EECT/CE 6325 VLSI DESIGN

Final Project Milestones 5

Name: Yuyang Hsieh

UTD ID: 2021775024

Partner: Xiongtao Zhang

1. Performance (based on operating the provided test pattern and generating expected output values)

Minimum Clock Period TMIN	Total Energy for Operating 8 Test Patterns, E	Layout Area, A	AEDP (EDP× A) (pJ.ns.um2)
0.67ns	12.5pJ	1908.48778um2	15,983

2. Description

The final design (4-bit calculation) meets all the requirements, verified by HSPICE simulation result with expected output values. All the subcircuits and top level of the design passed DRC and LVS with 0 error.

3. Overall Design

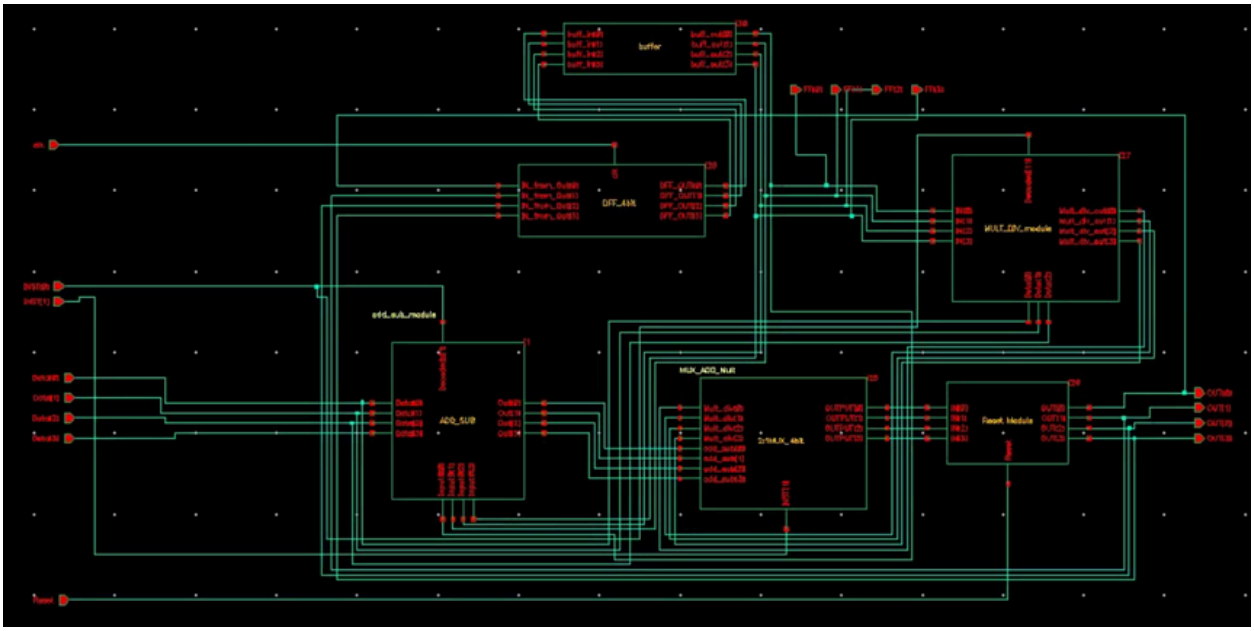


Figure 1.1 schematic

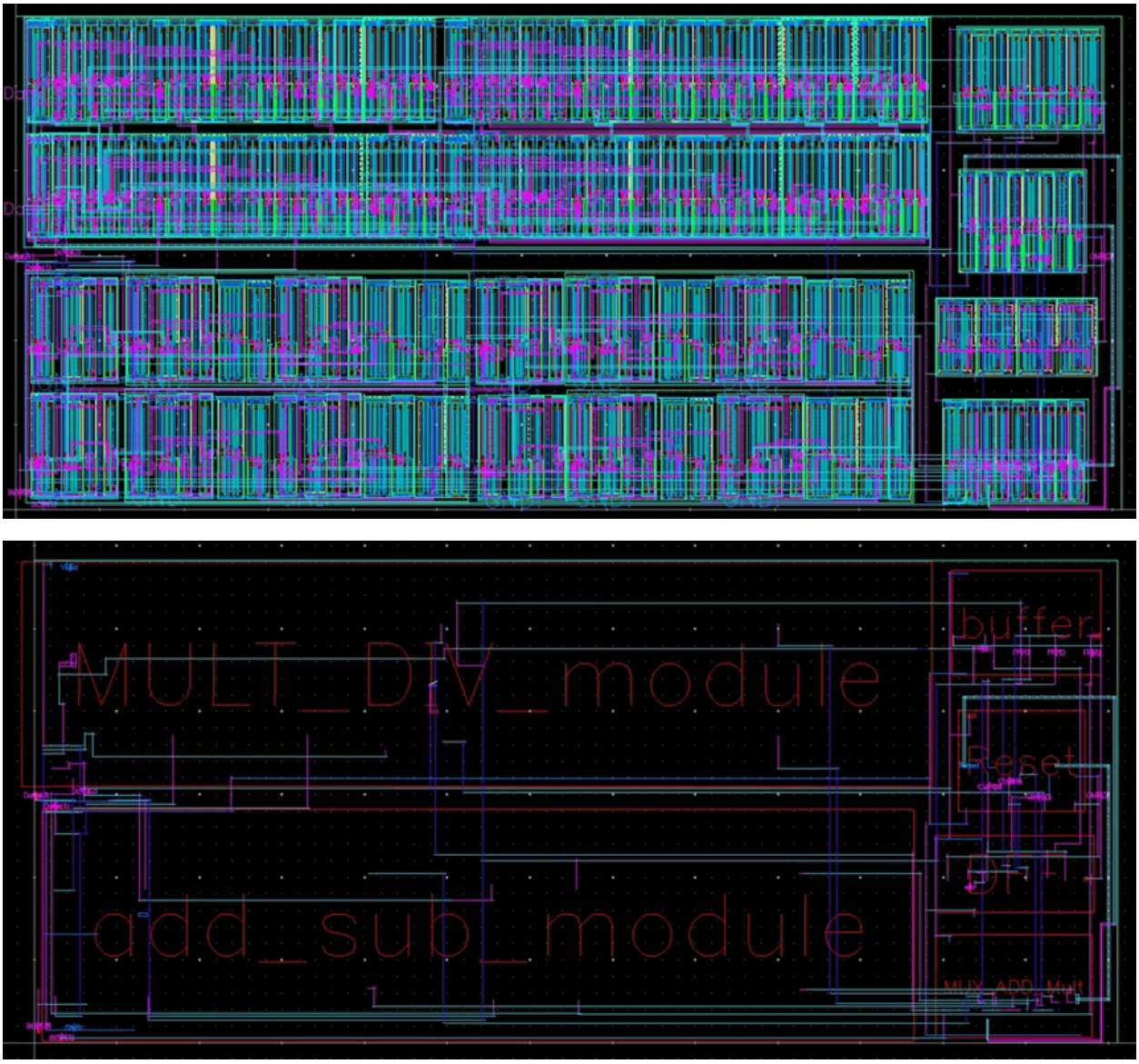


