Introduction to Integrated Circuit Design Final Project
Coin Bank

Student ID : 110006267
Name : Clarista Natasha Thio 張嘉倓

I. Introduction

In this assignment, we are asked to design a Coin Bank which can take the inputs of a 4-bit money and outputs the money stored in the machine or the money inputted in. Since this machine is an FSM, we are also required to output the states in 2-bit and 4-bit form.

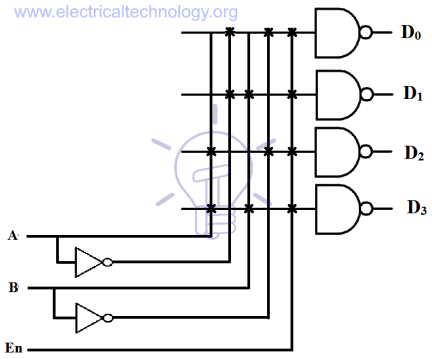
II. Pre Simulation

Since we are already given the states diagram in the specs file, we are going to base our machine on the given diagram. Looking at the diagram, I started to decide on the subcircuits needed to build the machine. The subcircuits are as follows.

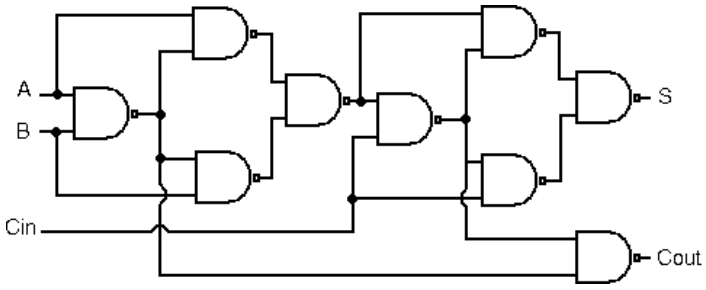
- Basic logic gates (NOT, NAND2, NAND3, NAND4, NOR2, NOR3)
- MUX (2-to-1 MUX, 4-to-1 MUX)
- DFF
- DECODER
- FSM logic

For the MUX, I used the one I used in the HW3. As for the DFF, I used a different DFF called TSPC DFF layout I found on the internet. I used this layout because it takes less area with the performance not being far behind the conventional one. For the 4-to-1 MUX, I used 3 2-to-1 MUXes to combine it into a single 4-to1 MUX. This MUX is then used for selecting the information shown in the monitor (*Mo3 ~ Mo0*). On the other hand, the 2-to-1 MUX is used for selecting the information inputted into the DFF before the 4-bit adder.

After procuring the basic gates mentioned above, we can then base our design for more complicated subcircuits based on the logic design level circuits. I based my decoder on this circuit.



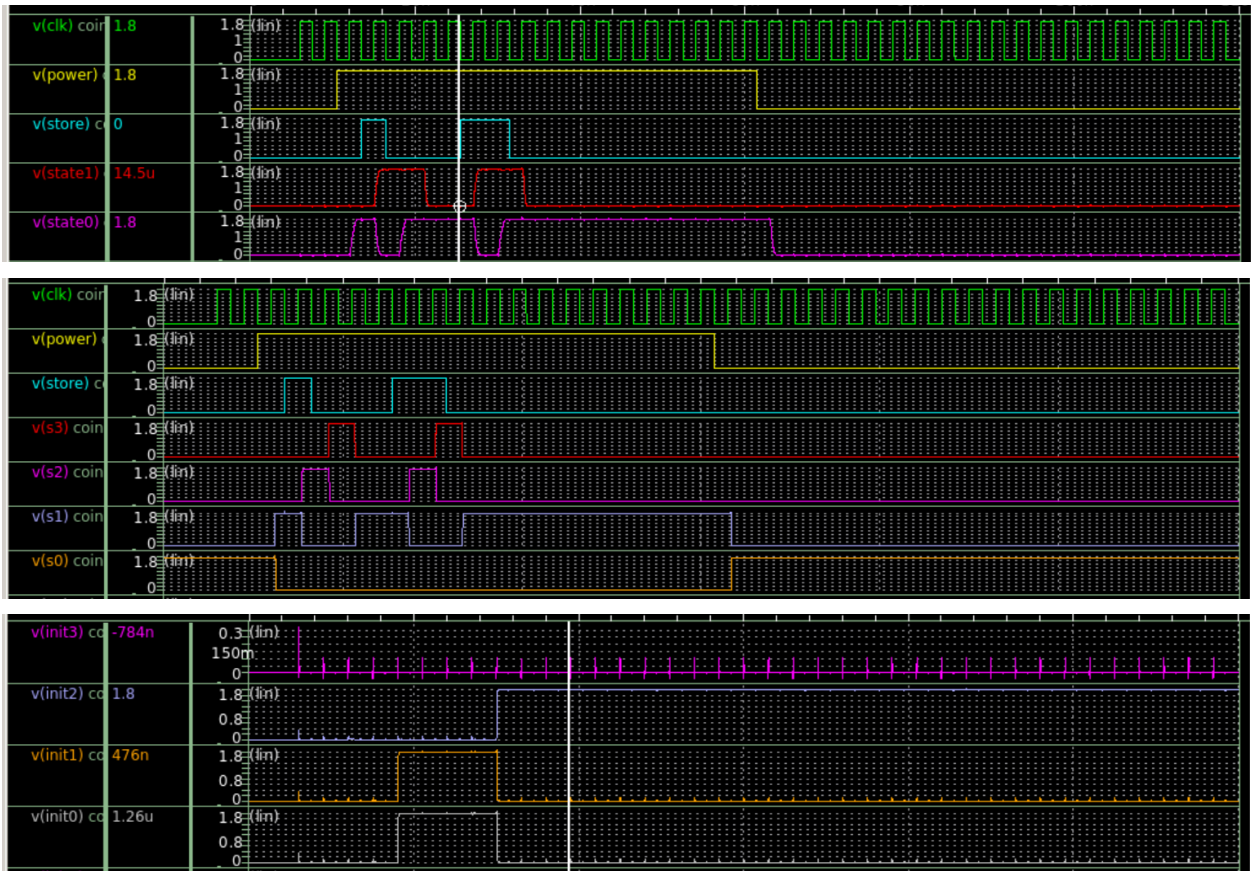
Similar to the decoder, I also based my 1-bit adder on the circuits on the logic design level shown in the picture below.

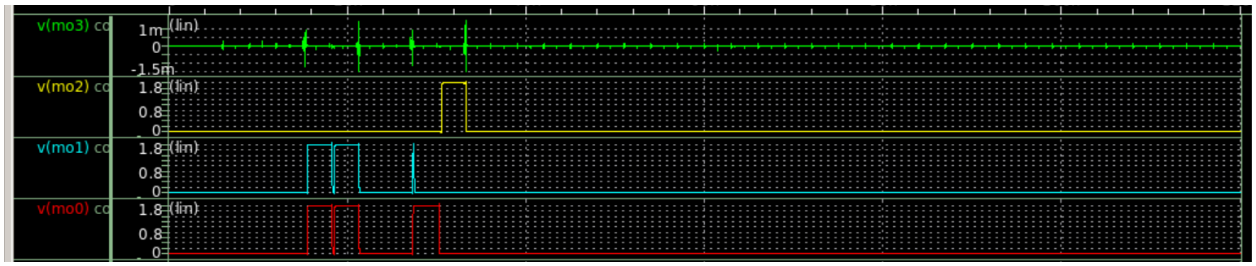


I then use RCA (Ripple Carry Adder) to build my 4-bit adder.

