Chenyu Song (cs4493)

Hang Ye (hy2891)

Project Group 2

EECS E4321

Final Project Report

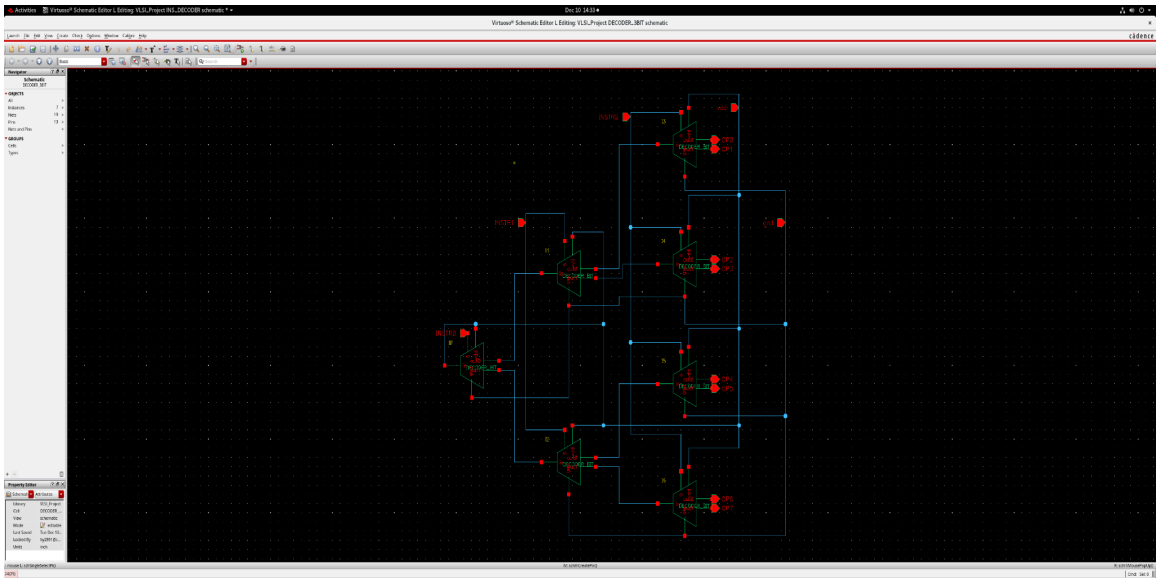
MACRO:

PS_9 has a poor description of instruction and operation. Here is a detailed description.

Opcode	Assembly	Description
000	NOP	Do nothing
001	LOAD	Mem[i] ← External bus
010	STORE	External bus ← Mem[i]
011	GET	Acc ← Mem[i]
100	PUT	Mem[i] ← Acc
101	ADD	Acc ← Acc + Mem[i]
110	SUB	Acc ← Acc - Mem[i]
111	SHIFT	Left logical shift of Acc

So basically the shifter only works with mem, so does the adder. This is important because the shifter is not able to store any data, while the datapath makes it possible to store data. There are 2 latches on the path of the shifter, which can be a temporary register. However, the opcode doesn't include some weird operation like acc <- shifter_out + mem[i], which makes everything simpler.
About the 3-1 mux, I choose not to use 3 nfet basic cell, but use a 3-stack high pass transistor, which is basically 3 mux sequentially.

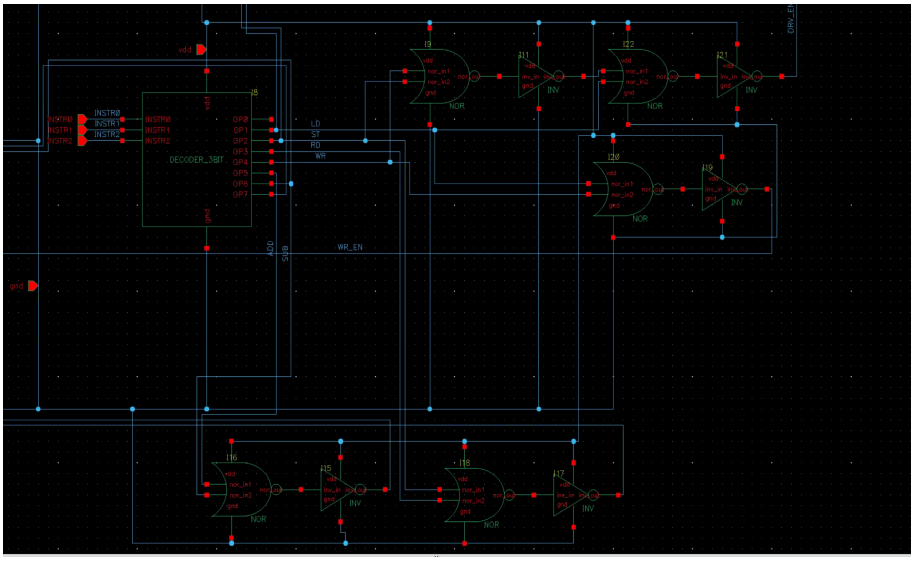
For the decoder, a simpler way is to use mux reversibly. This is not what the PS9 asks us to do, but much easier.

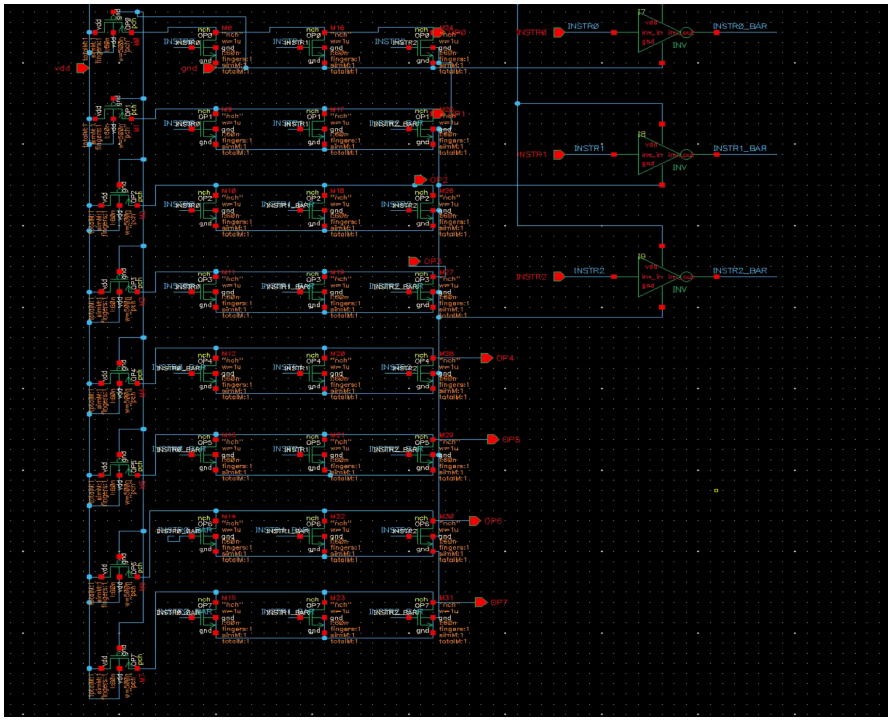


However, after testing, the pull down issue is critical under this situation. When OP is not selected, floating out will show up as a random value, which is out of controls.

