**Oscar Alejandro Manzano**

**EE 4513 VLSI**

**Report Lab 9**

**October 29, 2019**

For this lab, I created a Multiply add fused with two stages, as seen in the schematic. After creating this and successfully getting the waveform, I was able to use the code for encounter. After that I completed the assignments requirements. In conclusion, this was fun and I didn’t have any problems.

Fall 2019

**EE 4513 – Introduction to VLSI Design**

**Lab Assignment 9**

1. Implement the Multiply-Add-Fused (MAF) Functional Unit shown in Figure 1

Pipe

Multiplier

Adder

A

B

C

Pipeline stage registers

1

0

Result

3

2

3

2

3

2

Stage

1

Pipe

Stage

2

mult\_ou

t

mult\_out\_reg

c\_reg

func

f

unc\_reg

2

Figure 1: Fused Mult-Add Functional Unit

Table 1: Control signal for function and computation in the pipe stages

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Pipe Stage 1 | Pipe Stage 2 | Desired Output |
| 00 | A × B | (A × B) + 0 | A × B |
| 01 | A × 1 | (A × 1) + C | A + C |
| 10 | A × B | (A × B) + C | (A × B) + C |
| 11 | Don’t Care | Don’t Care | Don’t Care |

1. Once you have completed the behavioral simulation (for example, using Xilinx

Vivado), synthesize your code to generate the netlist using Cadence RTL Compiler (RC).

1. Then, use the Cadence Encounter platform to generate the schematic and 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 MAF unit
  2. Verilog Code and Simulation waveforms for your MAF unit

`timescale 1ns / 1ps

module MAF(

input clk,

input [31:0] A, B,C,

input [1:0] func,

output [63:0] Result

);

wire [63:0] mult\_out\_reg;

Pipeline1 P1(clk,A, B, func, mult\_out\_reg);

Pipeline2 P2(clk,C, mult\_out\_reg, func, Result);

endmodule

module Pipeline1(

input clk,

input [31:0] A,

input [31:0] B,

input [1:0] func,

output [63:0] mult\_out

);

reg [31:0] mux\_w;

always @(\*) begin

case(func)

2'b00: mux\_w <= B;

2'b01: mux\_w <= 32'h1;

2'b10: mux\_w <= B;

2'b11: mux\_w <= 32'h1;

endcase

end

assign mult\_out = A \* mux\_w;

endmodule

module Pipeline2(

input clk,

input [31:0] c\_reg,

input [63:0] mult\_out\_reg,

input [1:0] func\_reg,

output reg [63:0] Result

);

reg [31:0] mux\_w;

always @(\*) begin

case(func\_reg)

2'b00: mux\_w <= 0;

2'b01: mux\_w <= c\_reg;

2'b10: mux\_w <= c\_reg;

2'b11: mux\_w <= 0;