As for the combinational part of the FSM, I used K-map to simplify the expected output of all the different combinations of inputs. The simplified boolean function is as follows.
Next_State1 = State1' State0 Store Power + State1 State0' Power
Next_State0 = State0' Power + Power Store' + State1 Power
Since we only use NAND, NOT, and NOR to define our functions, we then modify the above functions using the said logic gates.

There's also another combinational part in this machine which is before inputting the Mo3 ~ Mo0 into the DFF. We have to decide whether the current money should be stored into the DFF. For this purpose we will use a 2-to-1 MUX with the selector being the AND of Idle State & Store. After defining all the subcircuits, we can then construct the machine based on the states diagram.
The waveform for this pre-simulation is shown below.

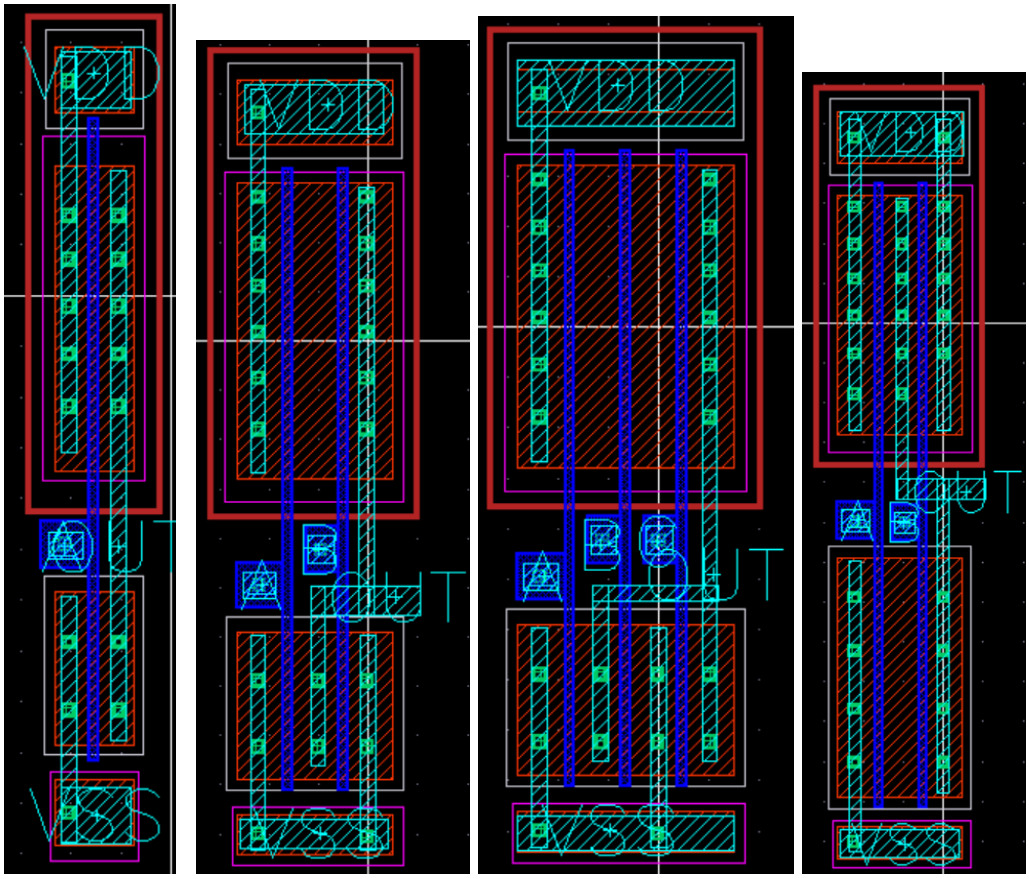


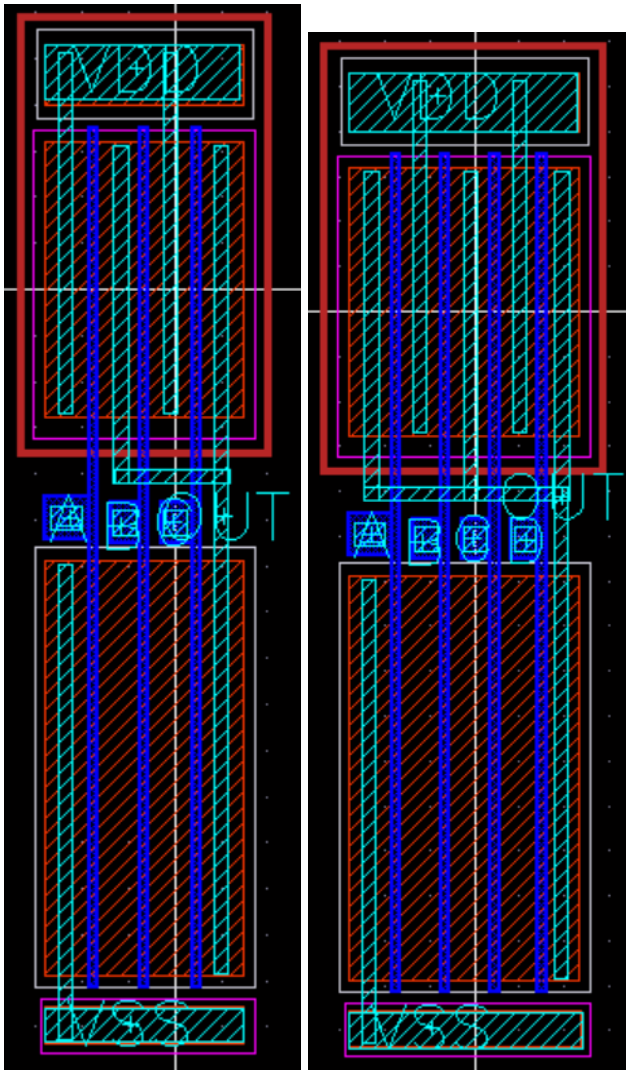


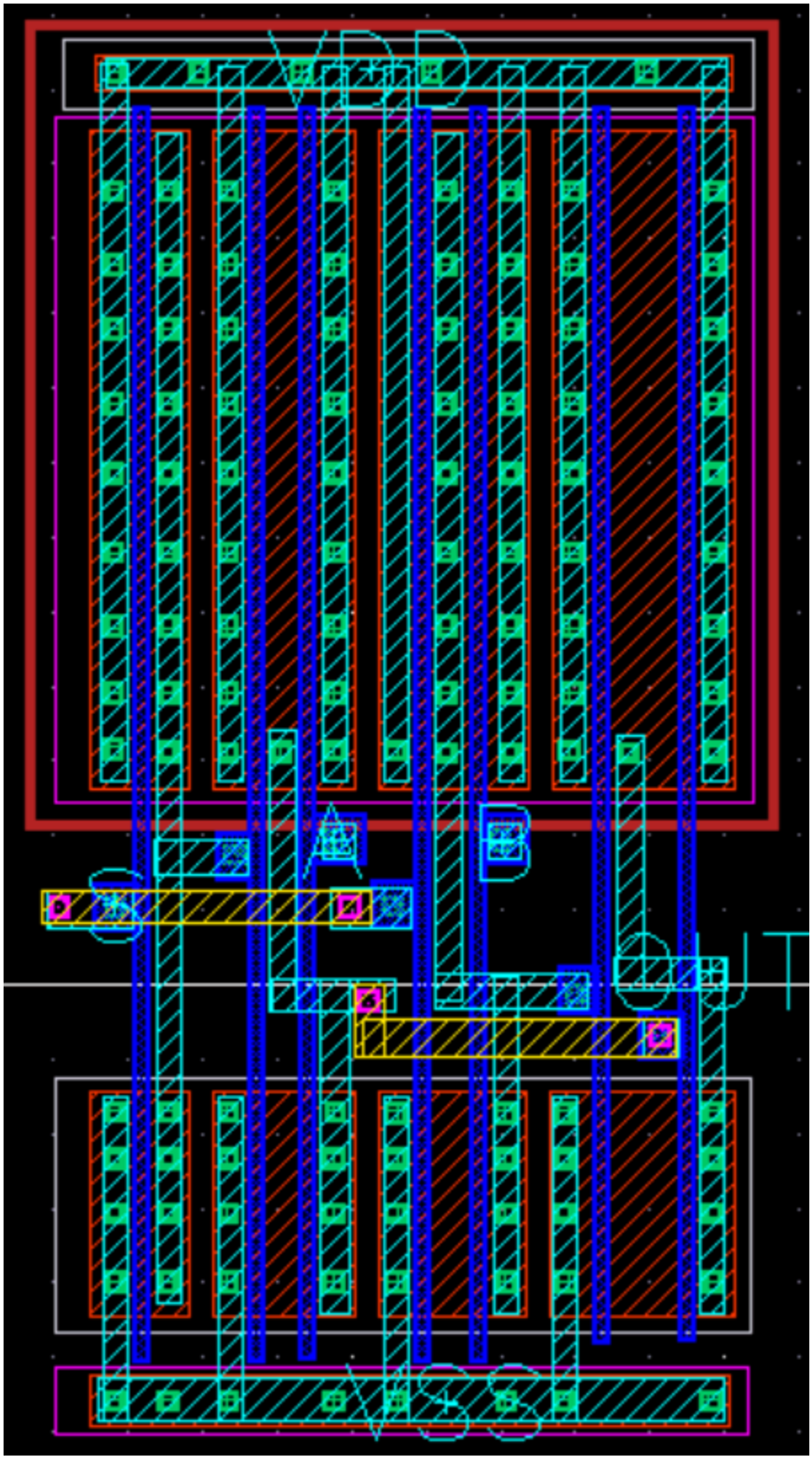
My period for the pre-simulation clock is 3ns.

