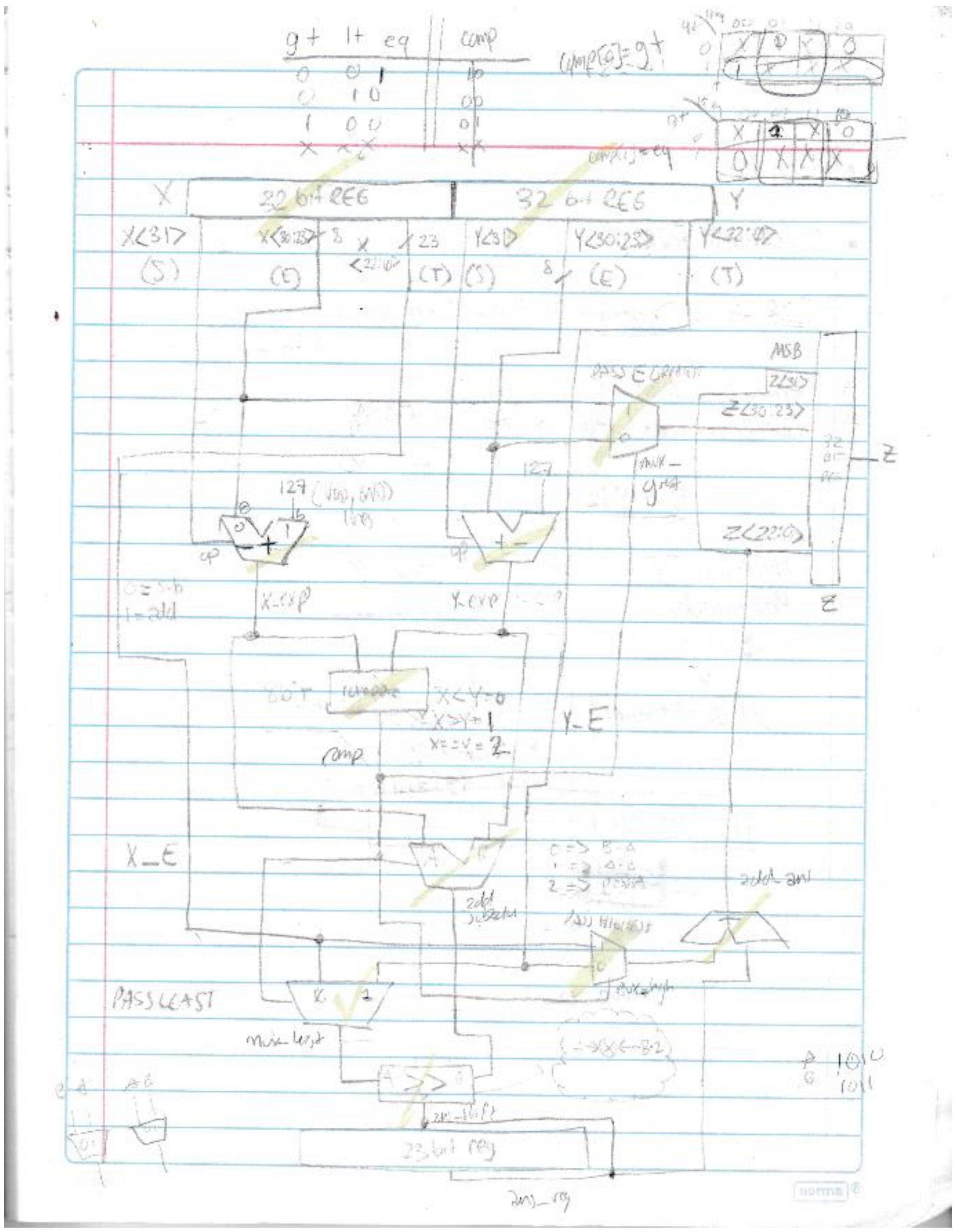
Bruno E. Gracia Villalobos

EE 4513

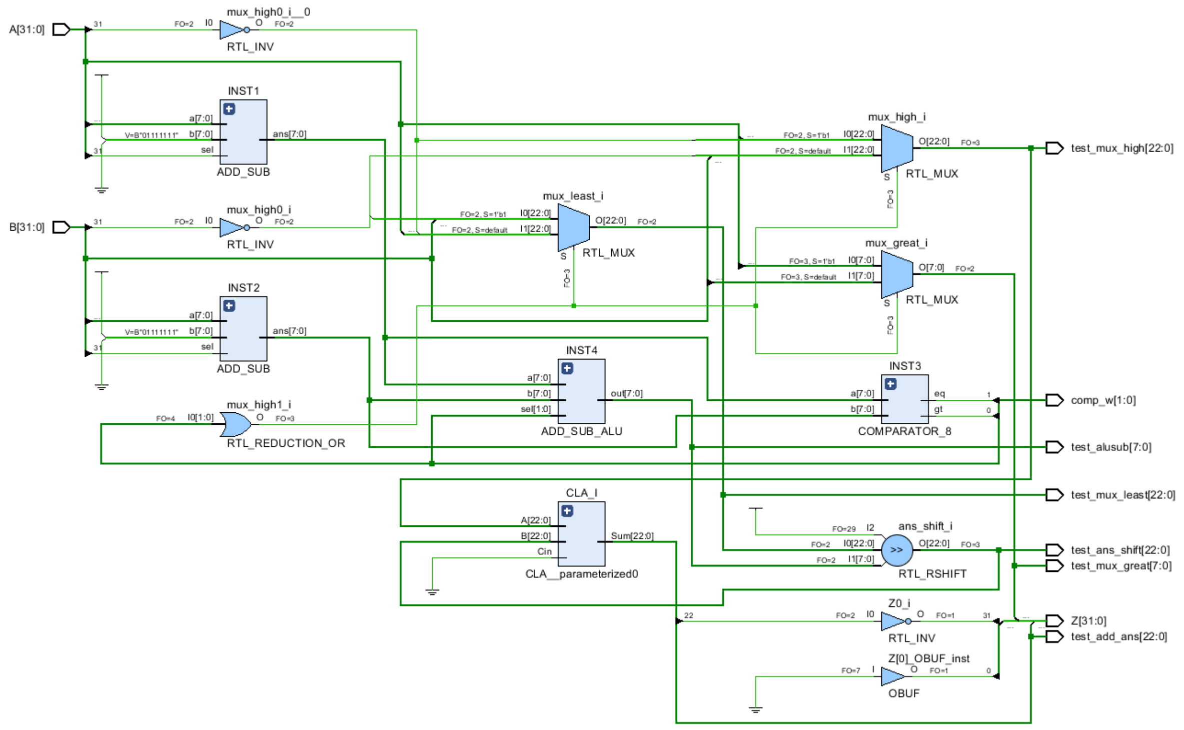
IEEE 754 Single-Precision Floating Point Adder