endcase

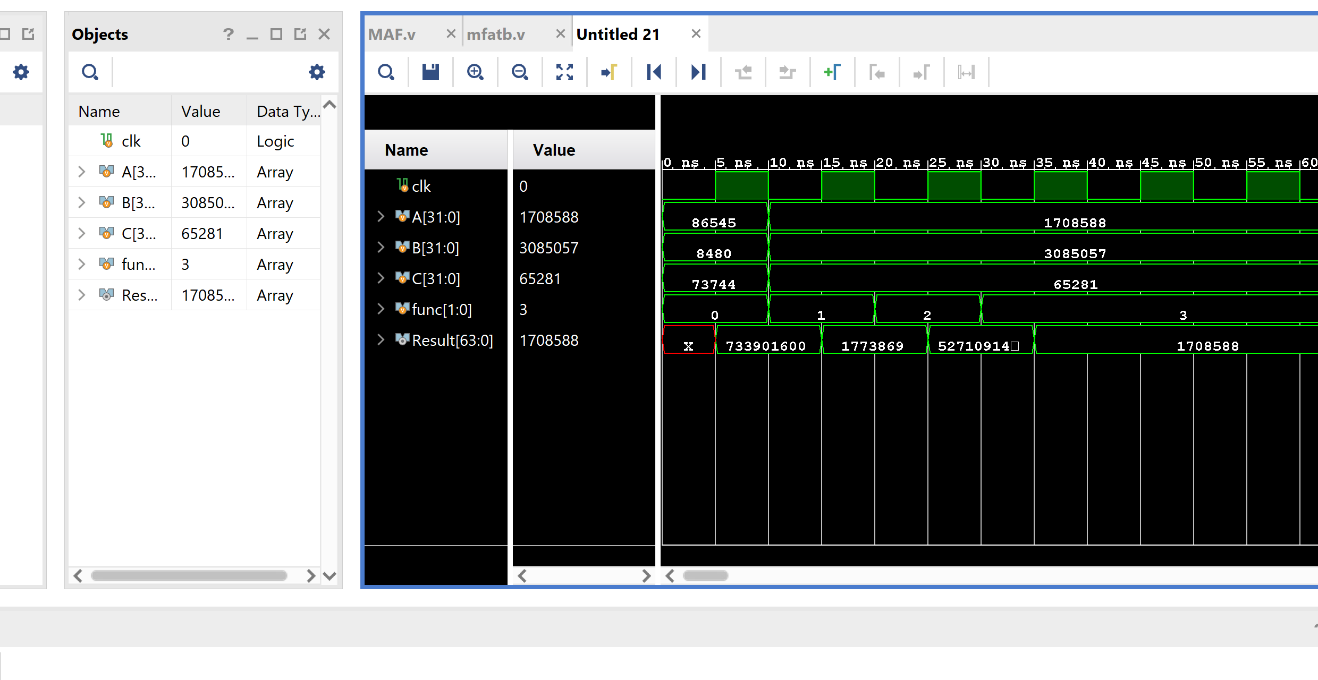
end

always @(posedge clk) begin

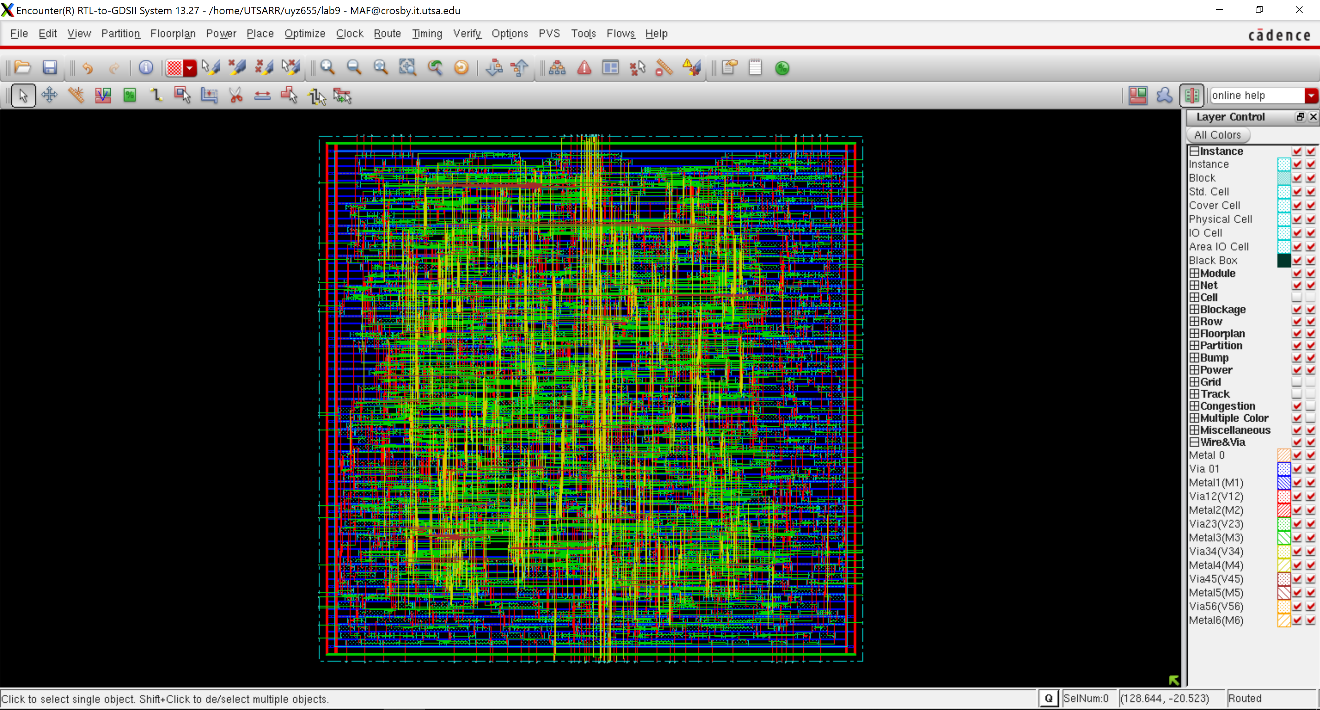
Result <= mult\_out\_reg + mux\_w;

end

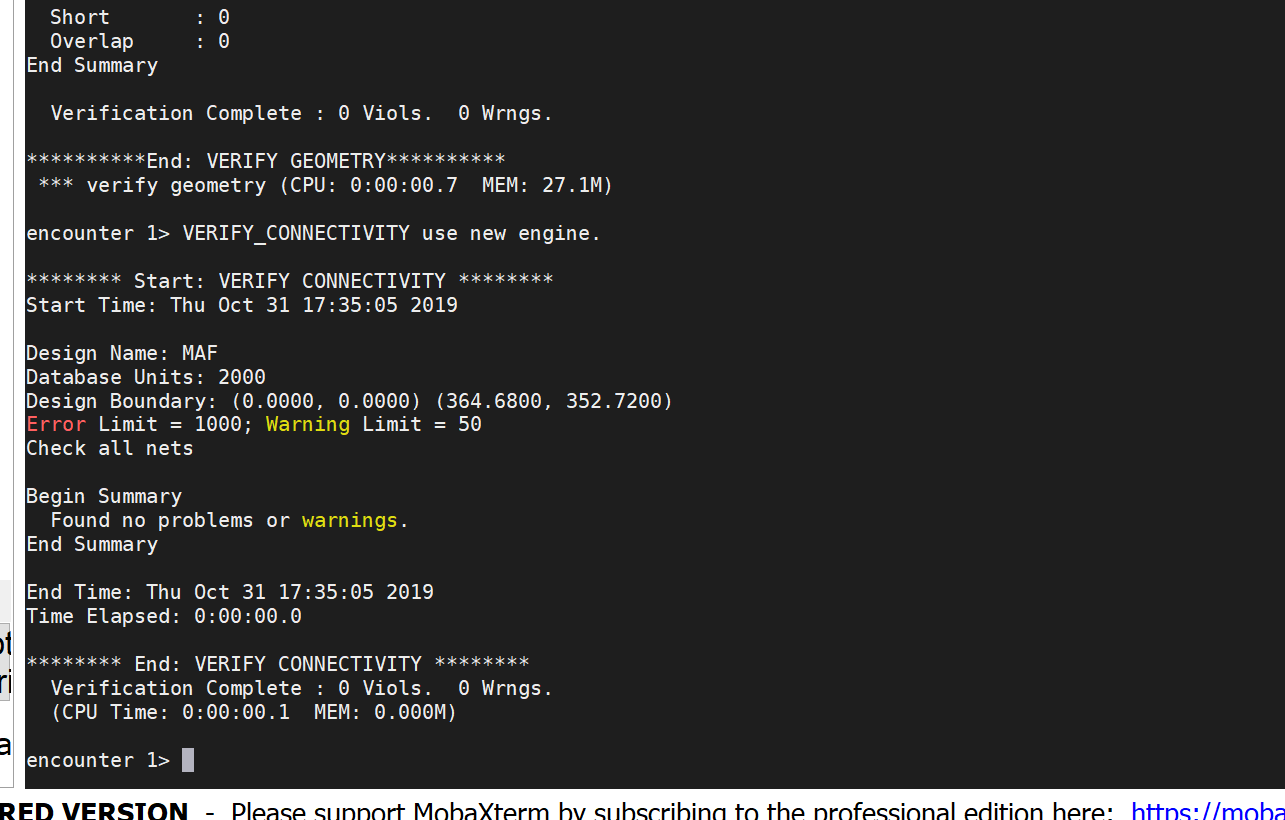
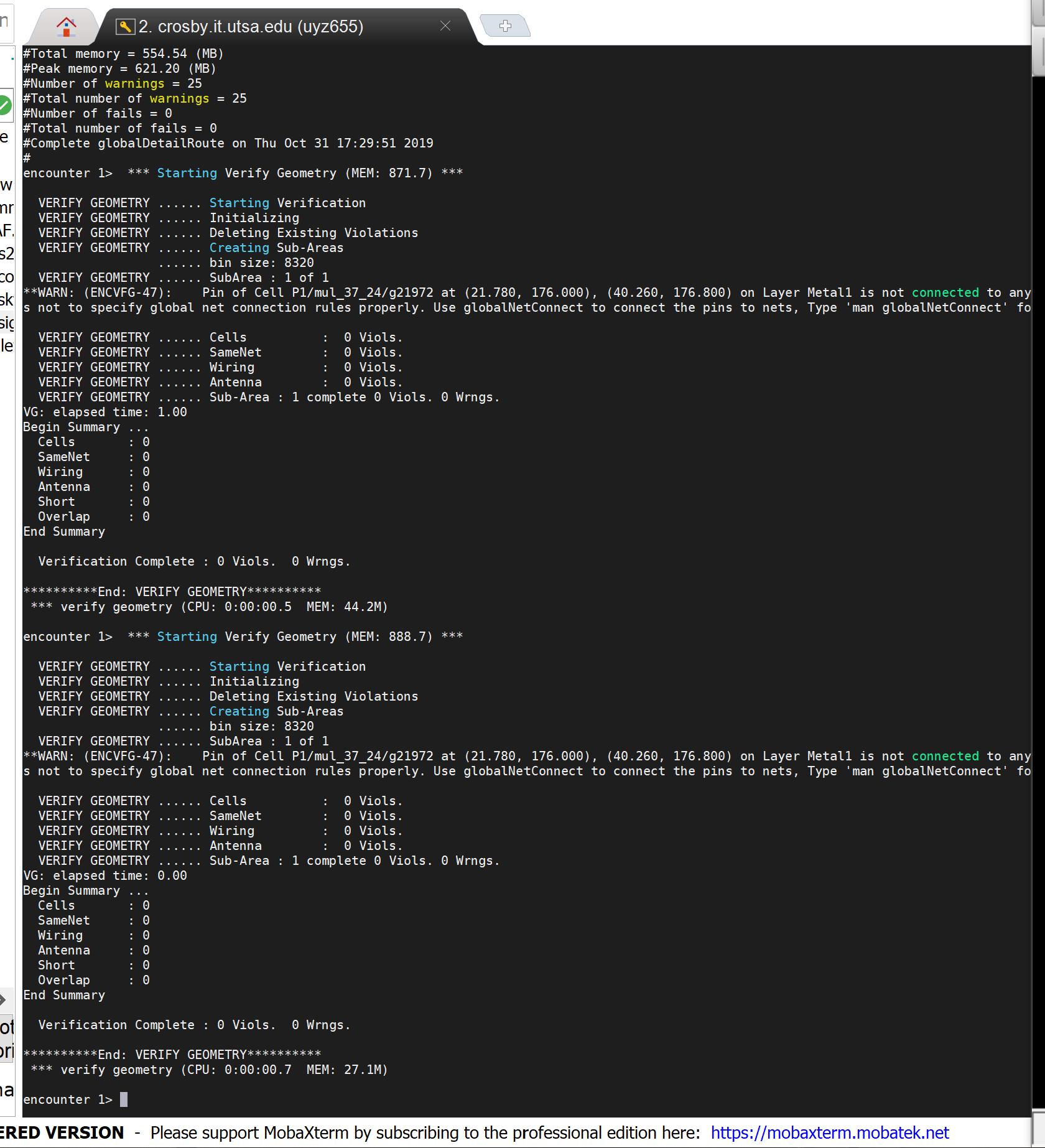
endmodule