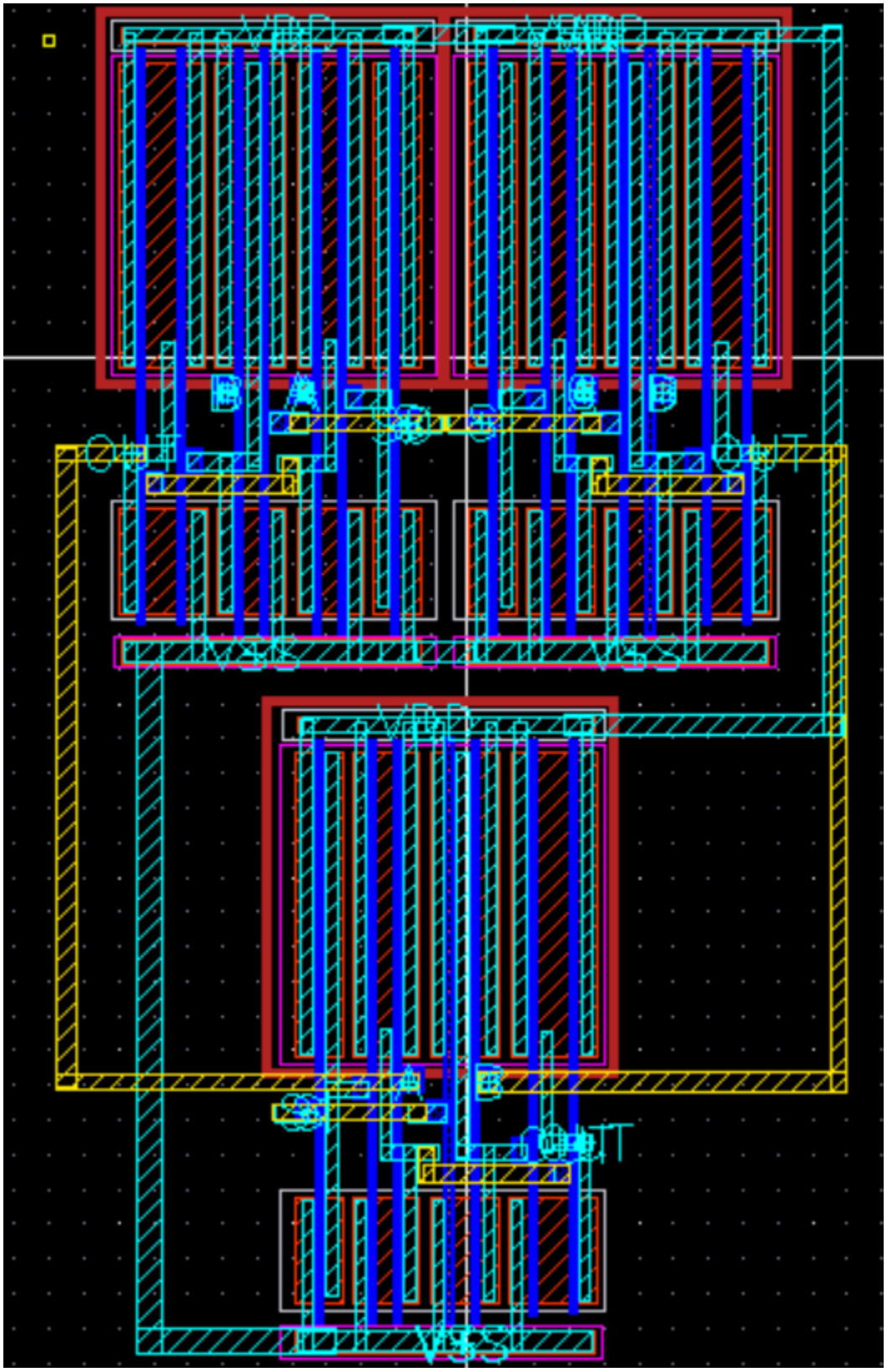
III. Layouting

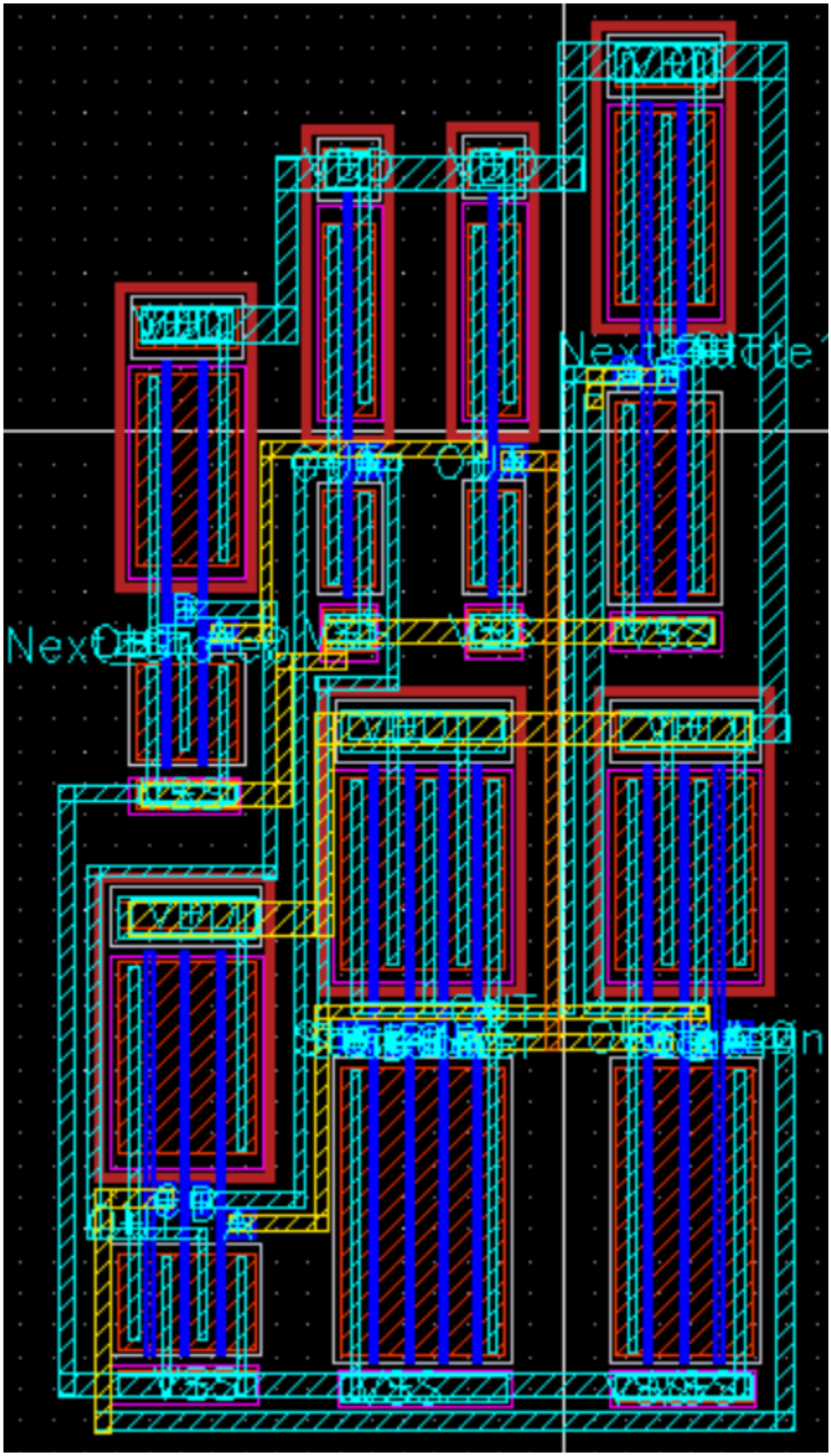
The steps in layouting are pretty much the same as in the 2 previous homeworks. As mentioned before, this machine is divided into different subcircuits, therefore, I first make different files for each of these subcircuits and check the DRC and the LVS rules to ease the DRC and LVS checking in the main circuit. Listed below are the layout as well as some of the LVS passing messages for each of the subcircuits. In the order from left to right, above to bottom, the layouts are NOT, NOR2, NOR3, NAND2, NAND3, NAND4, MUX21, MUX41, FSM Combinational, DFF, Decoder, ADD1, ADD4.

