

# Project Report

朱宇轩 2020531016 胡天衢 2020531038

*The two authors contributed equally.*

## 1. Basic Part

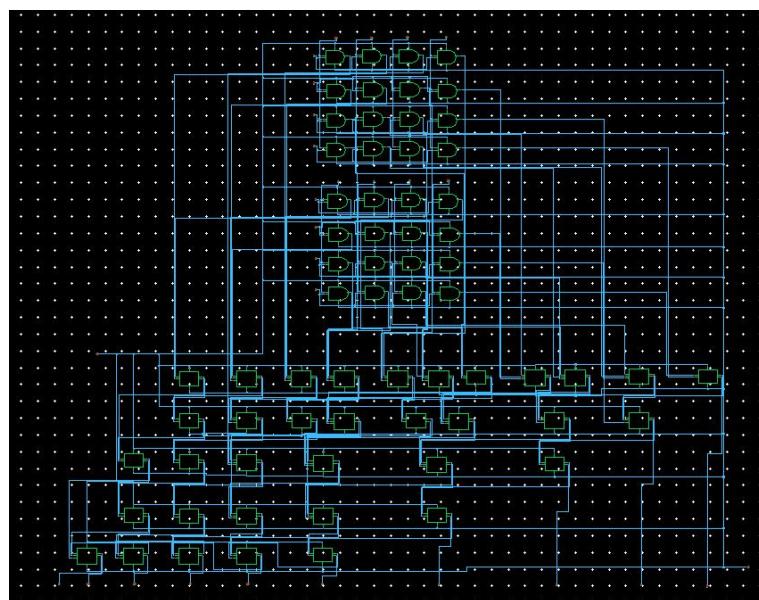
In the basic part, we are required to design a unit which can complete the vector multiplication.

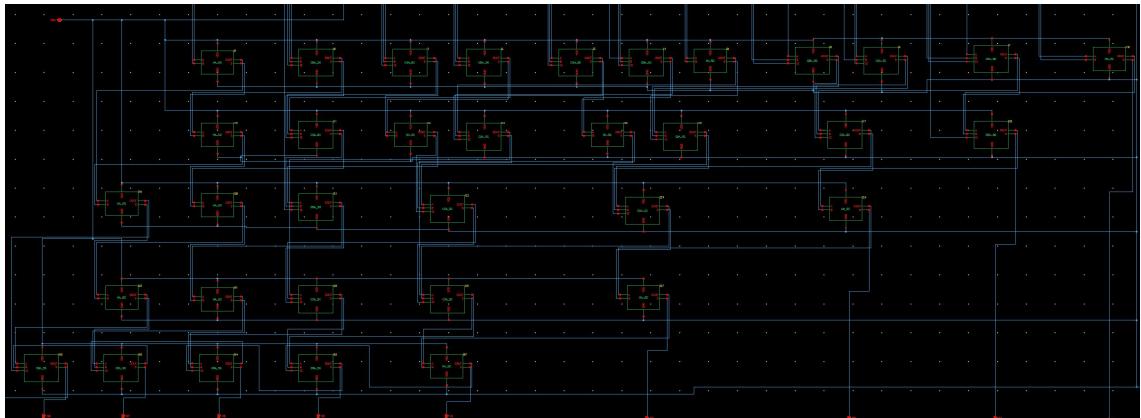
### 1.1 Structure

The guideline gives out the structure of vector unit, consisting of 2 multipliers and an 8-bit adder.

#### **First design:**

As the Wallace tree gives out a perfect way to add data with the same weight, we hope to use this structure (to be discussed in the next part) to speed up the circuit. We firstly change the total structure and make the 2 multipliers and the adder an overall unit.



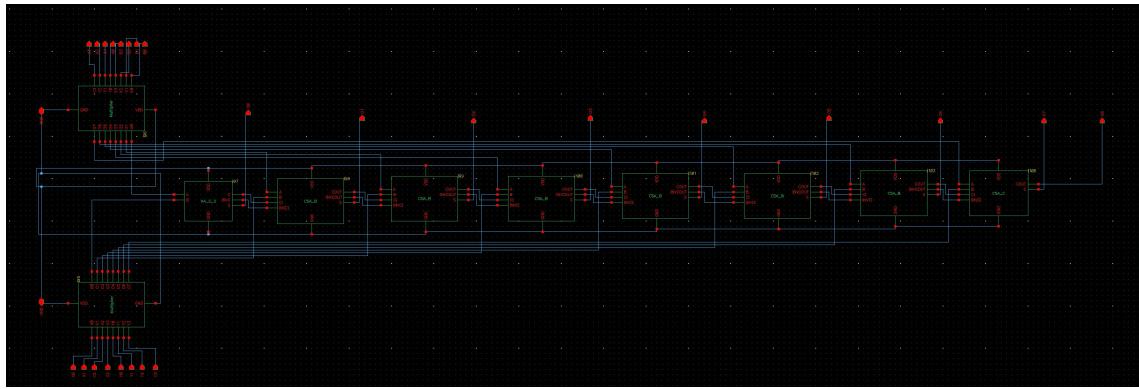


After testing, though the number of circuit parts seems to decrease, the delay time is even longer. Because in the original structure, the 2 multipliers calculate parallelly at first and the Wallace tree's  $\log_{\left(\frac{3}{2}\right)}$  time complexity advantage doesn't seem apparent.

### **Final design:**

Though the previous design turns out to be a failure, the effect of Wallace tree in multiplication still exists.

Therefore, we just change the structure of multiplication and set up a 4-bit Wallace tree multiplier.



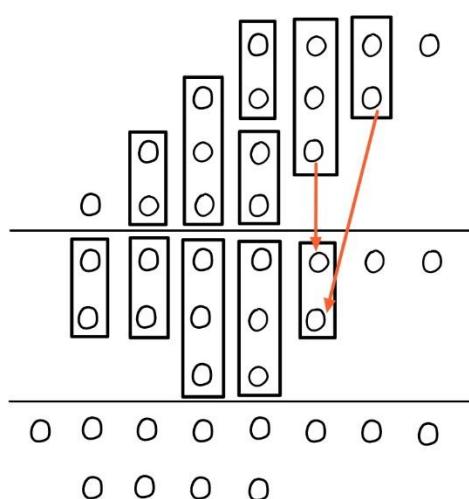
**The selection and design of components will be discussed in another part.**

## 1.2 Algorithm

Here, we lay emphasis on the principle of Wallace tree.

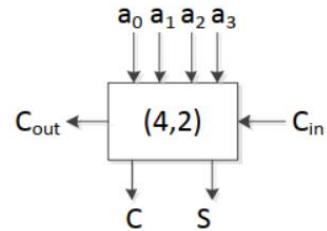
When do multiplication, we use every bit of a multiplier to multiply the multiplicand, the result is called partial product.

For a 4-bit multiplier, we can list out all the total partial product. Then, the Wallace tree takes effect, for the partial products with the same weights, it merges 3 of them and gives out 2 new bits, one for the current weight and one for the higher weight.



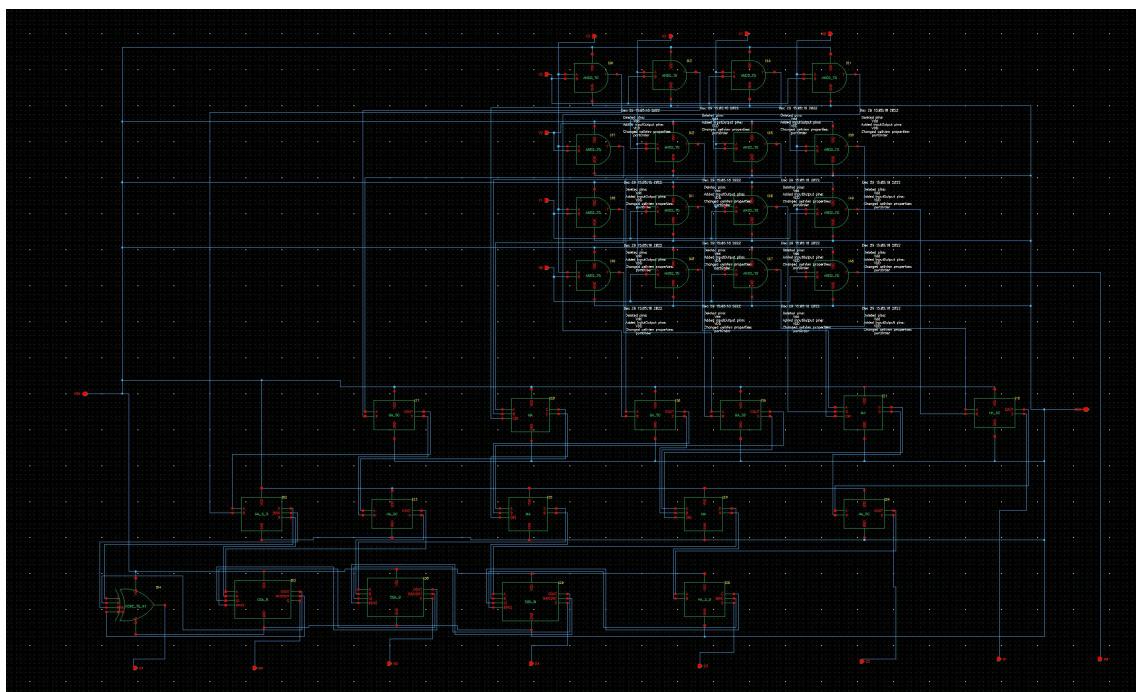
Consider the above process, it can actually be realized by a full adder (which has the same function).