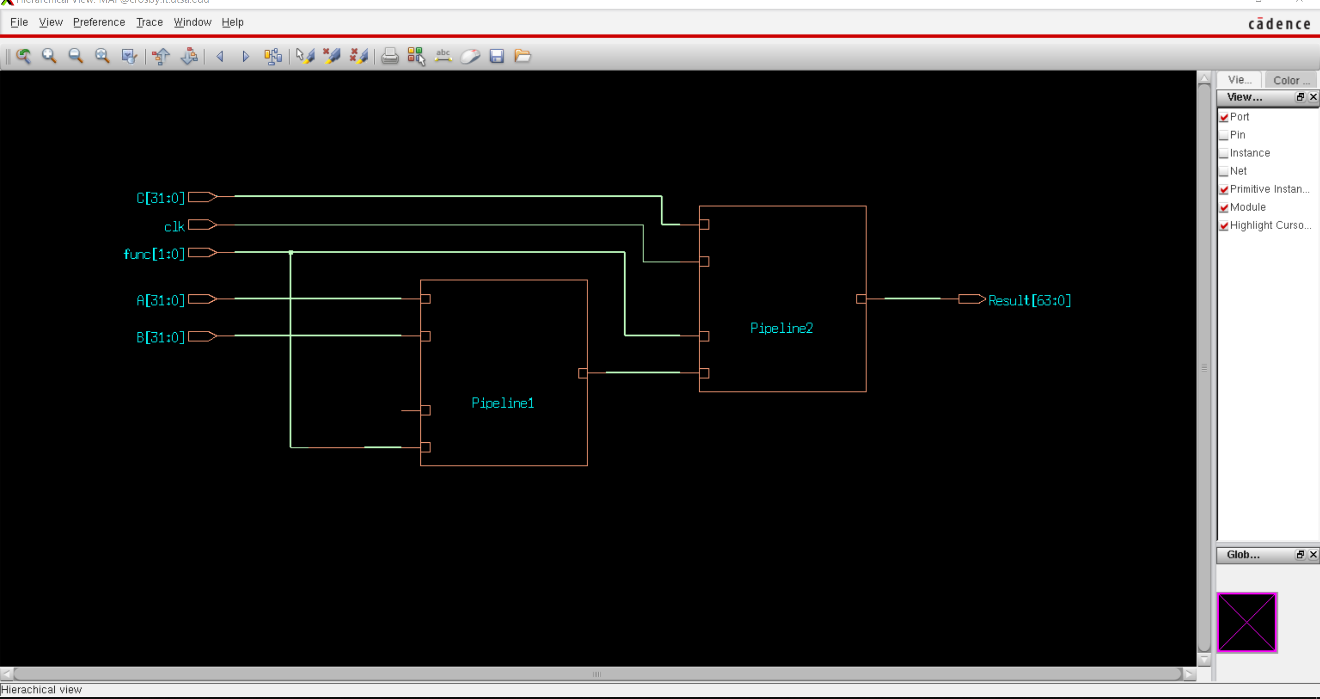
* 1. Physical Layout snapshot of your design



* 1. Snapshots of portions of the connectivity and geometry verification reports, showing any violations, if present.



* 1. Schematic of your design (can be seen by clicking Tools -> Schematic Viewer)



* 1. Report Your Timing Analysis (Slew, Delay, Arrival...etc) and Total Area from your RTL compiler for your design.

============================================================

Generated by: Encounter(R) RTL Compiler RC13.12 - v13.10-s021\_1

Generated on: Oct 31 2019 05:21:44 pm

Module: MAF

Technology library: tsmc18 1.0

Operating conditions: slow (balanced\_tree)

Wireload mode: enclosed

Area mode: timing library

============================================================

Pin Type Fanout Load Slew Delay Arrival

(fF) (ps) (ps) (ps)

-------------------------------------------------------------------

func[0] in port 9 26.2 0 +0 0 F

P1/func[0]

g508/A +0 0

g508/Y INVX1 25 55.3 705 +395 395 R

g500/A +0 395

g500/Y AND2X1 43 101.2 1268 +919 1314 R

mul\_37\_24/B[5]

g23150/A +0 1314

g23150/Y INVX1 38 102.6 780 +708 2022 F

g23056/B0 +0 2022

g23056/Y OAI33XL 32 100.2 3821 +2286 4307 R

g22902/C0 +0 4307

g22902/Y AOI222XL 1 6.1 824 +544 4852 F

g22739/S0 +0 4852

g22739/Y MXI2XL 1 4.6 340 +349 5201 F

g22679/A +0 5201

g22679/S ADDHXL 1 6.9 234 +352 5553 F

g22512/A +0 5553

g22512/CO ADDFX2 1 6.2 145 +522 6075 F

g22468/CI +0 6075

g22468/CO ADDFX2 1 2.7 135 +335 6410 F

g23295/ICI +0 6410

g23295/S CMPR42X1 3 7.7 133 +454 6864 F

g22316/A0N +0 6864

g22316/Y AOI2BB1XL 1 3.2 110 +253 7117 F

g22302/B0 +0 7117

g22302/Y AOI21XL 3 8.8 358 +229 7347 R

g22276/B0 +0 7347

g22276/Y AOI2BB1XL 1 3.2 110 +97 7444 F

g22259/B0 +0 7444

g22259/Y AOI21XL 3 8.8 358 +229 7674 R

g22235/B0 +0 7674

g22235/Y AOI2BB1XL 1 3.2 110 +97 7771 F

g22221/B0 +0 7771

g22221/Y AOI21XL 3 8.8 358 +229 8000 R

g22191/B0 +0 8000

g22191/Y AOI2BB1XL 1 3.2 110 +97 8098 F

g22175/B0 +0 8098

g22175/Y AOI21XL 3 8.8 358 +229 8327 R

g22100/B0 +0 8327

g22100/Y AOI2BB1XL 1 3.2 110 +97 8425 F

g22086/B0 +0 8425

g22086/Y AOI21XL 3 8.8 349 +229 8654 R

g22053/B0 +0 8654

g22053/Y AOI2BB1XL 1 3.2 109 +97 8751 F

g22039/B0 +0 8751

g22039/Y AOI21XL 3 8.8 349 +229 8980 R

g22012/B0 +0 8980

g22012/Y AOI2BB1XL 1 3.2 109 +97 9076 F

g21997/B0 +0 9076

g21997/Y AOI21XL 3 8.8 349 +229 9305 R

g21961/B0 +0 9305

g21961/Y AOI2BB1XL 1 3.2 109 +97 9402 F

g21947/B0 +0 9402

g21947/Y AOI21XL 3 8.8 349 +229 9631 R

g21912/B0 +0 9631

g21912/Y AOI2BB1XL 1 3.2 109 +97 9728 F

g21897/B0 +0 9728

g21897/Y AOI21XL 3 8.8 349 +229 9957 R

g21869/B0 +0 9957

g21869/Y AOI2BB1XL 1 3.2 109 +97 10053 F

g21847/B0 +0 10053

g21847/Y AOI21XL 3 8.8 349 +229 10282 R

g21813/B0 +0 10282

g21813/Y AOI2BB1XL 1 3.2 109 +97 10379 F

g21797/B0 +0 10379

g21797/Y AOI21XL 3 8.8 349 +229 10608 R

g21765/B0 +0 10608

g21765/Y AOI2BB1XL 1 3.2 109 +97 10705 F

g21751/B0 +0 10705

g21751/Y AOI21XL 3 8.8 349 +229 10934 R

g21722/B0 +0 10934

g21722/Y AOI2BB1XL 1 3.2 109 +97 11030 F

g21704/B0 +0 11030

g21704/Y AOI21XL 3 8.8 349 +229 11260 R

g21666/B0 +0 11260

g21666/Y AOI2BB1XL 1 3.2 109 +97 11356 F

g21648/B0 +0 11356

g21648/Y AOI21XL 3 8.8 349 +229 11585 R

g21617/B0 +0 11585

g21617/Y AOI2BB1XL 1 3.2 109 +97 11682 F

g21600/B0 +0 11682

g21600/Y AOI21XL 3 8.8 349 +229 11911 R

g21574/B0 +0 11911

g21574/Y AOI2BB1XL 1 3.2 109 +97 12008 F

g21552/B0 +0 12008

g21552/Y AOI21XL 3 8.8 349 +229 12237 R

g21519/B0 +0 12237

g21519/Y AOI2BB1XL 1 3.2 109 +97 12333 F

g21500/B0 +0 12333

g21500/Y AOI21XL 3 8.8 349 +229 12562 R

g21471/B0 +0 12562

g21471/Y AOI2BB1XL 1 3.2 109 +97 12659 F

g21453/B0 +0 12659