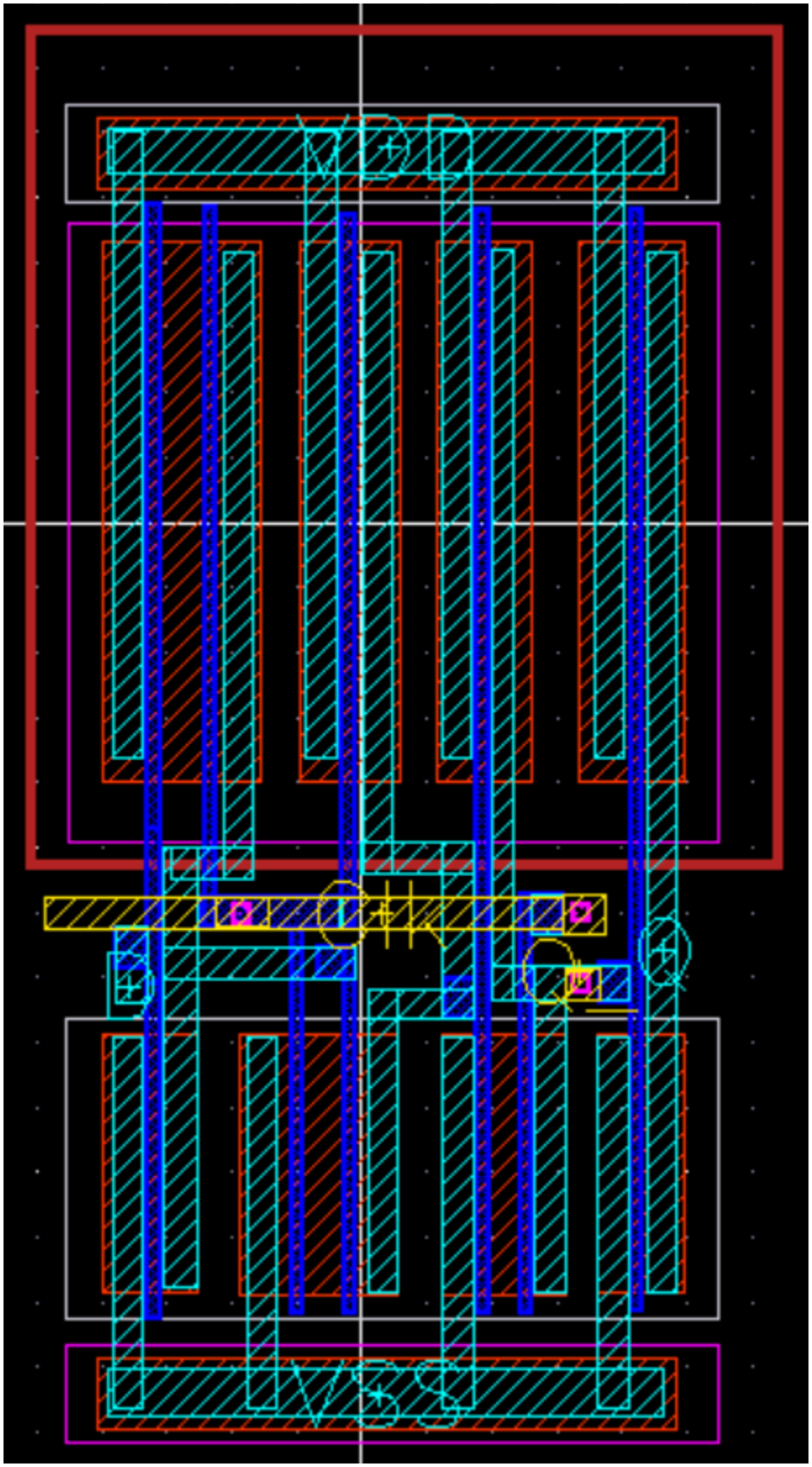

VDD
A
OUT
VSS
VDD
A
B
OUT
VSS
VDD
A
B
OUT
VSS
VDD
A
B
OUT
VSS

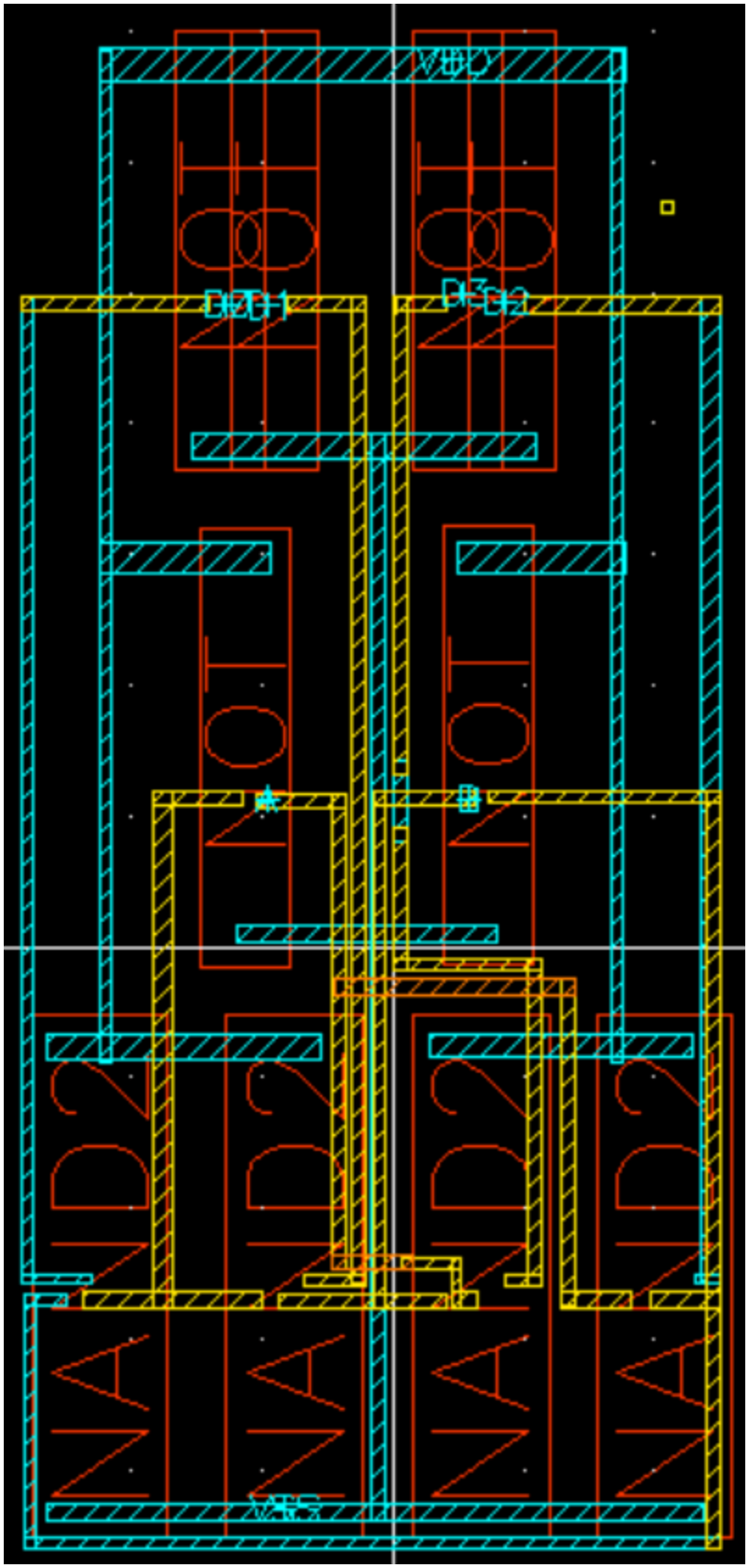

VDD
A
B
OUT
VSS
VDD
A
B
OUT
VSS

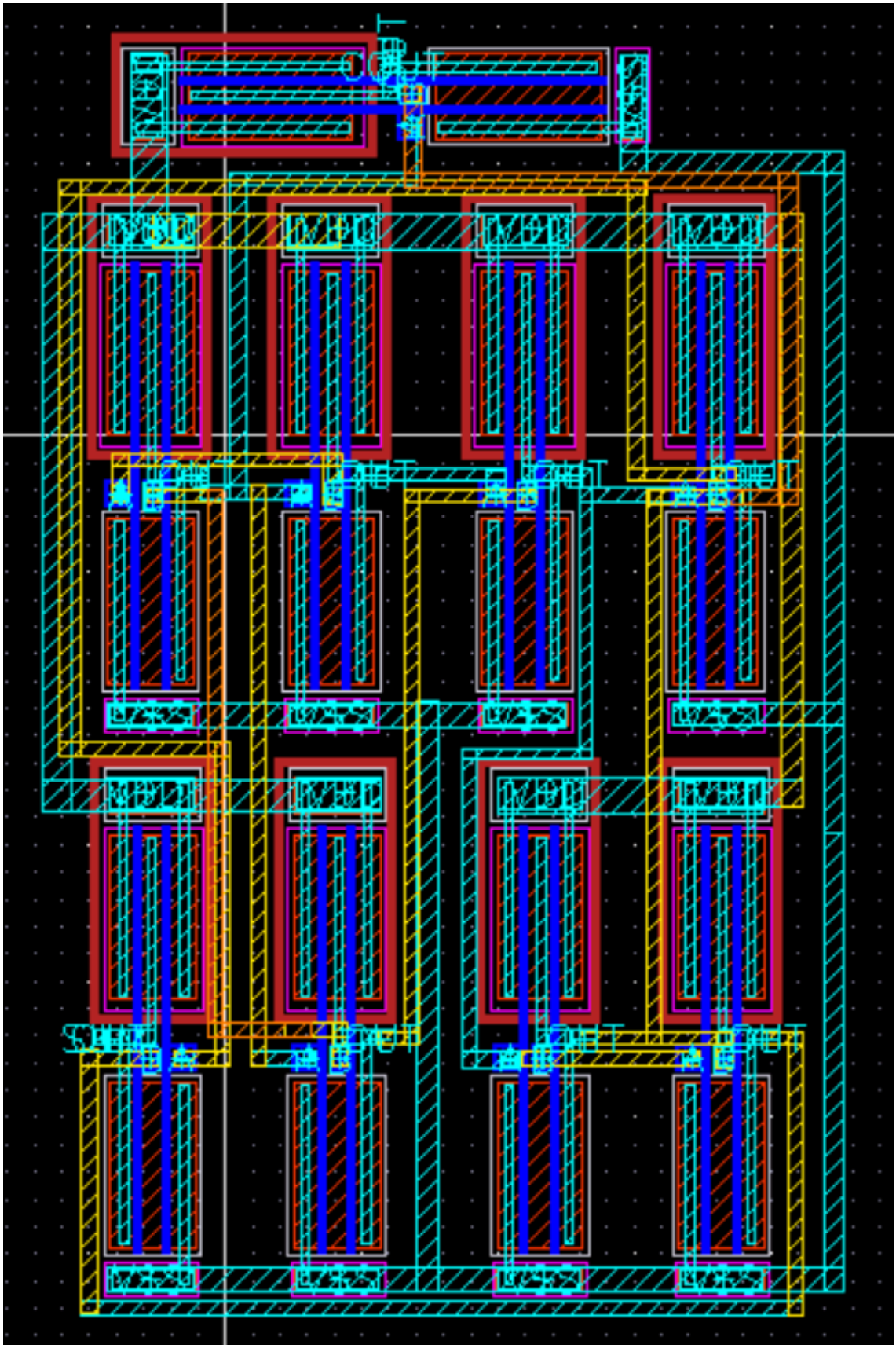

VDD
A
B
OUT
VSS

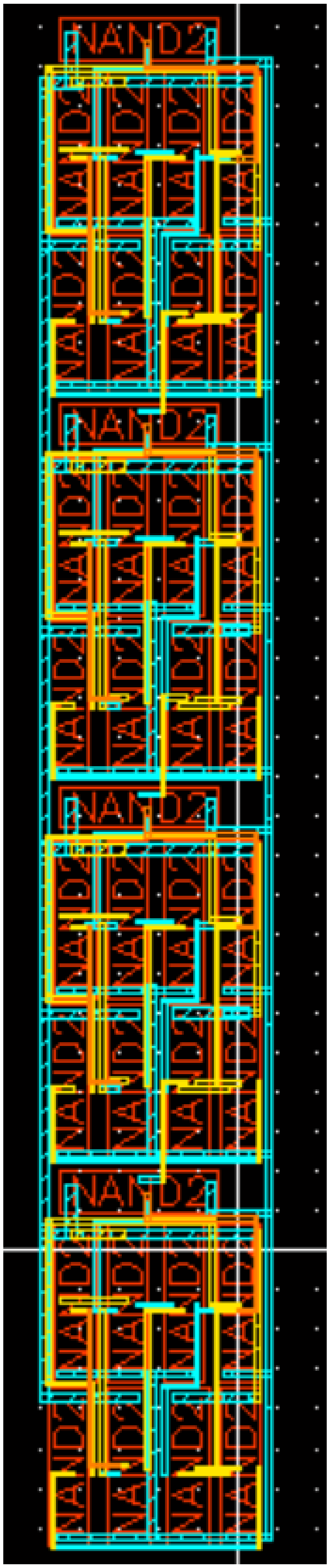

VDD
VDD
B
A
B
OUT
VSS
VSS
VDD
B
OUT
VSS