There also exists 4-2 compressor, which can take in 5 inputs in total and gives out 3 outputs. It boosts the efficiency of Wallace tree further, but the layout can be quite more complicated.



### 1.3 Circuit design

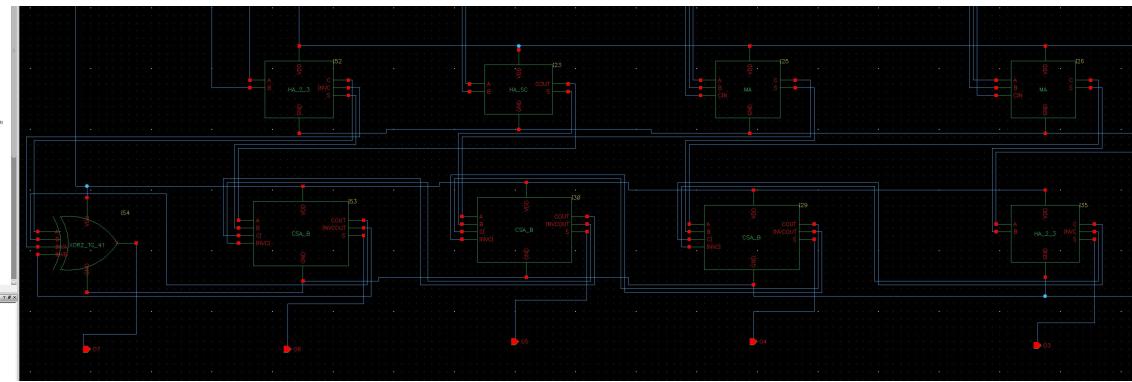
From the 4-bit Wallace tree structure in the last part, we turn it into circuits.



In our test, we find that if all the inputs for the adder are given at the same time, **the carry-select adder is much slower than the mirror full adder.**

However, the carry-select adder has an advantage in parallel calculation. For more-bit calculation, if all the inputs are given at the same time, the delay for higher bits is just the same as a MUX.

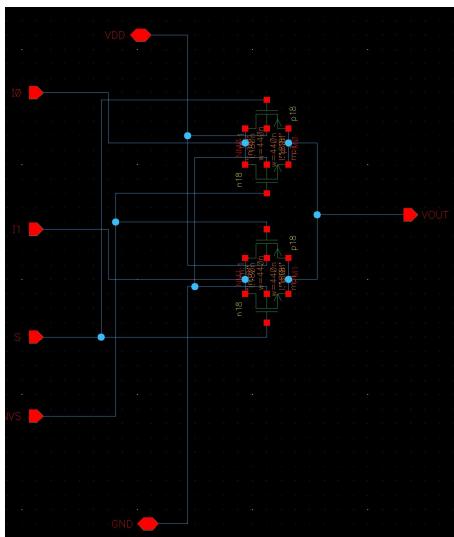
Therefore, for the intermediate adders in multipliers, we use mirror adder and for parallel calculation, we use carry-select adder.



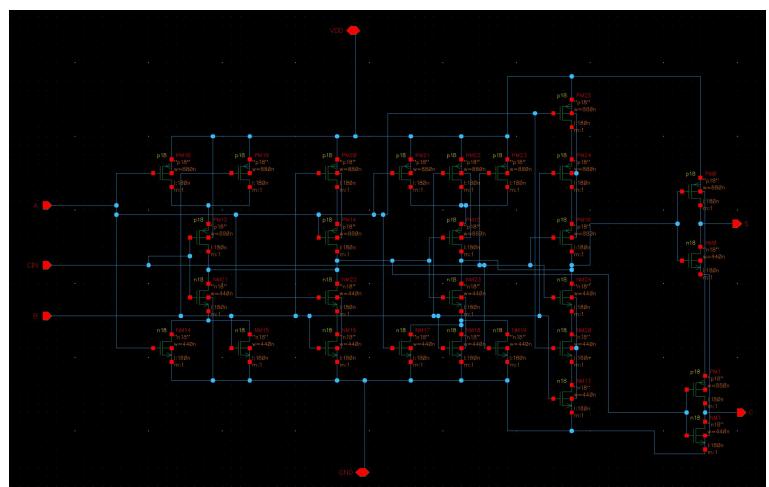
Then, for the MUX, we find that TG MUX has a better performance than Static CMOS.

As most calculation in half adder and full adder not only take use of the signal of carry, but also its inv  $\overline{\text{Carry}}$

To reduce and reuse the inverters in components, we change their structure and give out S,  $\bar{S}$ , Carry,  $\overline{\text{Carry}}$