November 7, 2019

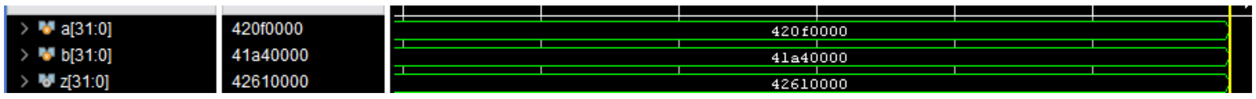
ARCHITECTURE DESIGN FROM SCRATCH



ELABORATED DESIGN

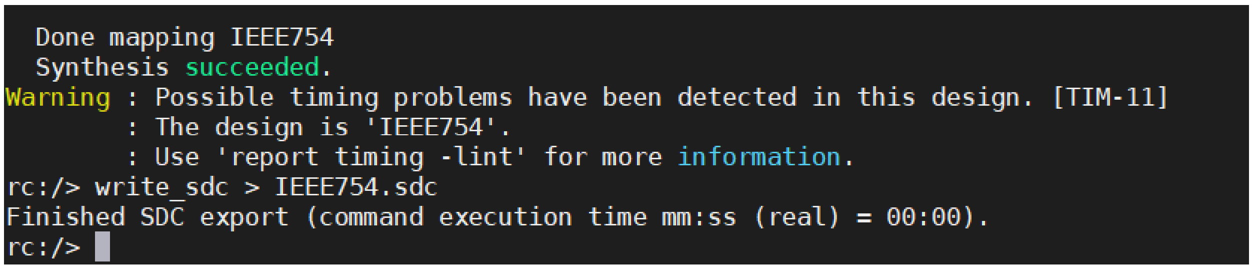


SIMULATION



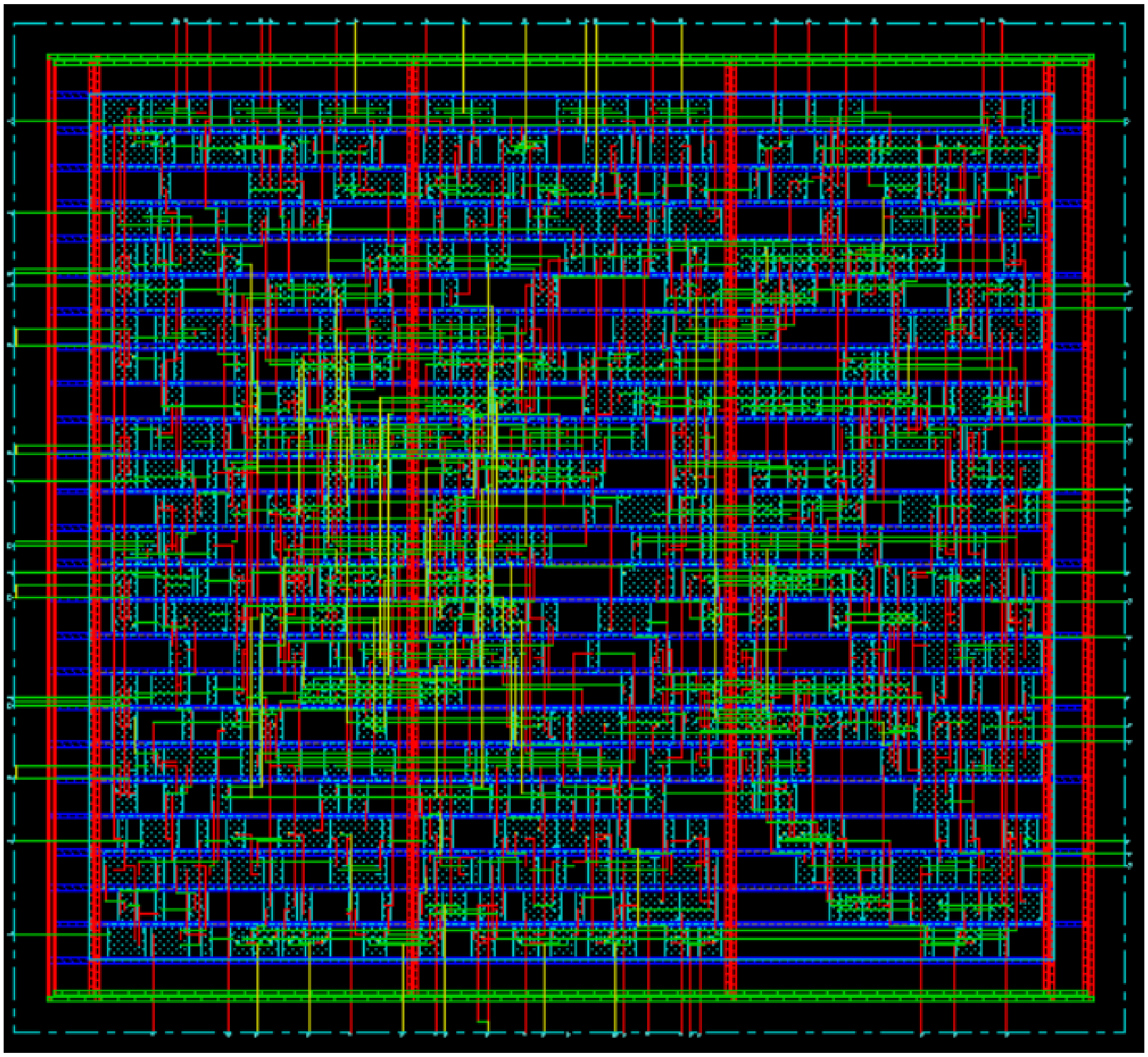
```
> a[31:0]    420f0000    420f0000
> b[31:0]    41a40000    41a40000
> z[31:0]    42610000    42610000
```

RTL COMPILER