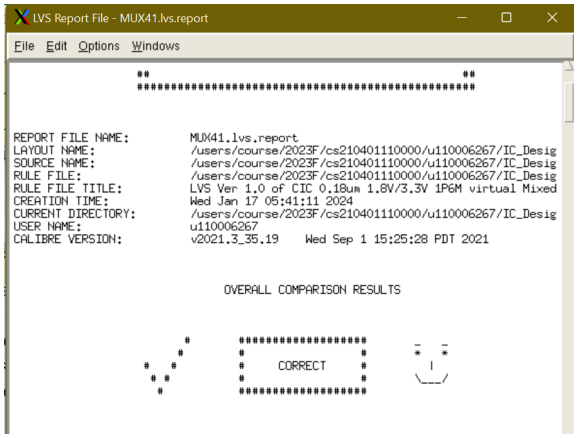


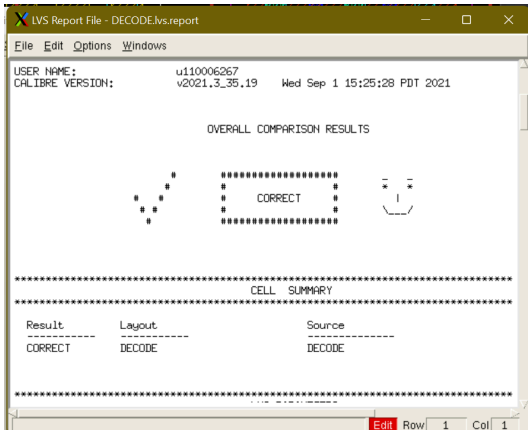


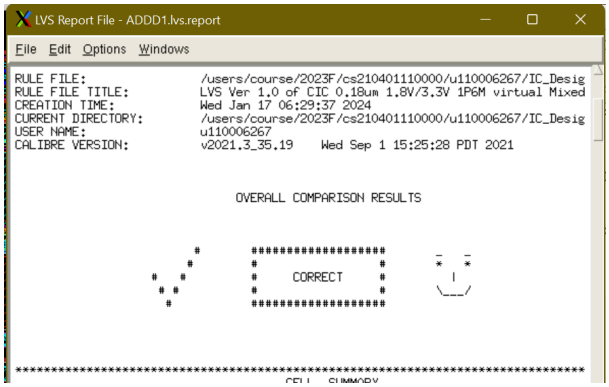

NOT
NOT
NAND2
NAND2
NAND2
NAND2

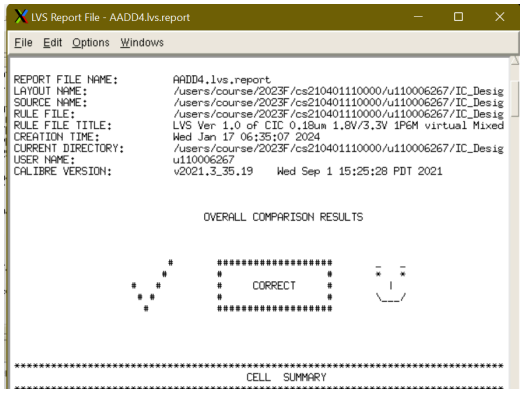


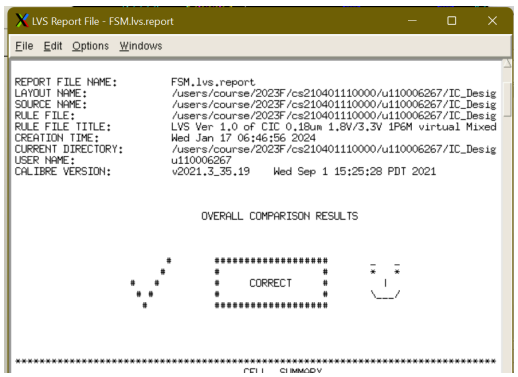

NAND2
NAND2
NAND2
NAND2



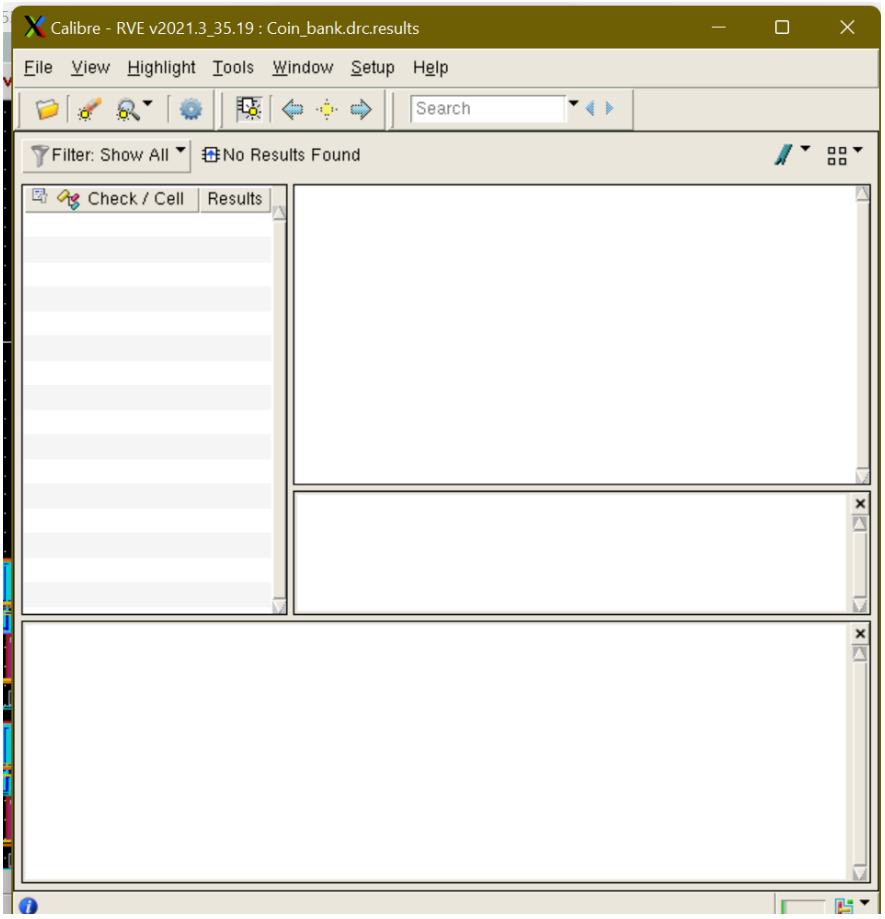