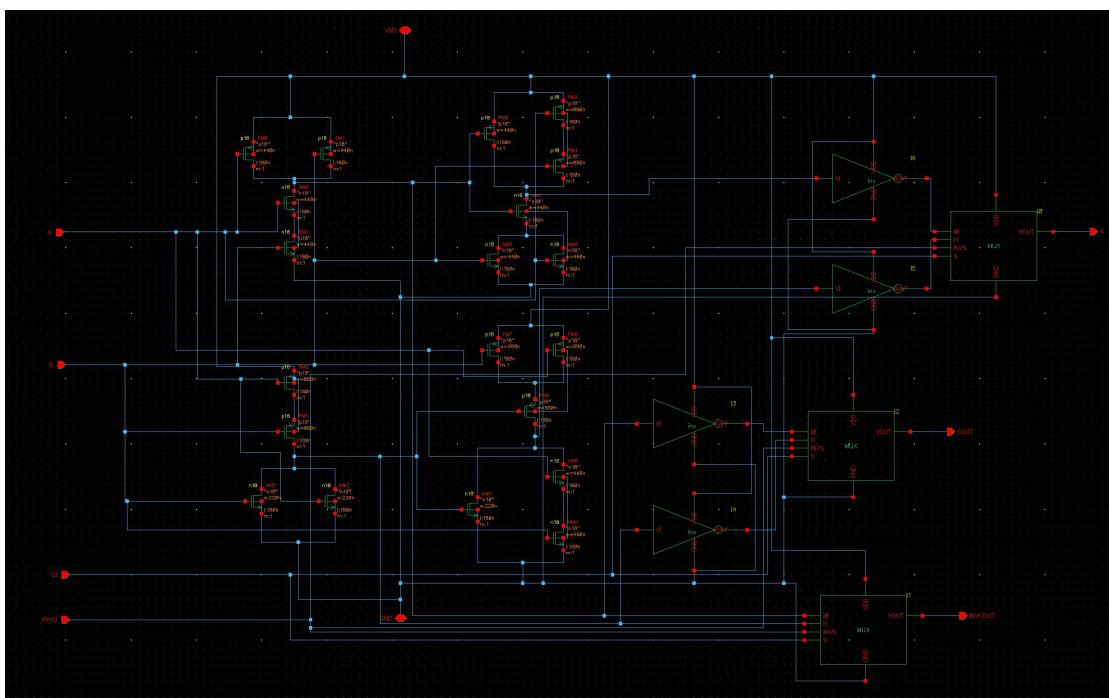


MUX

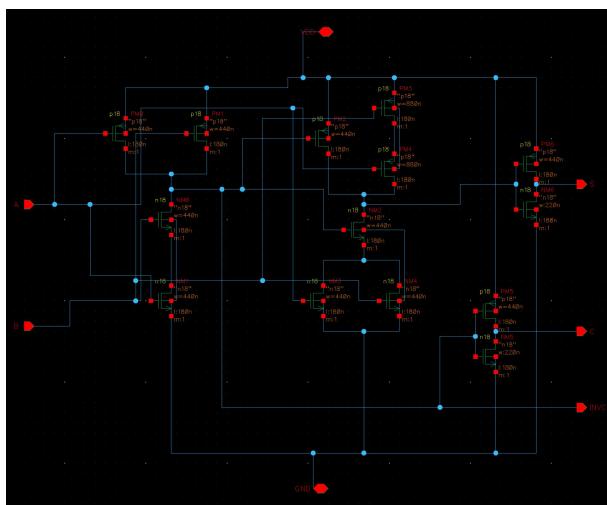


Mirror adder

Carry-select adder

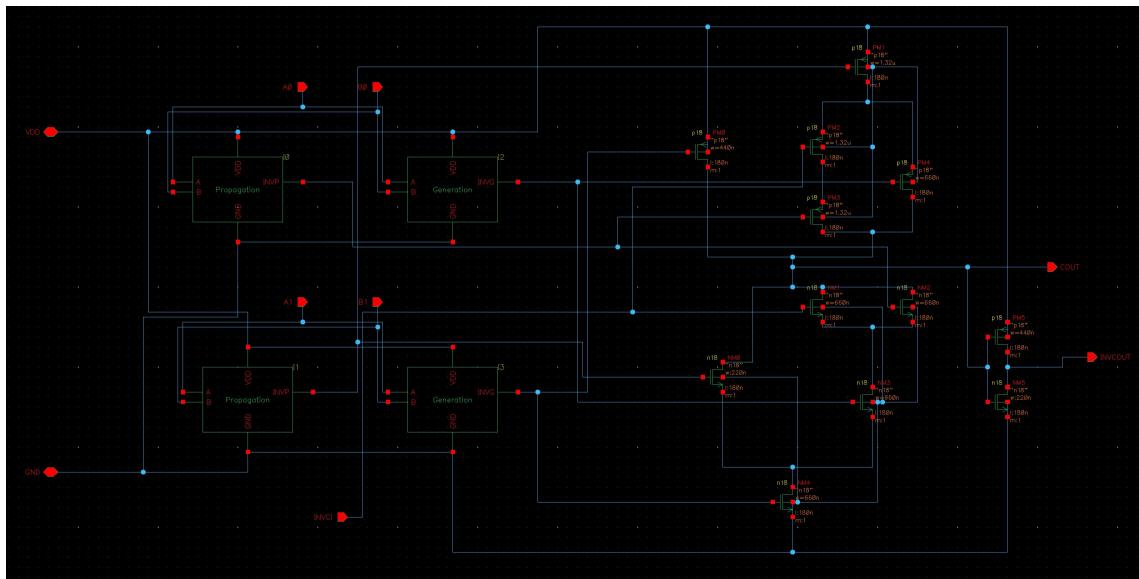


Half adder



Whether the speed of multiplier the faster, the better turns out to be a question. From the structure, we find that if the multiplier speeds up, the signal still needs to wait for the carry in the adder. Therefore, speeding up the adder is also necessary.

Though we have already choose CSA for parallel adding, I also tried 2-bit carry-look ahead adder in the adding part, but the effect is little, mainly due to its complex structure.



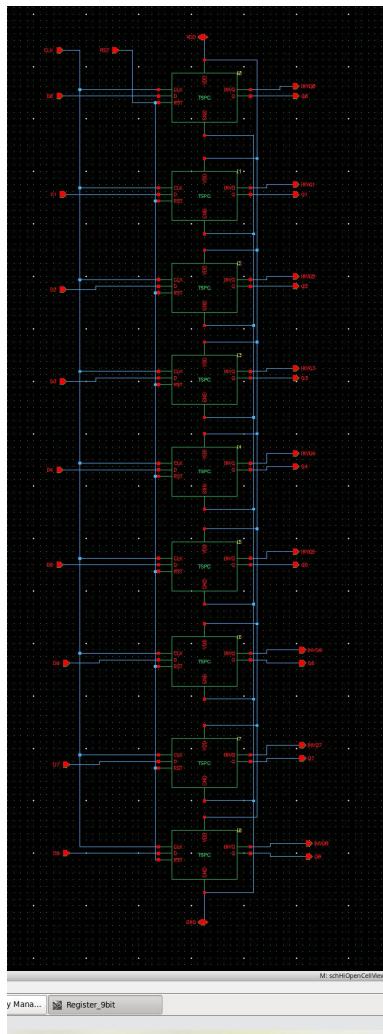
2-bit Carry-look ahead adder

However, due to time limit, I still convince that Carry-look ahead can be a choice. Rather than 2-bit CLA adder, the effect of 4-bit CLA may have better performance. But the cost of time and size of fan-in may also require further improvement.

For the register, as we have to consider the reset function, an AND gate is required. However, the TSPC gives out a method to

embed the AND into flip-flop. Then, I can remove the inverter at the output Q, which speeds up its response.

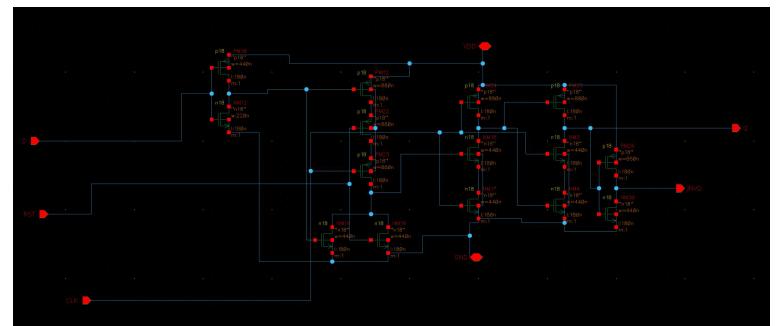
But such operation can cause another trouble: the output Q become dynamic, which presents a 0.1V shake.



9-bit Register

## 1.4 Layout

In layout level, to make the size as small as possible, we hope that all the components in the same level can have the