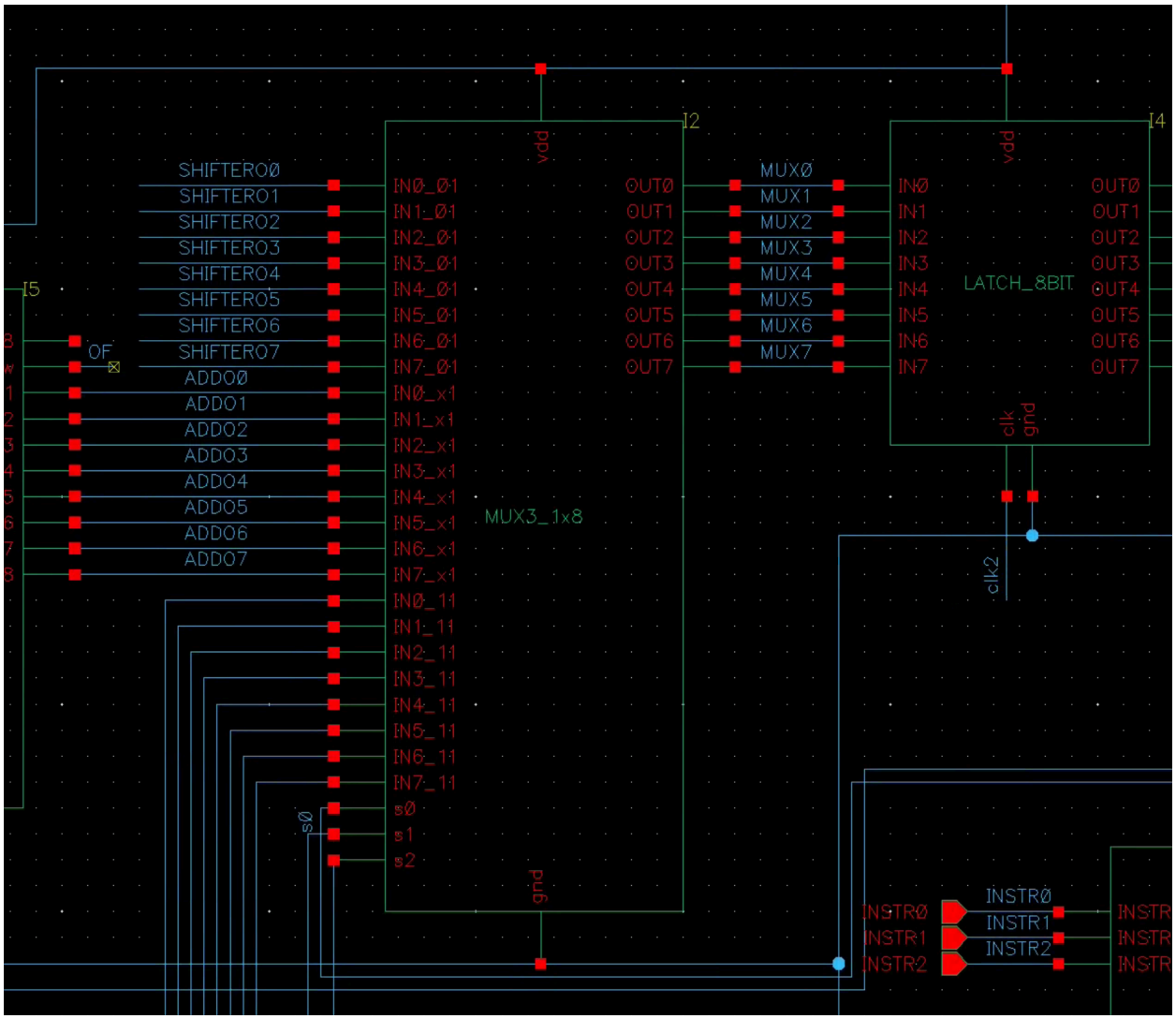
Now we turn to a PLA structure. The same as PS9 says, we use 3 nfet to do PLA. Seven output value has make collaborate effort to control blocks(exclude NOP):

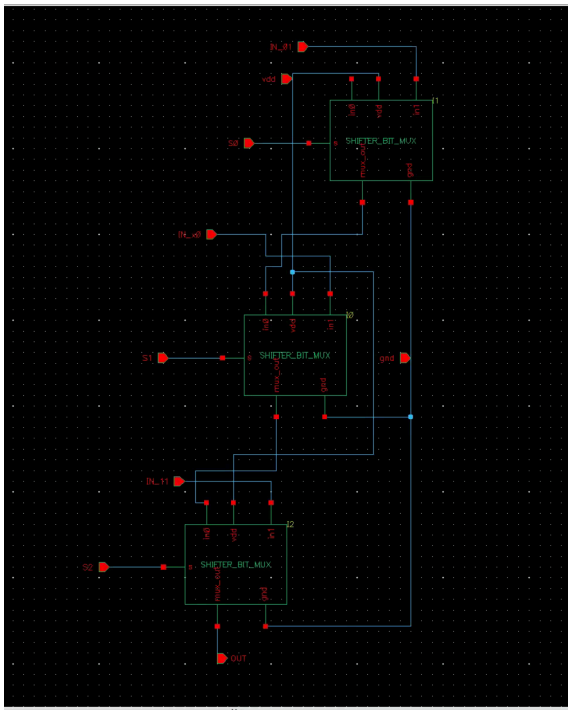
1. LD or ST or WR = BUS_EN
2. LD or WR = WR_EN
3. SHF = s0
4. ADD or SUB = s1
5. RD = s2
6. SUB = c0





S0, S1, S2 is the selection of 3-1 MUX. Of course we can use 2 bits to represent that. A 3 bit MUX makes it easy and clean. The basic unit is from the PS7 shifter.