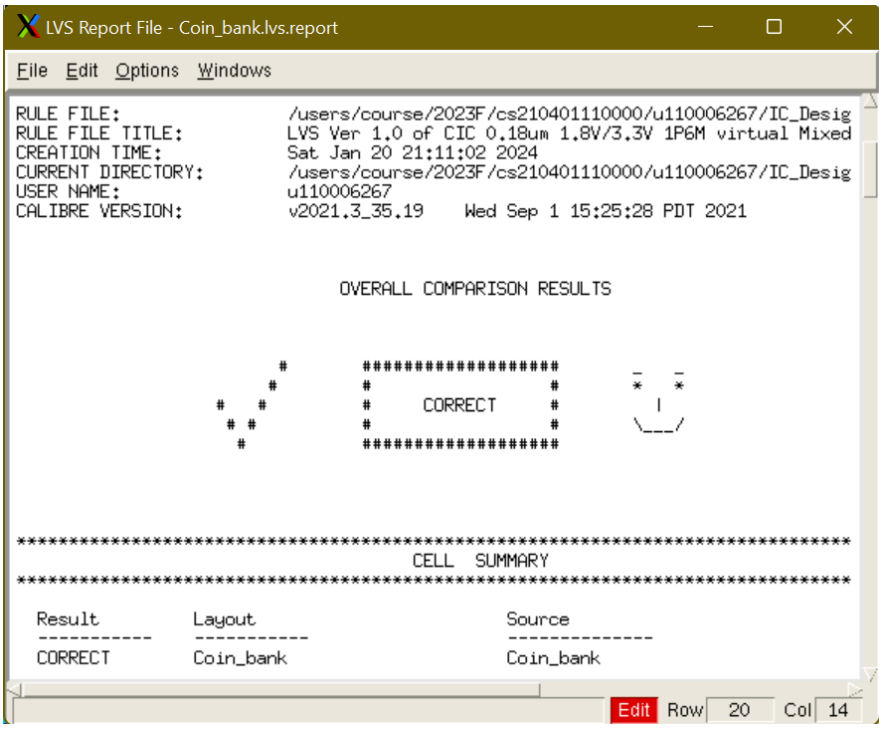


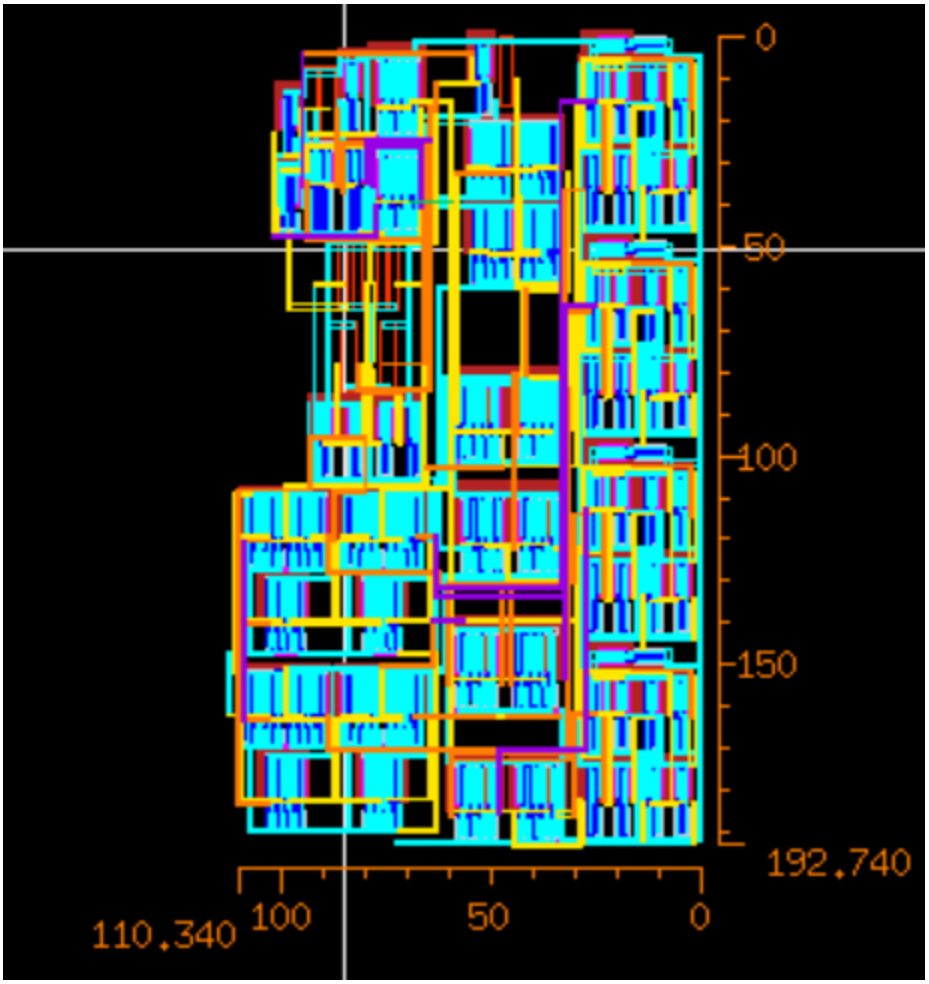




After I finished all the subcircuits listed above, I then started combining these different subcircuits which proved to be a difficult and repetitive task. I also ran into a problem when I'm doing the LVS checking in which the Virtuoso isn't able to highlight the nets with discrepancies which obviously hinders the fixing process. I tried to copy the layout into another file to solve this problem, which doesn't work. This causes me the need to redraw the whole circuit in which I am able to highlight the nets in the LVS checking process. The DRC & LVS message of passing is shown below.