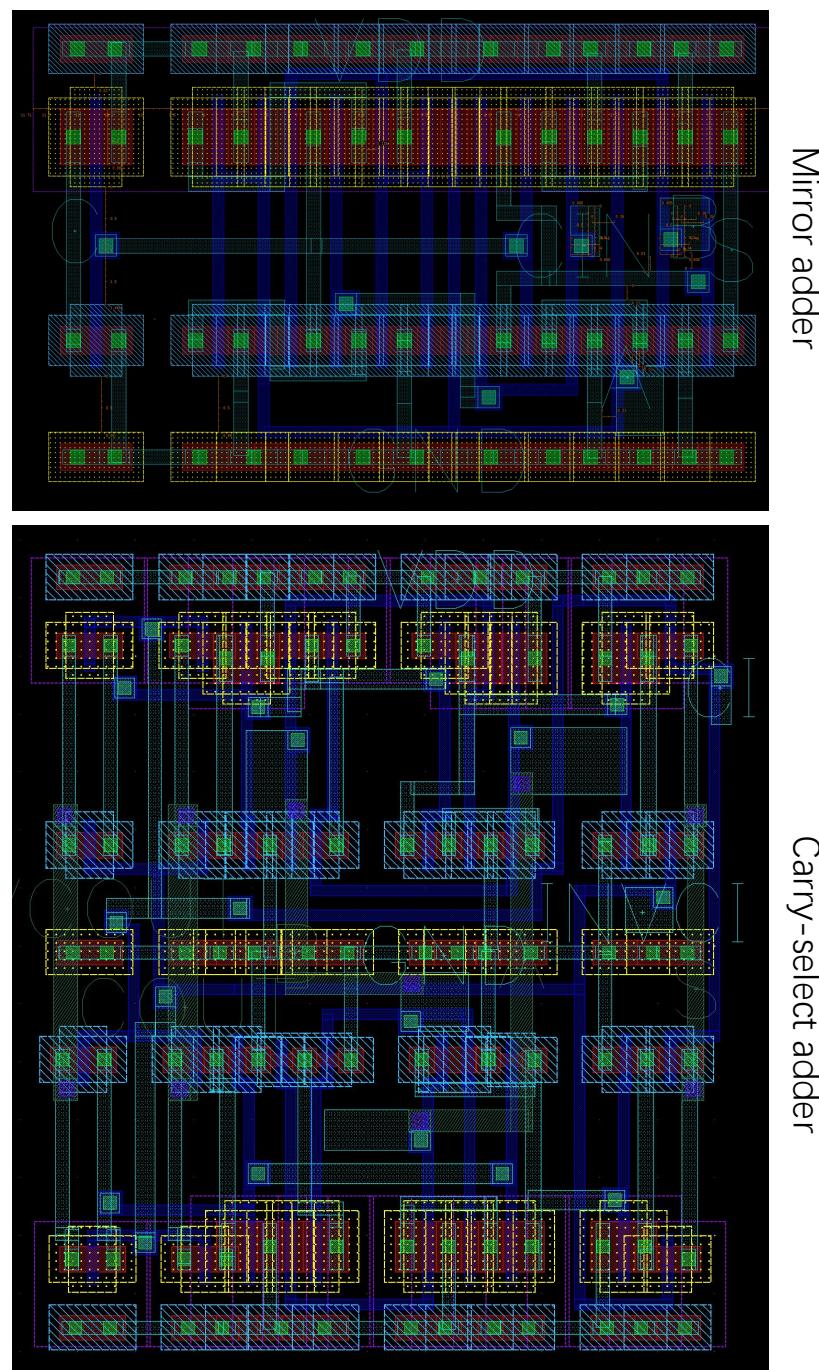


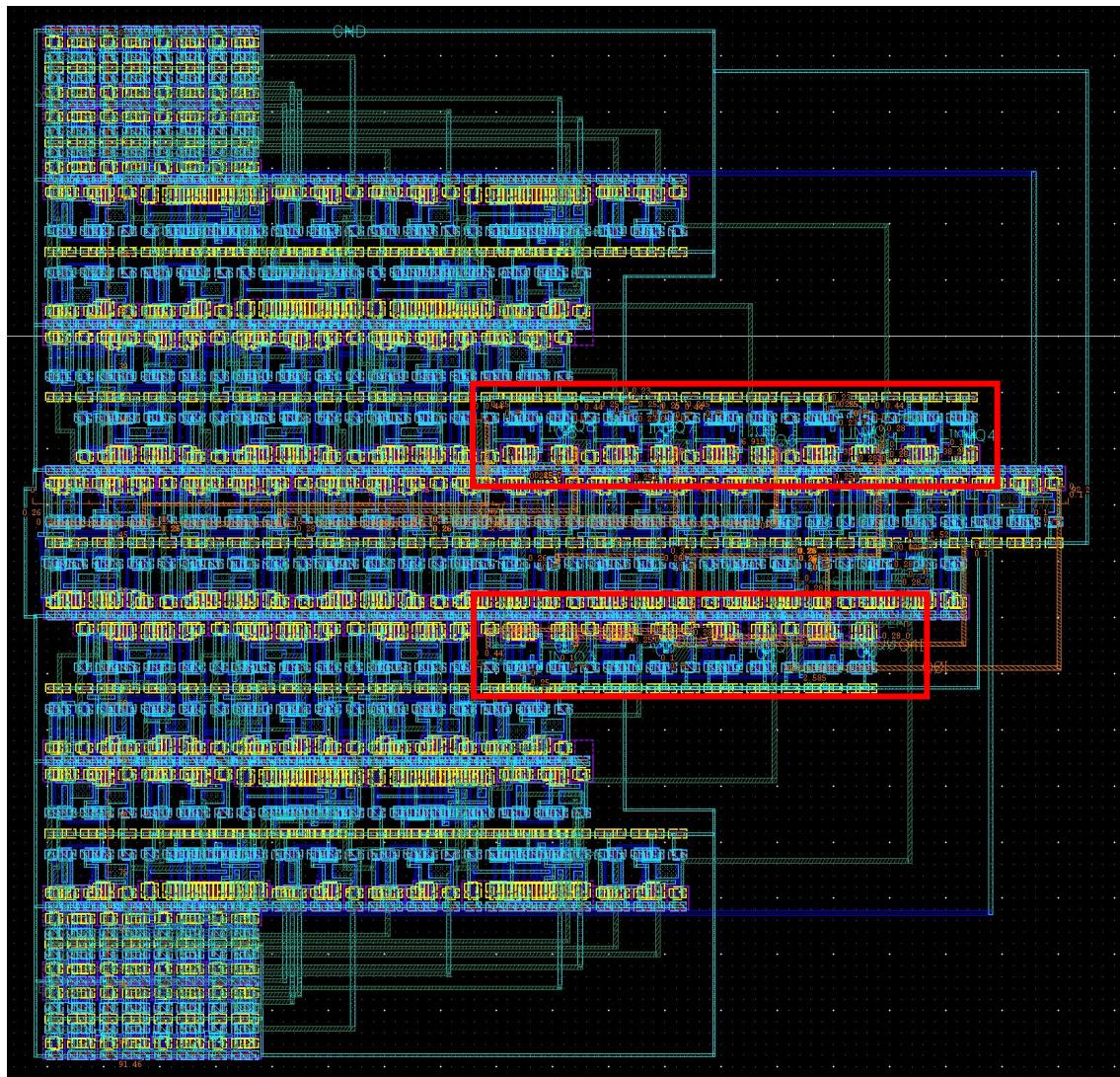
TSPC

same height and the upper and lower levels can share the same VDD or GND.

In this way, we take the most complex structure mirror adder as a sample for all the other components.

For components with TG like CSA, we make a two-level structure, which can also leave space for other components.





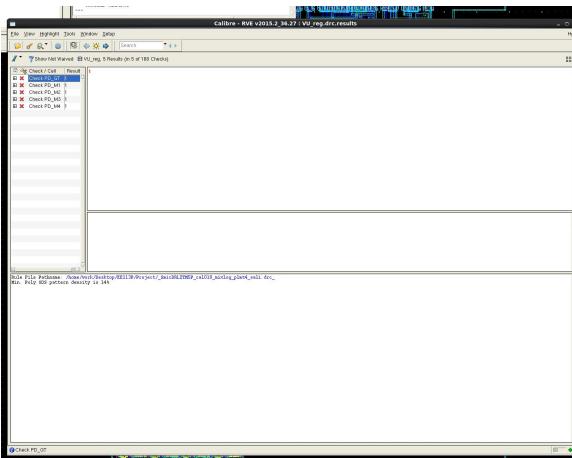
Total layout with area  $91.46 \times 92.2 \text{ } \mu\text{m}^2$

To save area, I separate the 9 TSPC into 2 parts, 5 above and 4 below.

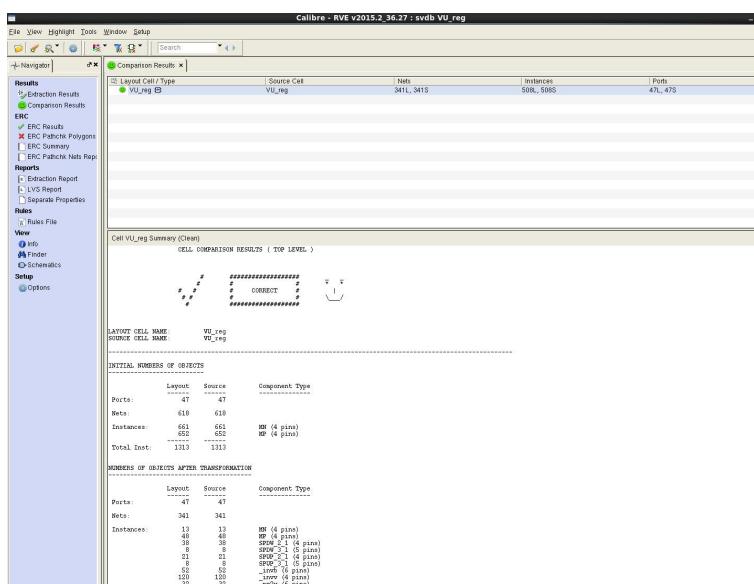
## 1.5 Simulation

Here, we give out our simulation results of Basic part (including lvs and drc).

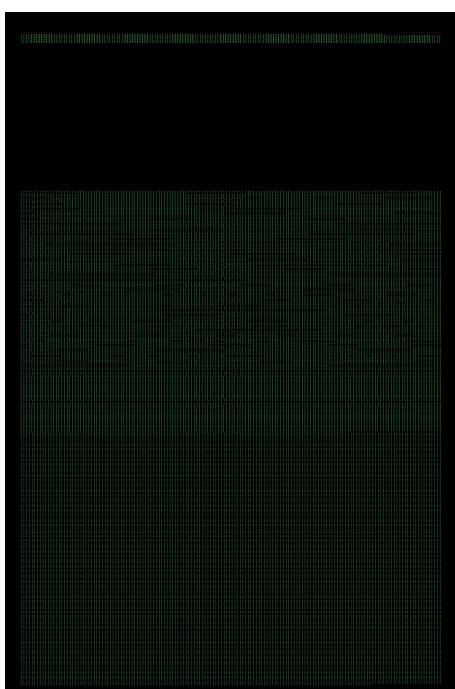
DRC:



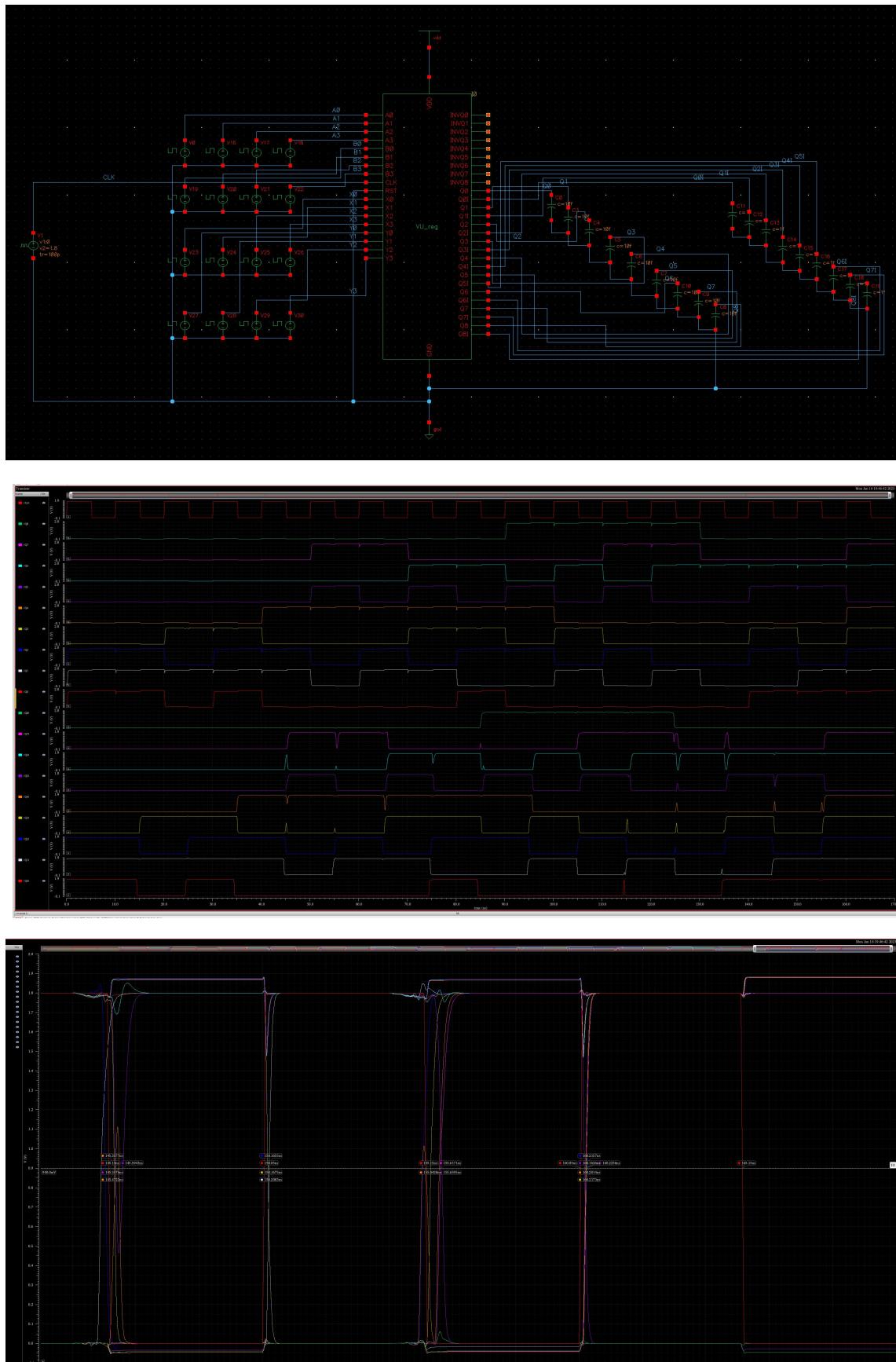
LVS:



PEX:



## Waveform of testbench:

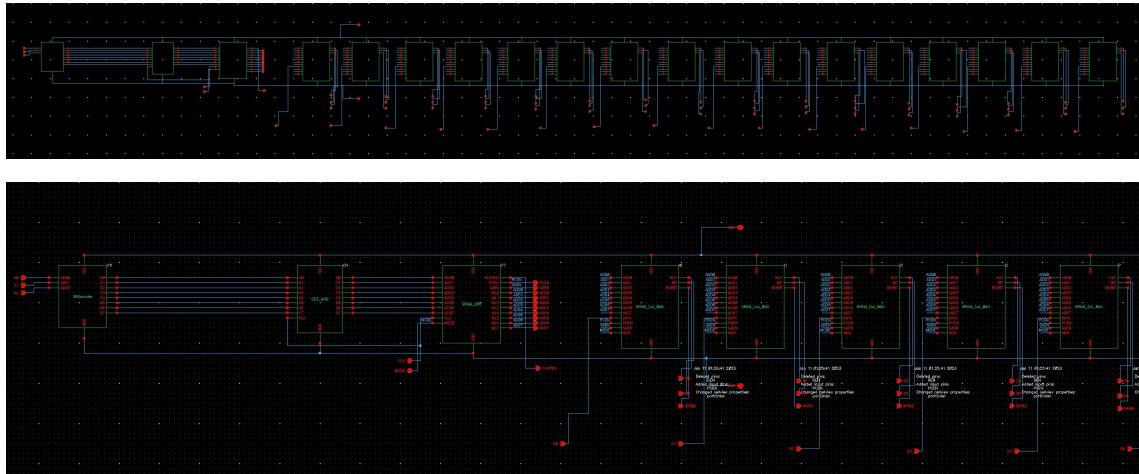


## 2. BONUS part

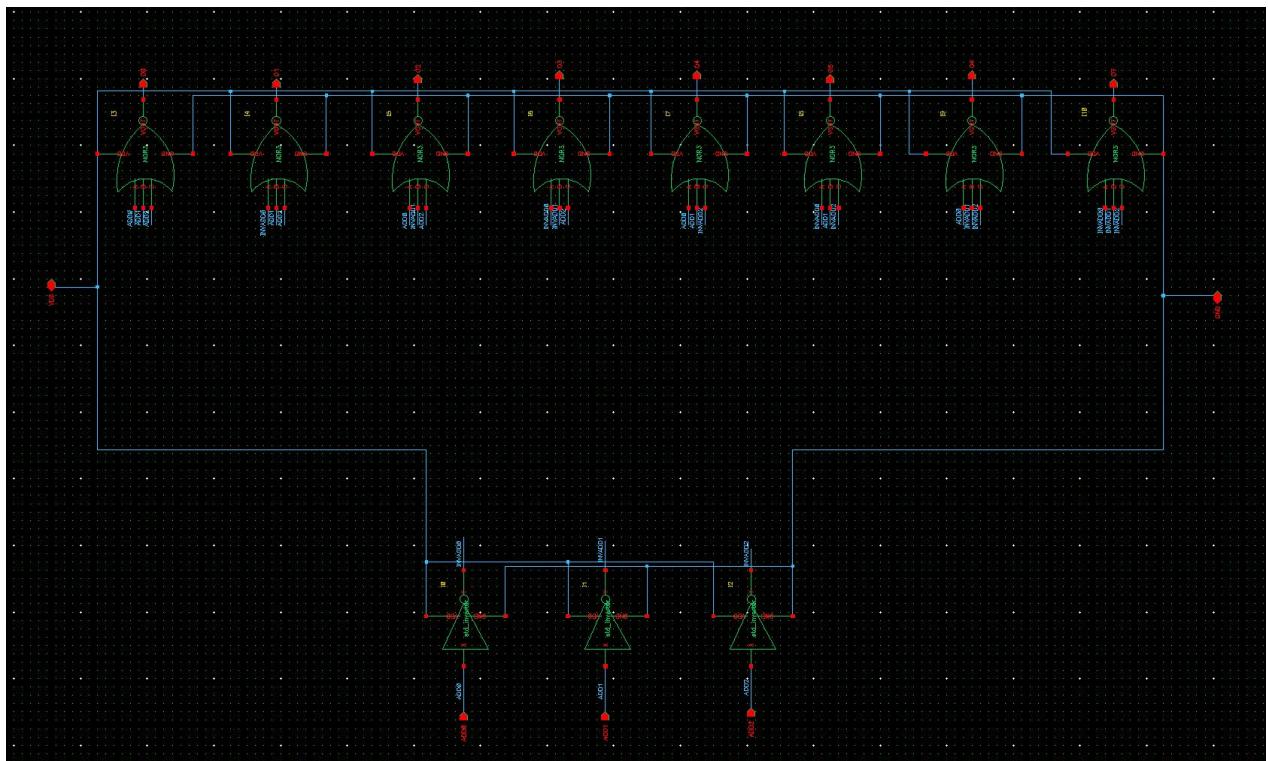
### 2.1 BONUS 1

#### 2.1.1 SRAM structure design

The SRAM consists of 3 parts: 3-8 decoder, SRAM driver circuits and SRAM bit arrays.