Figure 1.2 layout and floorplan

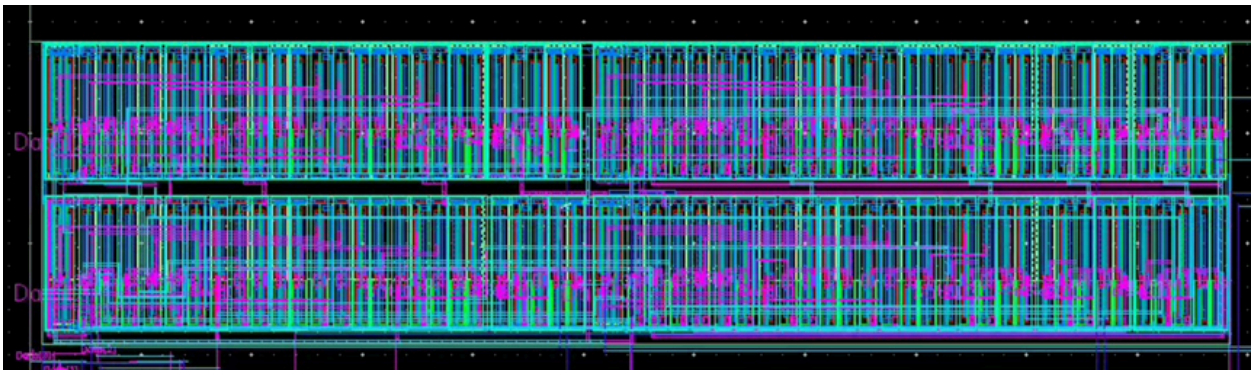


Figure 1.3 Mult_Div_module

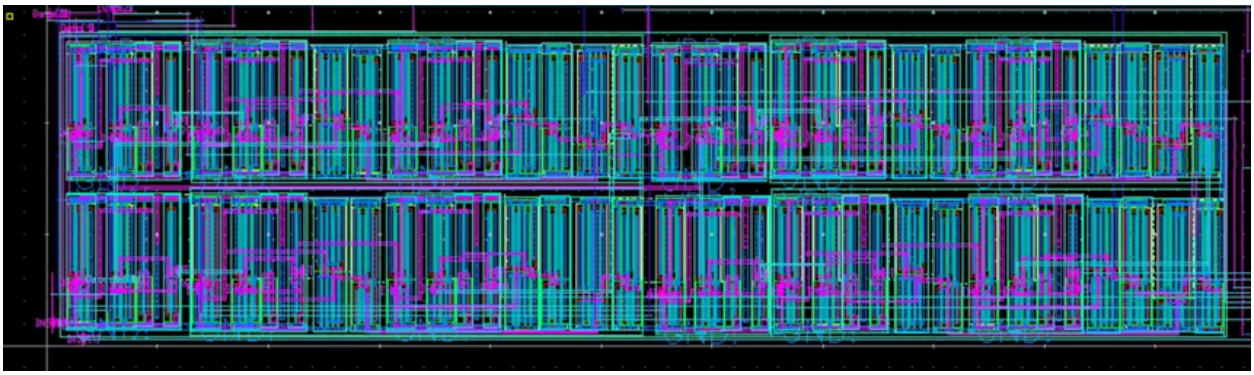


Figure 1.4 add_sub_module

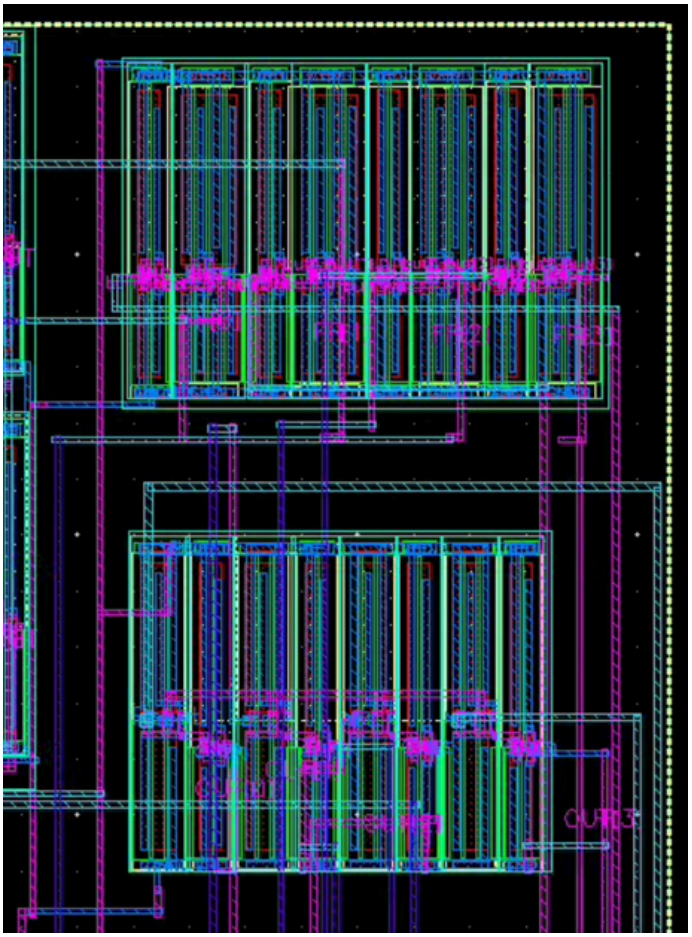


Figure 1.5 Reset and buffer

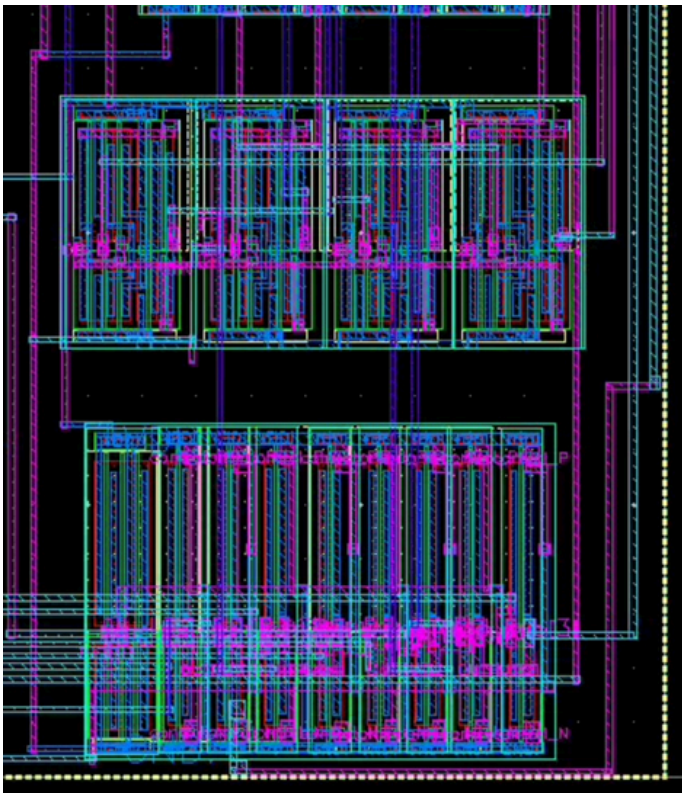


Figure 1.6 MUX_ADD_Mult and Flipflop