Calibre - RVE v2021.3_35.19 : Coin_bank.drc.results
File View Highlight Tools Window Setup Help
Search
Filter: Show All
No Results Found
Check / Cell
Results

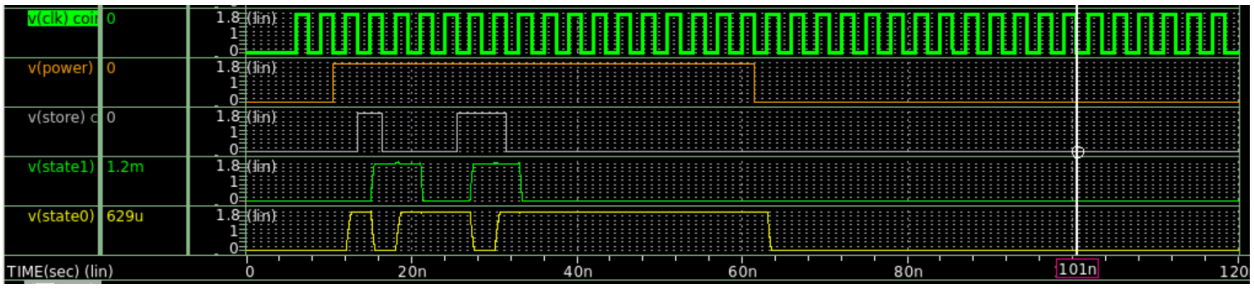

LVS Report File - Coin_bank.lvs.report
File Edit Options Windows
RULE FILE: /users/course/2023F/cs210401110000/u110006267/IC_Desig
RULE FILE TITLE: LVS Ver 1.0 of CIC 0.18um 1.8V/3.3V 1P6M virtual Mixed
CREATION TIME: Sat Jan 20 21:11:02 2024
CURRENT DIRECTORY: /users/course/2023F/cs210401110000/u110006267/IC_Desig
USER NAME: u110006267
CALIBRE VERSION: v2021.3_35.19 Wed Sep 1 15:25:28 PDT 2021
OVERALL COMPARISON RESULTS
CORRECT
CELL SUMMARY
Result Layout Source
CORRECT Coin_bank Coin_bank
Edit Row 20 Col 14

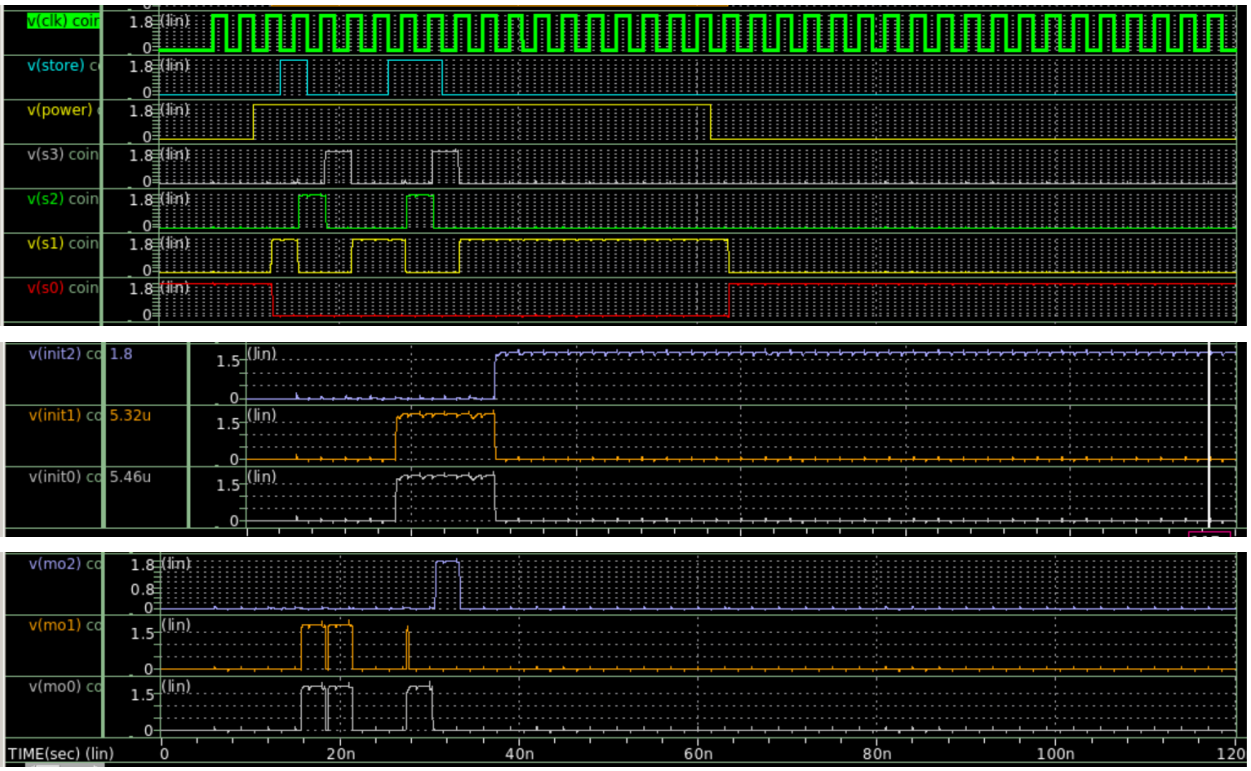


Shown above is the layout with the the sizing, 110.340 * 192.740.

IV. Post Simulation

After I'm done with the layout, I then moved on to the post simulation section. Using the same testbench, my layout is able to produce the following results.





The clock period for the post simulation is the same as the one I used in my pre-simulation, which is 3n. As seen here, there are some more glitches in the signal. The latency of my circuit is 120 ns since I have a 40 clock period in my testbench.

Problems I Encountered and How I Overcome It & the Hardness of This Assignment

In my opinion, the goal of this final project is for us to employ all the skills learnt in all the previous assignments in a step-by-step sequence. Therefore this final project is not that hard compared to the other assignments in terms of the techniques. However, the scale of the circuits that are assigned is greatly increased causing the great amount of time needed to finish this final project. Some of the problems I came across in this final project:

1. As mentioned before, because of the huge scale of the machines, combining the combinational & sequential circuit to have a fully functional machine, in every step of this assignment, I have to consider how it would look in the final layout. As an example, I have to think of the shape of each of the individual circuits ahead so as to not waste space in the final layout.
2. Even after checking the DRC and LVS for the individual subcircuits, I still spent a lot of time in the main circuit LVS, since my number of errors amounted to 110+ (QQ). These are mostly caused by the metal of the same unit intersecting each other.
3. I originally thought that the routing that connects the input and output ports will be one of the easier parts of this assignment. This is especially not true since I spent a lot of the

time & brainpower in routing the wires, as well as thought ahead to have more possible routes for the yet-to-routed ports.

4. As mentioned above, I have to draw my layout 2 times. This is because of some issue with the Virtuoso program which won't highlight the nets that are causing discrepancies in the LVS section. This obviously makes the fixing process harder. The Virtuoso program gives out an error message saying the results_query couldn't be rewritten. After spending quite some time trying to fix this problem, I decided to redraw the layout since I have yet to find the solution. This causes me to spend another 7 hours doing so.

Suggestions

I don't really have a suggestion for this assignment. Partly because I'm more used to the layout of the homework specs. Also because I'm used to the systems of the assignments now.