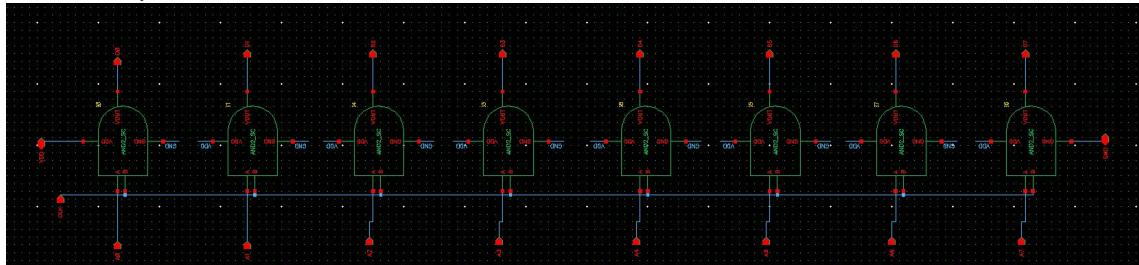


3-8 decoder is made up by Inverters and NORs



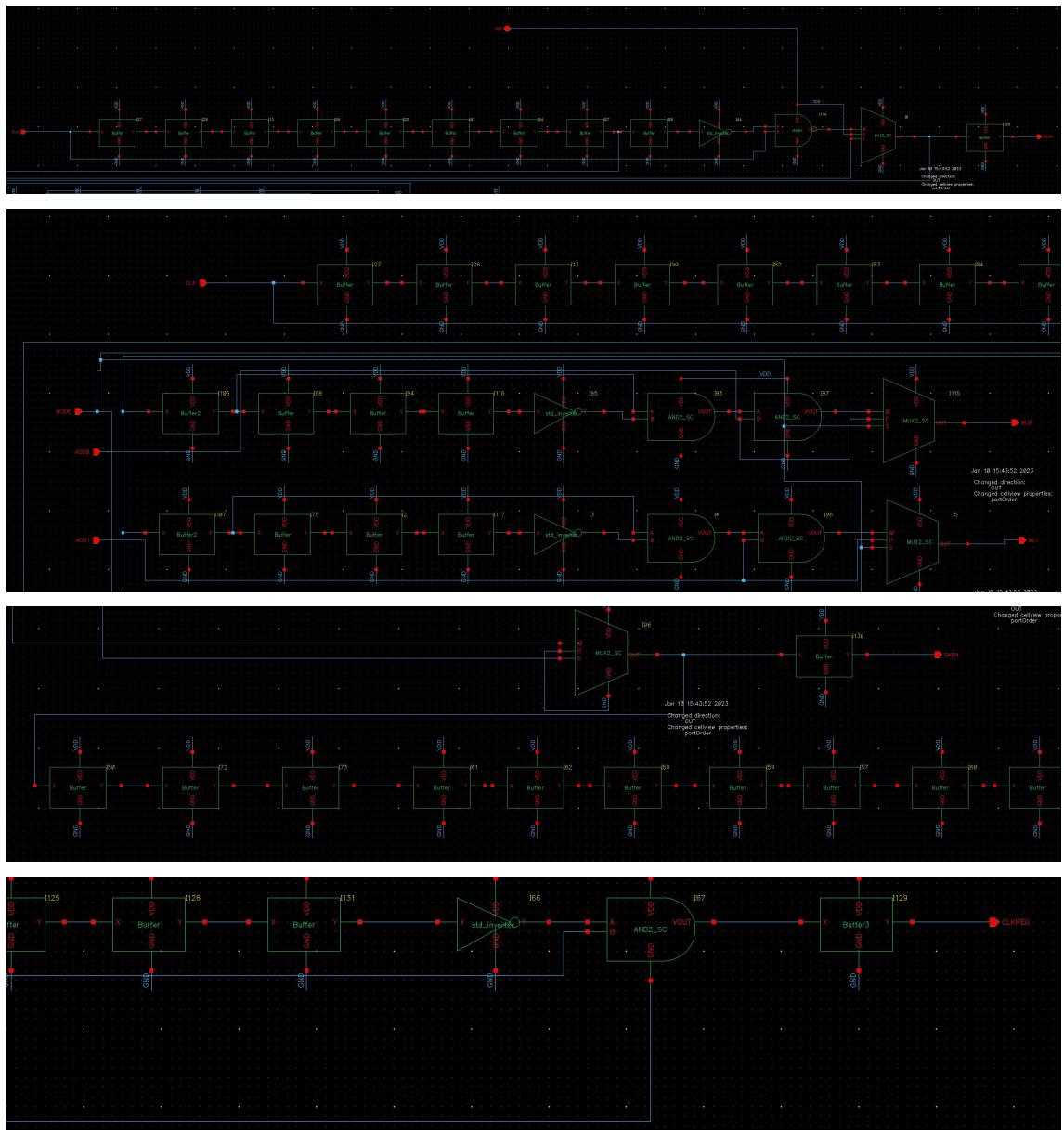
SRAM driver has actually two parts: AND arrays and SRAM\_DRV.

The AND arrays are mainly used for write function. When testing, we find that the address change of 3-8 decoder has certain delay, which may cause error write in previous address. The AND array is connected to CLK on one port which forces the WL signal only open for half clock cycle when writing and solves the problem.



The requirement for timing is quite tricky in SRAM (discussed in detail latter). As there exists continual reading and writing, we can only make pulses through CLK rather than MODE by adding buffers and do AND or NAND operations.





The bit arrays are composed of 16 lines. Each line has a Precharger, Write driver, Sense amplifier and 8 bitcells.

To avoid racing and shut down necessary transistors, I set the size of upper transistors as 220nm, transistors for WL control 440nm and transistors for pull down 800nm.

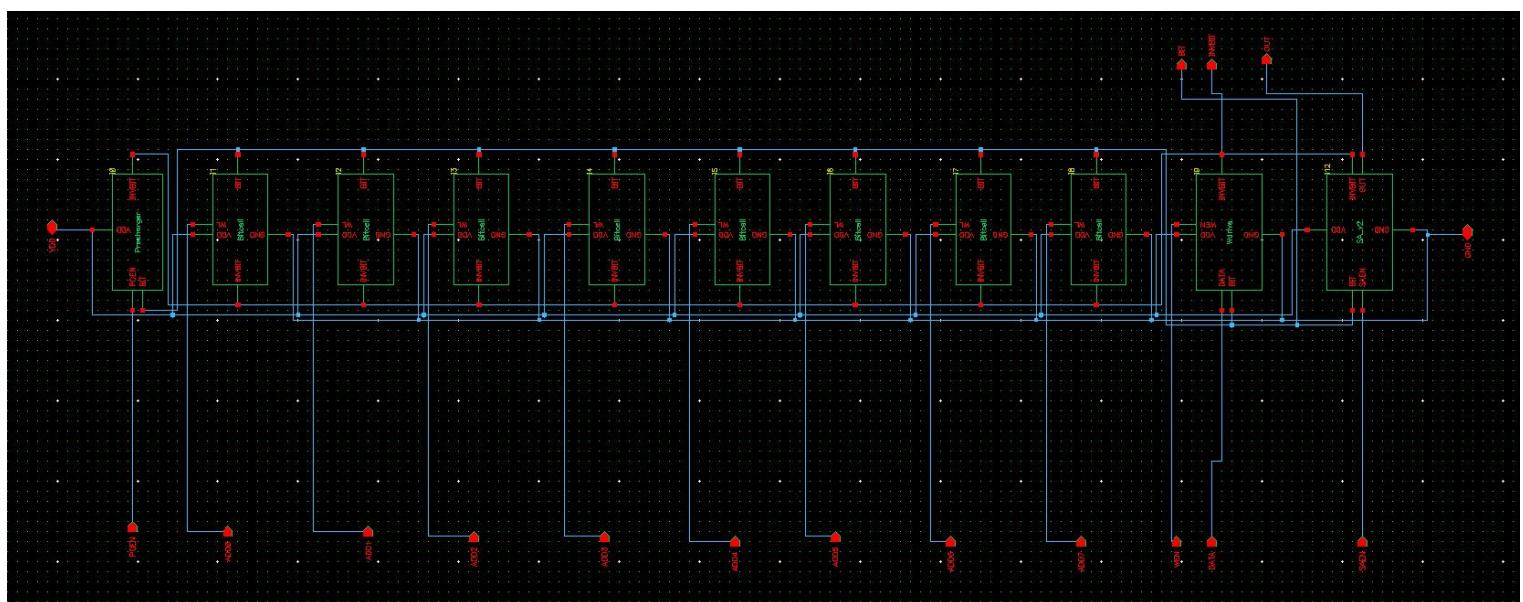
The Sense amplifier is a traditional current mirror amplifier.

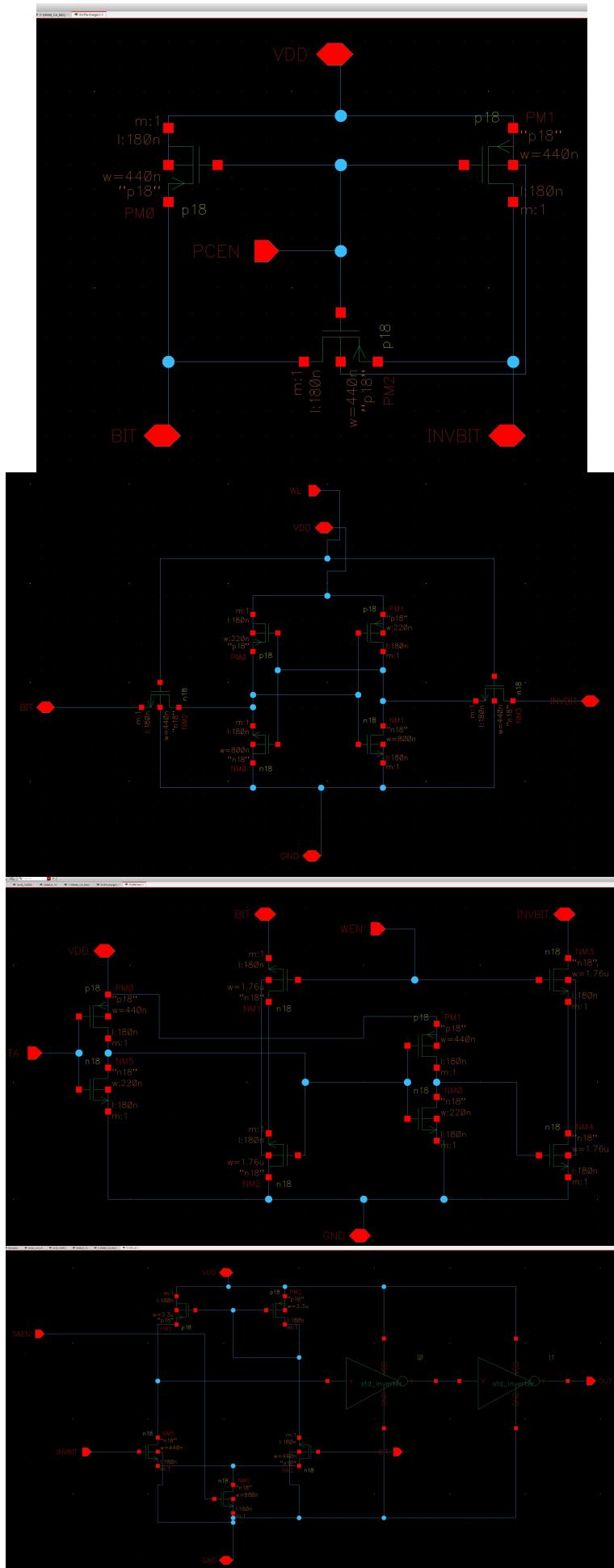
When check with TA, we mainly found 2 problems:

The SA helps the circuit work, but a larger gain is preferred.