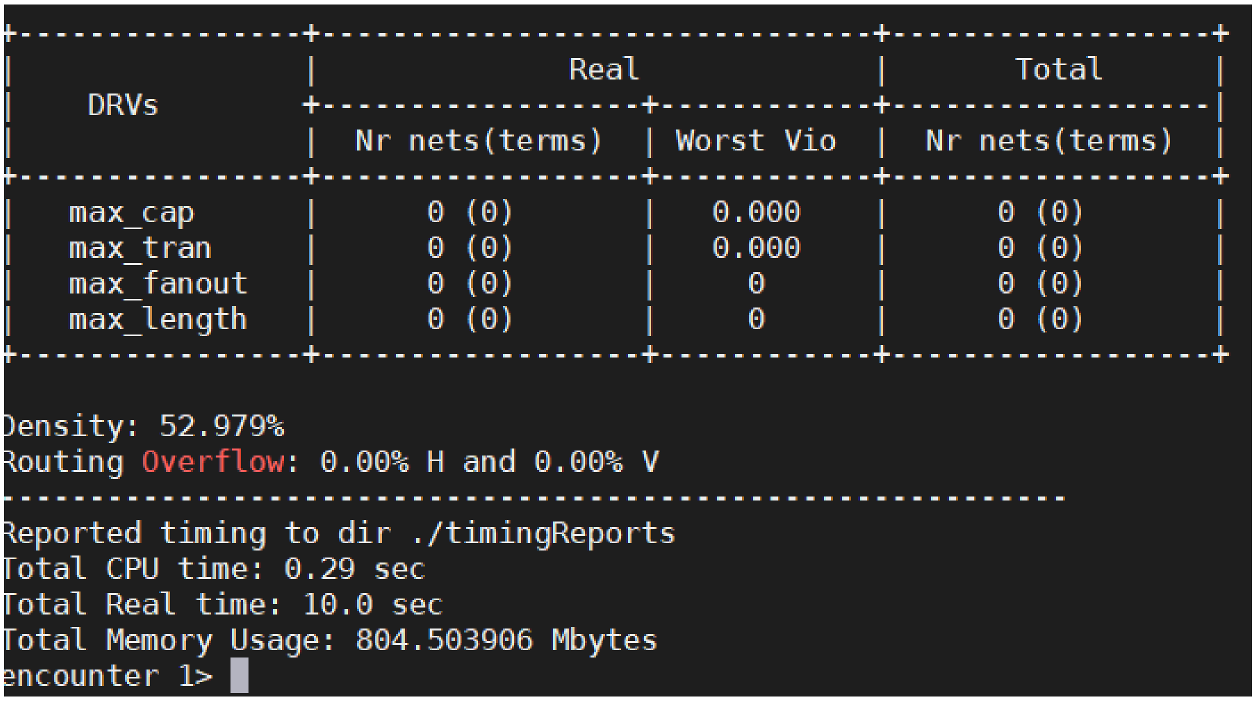


```
  Done mapping IEEE754
  Synthesis succeeded.
Warning : Possible timing problems have been detected in this design. [TIM-11]
        : The design is 'IEEE754'.
        : Use 'report timing -lint' for more information.
rc:/> write_sdc > IEEE754.sdc
Finished SDC export (command execution time mm:ss (real) = 00:00).
rc:/>
```