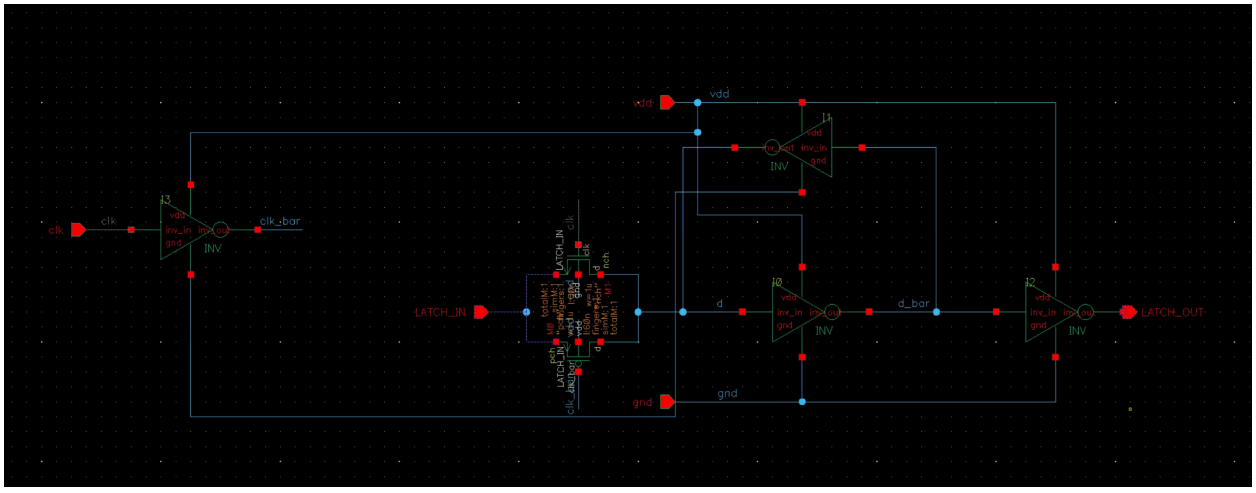




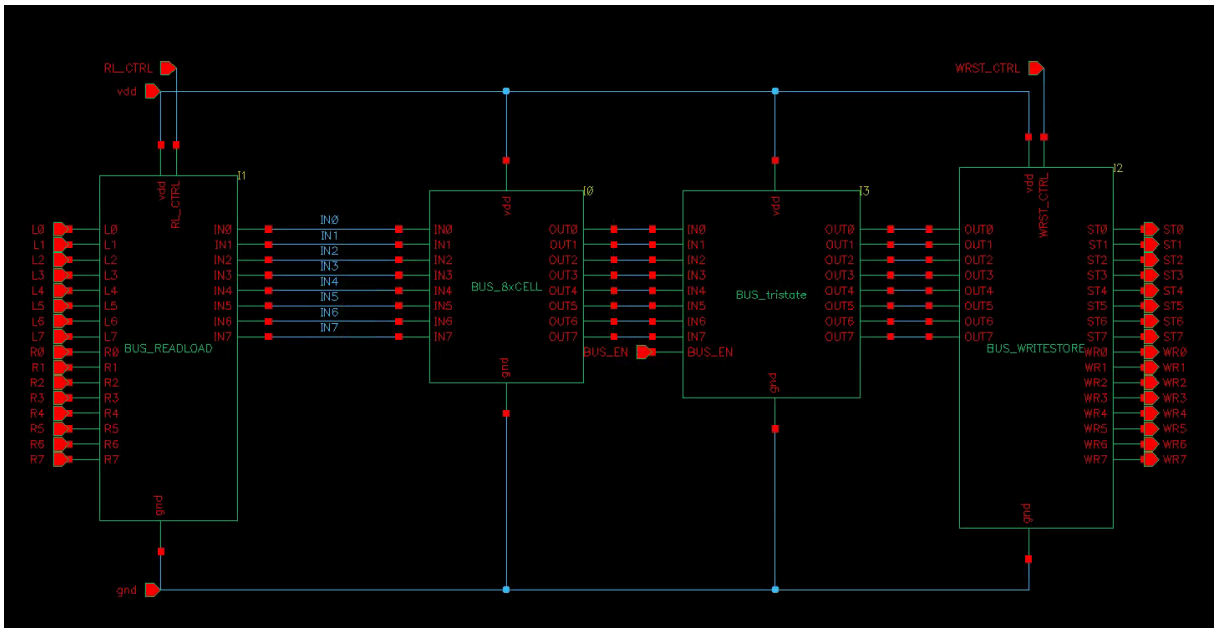
When designing a latch, we found that we can't just use the default INV we design. This is due to the fact that we need a weak INV so that the pass transistor can overwrite it. Our default INV is 1um width, we need to change it to 200nm to make it work.

The biggest load issue happens on the shifter into MUX to latch, where it is likely 4 high stakes. One way to solve this is by reducing the INV in the latch, so that a small voltage strength can pull it up/down. The other way is to add a buffer/repeater into the critical path, which will definitely increase the delay. We choose the first method.

Here is what we design:



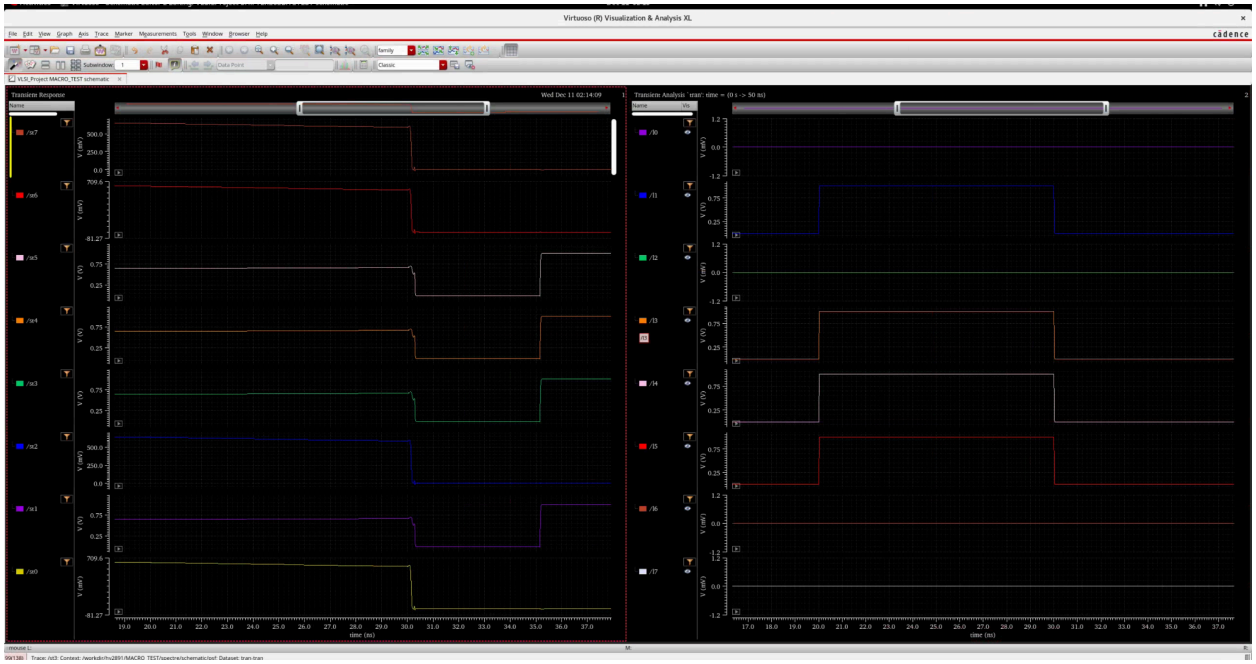
The bus driver is also a repeater, which makes it very easy to understand. Below is the schematic of the bus driver:



Here is our test on every function of our MACRO. To begin a test, at least 1 ‘LD’ is needed, since there is 0 valid value in sram. Clk period: 10ns

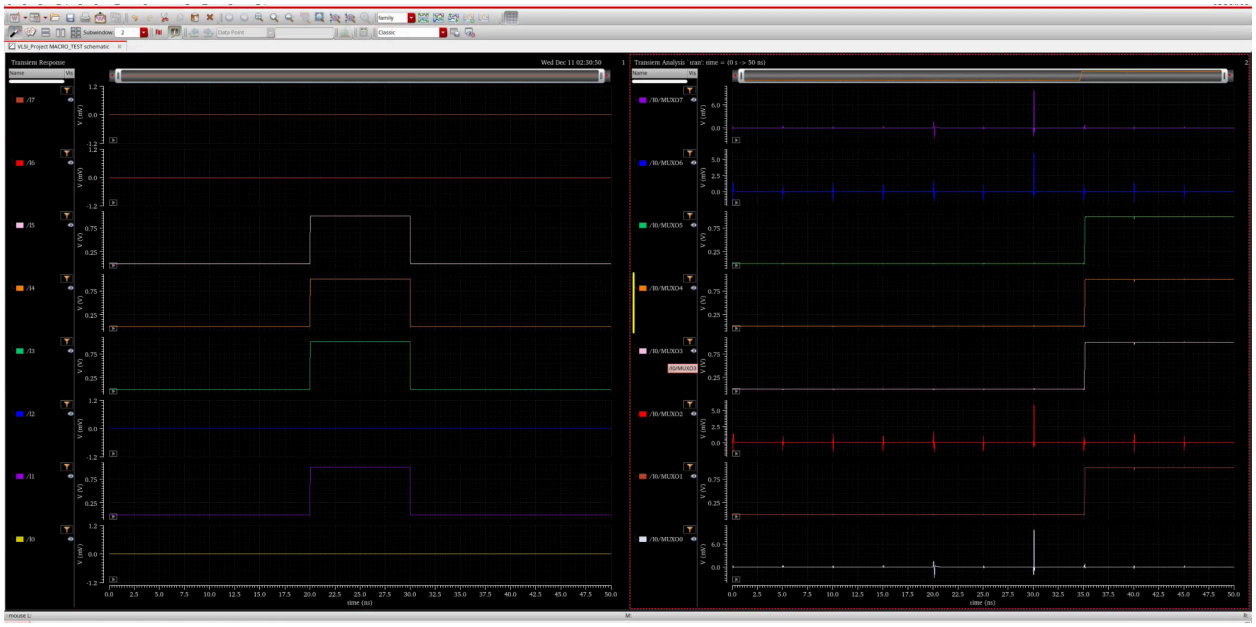
1. LD+ST: This is basically print what you input. To be noticed, the valid phase is only a small period of time – when clk2==0. Since there are no components like latch to store our temporary output, the value will change to a weird volt after the evaluation phase(clk2).

I: 00111010 (20-30 ns)
O: 00111010(30-35 ns)



2. LD+RD+SHF: There was s test on ‘LD+RD’, but RD only output on a different place compared to ST. However, if we use the combination of ‘RD+SHF’, while making the shift factor to be 0, we can force the datapath to be a temporary register.

I: 00111010

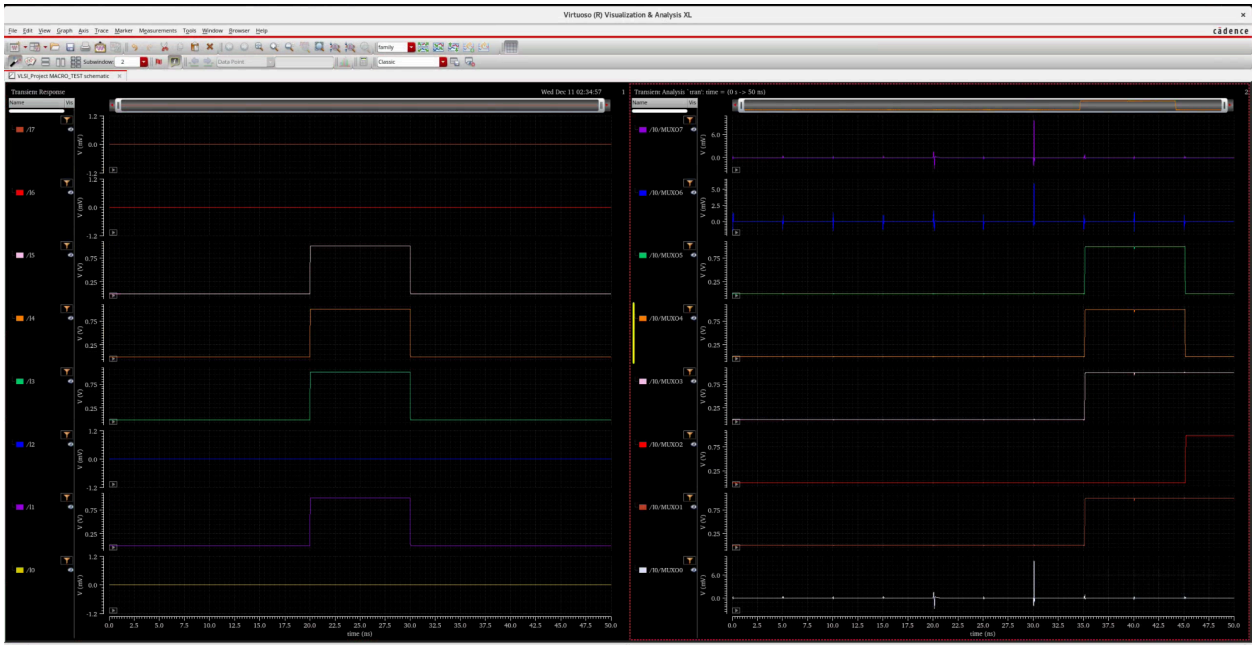


3. When talking about shifters, the shifter factor also comes from INSTR<3:5>, which is just the same as the MEM_ADDR. This is not mentioned in the PS9 description. Using ‘LD+RD+SHF’ but with a valid shift factor. The shift OP valid after 45ns.