While the current mirror SA requires quite large size to achieve this.

Certain buffers in SRAM\_DRV can be reused so that the number of buffers can be reduced.





### 2.1.2 SRAM sequential control

The SRAM's timing is quite complex.

It has 2 modes: write and read.

Write: The WEN and WL of certain address are pull up so that the 0 data are sent to the corresponding bitline or *bitline*.

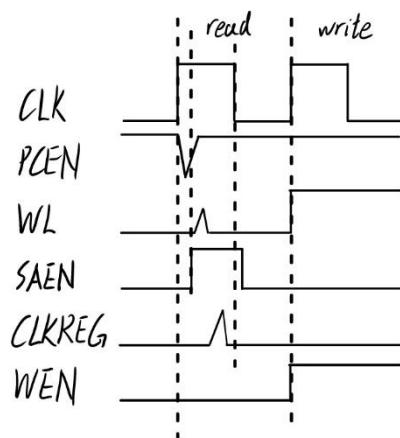
Read: When reading, the Precharger firstly charge bitline or *bitline* to VDD. Then WL will pull down the 0 bitline. But the WL must be a pulse to stop bitline pulling down lower than 70% VDD.

Finally, the sense amplifier amplifies the difference and output to register.

For the SRAM register, the clock should also be generated by SRAM\_DRV so that we can achieve the read-after-write function.

As the output load for certain signals are quite large, buffers are added to help keep signal shape.

The logical expression and timing diagram for control signals are listed below.



### 2.1.3 Testbench

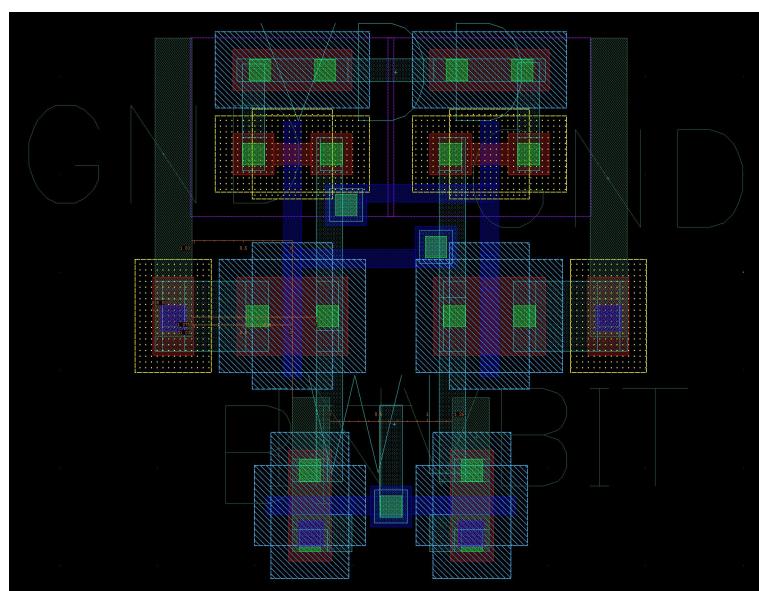


## 2.2 BONUS 2

### 2.2.1 Layout design

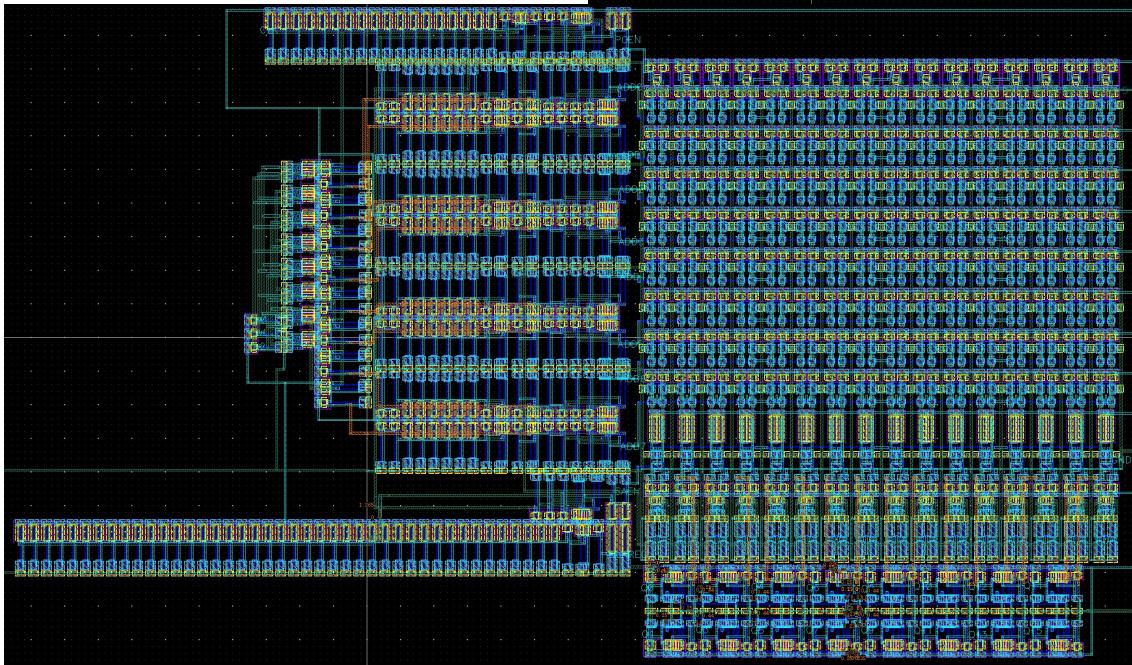
The layout for bitcell uses M2 for GND (on 2 sides) and bitline, *bitline* so that the whole column can share the same GND and bitline, *bitline*

M1 is set for VDD on top and the WL transistors are placed in horizontally. In this way, total row shares the same VDD by M1 and WL by poly (so that the parasitic capacitance on WL can be reduced).



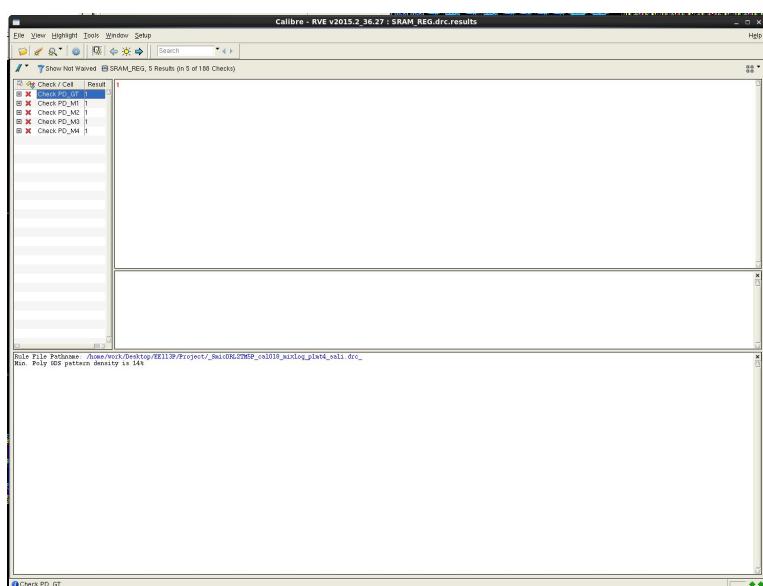
For SRAM\_DRV, as buffer for CLKREG is quite large, we take its size as a sample and draw all the other components.