IC LAYOUT

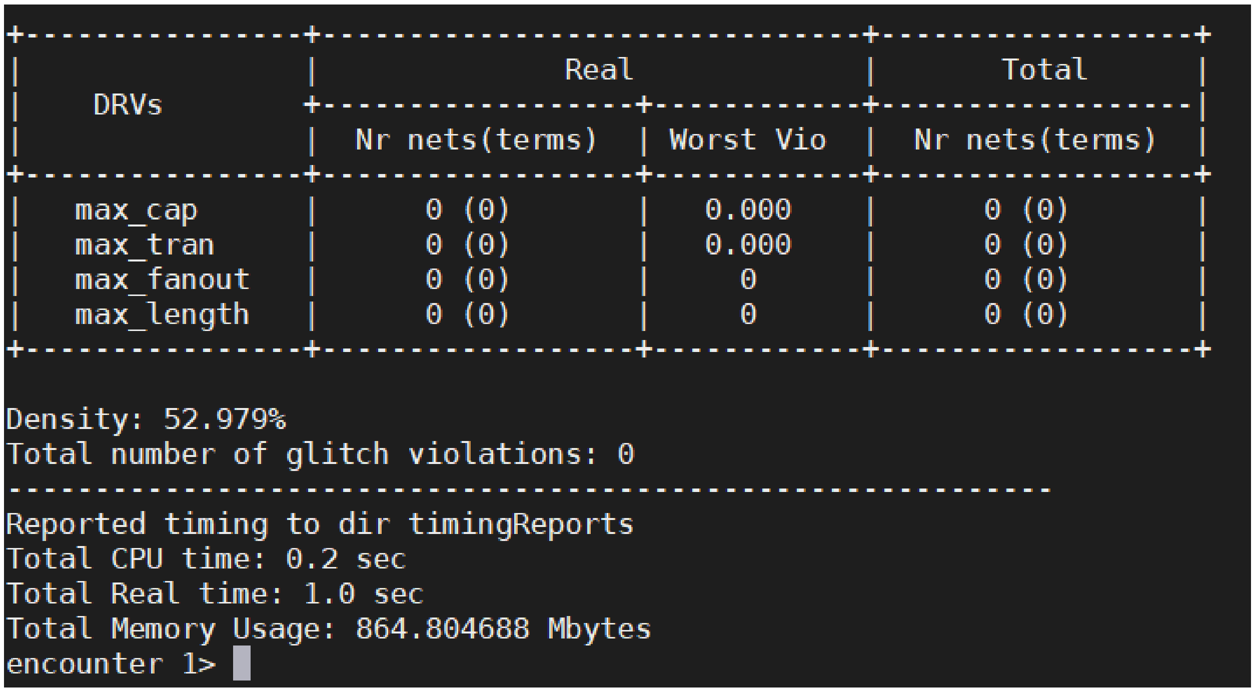


TIMING REPORT



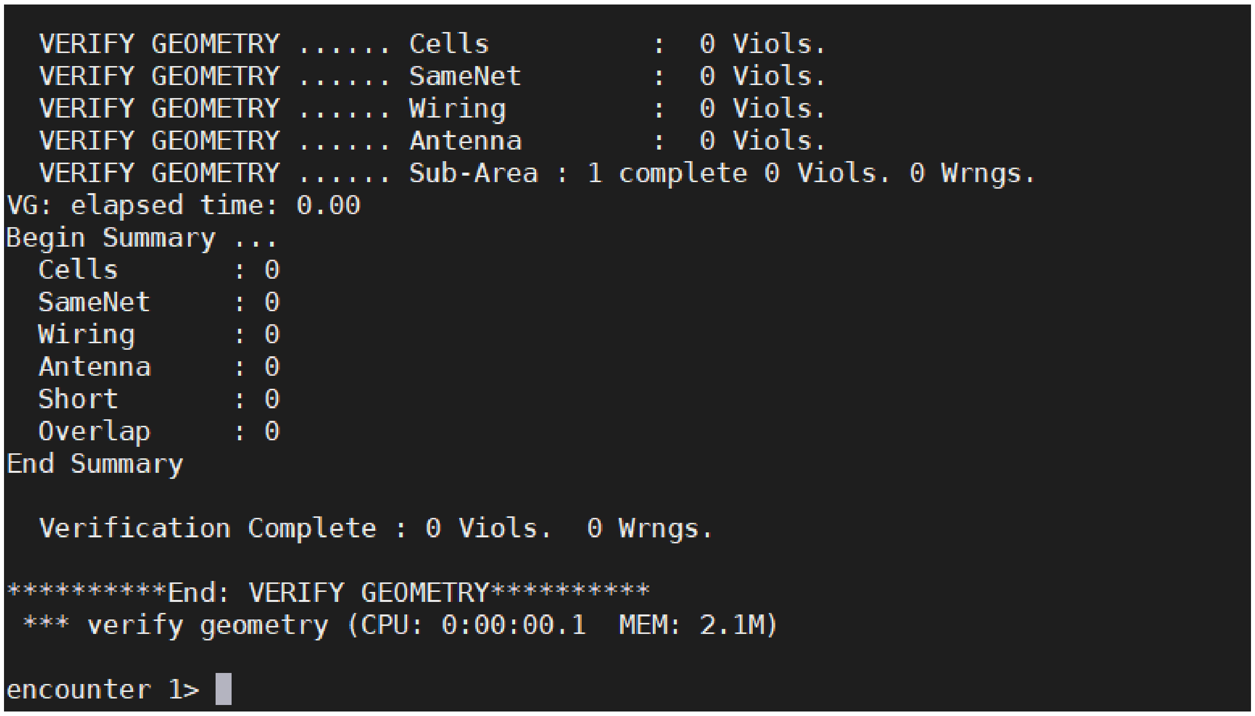
```
+----------------+-------------------------------+------------------+
|                |             Real              |      Total       |
|     DRVs       +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

ensity: 52.979%
outing Overflow: 0.00% H and 0.00% V
------------------------------------------------------------
eported timing to dir ./timingReports
otal CPU time: 0.29 sec
otal Real time: 10.0 sec
otal Memory Usage: 804.503906 Mbytes
ncounter 1>
```

POST ROUTE TIMING



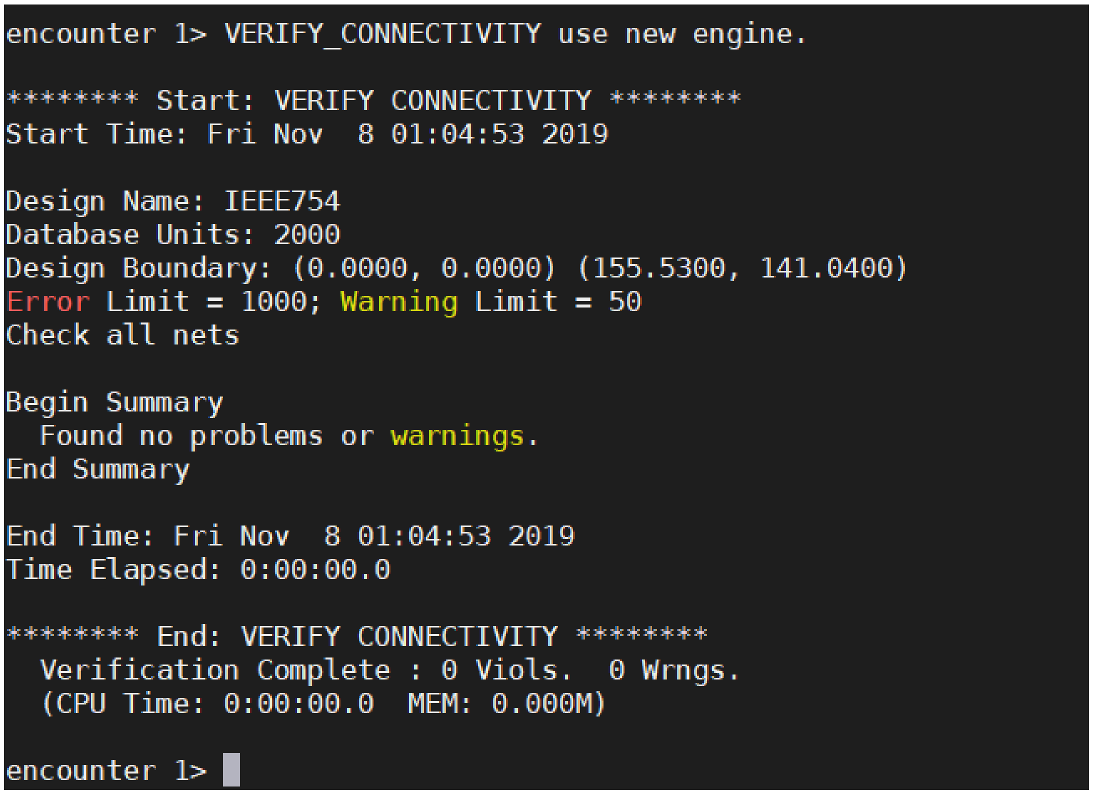
```
+----------------+------------------+------------+------------------+
|                |             Real              |      Total       |
|     DRVs       +------------------+------------+------------------|
|                |  Nr nets(terms)  | Worst Vio  |  Nr nets(terms)  |
+----------------+------------------+------------+------------------+
|   max_cap      |      0 (0)       |   0.000    |      0 (0)       |
|   max_tran     |      0 (0)       |   0.000    |      0 (0)       |
|   max_fanout   |      0 (0)       |     0      |      0 (0)       |
|   max_length   |      0 (0)       |     0      |      0 (0)       |
+----------------+------------------+------------+------------------+

Density: 52.979%
Total number of glitch violations: 0
------------------------------------------------------------
Reported timing to dir timingReports
Total CPU time: 0.2 sec
Total Real time: 1.0 sec
Total Memory Usage: 864.804688 Mbytes
encounter 1>
```