I: 00111010

Shift factor: 010

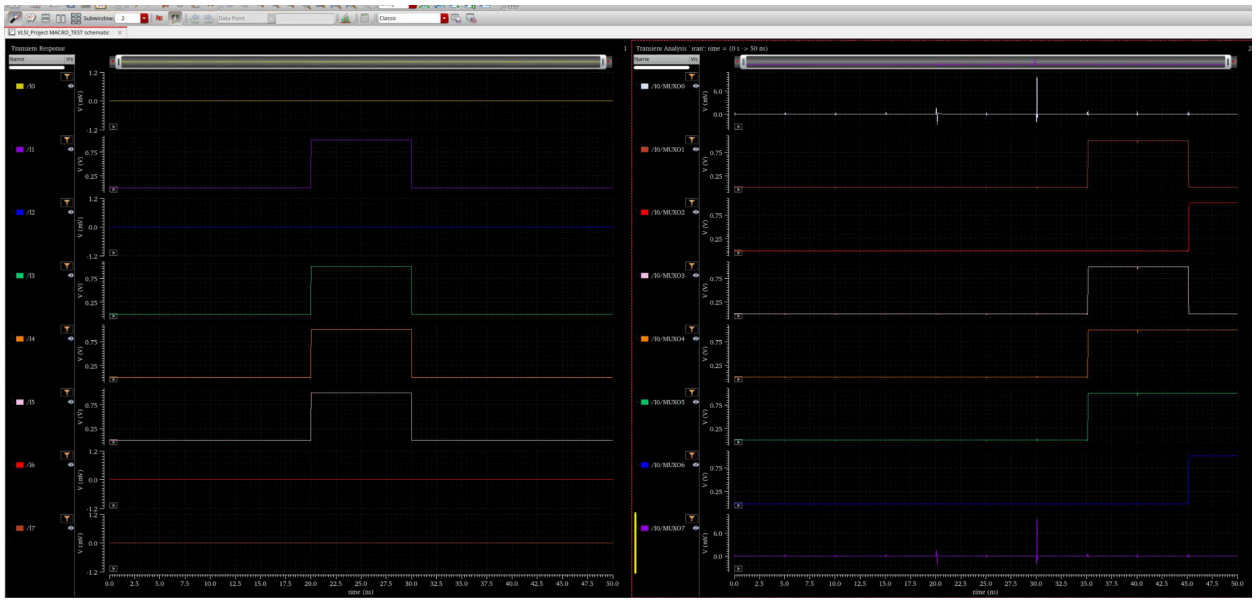
O: 00001110



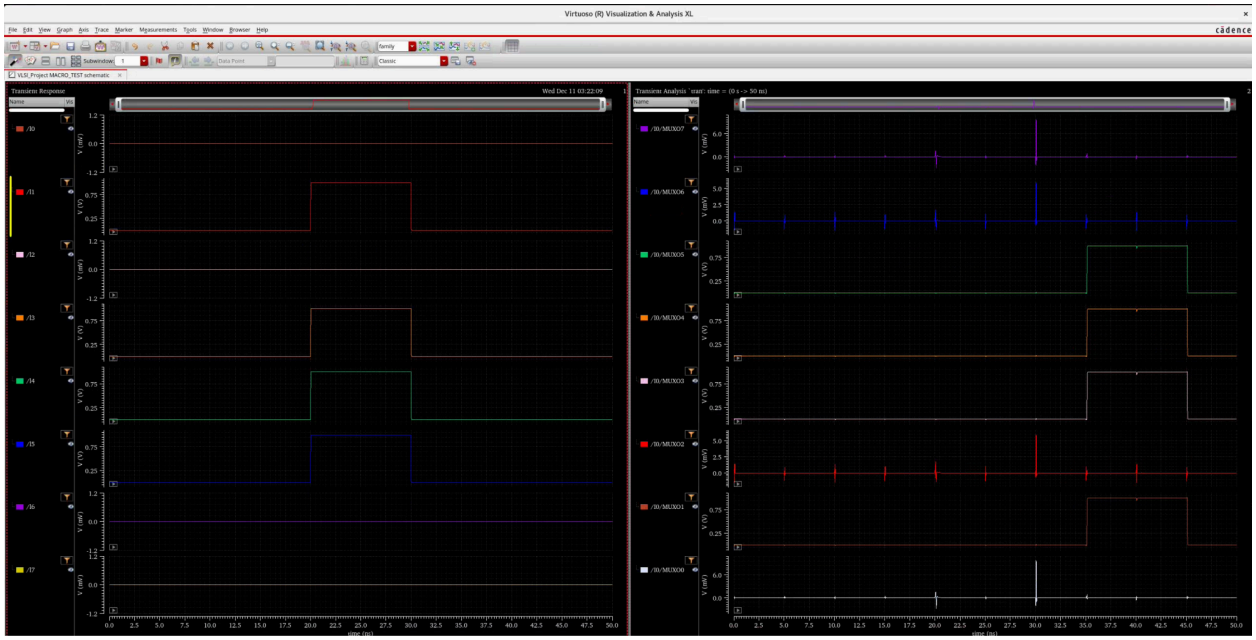
4. ‘LD+RD+ADD’. This is basically self-add. This is a simple test. Because we only store 1 valid value in sram, we can only do MEM[i] + MEM[i].