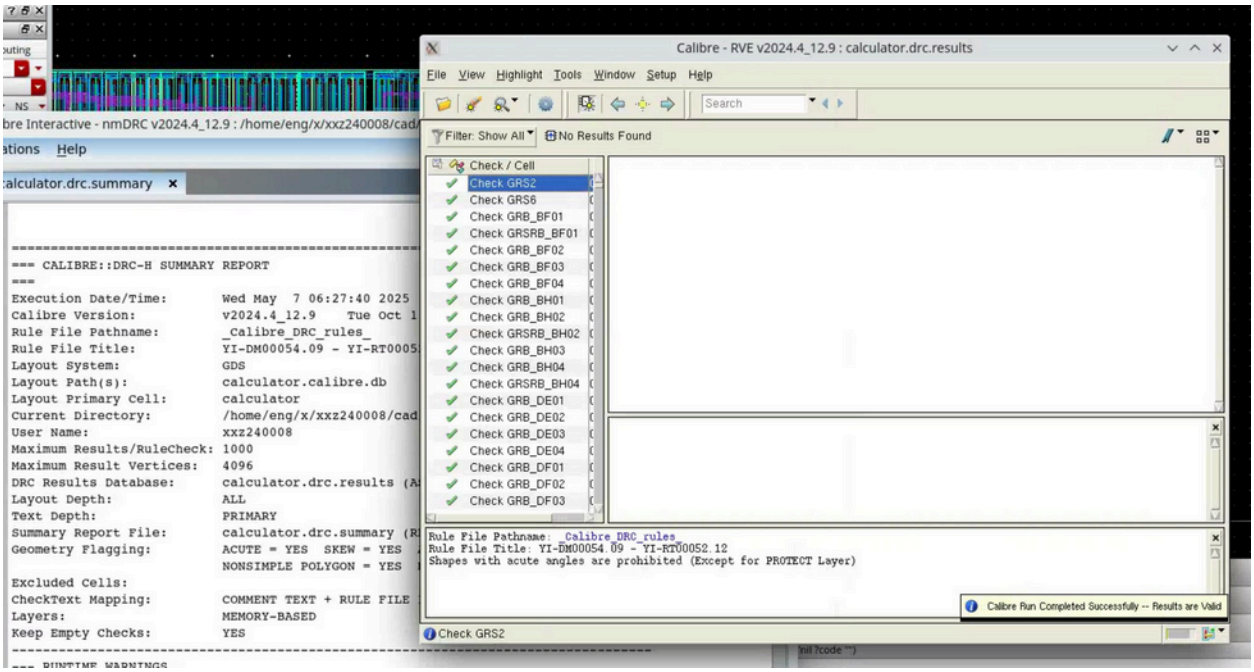


Figure 1.7 top level passed DRC

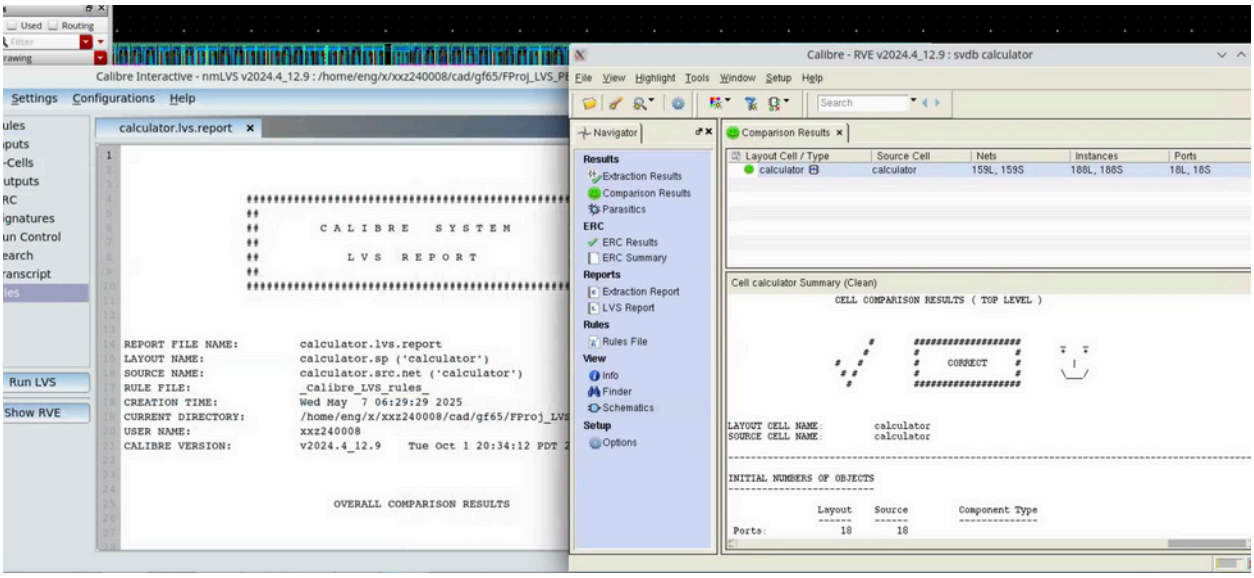


Figure 1.8 top level passed LVS

4. Simulation

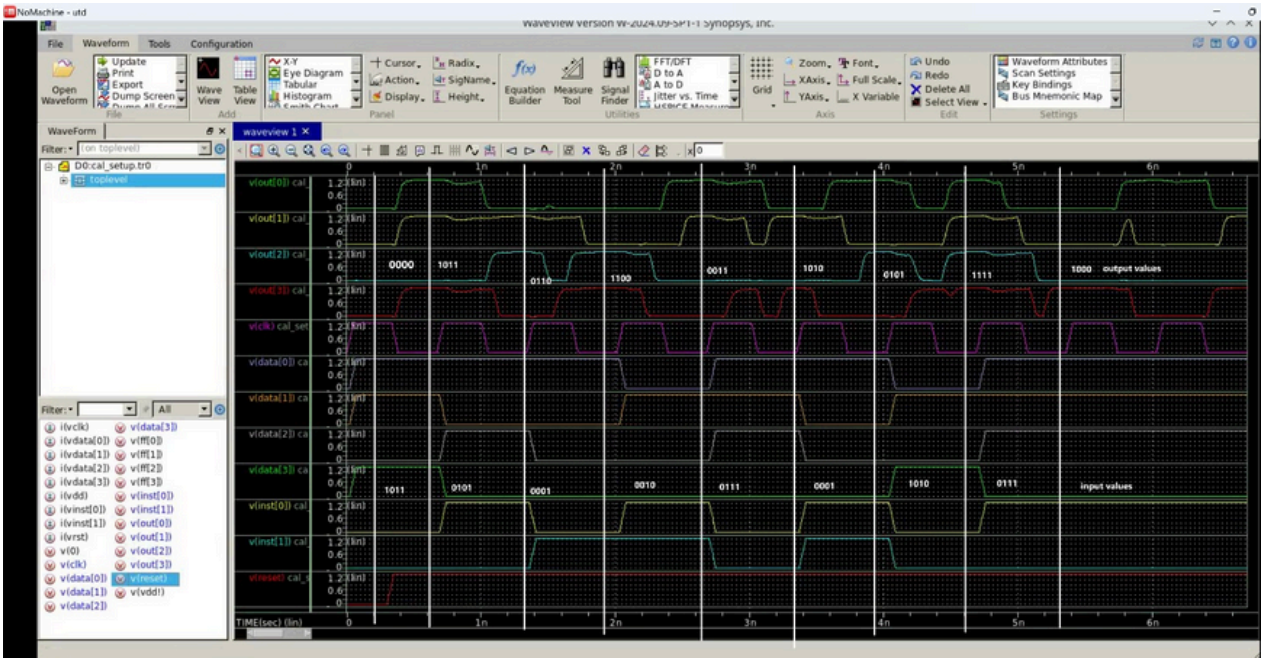


Figure 2.1 Simulated waveforms using provided test patterns

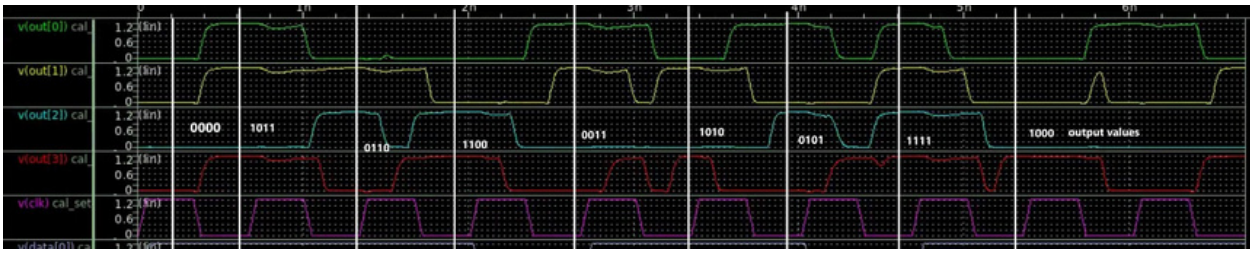


Figure 2.2 the design meets all the expected outputs and all processing completed within one clock cycle

```
******
$generated from generate_tb.py

****** transient analysis tnom=  25.000 temp=  25.000 ******
iavg=  -1.9453m  from=   0.             to=   5.3600n
energy= -12.5119p

          ***** job concluded
******
$generated from generate_tb.py

****** job statistics summary tnom=  25.000 temp=  25.000 ******
```

Figure 2.3 Total Energy for operating 8 test patterns

```
$Generated from generate_tb.py
$
.include "/proj/cad/library/mosis/GF65_LPe/cmos10lpe_CDS_oa_d1064_11_20160415/models/YI-SM00030/Hspice/models/design.inc"

$.include dff.pex.sp $include your design
.include "calculator.pex.netlist"
.option post runlv1=5

.param Tc=670p $CLK preiod
.param Di=20p  $Input delay after rising edge of clock

$xi gnd! Q vdd! CLK R D design  $instantiate your design
X2 VDD! GND! FF[0] RESET OUT[0] INST[1] OUT[1] OUT[2] OUT[3]
+ INST[0] DATA[0] DATA[1] FF[3] DATA[2] DATA[3] FF[1] FF[2] CLK calculator

vdd VDD! GND! 1.2v

cout0 OUT[0] GND! 50f
cout1 OUT[1] GND! 50f
cout2 OUT[2] GND! 50f
cout3 OUT[3] GND! 50f

*vrst reset GND! PWL(0ps 0v 300ps 0v 350ps 1.2v 600ps 1.2v 650ps 0v)
vrst reset GND! PWL(0ns 0v 0.3ns 0v 0.35ns 1.2v)
vclk clk GND! PULSE(0v 1.2v 0ps 50ps 50ps 'Tc/2-50ps' Tc)

$Expected Output:          D1:1011-D2:1011-   D3:-err-D4:0000-D5:0000-D6:0000-                D7:1111-D8:-err-D9:0111-D10:1101
vinst[1] inst[1] GND! PWL(0ps 0v Di 0v 'Di+0*Tc+50ps' 0v 'Di+1*Tc' 0v 'Di+1*Tc+50ps' 0v 'Di+2*Tc' 0v 'Di+2*Tc+50ps' 1.2v 'Di+3*Tc' 1.2v 'Di+3*Tc+50ps' 1.2v 'Di+4*Tc' 1.2v 'Di+4*Tc+50ps' 0v 'Di+5*Tc' 0v 'Di+5*Tc+50ps' 1.2v
~ 'Di+6*Tc' 1.2v 'Di+6*Tc+50ps' 0v 'Di+7*Tc' 0v 'Di+7*Tc+50ps' 0v 'Di+8*Tc' 0v)
vinst[0] inst[0] GND! PWL(0ps 0v Di 0v 'Di+0*Tc+50ps' 0v 'Di+1*Tc' 0v 'Di+1*Tc+50ps' 1.2v 'Di+2*Tc' 1.2v 'Di+2*Tc+50ps' 0v 'Di+3*Tc' 0v 'Di+3*Tc+50ps' 1.2v 'Di+4*Tc' 1.2v 'Di+4*Tc+50ps' 0v 'Di+5*Tc' 0v 'Di+5*Tc+50ps' 1.2v
~ 'Di+6*Tc' 1.2v 'Di+6*Tc+50ps' 0v 'Di+7*Tc' 0v 'Di+7*Tc+50ps' 1.2v 'Di+8*Tc' 1.2v)

$The following lines are only to get an example of what the output should look like.

vData[0] Data[0] GND! PWL(0ps 0v Di 0v 'Di+0*Tc+50ps' 1.2v 'Di+1*Tc' 1.2v 'Di+1*Tc+50ps' 1.2v 'Di+2*Tc' 1.2v 'Di+2*Tc+50ps' 1.2v 'Di+3*Tc' 1.2v 'Di+3*Tc+50ps' 0v 'Di+4*Tc' 0v 'Di+4*Tc+50ps' 1.2v 'Di+5*Tc' 1.2v 'Di+5*Tc+50ps'
~ 1.2v 'Di+6*Tc' 1.2v 'Di+6*Tc+50ps' 0v 'Di+7*Tc' 0v 'Di+7*Tc+50ps' 1.2v 'Di+8*Tc' 1.2v)
vData[1] Data[1] GND! PWL(0ps 0v Di 0v 'Di+0*Tc+50ps' 1.2v 'Di+1*Tc' 1.2v 'Di+1*Tc+50ps' 0v 'Di+2*Tc' 0v 'Di+2*Tc+50ps' 0v 'Di+3*Tc' 0v 'Di+3*Tc+50ps' 1.2v 'Di+4*Tc' 1.2v 'Di+4*Tc+50ps' 1.2v 'Di+5*Tc' 1.2v 'Di+5*Tc+50ps' 0v
~ 'Di+6*Tc' 0v 'Di+6*Tc+50ps' 1.2v 'Di+7*Tc' 1.2v 'Di+7*Tc+50ps' 1.2v 'Di+8*Tc' 1.2v)
vData[2] Data[2] GND! PWL(0ps 0v Di 0v 'Di+0*Tc+50ps' 0v 'Di+1*Tc' 0v 'Di+1*Tc+50ps' 1.2v 'Di+2*Tc' 1.2v 'Di+2*Tc+50ps' 0v 'Di+3*Tc' 0v 'Di+3*Tc+50ps' 0v 'Di+4*Tc' 0v 'Di+4*Tc+50ps' 1.2v 'Di+5*Tc' 1.2v 'Di+5*Tc+50ps' 0v
~ 'Di+6*Tc' 0v 'Di+6*Tc+50ps' 0v 'Di+7*Tc' 0v 'Di+7*Tc+50ps' 1.2v 'Di+8*Tc' 1.2v)
vData[3] Data[3] GND! PWL(0ps 0v Di 0v 'Di+0*Tc+50ps' 1.2v 'Di+1*Tc' 1.2v 'Di+1*Tc+50ps' 0v 'Di+2*Tc' 0v 'Di+2*Tc+50ps' 0v 'Di+3*Tc' 0v 'Di+3*Tc+50ps' 0v 'Di+4*Tc' 0v 'Di+4*Tc+50ps' 0v 'Di+5*Tc' 0v 'Di+5*Tc+50ps' 0v 'Di+6*Tc'
~ 0v 'Di+6*Tc+50ps' 1.2v 'Di+7*Tc' 1.2v 'Di+7*Tc+50ps' 0v 'Di+8*Tc' 0v)

.tr 10ps '10*Tc'  $Run for number of input clock cycles plus 2

.measure tran iavg avg i(vdd) from=0 to='8*Tc'
.measure energy param = 1.2*iavg*8*Tc


.end
```

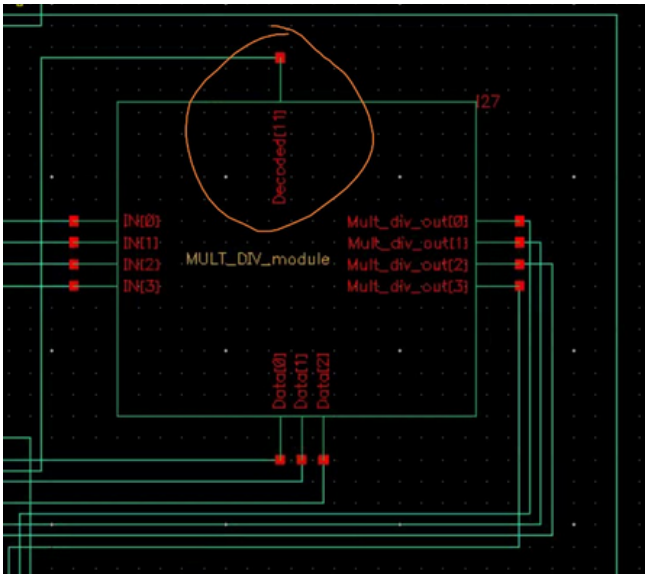
Figure 2.4 sp file, which includes clock cycle = 670ps, and test patterns

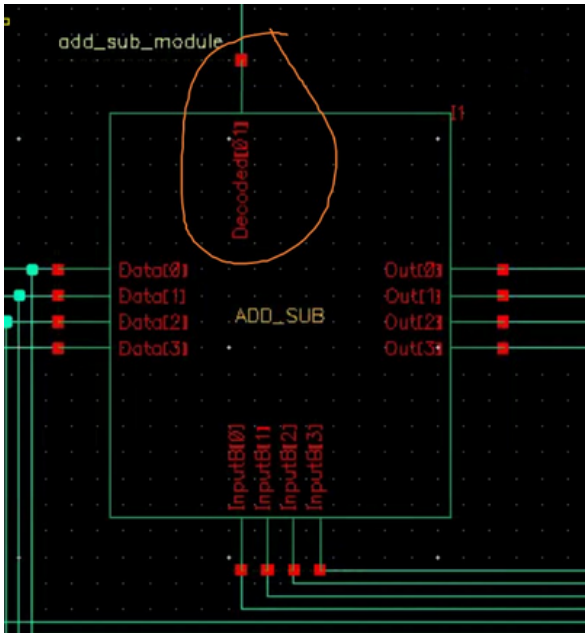
5. Project Decisions

The design contains 2 computation modules to perform addition, subtraction, multiplication, division; 1 MUX to select output result from computation modules; 1 Flipflop module to retain previous result; 1 reset module to reset the value in flipflop to all zero; 1 buffer module between Flipflop module and computation modules to improve the performance of driving fanout circuits.

6. AEDP Optimization

1. Optimize and reduce redundant modules/circuits. E.g1. In initial design, a decoder is used. But after finish the design of add_sub module and mult_div module, there is no need to contain a decoder in the circuit since the INST[0] bit can select add/sub or mult/div in each module. And INST[1] can select add_sub or mult_div in the MUX to provide output.





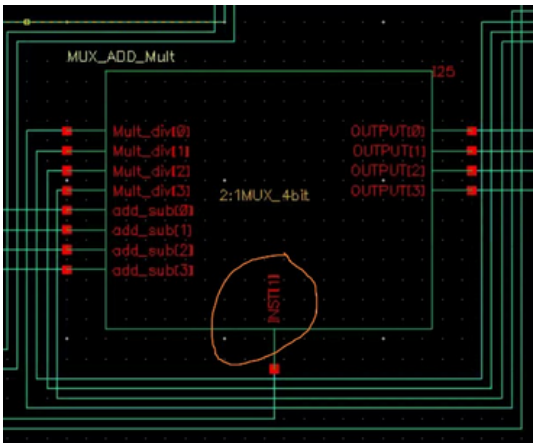


Figure 6.1 INST bits feed directly to modules

E.g.2. Reuse repeated circuit. In Mult_div module, there 12 MUX used in initial design, by rerouting and reuse MUX, the final design has 10 MUX.

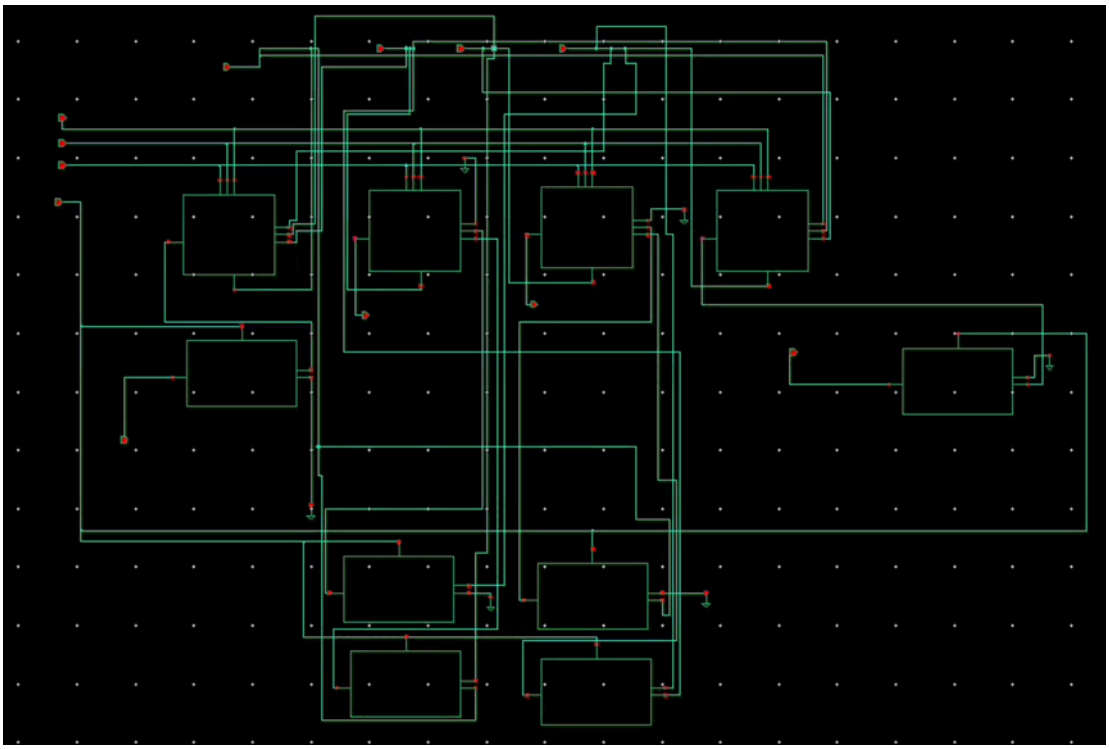


Figure 6.2 Mult_div module schematic

2. Gate level optimization: Use transmission gate instead of static CMOS gate to reduce the number of transistors. E.g. use transmission gate to build a 2x1 MUX(without buffers) requiring 6 transistors instead of 16 transistors with static CMOS gate.

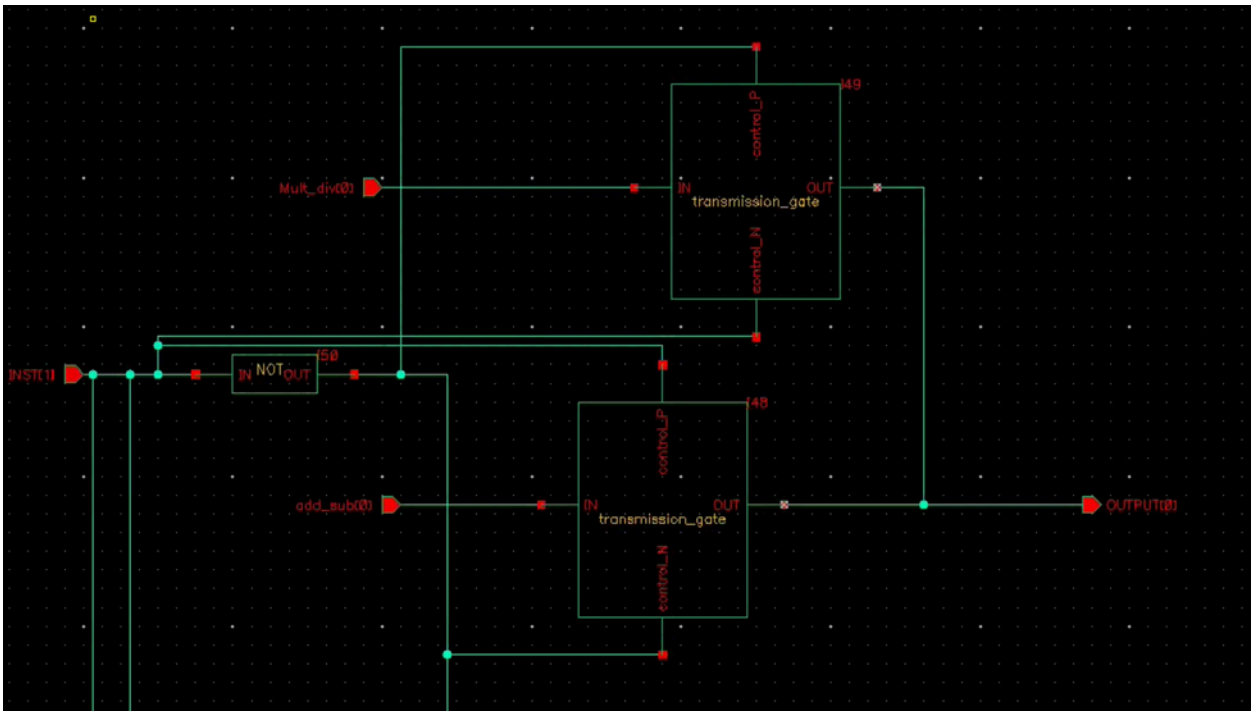


Figure 6.3 transmission gates are used in MUX_ADD_Mult module

3. Routing: M3 interconnection layer are used in subcircuit, M4 layer are used in top level; Upper metal layer reduces resistance when the wire has long distance.

4. Buffer module between Flipflop module and computation modules is added to reduce delay, and it improves the circuit speed (clock cycle) in HSPICE simulation.

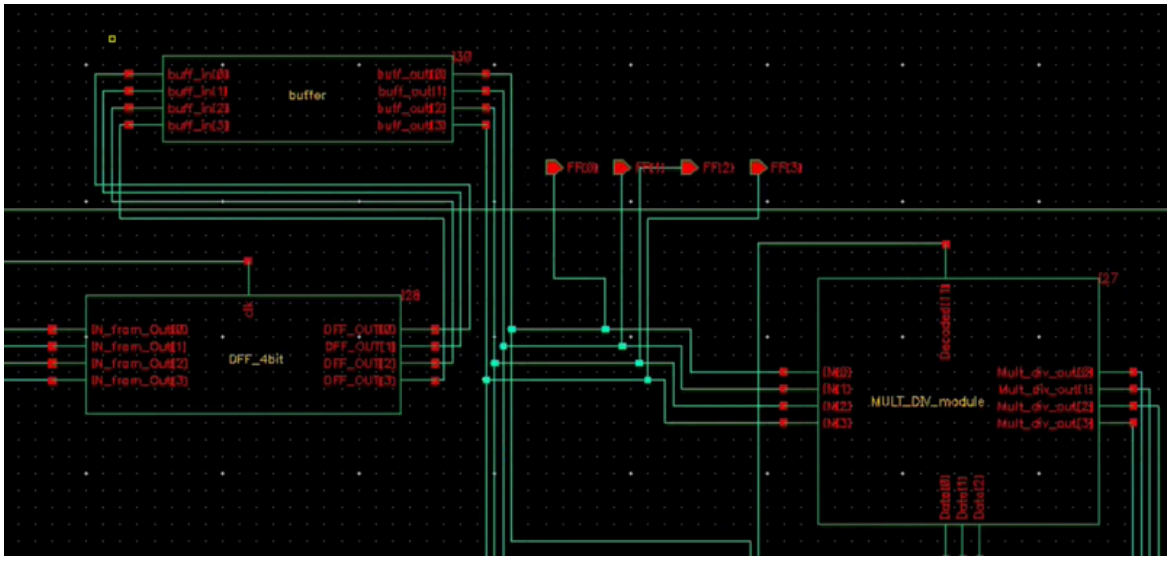


Figure 6.4 buffer module