VERIFY GEOMETRY



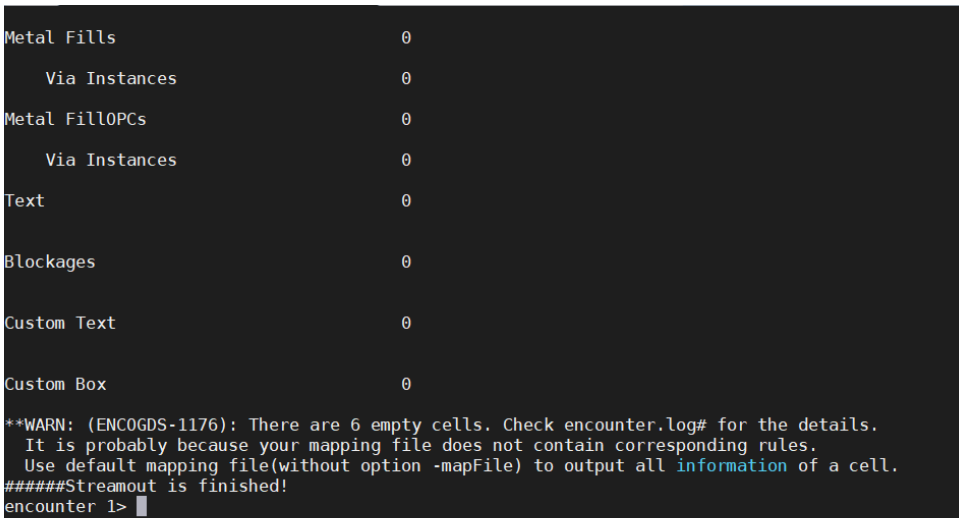
```
  VERIFY GEOMETRY ...... Cells          :  0 Viols.
  VERIFY GEOMETRY ...... SameNet        :  0 Viols.
  VERIFY GEOMETRY ...... Wiring         :  0 Viols.
  VERIFY GEOMETRY ...... Antenna        :  0 Viols.
  VERIFY GEOMETRY ...... Sub-Area : 1 complete 0 Viols. 0 Wrngs.
VG: elapsed time: 0.00
Begin Summary ...
  Cells       : 0
  SameNet     : 0
  Wiring      : 0
  Antenna     : 0
  Short       : 0
  Overlap     : 0
End Summary

  Verification Complete : 0 Viols.  0 Wrngs.

**********End: VERIFY GEOMETRY**********
 *** verify geometry (CPU: 0:00:00.1  MEM: 2.1M)

encounter 1>
```

VERIFY CONNECTIVITY



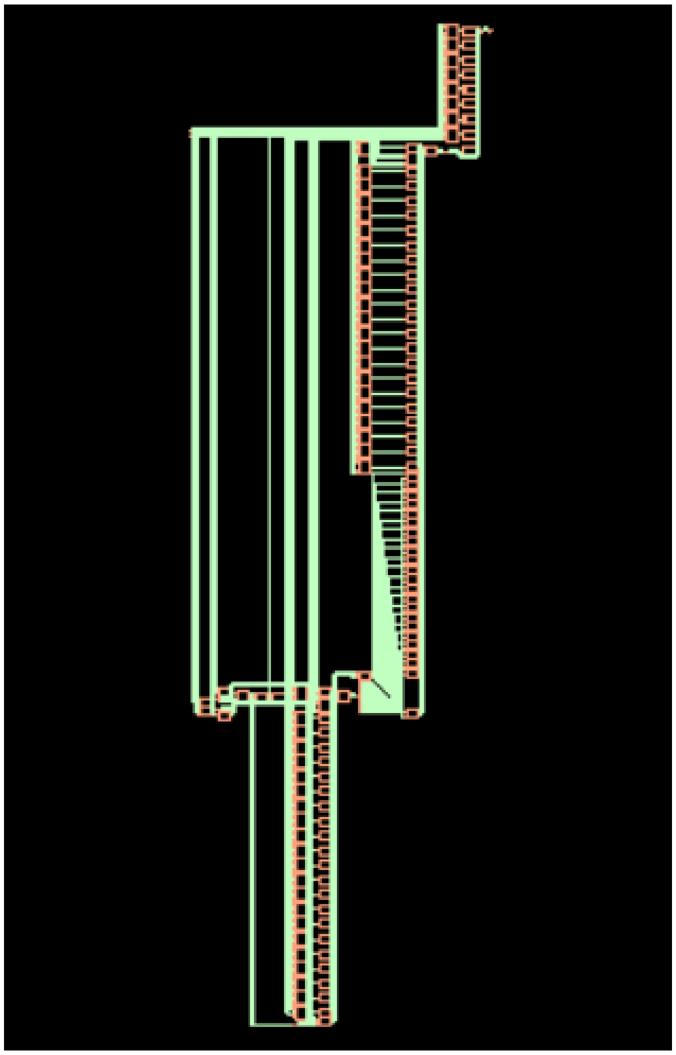
```
encounter 1> VERIFY_CONNECTIVITY use new engine.

******** Start: VERIFY CONNECTIVITY ********
Start Time: Fri Nov  8 01:04:53 2019

Design Name: IEEE754
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (155.5300, 141.0400)
Error Limit = 1000; Warning Limit = 50
Check all nets

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Fri Nov  8 01:04:53 2019
Time Elapsed: 0:00:00.0

******** End: VERIFY CONNECTIVITY ********
  Verification Complete : 0 Viols.  0 Wrngs.
  (CPU Time: 0:00:00.0  MEM: 0.000M)

encounter 1>
```

GDS FILE



```
Metal Fills                            0

    Via Instances                      0

Metal FillOPCs                         0

    Via Instances                      0

Text                                   0


Blockages                              0


Custom Text                            0


Custom Box                             0

**WARN: (ENCOGDS-1176): There are 6 empty cells. Check encounter.log# for the details.
  It is probably because your mapping file does not contain corresponding rules.
  Use default mapping file(without option -mapFile) to output all information of a cell.
######Streamout is finished!
encounter 1> 
```

SCHEMATIC VIEWER



VERILOG CODE

```
`timescale 1ns / 1ps
module IEEE754(
input [31:0] A,B,

output [31:0] Z,
output [1:0] comp_w,
output [7:0] test_alusub, test_mux_great,
output [22:0] test_mux_least, test_ans_shift, test_mux_high, test_add_ans
);

wire [7:0] x_exp, y_exp;

ADD_SUB INST1(
.a(A[30:23]),
.b(8'd127),
.sel(A[31]),

.ans(x_exp),
.carry()
);

ADD_SUB INST2(
.a(B[30:23]),
.b(8'd127),
.sel(B[31]),

.ans(y_exp),
.carry()
);
wire gt,lt,eq;
wire [1:0] comp;

COMPARATOR_8 INST3(
.a(x_exp),
.b(y_exp),
.gt(gt),

.lt(lt),

.eq(eq)

);


assign comp = {eq,gt};

assign comp_w = comp;


wire [7:0] addsubalu;

assign test_alusub = addsubalu;


ADD_SUB_ALU INST4(

.a(x_exp),

.b(y_exp),

.sel(comp),


.out(addsubalu)

);


wire [22:0] mux_least;

wire [22:0] ans_shift;


assign mux_least = comp ? {~B[31], B[22:1]} : {~A[31], A[22:1]};

assign test_mux_least = mux_least;


assign ans_shift = mux_least >> addsubalu;

assign test_ans_shift = ans_shift;


wire [22:0] mux_high;

assign mux_high = comp ? {~A[31], A[22:1]} : {~B[31], B[22:1]};

assign test_mux_high = mux_high;


wire [22:0] add_ans;

assign test_add_ans = add_ans;


CLA #(23) CLA_I(

.A(mux_high),
.B(ans_shift),
.Cin(1'b0),

.Sum(add_ans),
.Cout()
);

wire [7:0] mux_great;
assign mux_great = comp ? A[30:23] : B[30:23];
assign test_mux_great = mux_great;

assign Z = ({~add_ans[22], mux_great, {add_ans[21:0], 1'b0}});

endmodule

module COMPARATOR_8(
input [7:0] a, b,
output gt,lt,eq

);

wire [1:0] gt_in, lt_in, eq_in;

COMPARATOR_4 COMP_1(
.a(a[7:4]),
.b(b[7:4]),

.gt(gt_in[1]),
.lt(lt_in[1]),
.eq(eq_in[1])
);

COMPARATOR_4 COMP_2(
.a(a[3:0]),
.b(b[3:0]),

.gt(gt_in[0]),
.lt(lt_in[0]),
.eq(eq_in[0])
);

GLUE GLUE_1(
.gt_in(gt_in),
.lt_in(lt_in),
.eq_in(eq_in),

.gt_out(gt),
.lt_out(lt),
.eq_out(eq)
);

endmodule

module COMPARATOR_4(
    input [3:0] a, b,
    output gt,lt,eq
);

wire [1:0] gt_in, lt_in, eq_in;

COMPARATOR HI(
.a(a[3:2]),
.b(b[3:2]),

.gt(gt_in[1]),
.lt(lt_in[1]),
.eq(eq_in[1])
);

COMPARATOR LO(
.a(a[1:0]),
.b(b[1:0]),

.gt(gt_in[0]),
.lt(lt_in[0]),
.eq(eq_in[0])
);

GLUE GLUE_I(
.gt_in(gt_in),
.lt_in(lt_in),
.eq_in(eq_in),

.gt_out(gt),
.lt_out(lt),
.eq_out(eq)
);


endmodule

module GLUE (
    input [1:0] gt_in, lt_in, eq_in,
    output gt_out, lt_out, eq_out
);

wire [1:0] and_w;

//lvl 1
and(and_w[1], eq_in[1], gt_in[0]);
and(and_w[0], eq_in[1], lt_in[0]);

//lvl 2
or(gt_out, gt_in[1], and_w[1]);
or(lt_out, lt_in[1], and_w[0]);
and(eq_out, eq_in[1], eq_in[0]);

endmodule

module COMPARATOR(
input [1:0] a, b,
output gt, lt, eq

);


assign lt = ~a[1] & b[1] | ~a[1] & ~a[0] & b[0] | ~a[0] & b[1] & b[0];

assign gt = a[1] & ~b[1] | a[0] & ~b[1] & ~b[0] | a[1] & a[0] & ~b[0];

assign eq = ~a[1]&~a[0]&~b[1]&~b[0] | ~a[1] & a[0] & ~b[1] & b[0] | a[1] & a[0] & b[1] & b[0] | a[1] &
~a[0] & b[1] & ~b[0];



endmodule


//Wrapper for Carry-Lookahead-Logic and Partial Full Adders

module CLA #(parameter bits = 8) ( //parameter to specify number of bits for adder

input [bits-1:0] A, B, //augend and addend

input Cin,  //first carry in


output [bits-1:0] Sum, //sum output

output Cout //carry out bit


//output [bits-1:0]carries //for testbench

);


wire [bits-1:0]P_in, G_in; //wire bus for propagate and generate bits

wire [bits-1:0]carries; //wire bus for carry bits


assign Cout = carries[bits-1]; //propagate last carry out from msb of carries bus


//create partial full adder instance for the first bit to incorporate Cin

PFA PFA0(

.A(A[0]),

.B(B[0]),

.Cin(Cin),


.P(P_in[0]),

.G(G_in[0]),

.S(Sum[0])


);
```

```
genvar i;
generate //Instantiate N PFA's for bits
    for(i=1; i<bits; i=i+1) begin : PFAS
        PFA PFA_I(
            .A(A[i]),
            .B(B[i]),
            .Cin(carries[i-1]),

            .P(P_in[i]),
            .G(G_in[i]),
            .S(Sum[i])
        );
    end
endgenerate

//Instantiate Carry-Lookahead-Logic block
CLL #(.bits(bits)) CLL_INST(
.Cin(Cin),
.P(P_in),
.G(G_in),

.Cout(carries)
);

endmodule
//////////////////////////////////////////////////////////
//This module handles the carry outs for each partial full adder
//Acts as a wrapper to individual carry_gen modules
module CLL #(parameter bits = 8) (
    input Cin,
    input [bits-1: 0] P, G,
    output [bits-1: 0] Cout
);

//Create carry generator block for first bit
CARRY_GEN INST0(
        .G(G[0]),
        .P(P[0]),
        .Cin(Cin),

        .Cout(Cout[0])
);

genvar i;
generate //Create N bits of carry generators
    for(i=1; i < bits; i=i+1) begin : CARRIES
        CARRY_GEN INST_I(
        .G(G[i]),
        .P(P[i]),
        .Cin(Cout[i-1]),

        .Cout(Cout[i])
        );

    end
endgenerate
endmodule

/////////////////////////////////////////////////////////
module ADD_HALF(
    output cout,
    output sum,
    input a,
    input b
);
    xor(sum, a, b);
    and(cout, a, b);
endmodule
/////////////////////////////////////////////////////////
//Calculates the propagate, generate, and sum bits
module PFA(
    input A, B, Cin,
    output P, G, S
```

```
);

    assign P = A ^ B;   //xor gate for propagate
    assign G = A & B;   // and gate for generate
    assign S = Cin ^ P; // xor gate for sum

endmodule
//////////////////////////////////////////////////////////
module FULL_ADDER(
    input a,b,cin,
    output cout,sum

);

wire half1_cout, half1_sum, half2_cout;

ADD_HALF HALF1(
.cout(half1_cout),
.sum(half1_sum),
.a(a),
.b(b)
);

ADD_HALF HALF2(
.cout(half2_cout),
.sum(sum),
.a(cin),
.b(half1_sum)
);

//for the carry out of both half adders
or(cout, half2_cout, half1_cout);

endmodule
//////////////////////////////////////////////////////////
//Calculates the carry using generate, propagate, and carry in bits
module CARRY_GEN(
    input G, P, Cin,
    output Cout
);

assign Cout = G | P&Cin; //wire out carry calculation

endmodule
//////////////////////////////////////////////////////////

module ADD_SUB_ALU(
input [7:0] a,b,
input [1:0] sel,
output reg [7:0] out
);

//0 is B-A
//1 is A-B
//2 is pass A


always@(*) begin
    case(sel)
    2'b00: begin
        out = b-a;
    end
    2'b01: begin
        out = a-b;
    end
    2'b10: begin
        out = a;
    end
    2'b11: begin
        out = out;
    end
    endcase
end
```

```
/*
CLA #(8) CLA_I(
.A(mux_1),
.B(mux_2),
.Cin(1'b0),

.Sum(m_b),
.Cout(carry)
);
*/

endmodule

//if sel ==0  subtract A-B
//if sel ==1 add A+B
module ADD_SUB(
input [7:0] a, b,
input sel,

output [7:0] ans,
output carry

);
wire [7:0] add, sub;
wire [7:0] b_cla;

//(8'h7F)
assign b_cla = sel ? b : (~b + 1);



CLA #(8) CLA_I(
.A(a),
.B(b_cla),
.Cin(1'b0),

.Sum(ans),
.Cout(carry)
);
endmodule
```