I: 00111010

O: 01110100



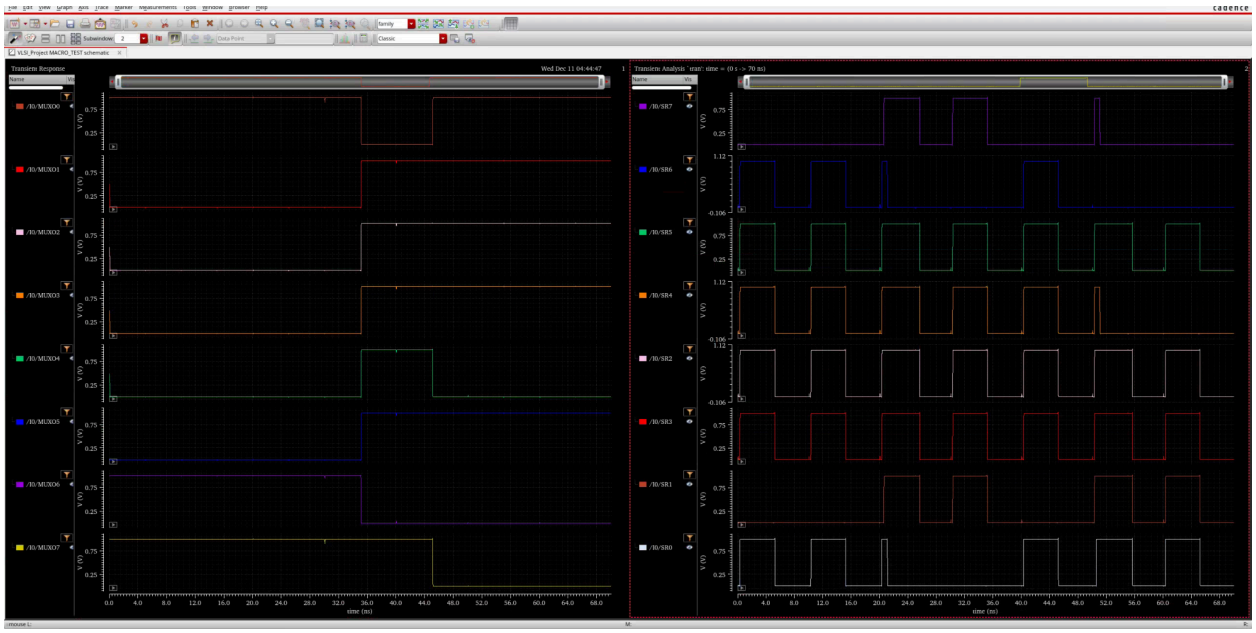
5. ‘LD+RD+SUB’. Answer should be 0. (Starting at 45ns)



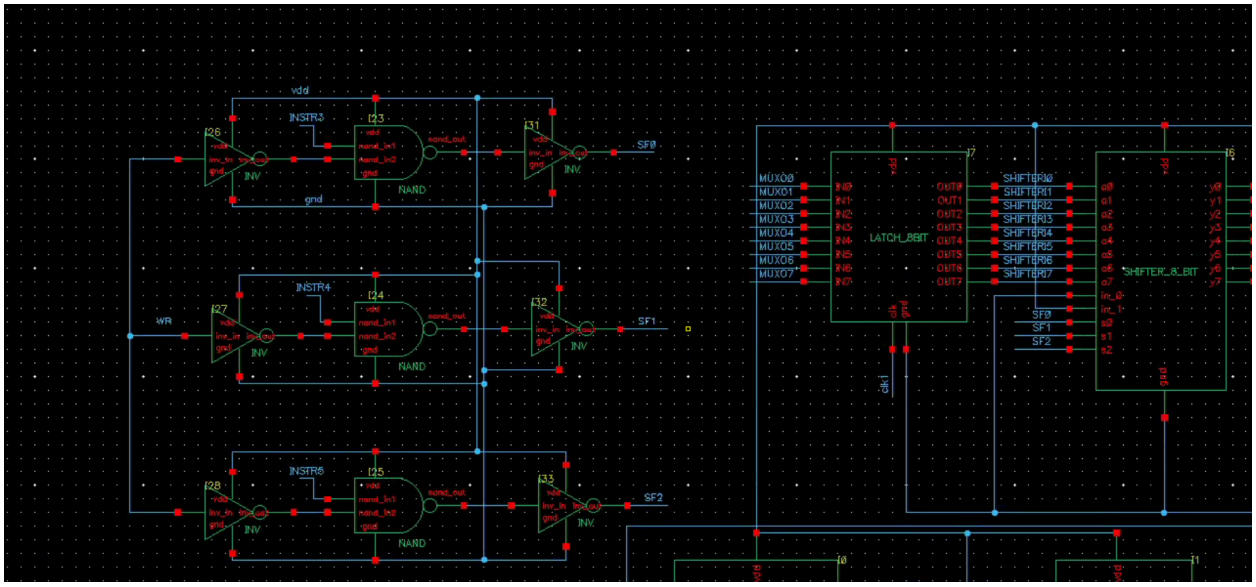
6. ‘LD+RD+SHF+WR+RD’. We need to test WR. Although basically if we get the right result on LD, WR should also be correct. But we still make an example to test. There is a small trick on WR. Since WR only gets data from the internal bus, in order to make sure there is no conflict and speed up the clk, we can do shift 0 at the same time. Because if we use a high frequency clk, there would be a setup time issue. A shift 0 operation can extend the data valid time, while it would not influence the operation of WR.

LD to 000 -> RD from 000 -> SHF 010 ->WR to 010(SHF 000 at the same time.)

I: 10111110



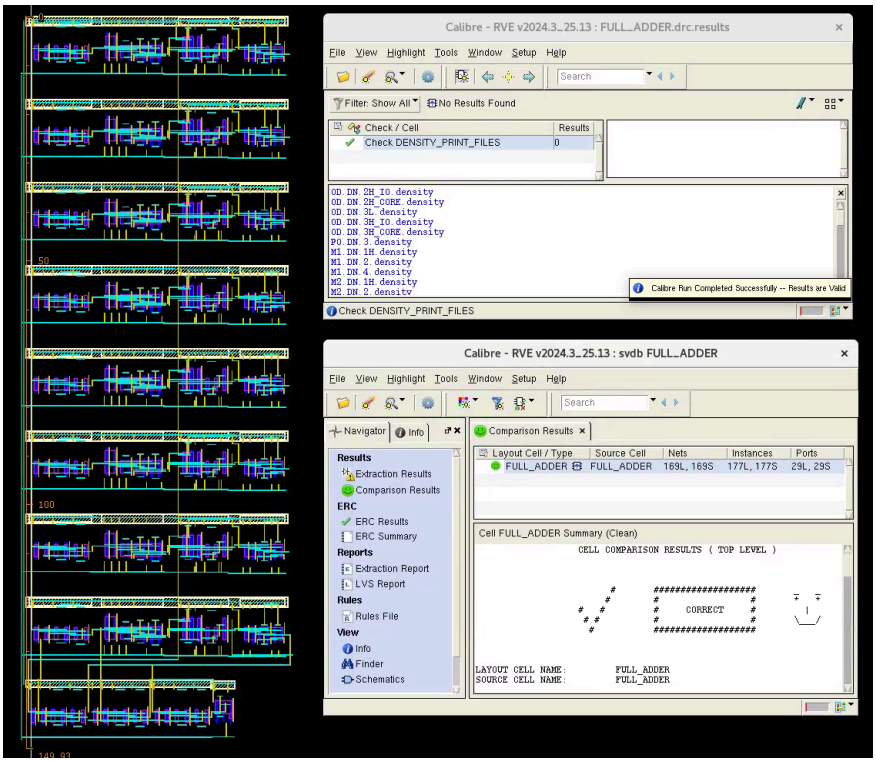
.Tricky shifter: (when WR is high, SF will pull down to 0)