Total layout:

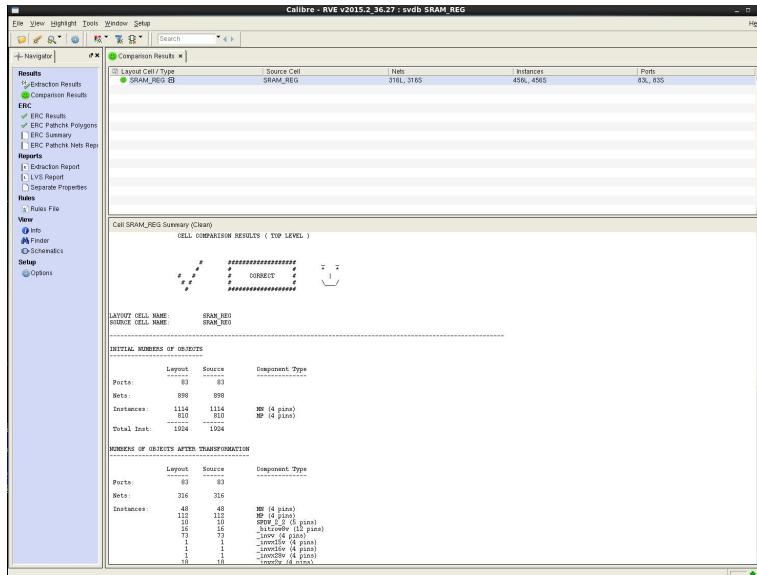


## 2.2.2 Layout test

DRC:



LVS:



PEX:



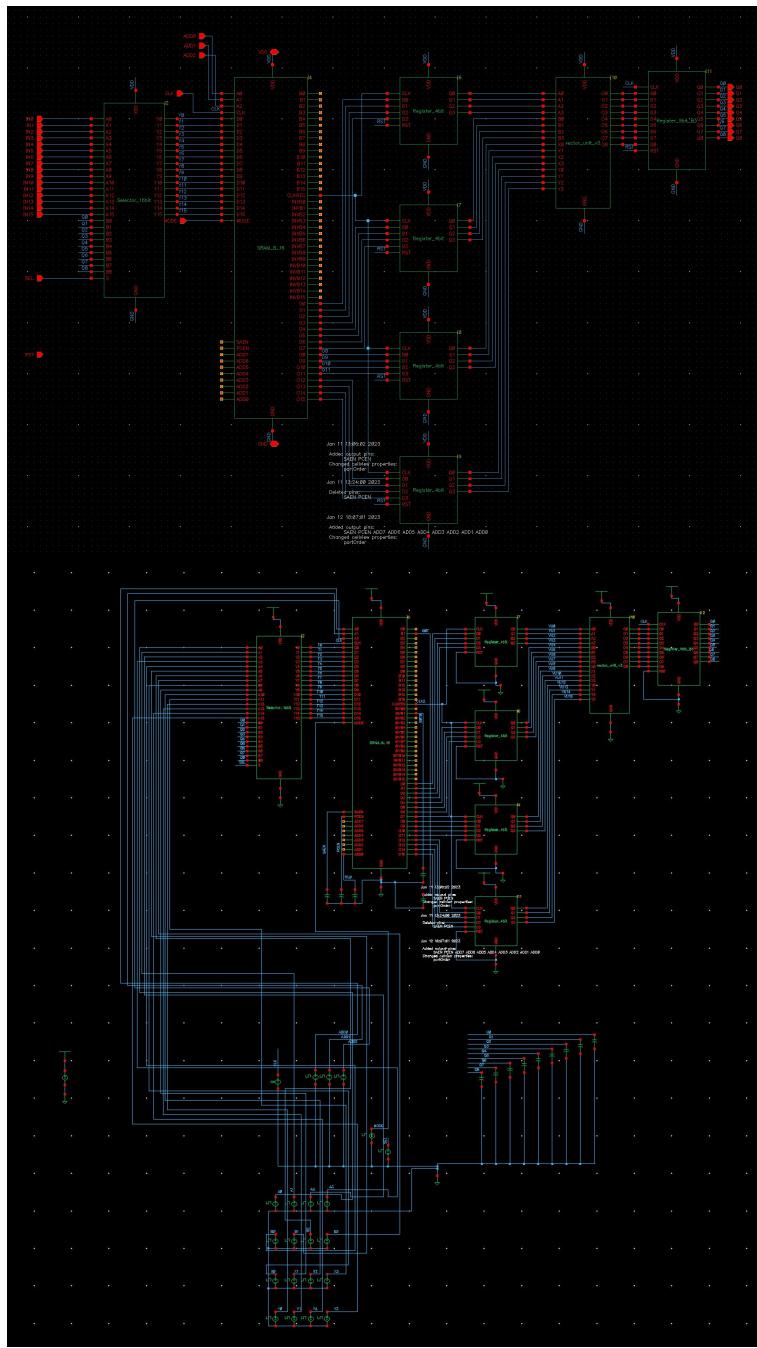
## 2.3 BONUS3

### 2.3.1 Circuit and symbol

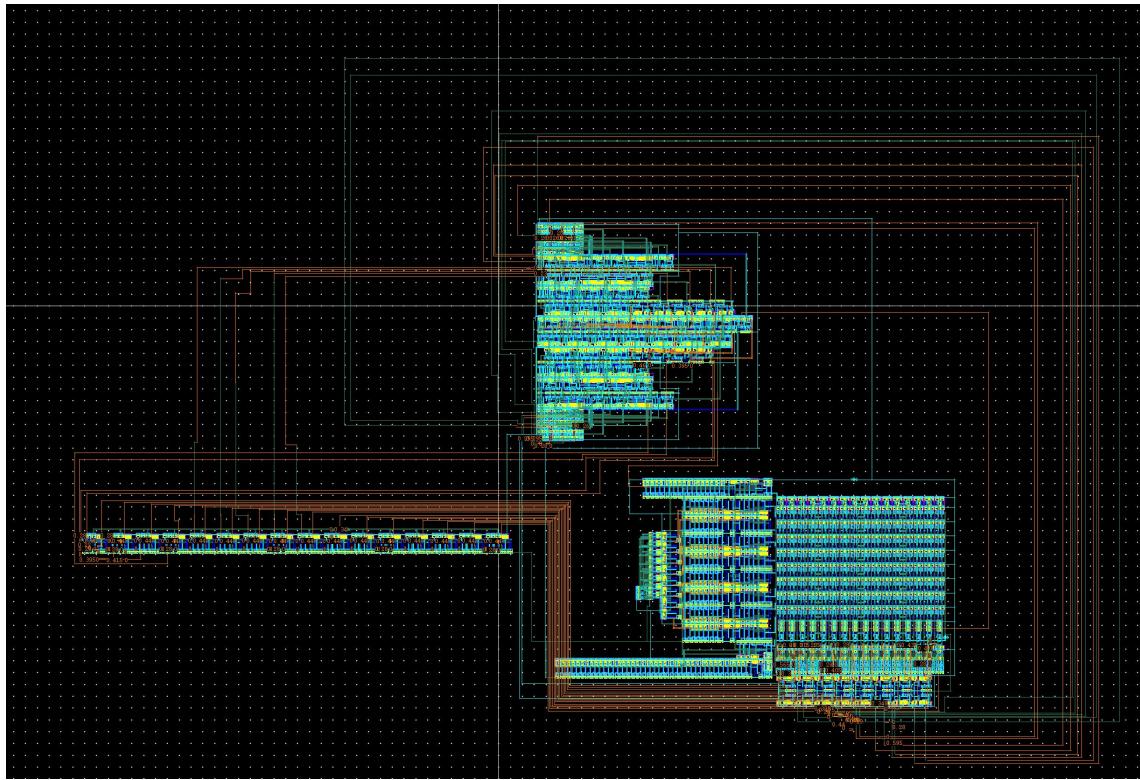
We set up the circuit as guideline.

In test cases, the 16 inputs are firstly written into the Selector to SRAM, then read from SRAM to Vector Unit. After that, vector unit outputs the calculation results and write to SRAM. Finally, the results are read out from SRAM again.

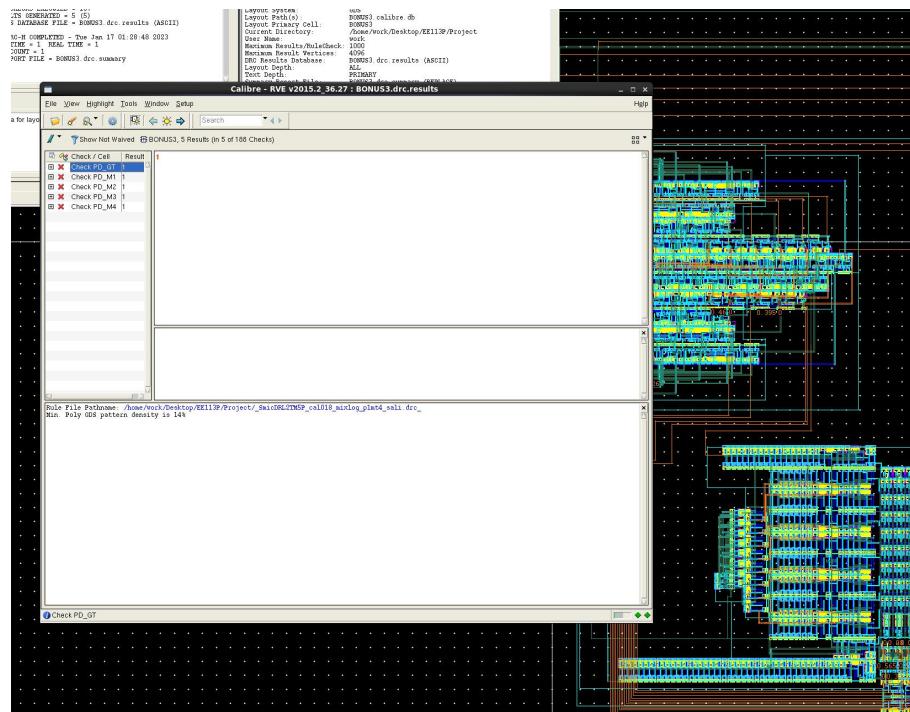
Such operation are repeated for 4 times, which represents the collaboration form of SRAM and vector unit. The circuit completed all the testcases correctly.



### 2.3.2 Layout and test



DRC:



LVS:

