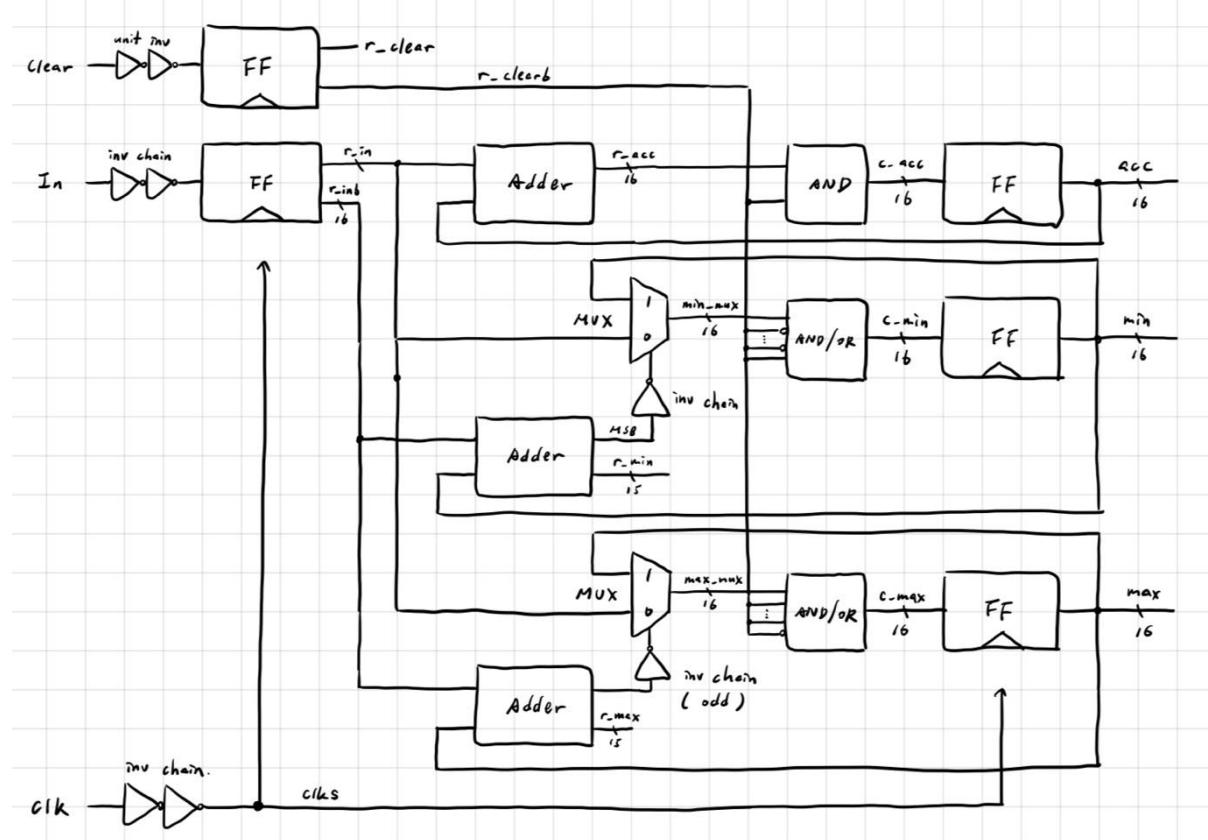


**EE3230 VLSI Design (2023 Fall) Final Project**  
**Team 30: 110061119 李昀達、110061217 王彥智**

**Block Diagram**

**Design:**

Top view block diagram:



**Operation of design:**

This project consists of the design and layout of a chip which accumulates and finds the minimum and maximum of a series of 16-bit input numbers. All 3 adders have 16-bit inputs and a 16-bit output. The "AND" and "AND/OR" blocks contain 16 2-input gates with one input of all gates connected to  $r\_clear$ . The "AND" blocks contain 16 AND gates. The "AND/OR" block for min and max contains 1 AND (OR) gate connected to the MSB of "min\_mux" (max\_mux) and 15 OR (AND) gates with one input inverted tied to  $r\_clear$  (i.e., out = in1 OR in2). The output max is the maximum, min is the minimum, and acc is the accumulated sum of all inputs since the last time clear was asserted.

**Design consideration:**

For the design of this project, we set 0.5G Hz pre-layout simulation operating frequency as our goal, thus, we lean on choosing adder and block designs with faster operating frequencies (lower time delay).

For the design of the flip-flop for the 16-bit input, 3 sets of 16-bit output and clear signal, we opted for our flip-flop design used in hw4. We choose the inverter flip-flop, as it has a lower C2Q and D2Q time compared to the NAND2 flip-flop. We also increased the size of the 2 inverters on the critical path to 3 times unit size, as this could further lower the path delay.

For the design of the main adder, we can split our pre-simulation testing into 3 types: Carry lookahead adder valency [4, 4, 4, 4], carry lookahead adder [2, 3, 4, 4, 3], MUX adder (from paper “Fast Mux-based Adder with Low Delay and Low PDP” – DOI: 10.22044/JADM.2018.7177.1853). By running pre-sim on the following different cases using mode2 from the testbench (mode2 is a balance between run time and testing acc min and max) we can obtain the following Table. (testing at simcase2 at 100\*tclk)

Type	Optimization	Frequency (GHz)	Power (mW)	Freq / Power
CLA	Original	0.25	5.1439	0.0486
	2 inv at input m=8	0.41	2.9216	0.14033
	2 inv at input m=16	0.41	2.7605	0.14852
MUX adder	2 inv at input m=16	0.475	3.3911	0.14007
	FF m=3 with 2 inv at input m=16	0.48	4.9792	0.09640
	FF m=3 with 4 inv at input m=1 4 16 64	0.505	5.4505	0.92652
	Merge inv in FF with FF m=3	0.49	4.7193	0.10383
[2, 3, 4, 4, 3]	2 inv at input m=16	0.43	2.8882	0.14888
	Final design	0.565	3.7853	0.14926s

From the table above, we can see that although the Mux adder design might achieve a higher operating frequency after more optimization, the power is also higher, which results in a lower FoM. As for the carry lookahead adder using valency [4, 4, 4, 4] compared with using valency [2, 3, 4, 4, 3], although using [4, 4, 4, 4] is easier to implement when completing the design in layout, we are able to reach a higher frequency and FoM when using the valency [2, 3, 4, 4, 3] design. Thus, in the end we decided on using the optimized carry lookahead adder with valency of [2, 3, 4, 4, 3], as it balances operating frequency, power and FoM.

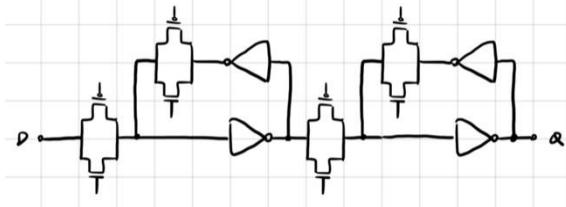
Additional optimizations are made to achieve the final design with a simcase2 operating frequency of 0.565 GHz. First, we inserted a 4-stage inverter chain at the clk input with sizing of 1, 3, 9, 18 to increase the pushing capacity of the clk signal. Second, from analyzing the operating capacitance of each node, we can see that the select node of the MUX block sees a high capacitance, thus, a 2-stage (m=4, 10) and a 3-stage (m=3, 9, 18) is inserted at the select of MUX for min and max respectively to increase the operating frequency. Odd number of inverters are added before the MUX of max to change the selection to match that of the min MUX. This lowers the complexity when drawing the layout as we only need to switch out the inverter chain and AND/OR blocks without changing the wiring for the min and max adders.

To lower the power of the entire design, we removed the last valency3 cell, as in this final project, overflow of the adder is neglected. This further lowers the power consumption of our design and increases FoM. The final pre-sim does indeed meet our personal goals.

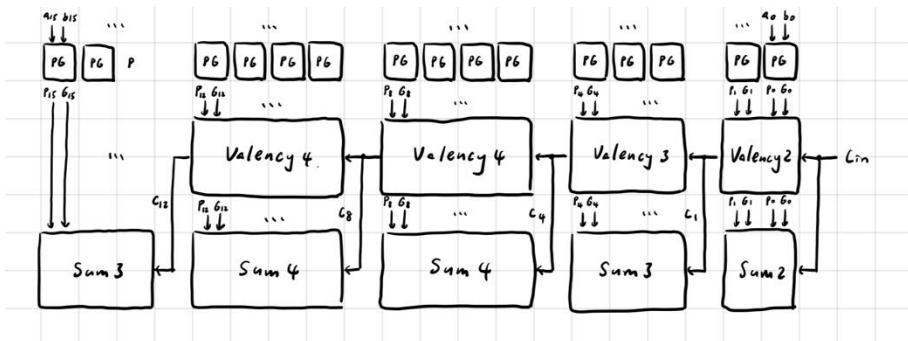
### Gate-level:

Figures below are the gate level design of each block used in the block diagram given above. However, they are only a brief representation on the workings of each design. Our final design switches all gates to either inv, nand or xor for easier simulation and layout implementation. Specifics will be further explained in the transistor-level design.

FF:

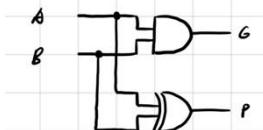


Adder:



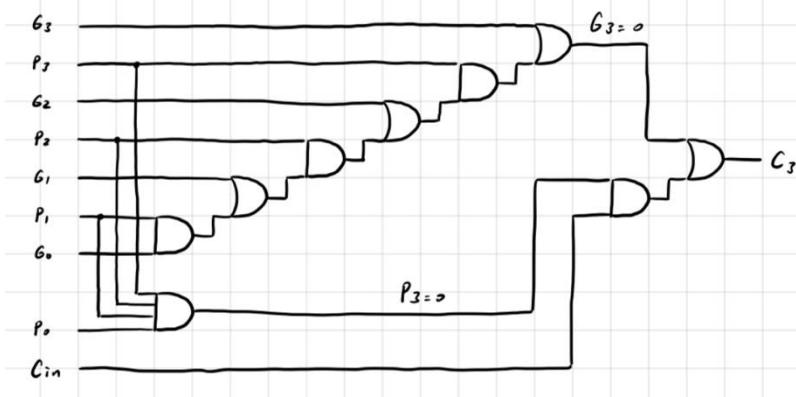
PG:

$$P = A \oplus B \quad G = A \cdot B$$



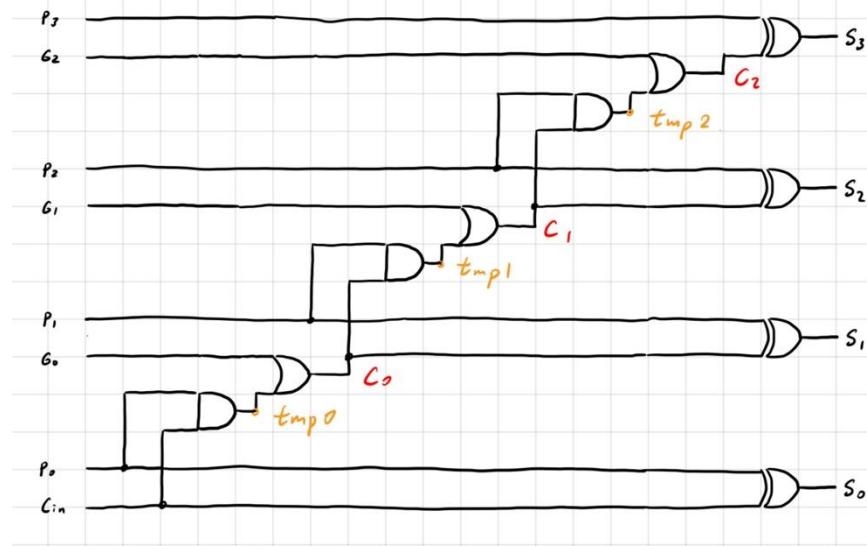
Valency:

$$C_{out} = G_{i=0} = G_i + P_i \cdot G_{i-1=0} = G_{i,j} + P_{i,j} \cdot G_{i-1=0}$$

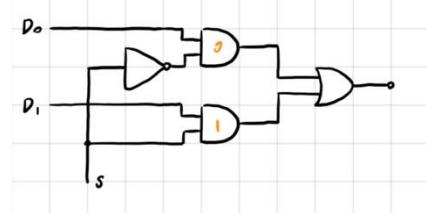


Sum:

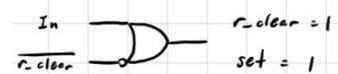
$$S_i = P_i \oplus G_{i-1,o} = P_i \oplus C_{i-1}$$



MUX:



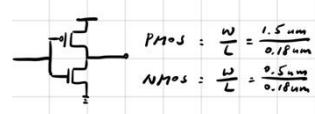
AND/OR:



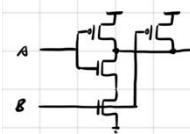
### Transistor-level:

Following the gate level design of each block, by using the bubble method, we can switch gates into NAND and inverter gates, which is easier for layout implementation, as we only need to complete the layout of a unit inverter, NAND2, NAND3, NAND4 and a XOR2. The transistor level design of these 5 gates is shown below. Figures below on the other complex blocks show the use of bubble technique to switch AND and OR gates into NAND gates.

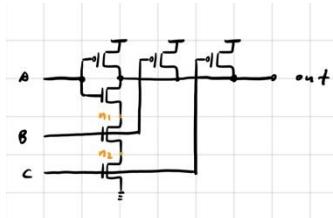
Inverter:



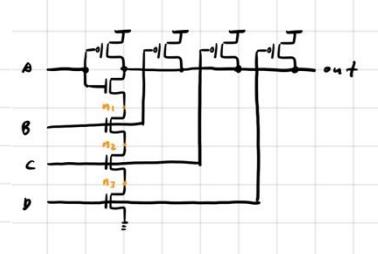
NAND2:



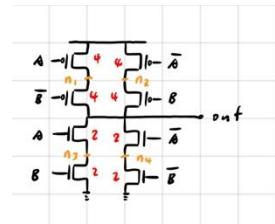
NAND3:



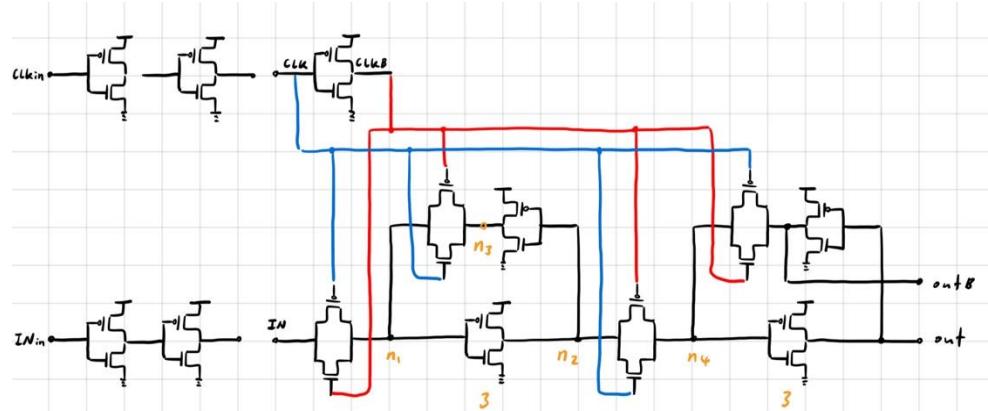
NAND4:



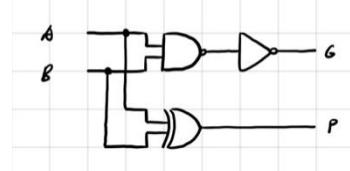
XOR:



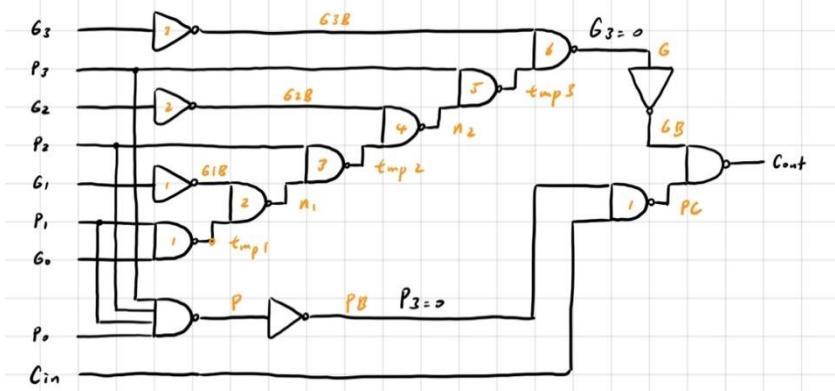
FF:



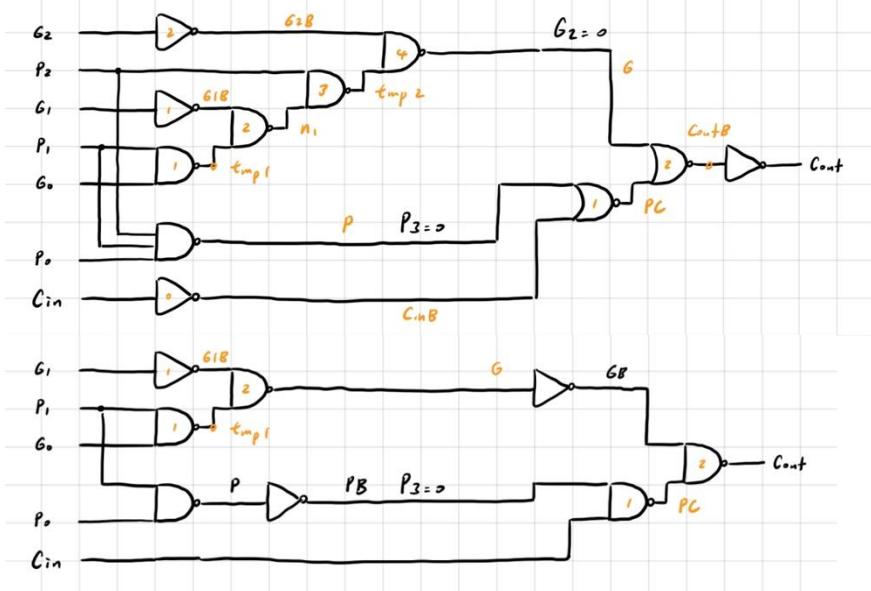
PG:



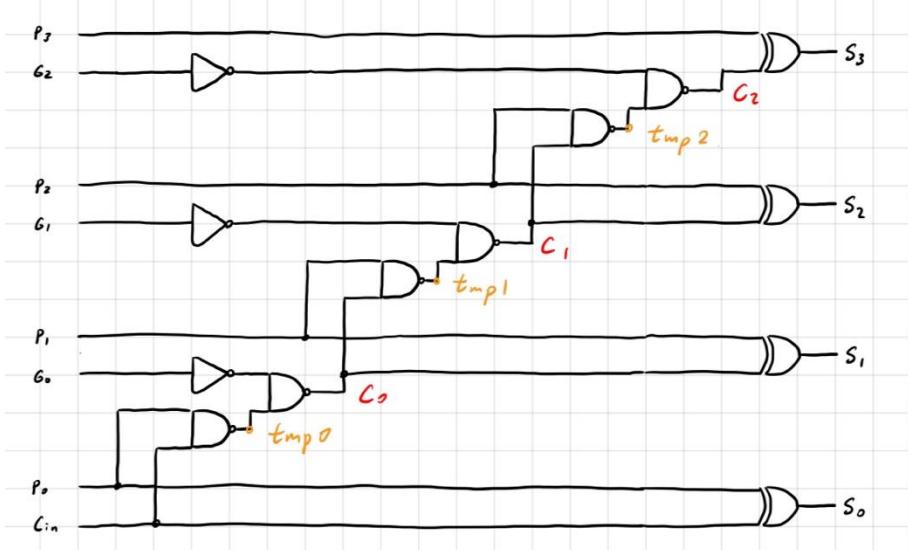
Vacency4:



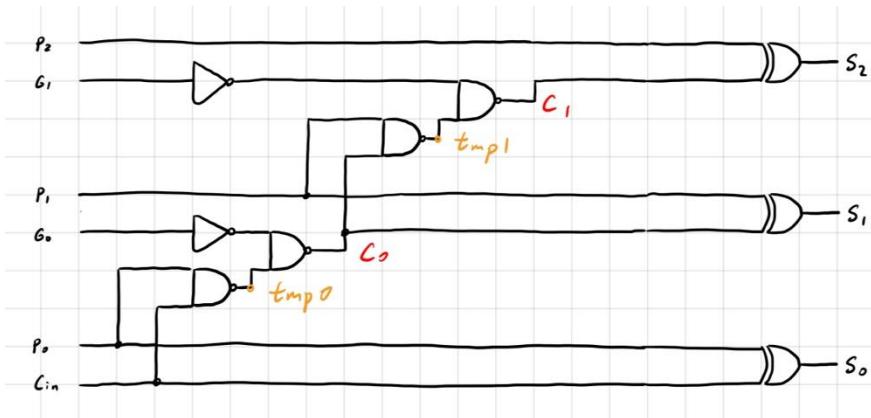
Vacency3:



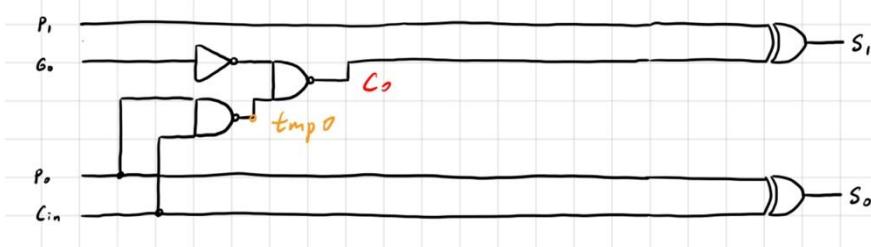
Sum4:



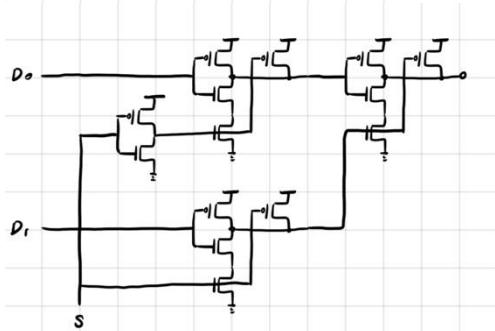
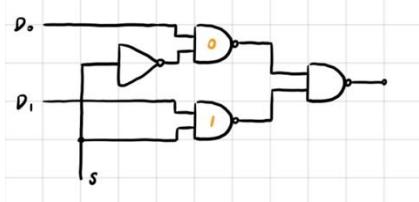
Sum3:



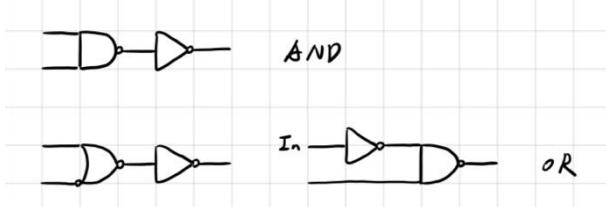
Sum2:



MUX:

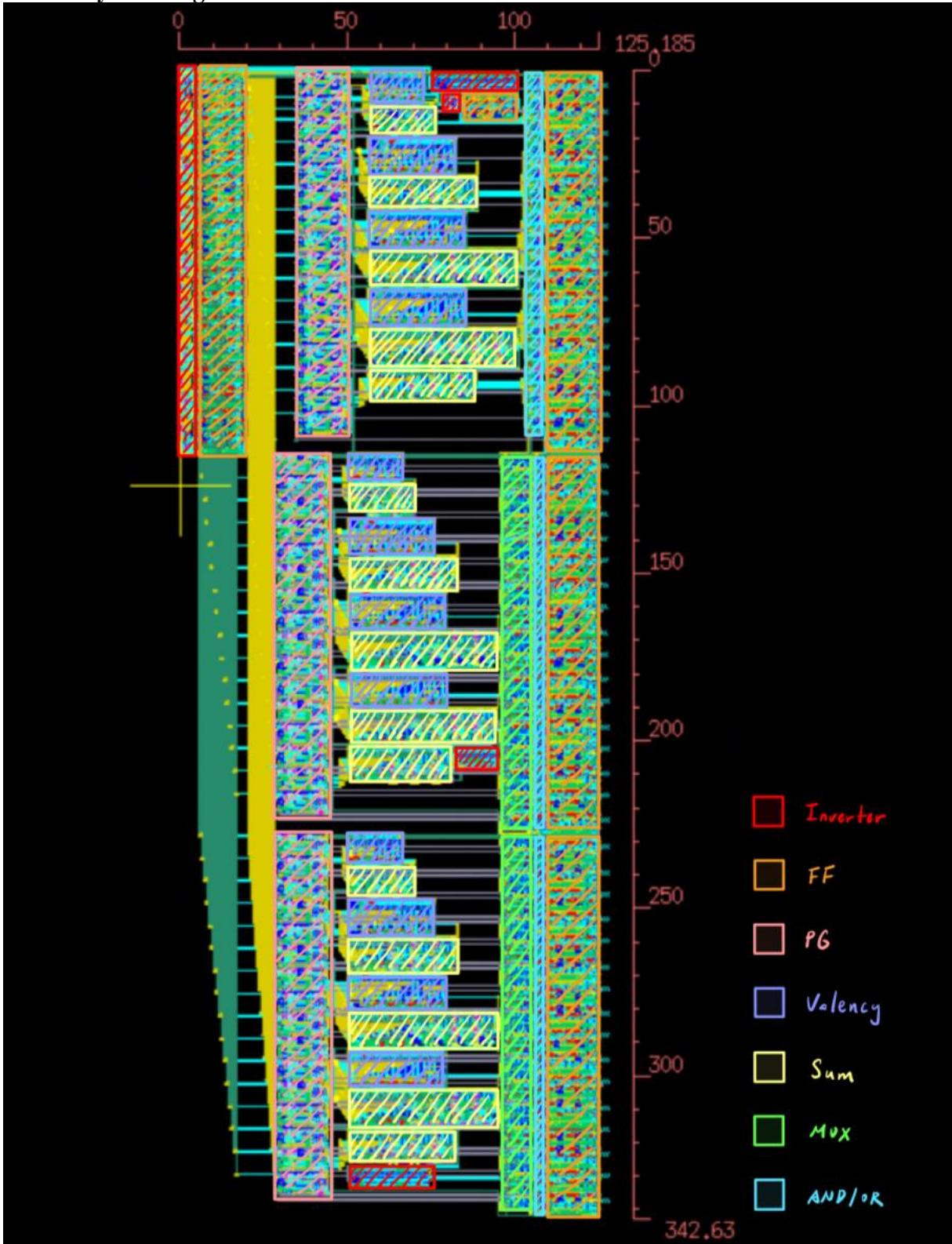


AND/OR:



## Layout

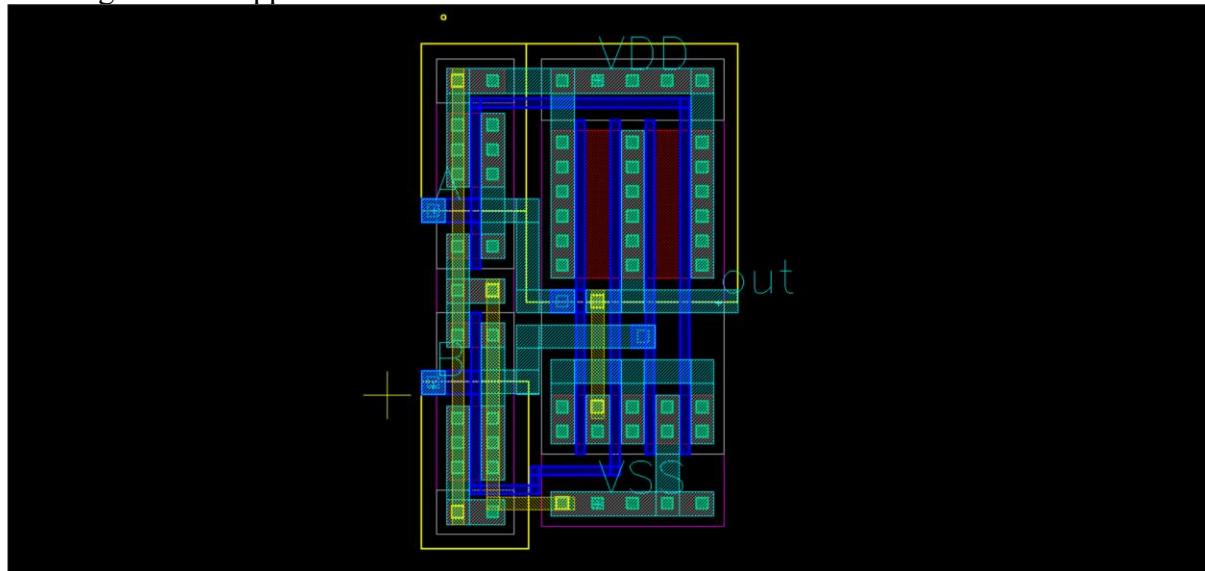
Entire layout design:



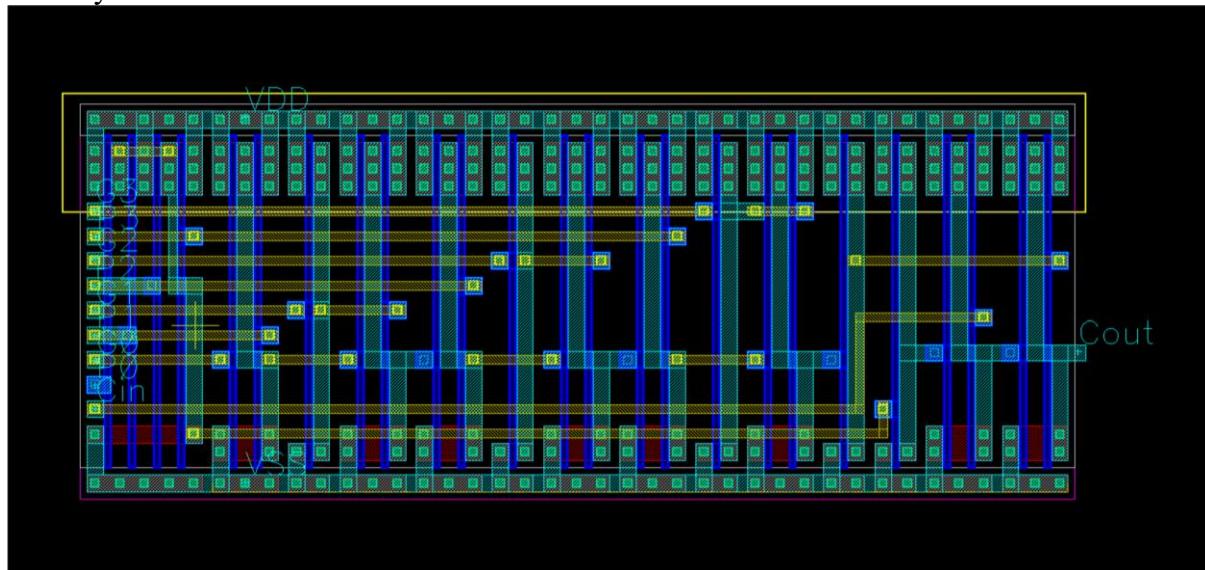
Layout area:  $125.185\mu\text{m} * 342.63\mu\text{m} = 42892.14 (\mu\text{m}^2)$

Below figures are individual layout of special block implementations, each are shared to the maximum, with PG block using a horizontal flipped XOR gate to match the height of NAND and inverter layout design. All inputs are connected at the left and output at the right.

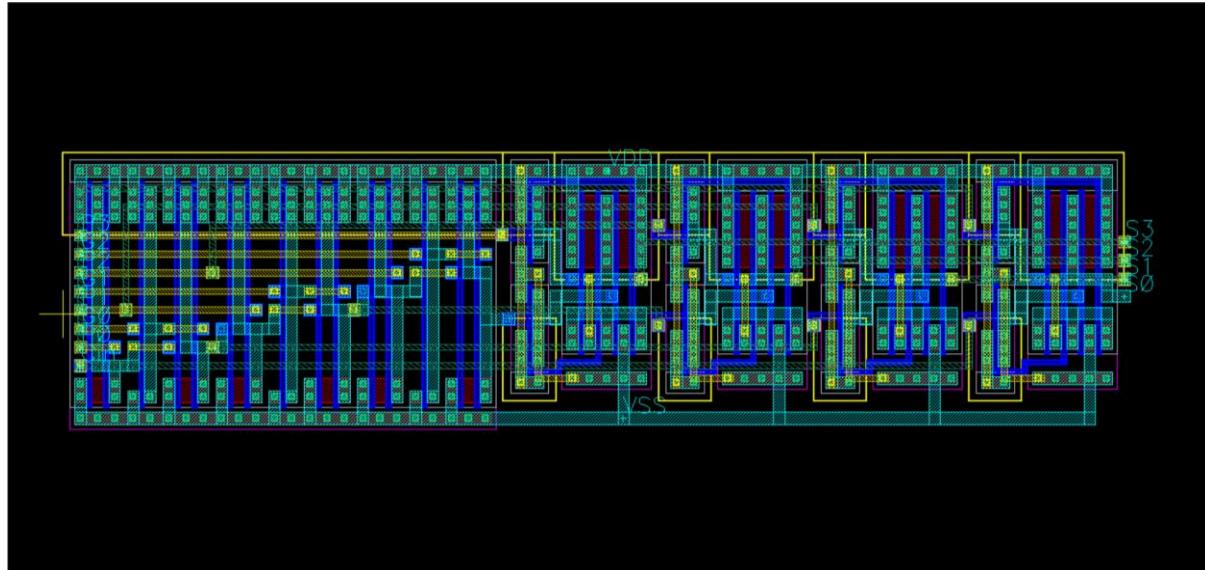
XOR2 gate with flipped and shared inverters:



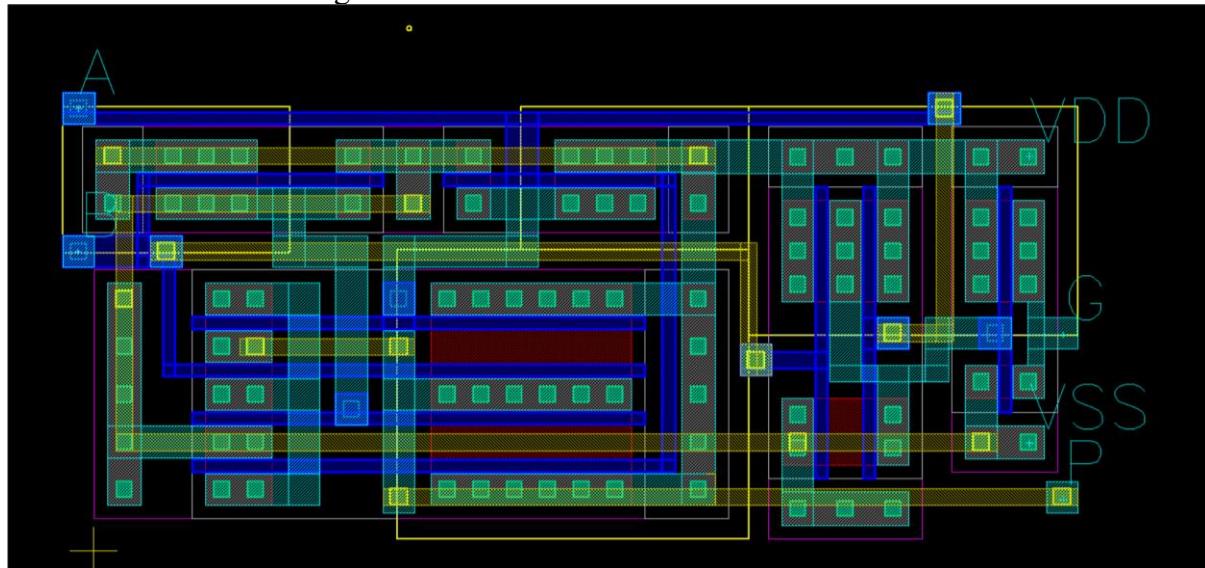
Valency4:



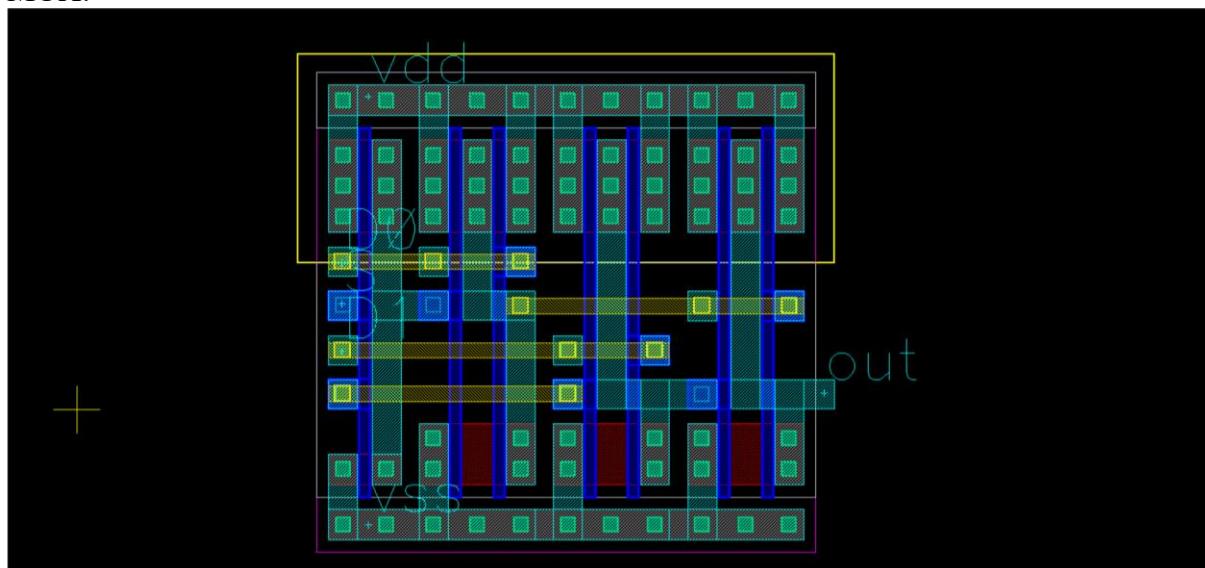
Sum4:



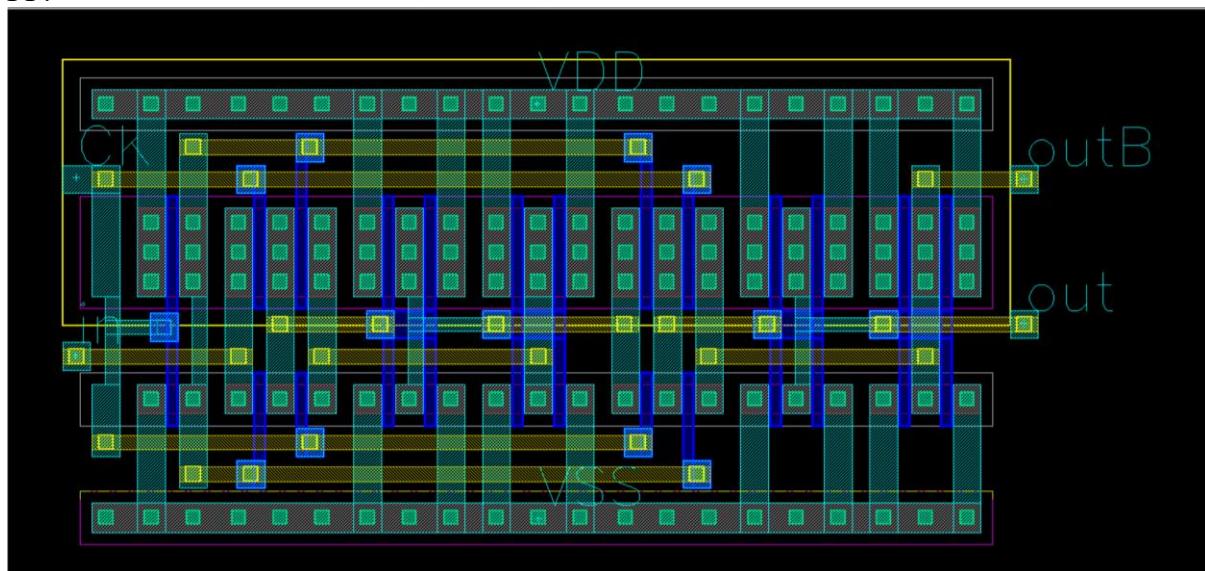
PG with horizontal XOR gate for minimized area:



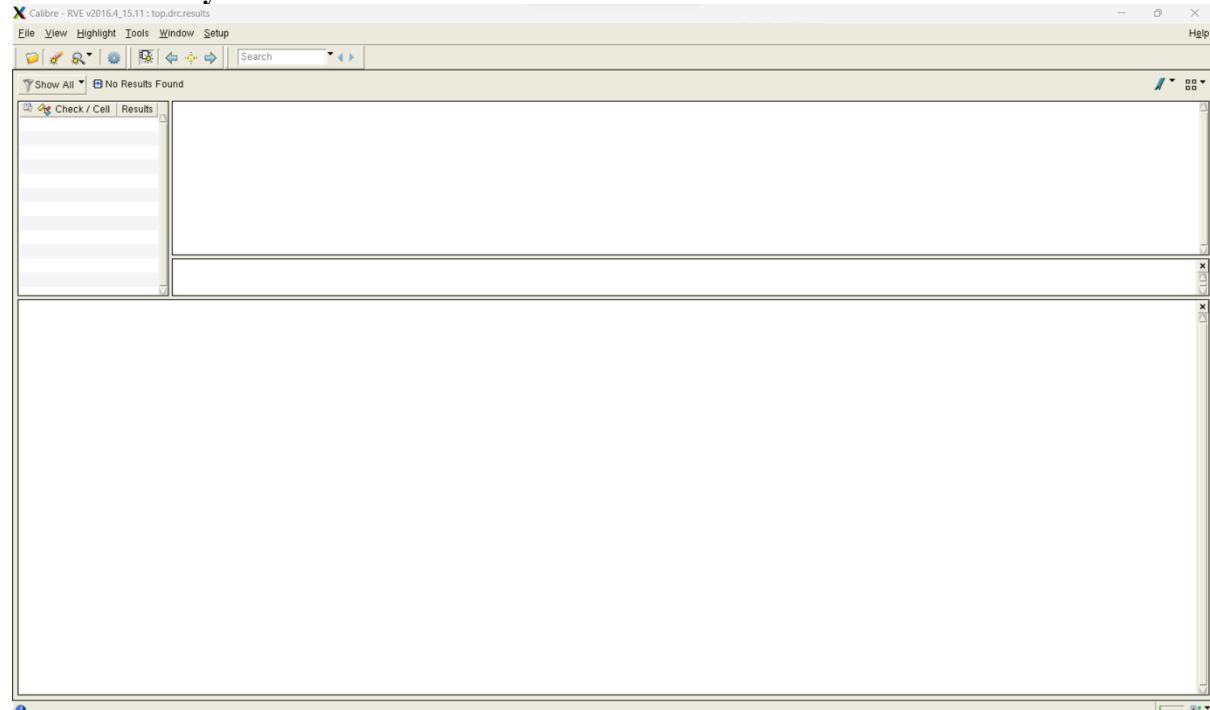
MUX:



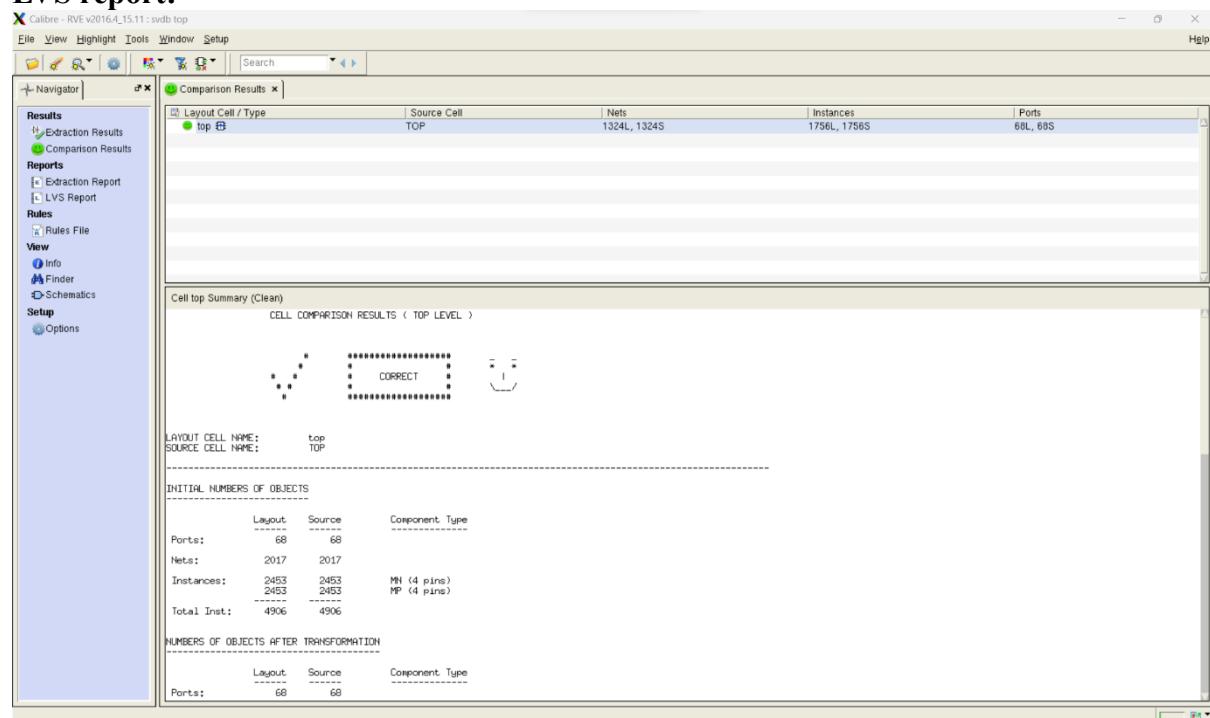
FF:



## DRC summary:



## LVS report:



## Layout consideration:

Since we completed the layout as a team, there are a few layout considerations and directions we try to follow to achieve a smaller layout area and to simplify our layout drawing process. When completing the layout, we split the process into 3 levels: simple gates (inv, NAND, XOR gates), individual blocks (FF, MUX, adder ...) and the top-level layout. As the design for each of the individual block only uses inv, NAND and XOR gates, we are able to complete each block by importing the 3 simple gates. Then, by connecting each of the blocks we are able to complete the entire layout design efficiently.

A key part of all our designs is to share as much as possible, as this significantly lowers the total layout area. Thus, in our design of inverter, NAND and XOR gates, we can see that all transistors share either drain and source nodes with nearby devices. We also try to avoid using metal 4 in the simple and lower gate designs, as we want to save it to use to complete the feedback from the output nodes to the adders.

When designing each block, we try to connect all inputs from the left side and all outputs from the right, with VDD on top and VSS on the bottom. This ensures that we wouldn't face overlapping wiring problems when connecting blocks together. This also takes effect to our top-level layout, as all 16-bit inputs with clear and clk signals are passed in from the left side, while signal outputs nodes for acc, min and max are all on the right.

Since we use similar blocks in the adder design for acc, min and max, when connecting with MUX and FFs, we try to align the same blocks for a compact and neat layout, this also further lower the total area. When connecting the different blocks with 16-bit input, we separate the horizontal lines by using metal 1 and vertical lines by using metal 2, this ensures that the connection is aligned properly without the chance of entanglement and the use of higher metals.

## Simulation Results

All following simulation are tested under  $10*tclk$  for a more efficient testing process. However, the power consumption under this testing condition is significantly larger than that under the condition using  $100*tclk$ .

FoM in the following tables are calculated by frequency / power. This gives us a better comparison between pre-layout simulation vs. post-layout simulation, as there is no area assumption in the pre-layout phase. (Formula for FoM given: frequency / area\*power is not suitable)

### Pre-layout simulation results:

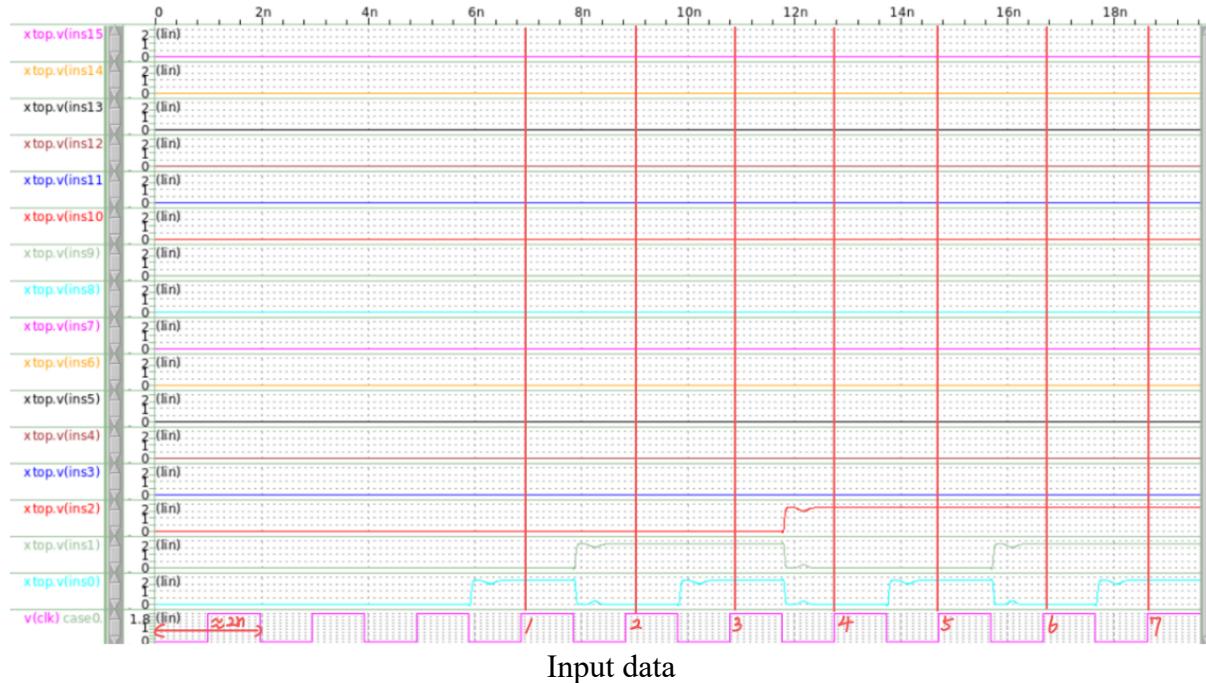
Table. 1

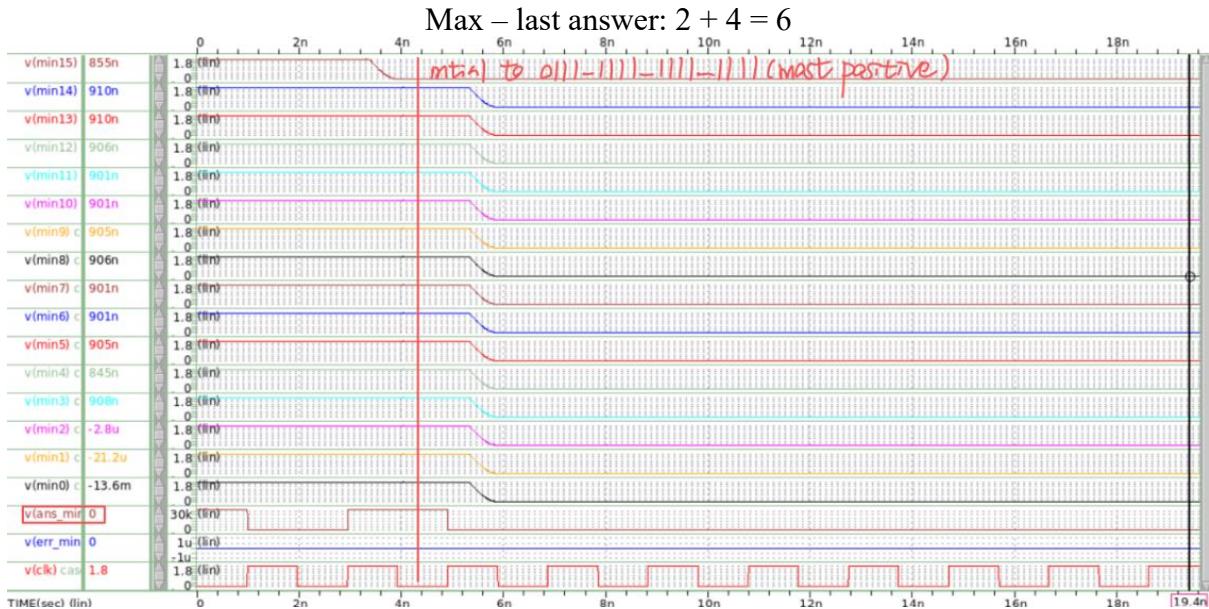
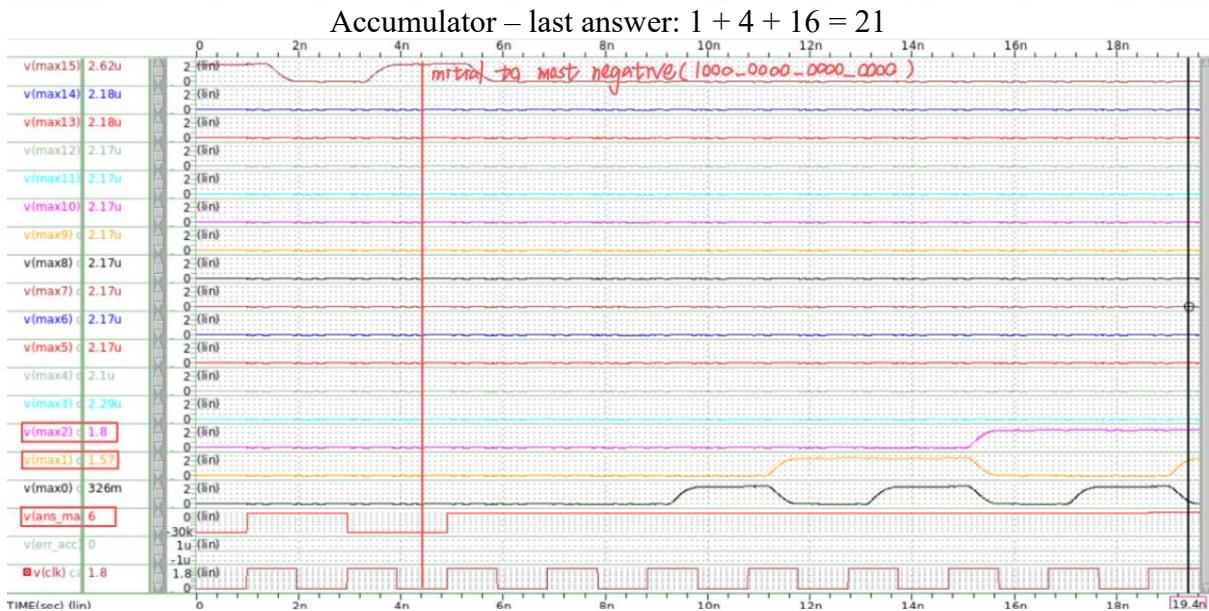
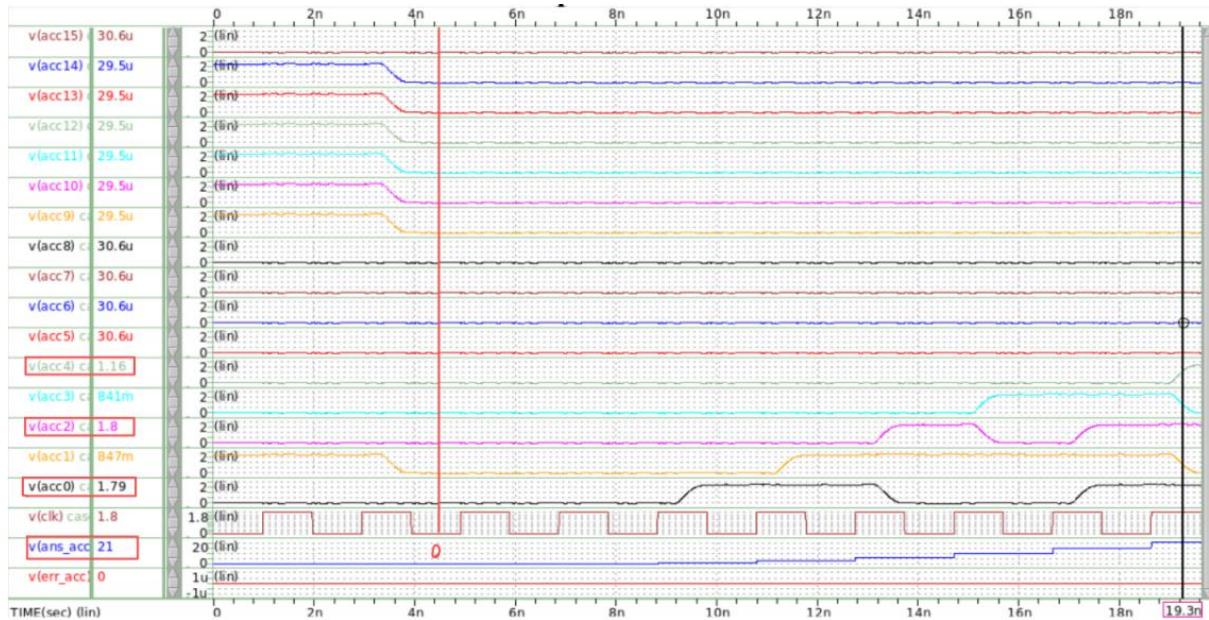
TT 25°C			
	Frequency (GHz)	Power (mW)	FoM (Freq / Power)
Sim case 0	0.51	4.9509	0.1030 M
Sim case 1	0.635	5.6421	0.1125 M
Sim case 2	0.565	10.8479	0.0521 M

(For sim case 2 at  $100*tclk$ , power = 3.7564mW)

Waveform analysis for TT corner 25°C:

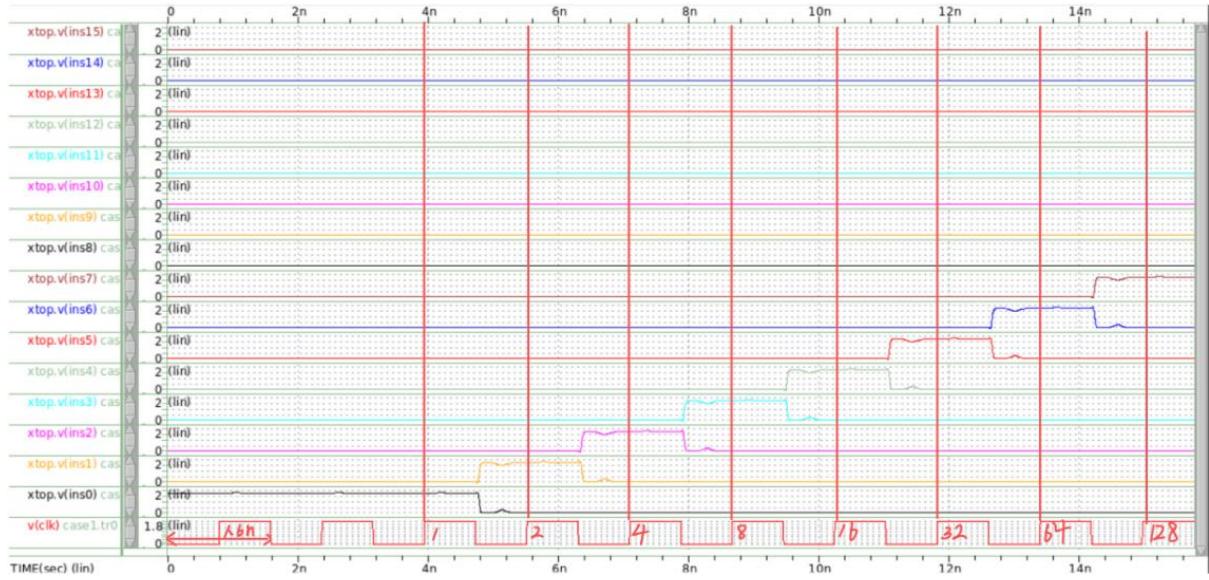
Sim case 0:



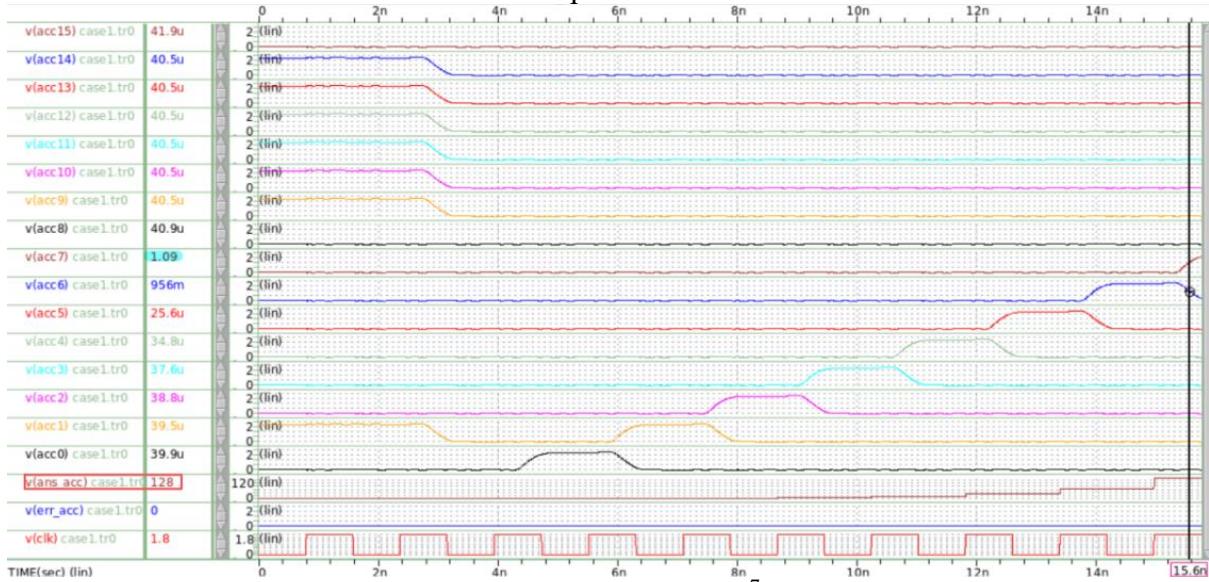


Min – last answer: 0

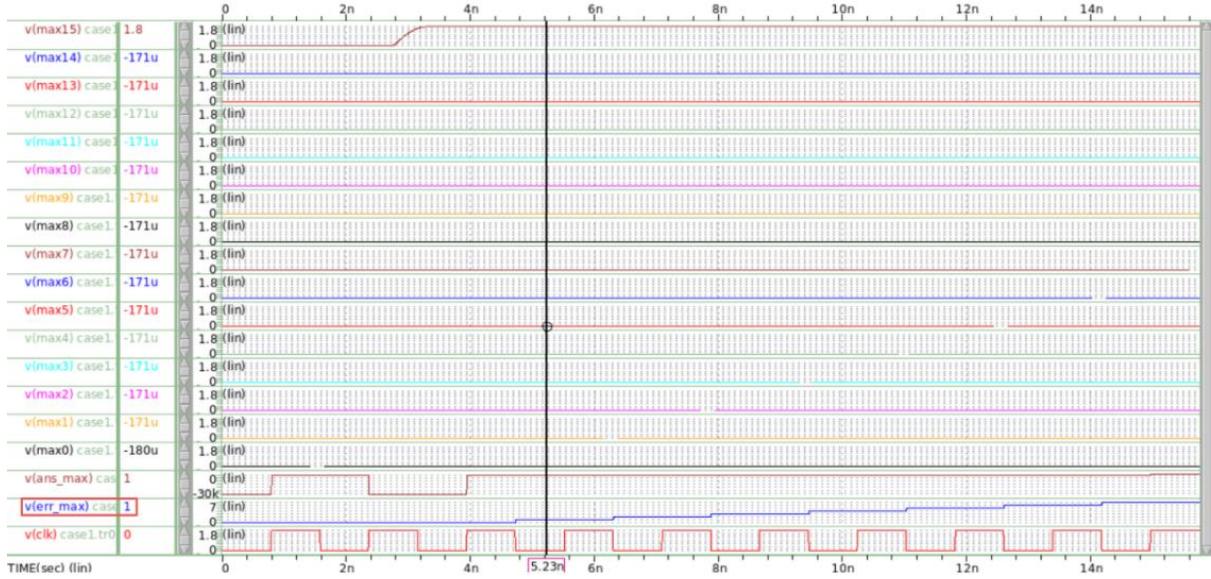
Sim case 1:



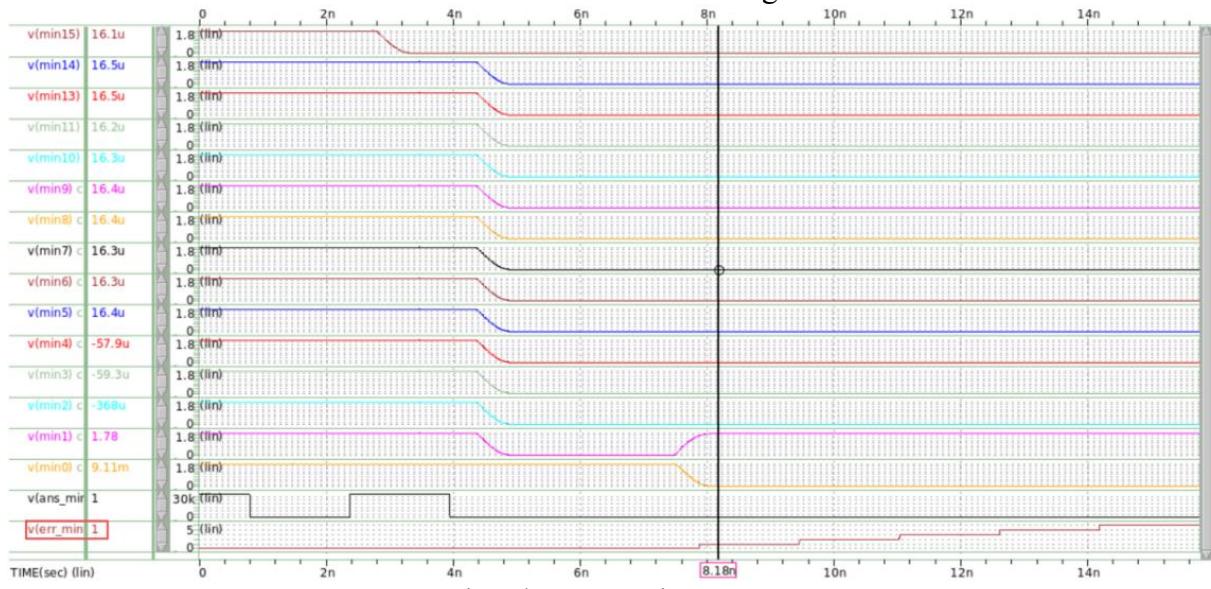
Input data



Accumulator – last answer:  $2^7 = 128$

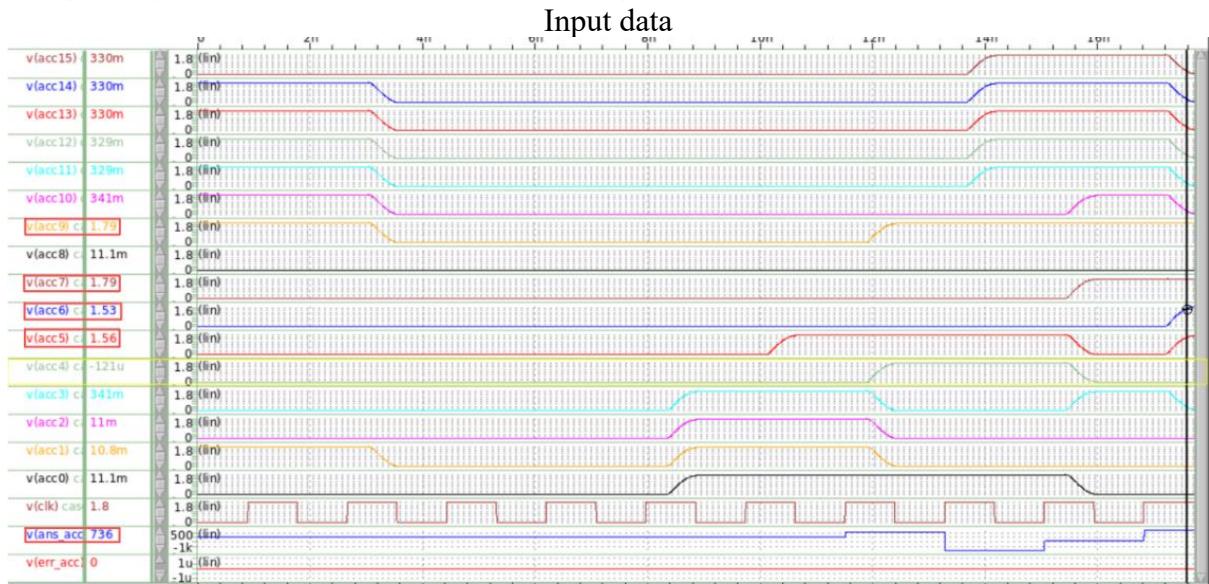
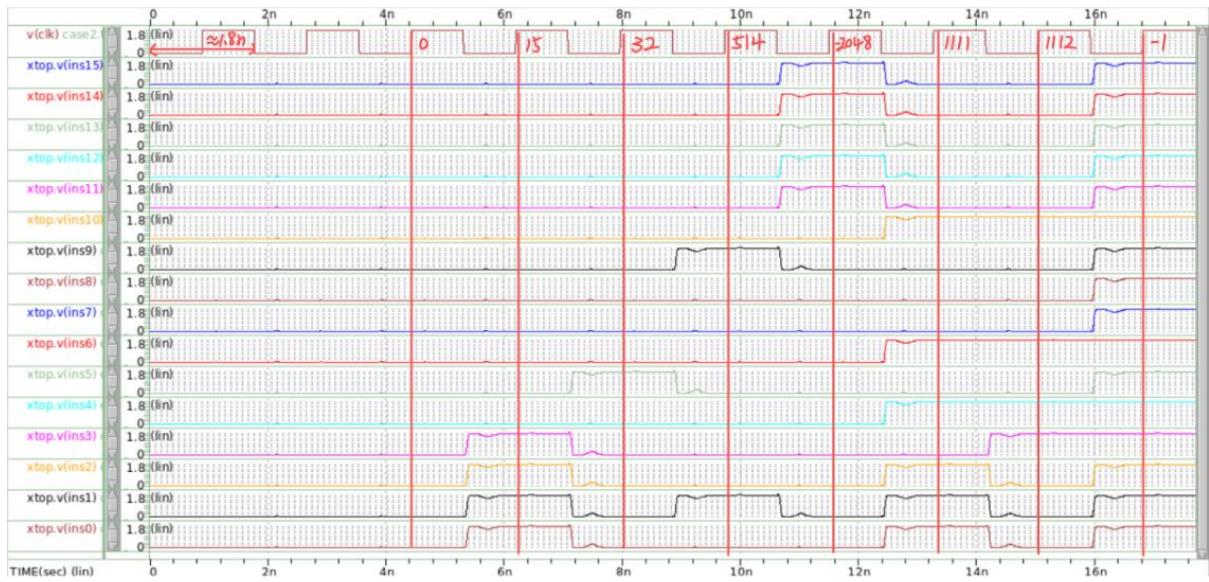


Max – the answer is wrong.

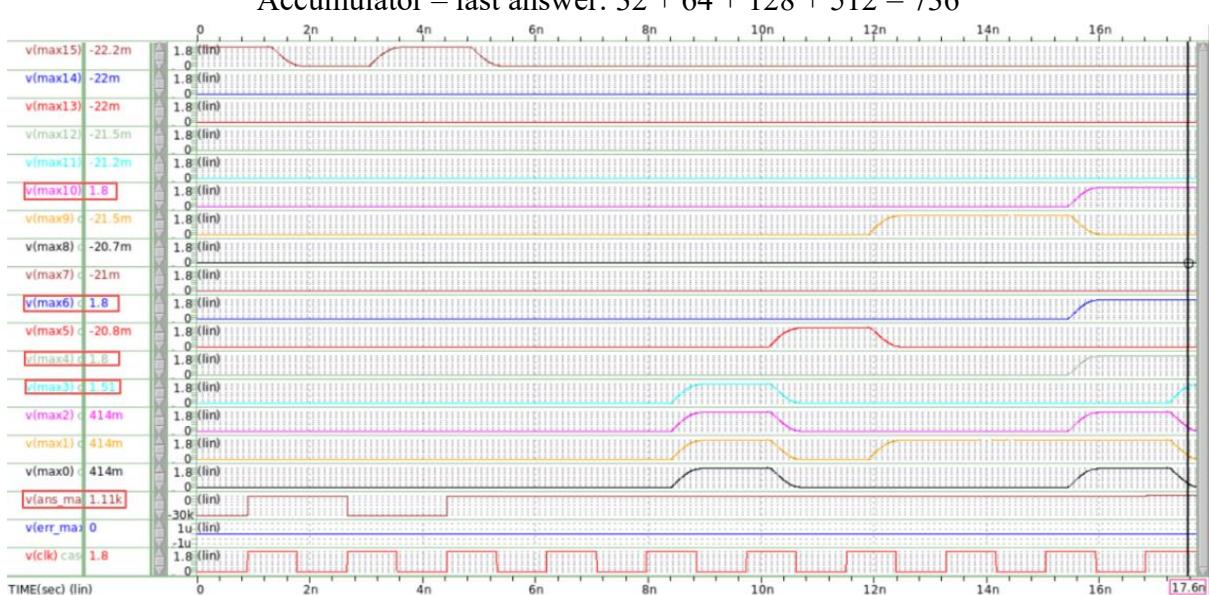


min – the answer is wrong.

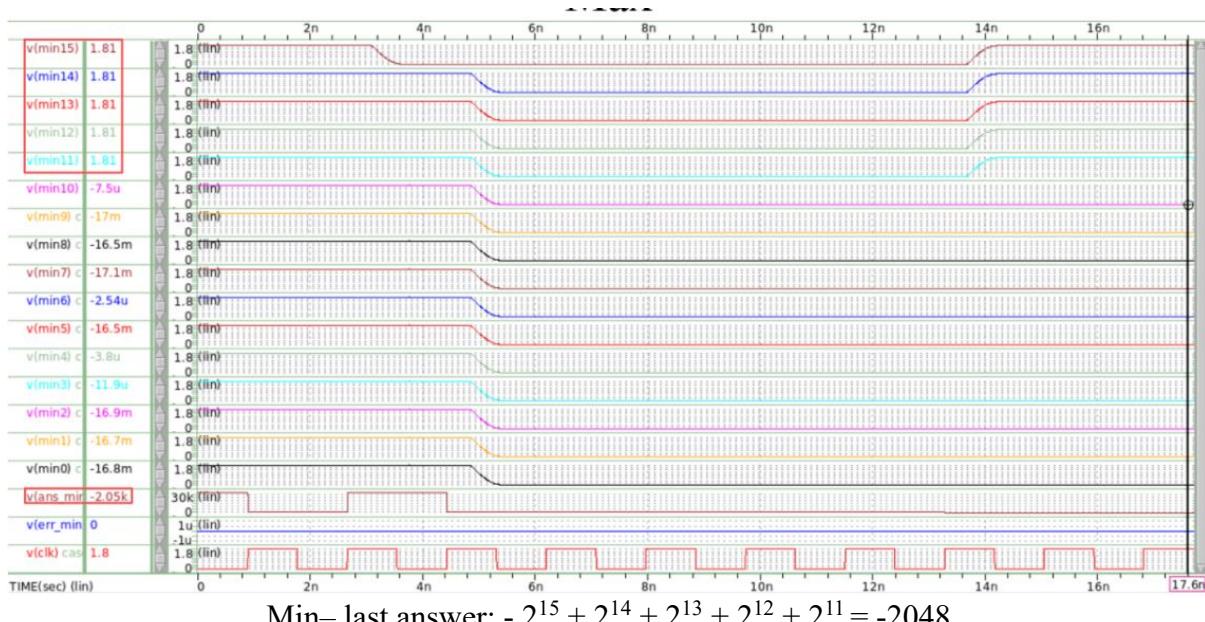
Sim case 2:



Accumulator – last answer:  $32 + 64 + 128 + 512 = 736$



Max – last answer:  $1024 + 64 + 16 + 8 = 1112$



### Post-layout simulation results:

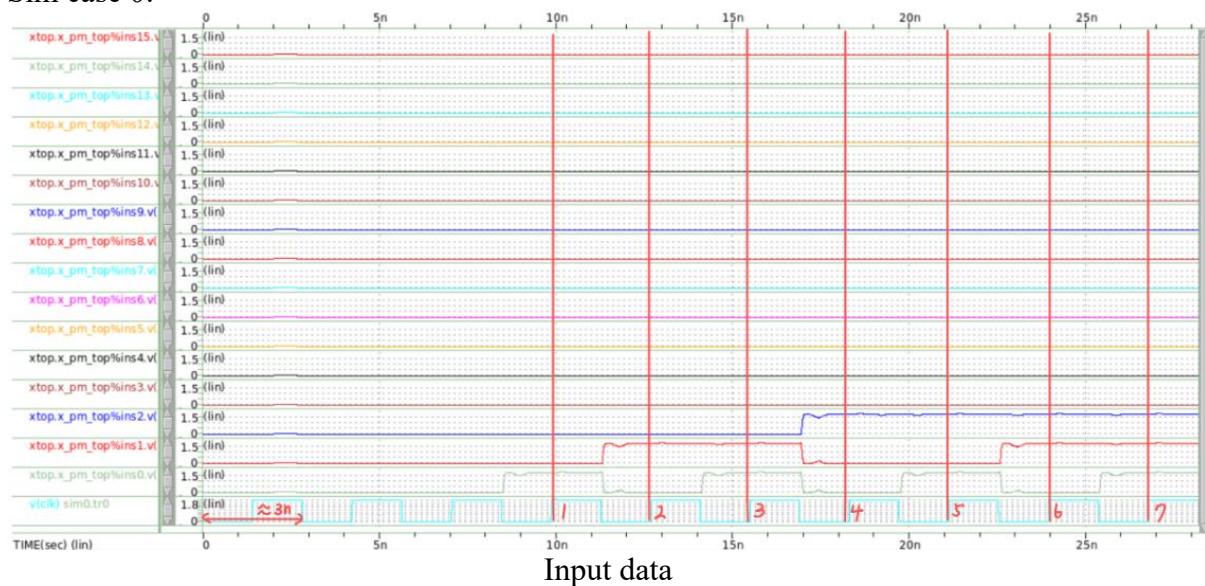
Table. 2

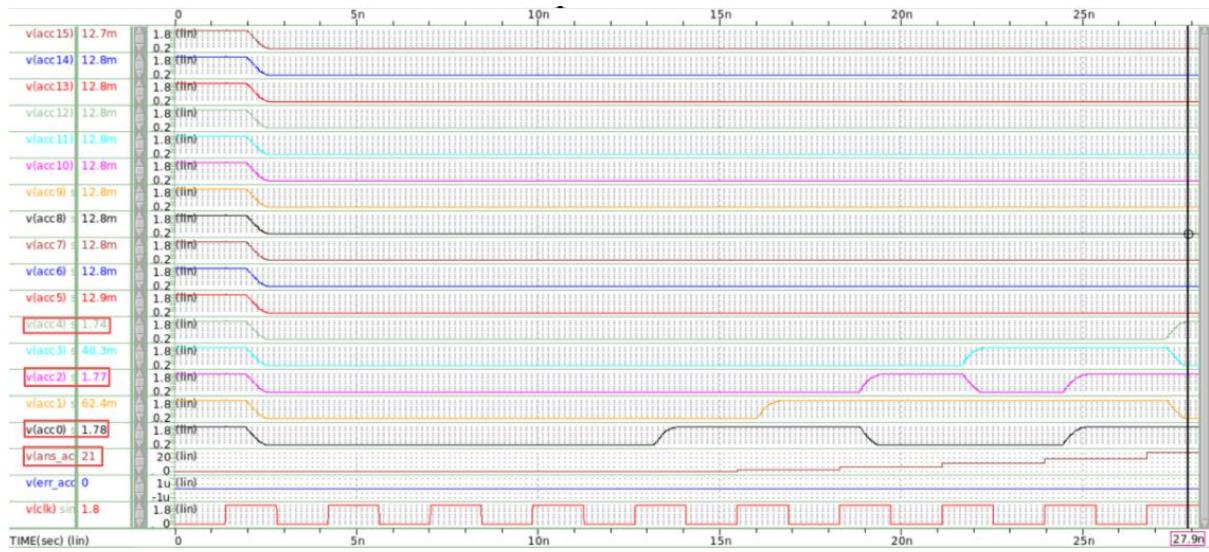
TT 25°C

	Frequency (GHz)	Power (mW)	FoM (Freq / Power)
Sim case 0	0.355	4.5829	0.0775 M
Sim case 1	0.63	7.5014	0.0840 M
Sim case 2	0.365	8.2254	0.0444 M
Layout Area ( $\mu\text{m}^2$ )			42892

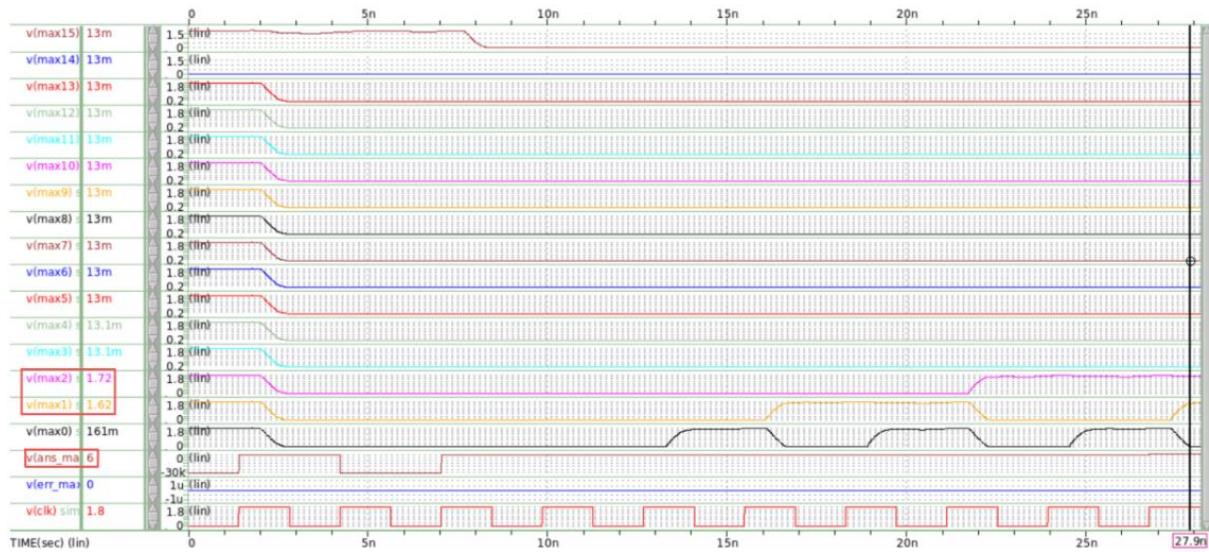
Waveform analysis for TT corner 25°C:

Sim case 0:





Accumulator – last answer:  $1 + 4 + 16 = 21$

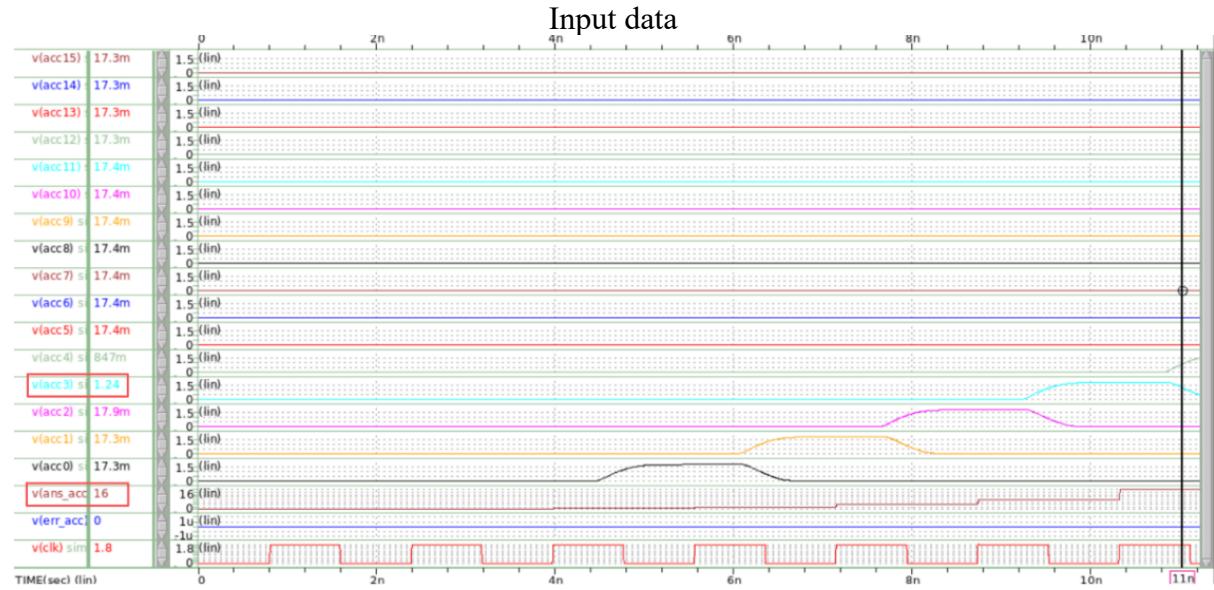
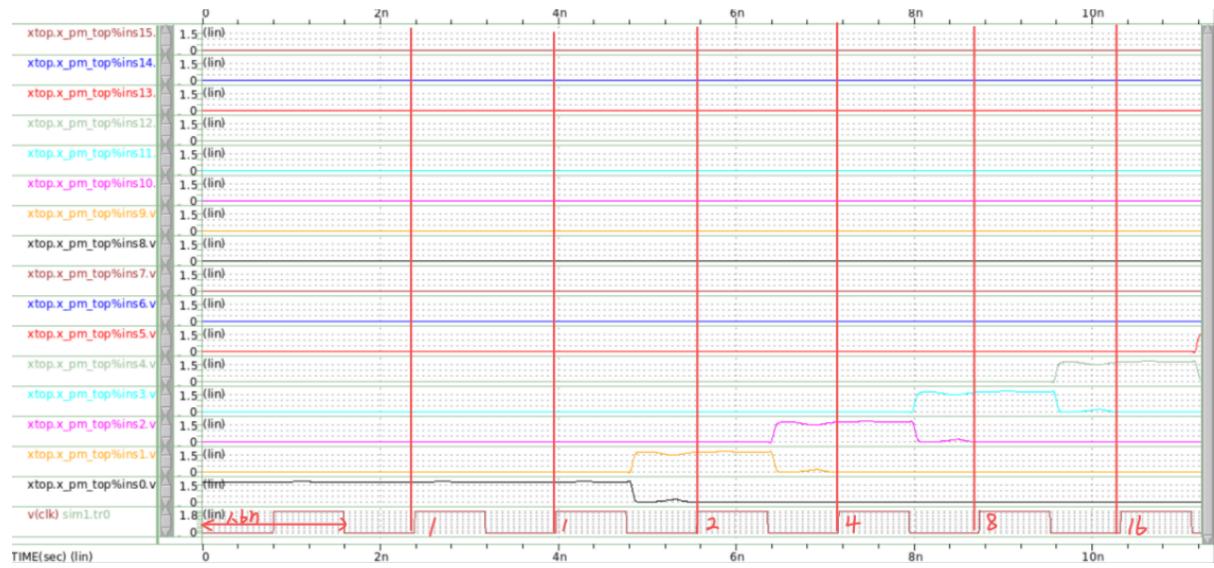


Max – last answer:  $2 + 4 = 6$



Min – last answer: 0

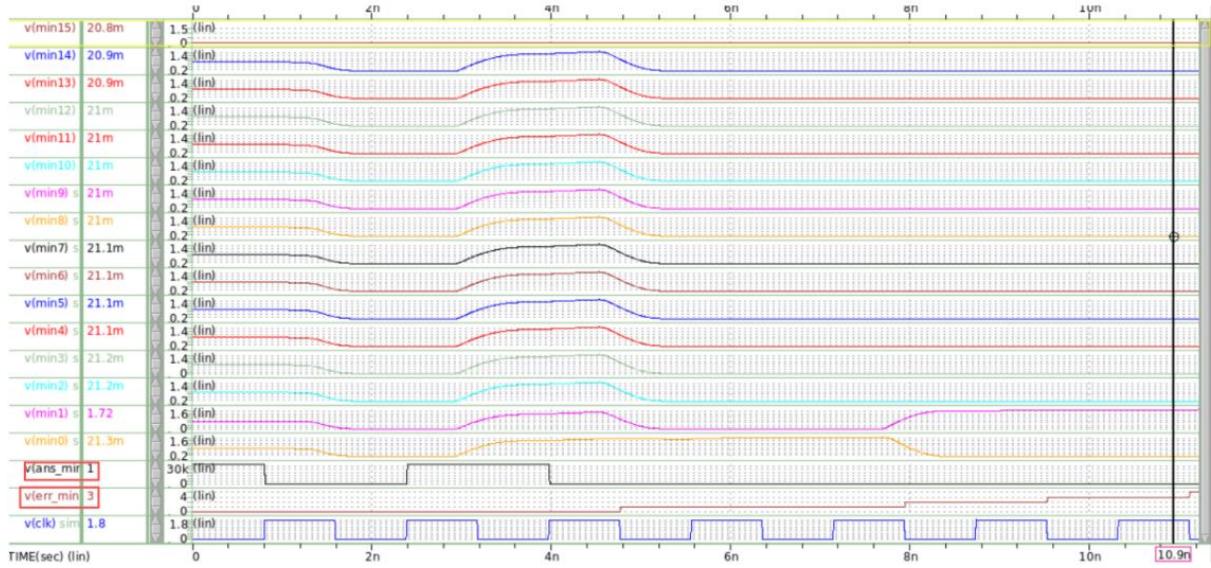
Sim case 1:



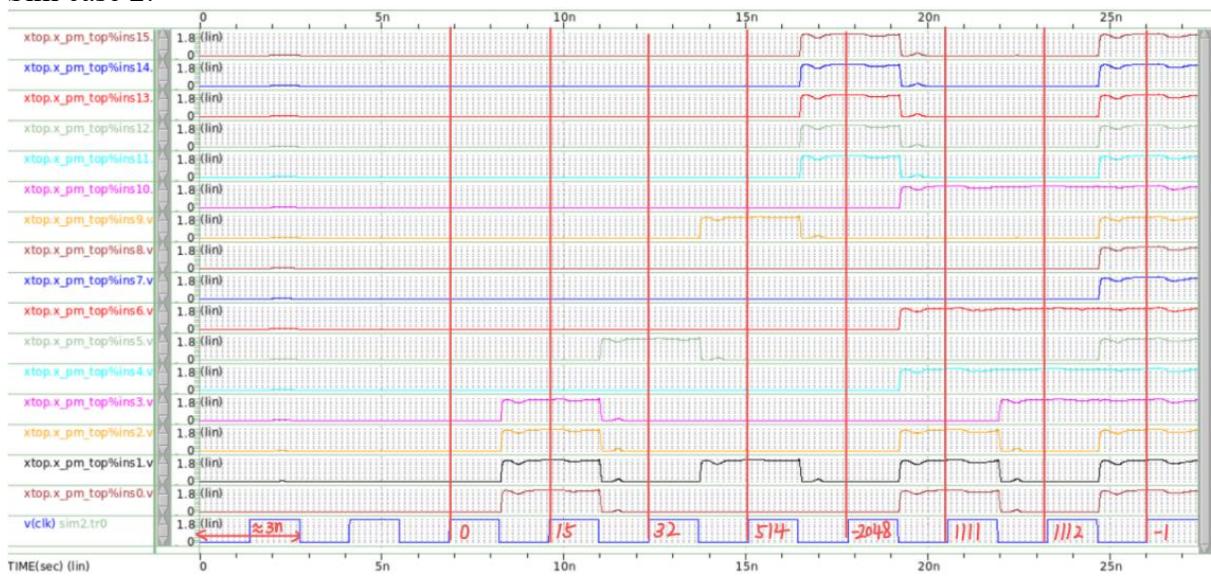
Accumulator – last answer:  $2^4 = 16$



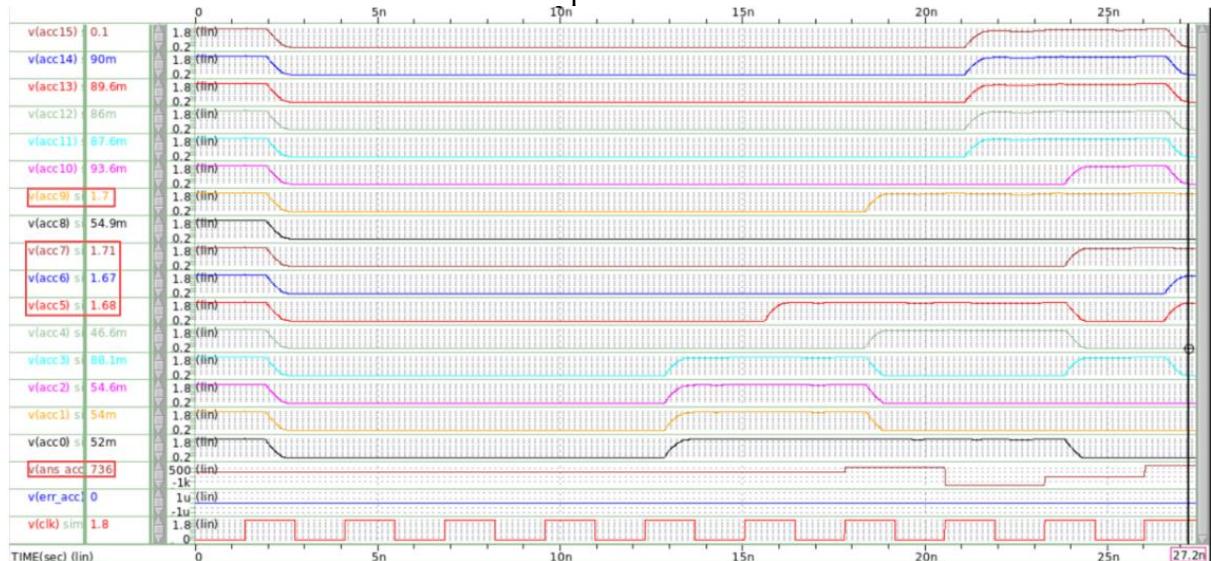
Max – the answer is wrong.



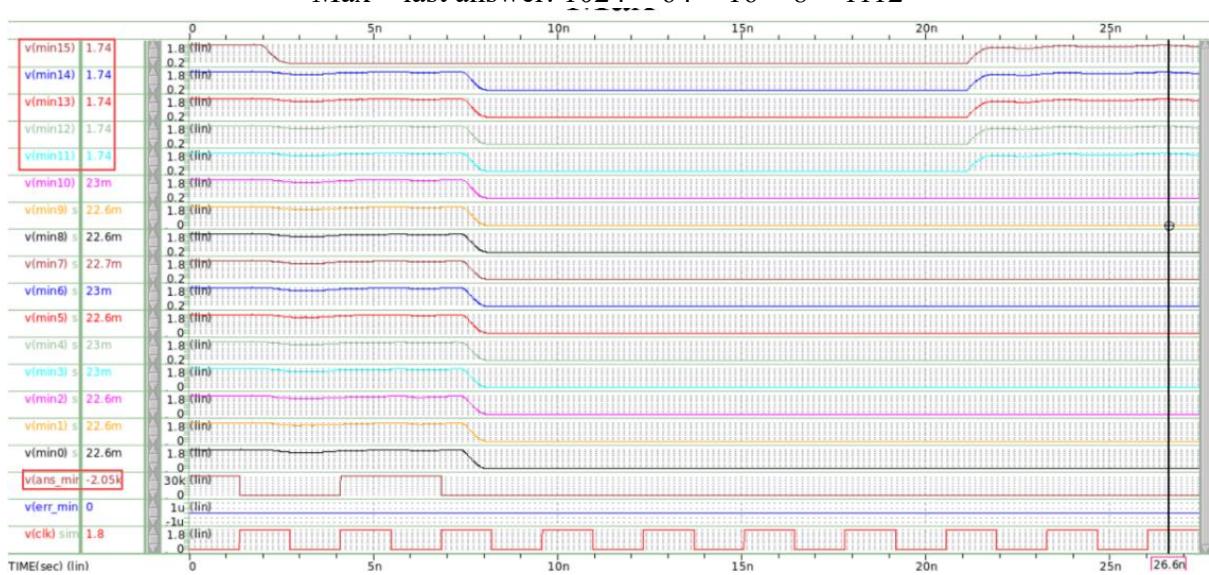
Sim case 2:



Input data



Accumulator – last answer:  $32 + 64 + 128 + 512 = 736$



#### Waveform observation:

- Since we only run  $10*tclk$ , we can not feed all values from testbench to our circuit.
- The speed is faster in pre sim. We can see that the clock period is about 3ns in case 2 post sim result and the clock period is about 1.8ns in case 2 post sim result.
- When the frequency is high, the voltage may not be perfectly 1.8V or 0V when sampling since there is not enough time to charge or discharge capacitors at each node.
- We can see that the results of accumulator, max and min are not included the input data on current cycle.

Table. 3

Sim case 2 at 0.1MHz		
Process	Temperature (°C)	Power (uW)
FF	-40	2.5170
SS	125	3.1144
SF	25	2.7871
FS	25	2.8810

### **Pre & Post-layout simulation comparison:**

From comparing the max operating frequency of each sim case for pre-layout and post-layout simulation, we can clearly see that there is a significant drop in the post-layout case. For sim case 2, the pre-layout max operating frequency is 0.565GHz, where it drops 0.2GHz to 0.365GHz for the post-layout case. This is inevitable as we assume an ideal situation for the pre-layout phase, and account in the parasitic capacitance that comes with parallel metals and internal resistance of wires by running PEX (R+C+CC) for post-layout simulation.

When looking at the power consumption for the post-layout simulations, we see that the total power consumption of sim case 0 and 2 sees a slight decrease compared to the pre-layout simulations. However, in theory, the power consumption of the post-layout case is supposed to be larger than the pre-layout case, as there are additional resistance and capacitance in the layout design, which consumes more energy. If we calculate the FoM by frequency / power, we can see that the post-layout simulations actually exhibit a lower FoM. This could be clearly explained, as the total power consumption increases with the operating frequency, thus, by comparing the FoM, the post-layout simulations does indeed have a worse performance compared to pre-layout simulations.

As for the simulations results of post-layout for different corners, we can see that at SS 125°C our design consumes the most power, while FF -40°C consumes the least power under the same frequency and sim case ( $f = 0.1\text{MHz}$ , sim case 2). This is due to the difference of corners and propagation delay of gates under different temperatures. At SS corner both NMOS and PMOS are operating at their slowest operational state. High temperatures (125°C) increases the leakage current in transistors as it slows the carrier mobility. Slow transistor switching (increasing dynamic power due to longer transition times and higher capacitance charging/discharging times) and high leakage currents (increasing static power) at high temperature leads to the highest power consumption in the SS 125°C case, vice versa for the FF -40°C case which has the lowest power consumption.

### **Personal opinions**

Overall, we quite enjoyed the problem of this final project, from the design of adder through reviewing designs mentioned in class and surveying papers for better designs, to analyzing the pushing capability and load of each node and adding inverter chains to better improve the performance of our design, it was a great ending to this semester of VLSI. However, there are a few problems we face which significantly affected our overall experience during the 2 weeks of completing this final. First, the specification mentioned in the final project pdf wasn't as clear as we would have liked, from how many buffers we should add at each input to the problem of adder overflow and the change of testbench, we find it quite confusing at each step as we wouldn't be sure whether to continue with our current design or wait for TA's or professor Hsieh's response. Second, when designing the gate sizes of each block, we found that not following the sizing taught in the lecture for NAND3, NAND4 leads to a better performance for frequency, which is quite interesting. Last, the problem with EE workspace at the last few days before the deadline is annoying to us student, as we weren't able to run simulations due to overflowing number of students connecting to the workspace, our testbench was once overwritten to completely blank, and laker couldn't connect to our finished library file. Although we ended up finding solutions to each of the problems mentioned above, we still think this is something that needs more work on from our department. At the end, we would still like to thank professor Hsieh and all TA's for this semester's VLSI course and all the effort put in.

### **Reference**

- Fast Mux-based Adder with Low Delay and Low PDP – H. Tavakolaee, Gh. Ardeshir, Y. Baleghi 2018. (DOI: 10.22044/JADM.2018.7177.1853)
- Design of Optimal Fast Adder – Pavan Kumar.M.O.V, Kiran.M 2013. (DOI: 10.1109/ICACCS.2013.6938692)
- The Fastest Carry Lookahead Adder – Yu-Ting Pai, Yu-Kumg Chen 2004. (DOI: 10.1109/DELTA.2004.10071)