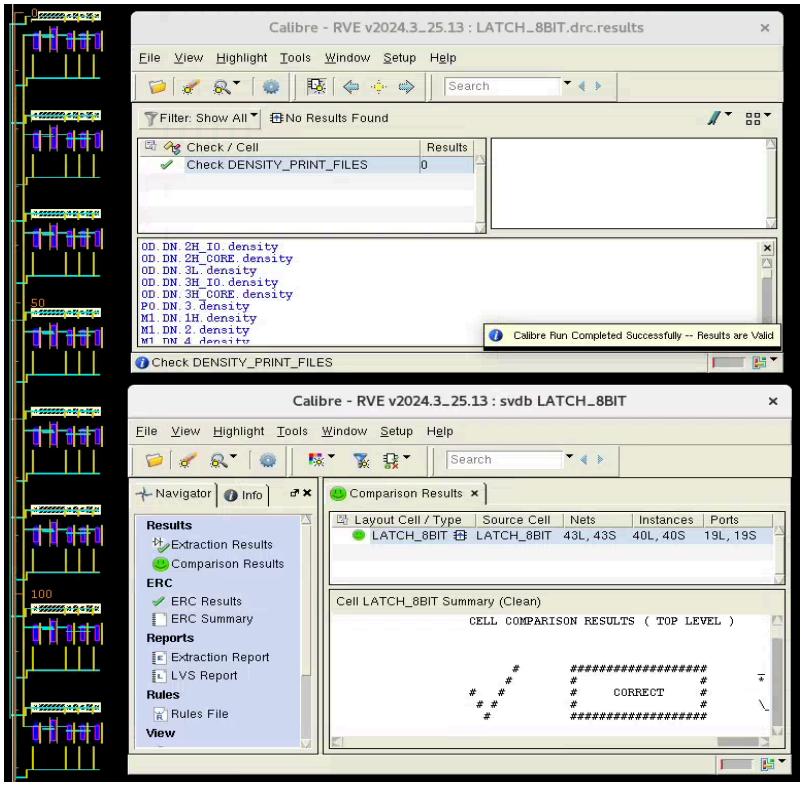
Function waveform end.

For this macro, we complete the whole schematic with partial layouts.

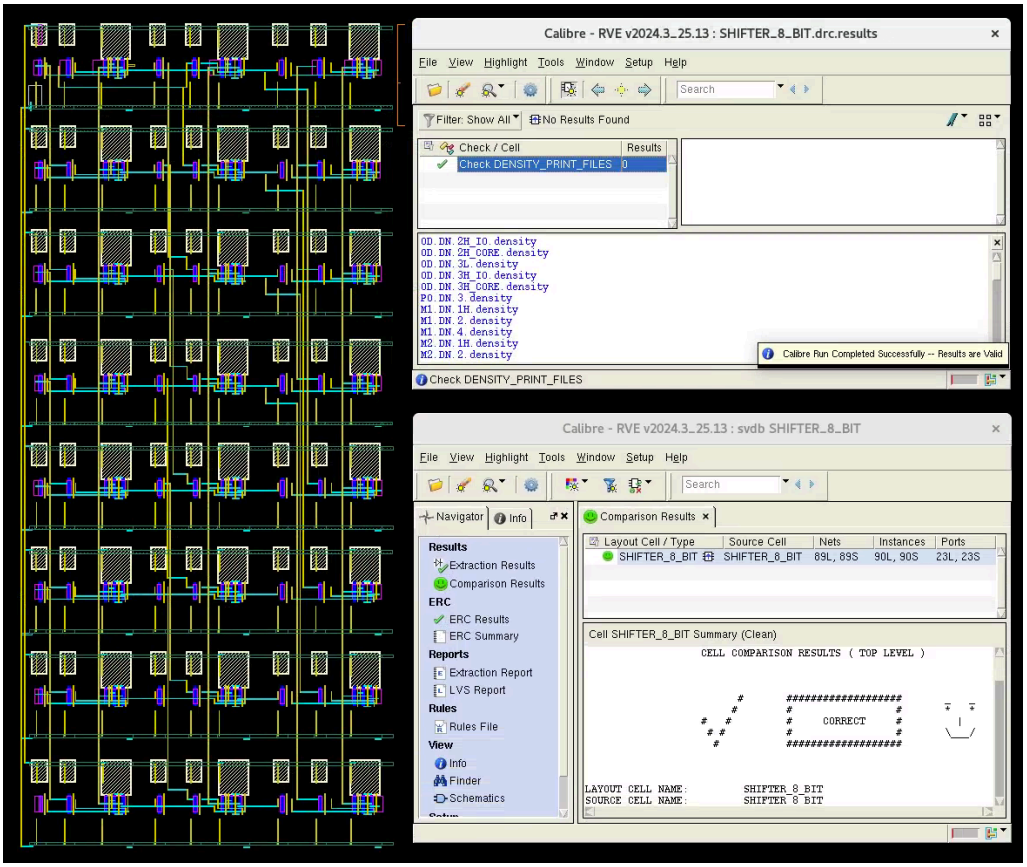
Layout of Full Adder:



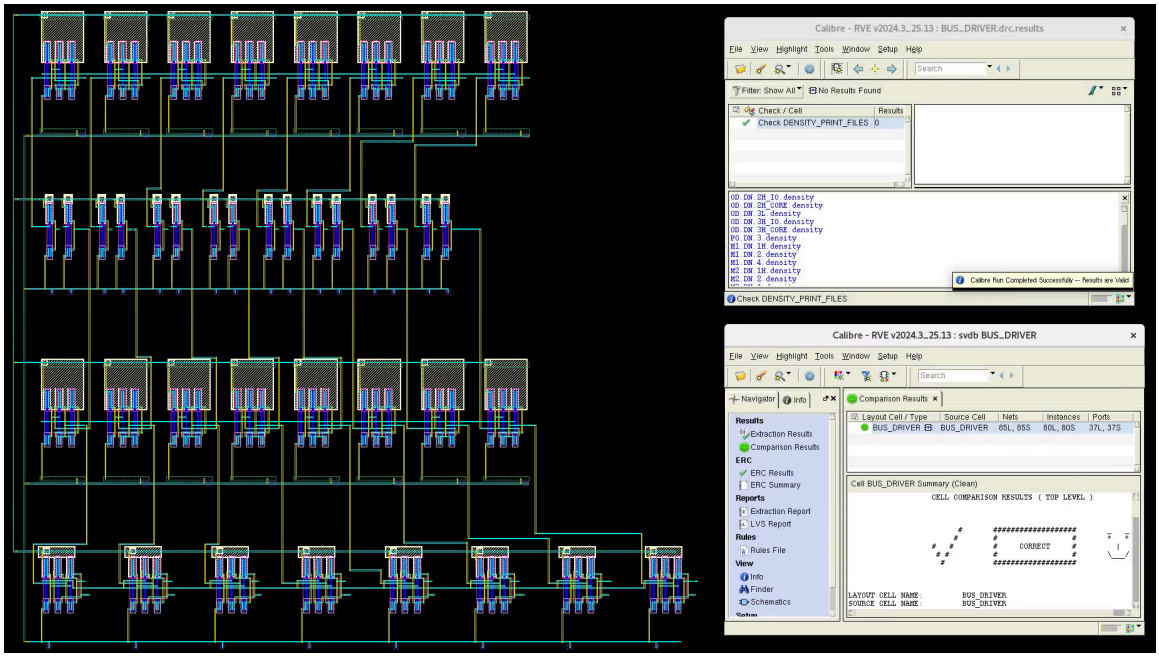
Layout of latch:



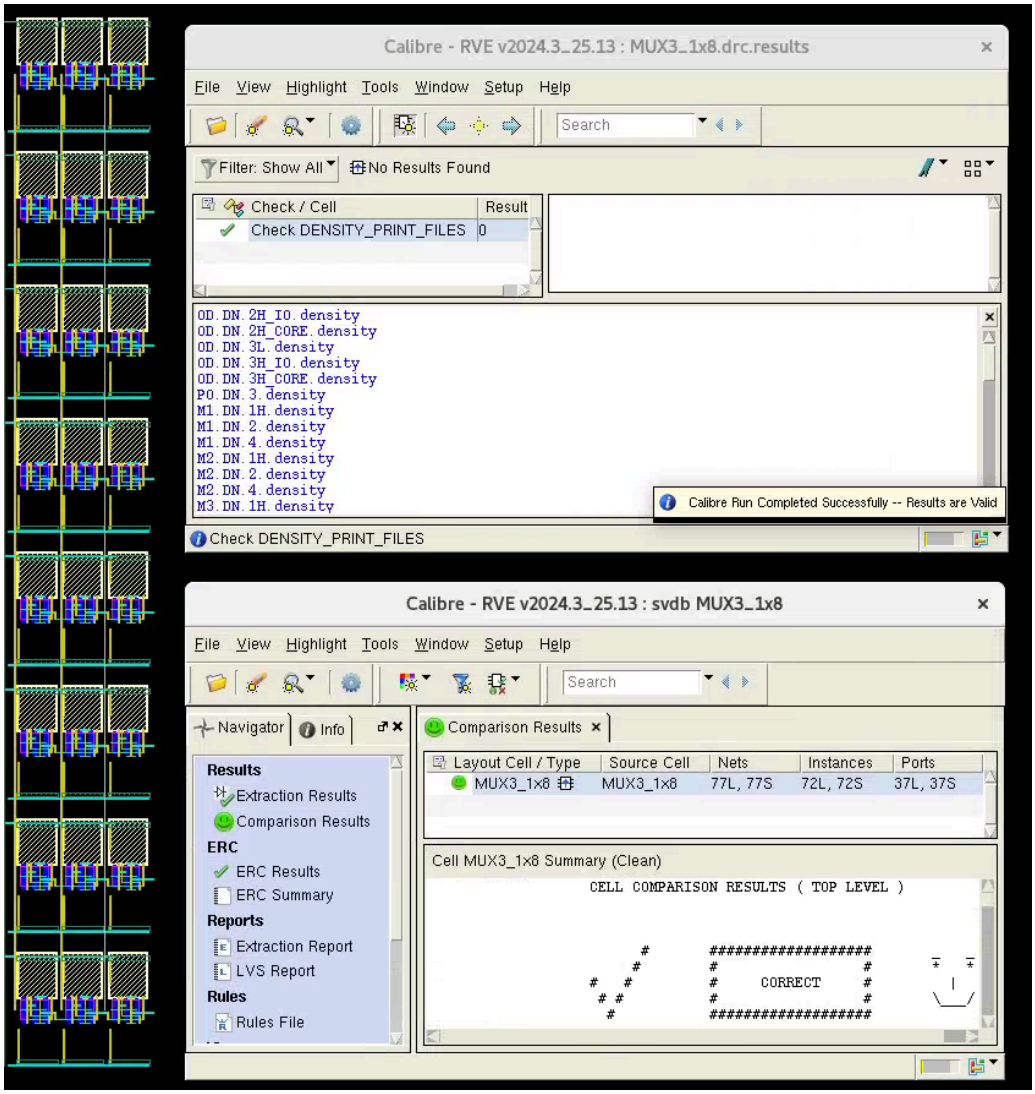
Layout of shifter:



Layout of bus driver:



Layout of 3 to 1 mux:


Calibre - RVE v2024.3_25.13 : MUX3_1x8.drc.results
File View Highlight Tools Window Setup Help
Search
Filter: Show All
No Results Found
Check / Cell
Result
Check DENSITY_PRINT_FILES
0
OD.DN.2H_IO.density
OD.DN.2H_CORE.density
OD.DN.3L.density
OD.DN.3H_IO.density
OD.DN.3H_CORE.density
PO.DN.3.density
M1.DN.1H.density
M1.DN.2.density
M1.DN.4.density
M2.DN.1H.density
M2.DN.2.density
M2.DN.4.density
M3.DN.1H.density
Calibre Run Completed Successfully -- Results are Valid
Check DENSITY_PRINT_FILES
Calibre - RVE v2024.3_25.13 : svdb MUX3_1x8
File View Highlight Tools Window Setup Help
Search
Navigator
Info
Results
Extraction Results
Comparison Results
ERC
ERC Results
ERC Summary
Reports
Extraction Report
LVS Report
Rules
Rules File
Comparison Results
Layout Cell / Type
Source Cell
Nets
Instances
Ports
MUX3_1x8
MUX3_1x8
77L, 77S
72L, 72S
37L, 37S
Cell MUX3_1x8 Summary (Clean)
CELL COMPARISON RESULTS (TOP LEVEL)
CORRECT