g21453/Y AOI21XL 3 8.8 349 +229 12888 R

g21430/B0 +0 12888

g21430/Y AOI2BB1XL 1 3.2 109 +97 12985 F

g21414/B0 +0 12985

g21414/Y AOI21XL 3 8.8 349 +229 13214 R

g21376/B0 +0 13214

g21376/Y AOI2BB1XL 1 3.2 109 +97 13310 F

g21364/B0 +0 13310

g21364/Y AOI21XL 3 8.8 349 +229 13540 R

g21320/B0 +0 13540

g21320/Y AOI2BB1XL 1 3.2 109 +97 13636 F

g21296/B0 +0 13636

g21296/Y AOI21XL 1 2.7 192 +139 13775 R

g23209/ICI +0 13775

g23209/CO CMPR42X1 1 2.7 100 +282 14057 R

g23207/ICI +0 14057

g23207/CO CMPR42X1 1 2.7 100 +261 14318 R

g23205/ICI +0 14318

g23205/CO CMPR42X1 1 6.2 140 +288 14606 R

g21251/CI +0 14606

g21251/CO ADDFX2 1 3.5 109 +287 14893 R

g21250/A +0 14893

g21250/Y INVX1 1 2.7 50 +46 14939 F

g23203/ICI +0 14939

g23203/CO CMPR42X1 1 2.7 100 +287 15226 F

g23202/ICI +0 15226

g23202/CO CMPR42X1 1 2.7 100 +301 15527 F

g23201/ICI +0 15527

g23201/CO CMPR42X1 1 2.7 100 +301 15828 F

g23200/ICI +0 15828

g23200/CO CMPR42X1 3 9.1 144 +345 16173 F

g21243/A0 +0 16173

g21243/Y AOI22XL 1 2.7 246 +156 16330 R

g23199/ICI +0 16330

g23199/CO CMPR42X1 1 6.2 140 +320 16650 R

g21240/CI +0 16650

g21240/CO ADDFX2 1 3.5 109 +287 16937 R

g21239/A +0 16937

g21239/Y INVX1 1 2.7 50 +46 16983 F

g23198/ICI +0 16983

g23198/CO CMPR42X1 1 2.7 100 +287 17270 F

g23197/ICI +0 17270

g23197/CO CMPR42X1 1 2.7 100 +301 17571 F

g23196/ICI +0 17571

g23196/CO CMPR42X1 1 2.7 100 +301 17872 F

g23195/ICI +0 17872

g23195/CO CMPR42X1 1 2.7 100 +301 18173 F

g23194/ICI +0 18173

g23194/CO CMPR42X1 1 2.7 100 +301 18474 F

g23193/ICI +0 18474

g23193/CO CMPR42X1 1 2.7 100 +301 18775 F

g23192/ICI +0 18775

g23192/CO CMPR42X1 1 2.7 100 +301 19076 F

g23191/ICI +0 19076

g23191/CO CMPR42X1 1 2.7 100 +301 19376 F

g23190/ICI +0 19376

g23190/CO CMPR42X1 1 2.7 100 +301 19677 F

g23189/ICI +0 19677

g23189/CO CMPR42X1 1 2.7 100 +301 19978 F

g23188/ICI +0 19978

g23188/CO CMPR42X1 1 2.7 100 +301 20279 F

g23187/ICI +0 20279

g23187/CO CMPR42X1 1 2.7 100 +301 20580 F

g23186/ICI +0 20580

g23186/CO CMPR42X1 1 2.7 100 +301 20881 F

g23185/ICI +0 20881

g23185/CO CMPR42X1 1 2.7 100 +301 21182 F

g23184/ICI +0 21182

g23184/CO CMPR42X1 1 2.7 100 +301 21483 F

g23183/ICI +0 21483

g23183/CO CMPR42X1 1 2.7 100 +301 21784 F

g23182/ICI +0 21784

g23182/CO CMPR42X1 1 2.7 100 +301 22084 F

g23181/ICI +0 22084

g23181/CO CMPR42X1 1 2.7 100 +301 22385 F

g23180/ICI +0 22385

g23180/CO CMPR42X1 1 6.2 124 +325 22710 F

g21218/CI +0 22710

g21218/CO ADDFX2 1 6.2 145 +343 23054 F

g21217/CI +0 23054

g21217/CO ADDFX2 1 6.2 145 +349 23402 F

g21216/CI +0 23402

g21216/CO ADDFX2 2 5.7 144 +347 23749 F

g21215/A1N +0 23749

g21215/Y OAI2BB2XL 2 5.2 143 +224 23973 F

mul\_37\_24/Z[63]

P1/mult\_out[63]

P2/mult\_out\_reg[63]

add\_63\_27/A[63]

g949/A0N +0 23973

g949/Y OAI2BB2XL 1 2.1 110 +234 24207 F

add\_63\_27/Z[63]

Result\_reg[63]/D DFFHQX1 +0 24207

Result\_reg[63]/CK setup 0 +340 24547 R

-------------------------------------------------------------------

Timing slack : UNCONSTRAINED

Start-point : func[0]

End-point : P2/Result\_reg[63]/D

============================================================

Generated by: Encounter(R) RTL Compiler RC13.12 - v13.10-s021\_1

Generated on: Oct 31 2019 05:21:44 pm

Module: MAF

Technology library: tsmc18 1.0

Operating conditions: slow (balanced\_tree)

Wireload mode: enclosed

Area mode: timing library

============================================================

Instance Cells Cell Area Net Area Total Area Wireload

--------------------------------------------------------------------

MAF 2060 57274 0 57274 <none> (D)

P1 1868 50056 0 50056 <none> (D)

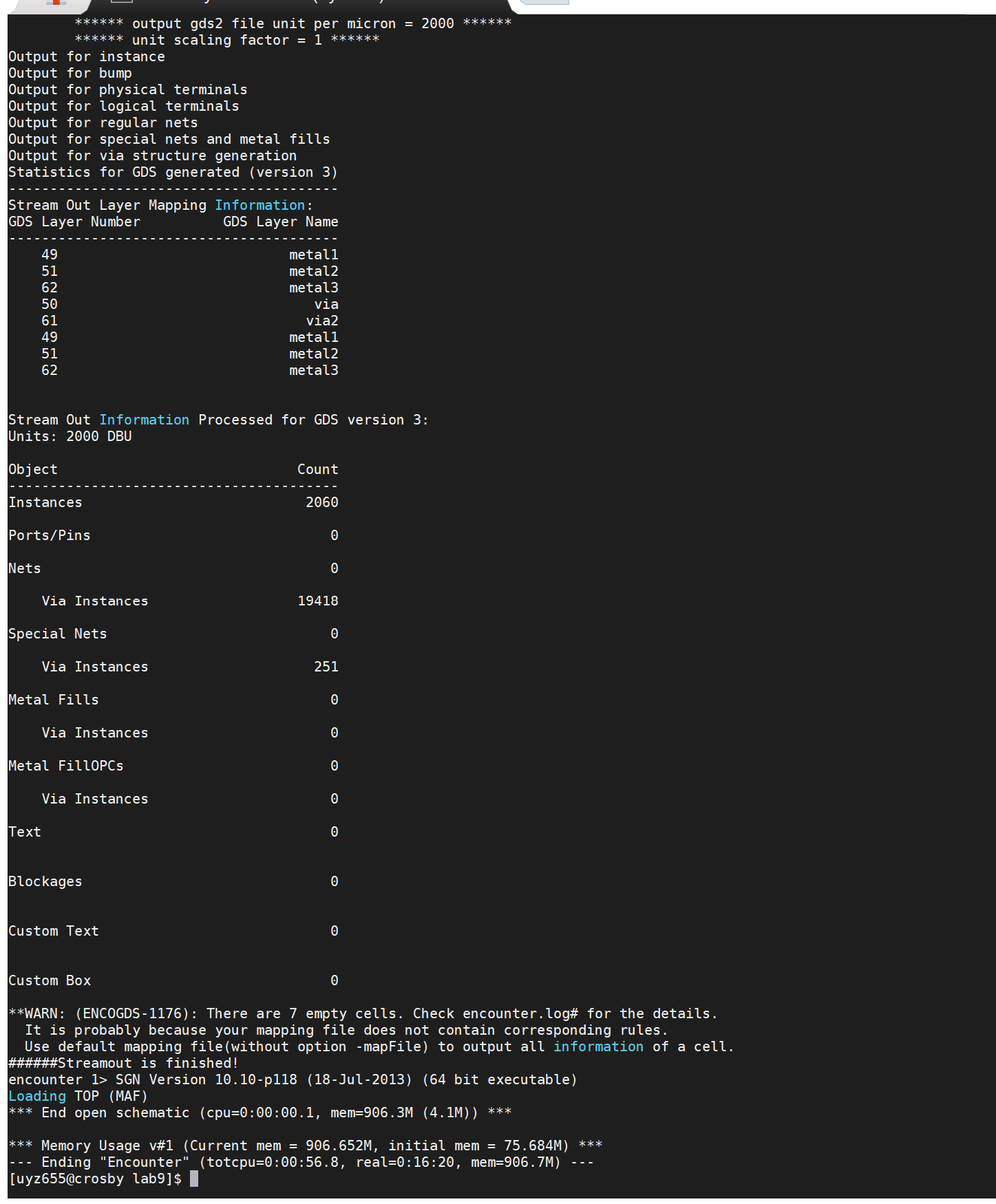
mul\_37\_24 1835 49623 0 49623 <none> (D)

P2 192 7218 0 7218 <none> (D)

add\_63\_27 95 3360 0 3360 <none> (D)

(D) = wireload is default in technology library

* 1. Snapshot of the message showing successful generation of GDS2 layout



* 1. Try to optimize your design in the Verilog (efficient /structured code) and post a conclusion for your design.

**Note**: -

Once you have logged in, you can make a new directory called **lab9hw** using the **mkdir** command (**mkdir lab9hw**) in the terminal. Then change over to the created directory using **cd lab9hw.** Inside this directory, make a new directory called “**maf\_unit**” using the **mkdir** command. Use this directory to create the layout of the Verilog module required in this assignment. Provide the path of this directory in the report.