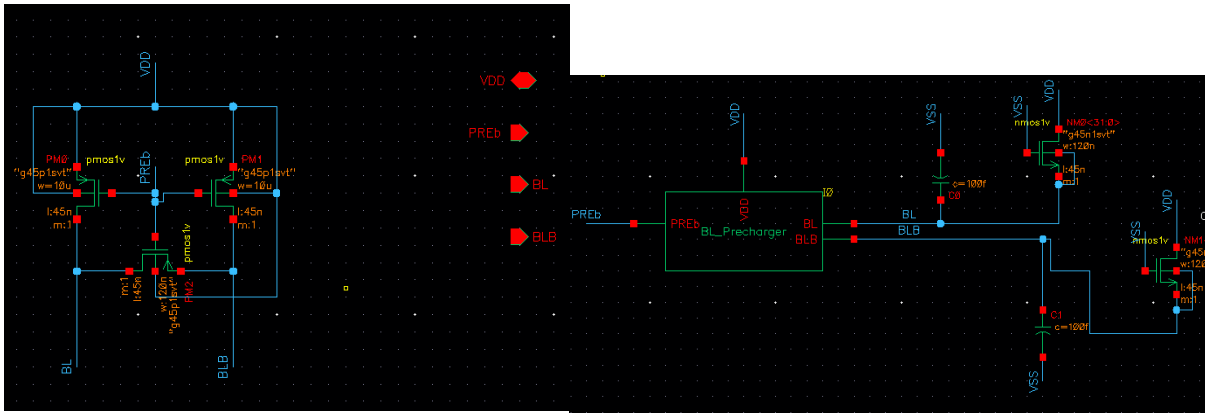
Digital Integrated Circuits Term Project

Part I. Schematic Design

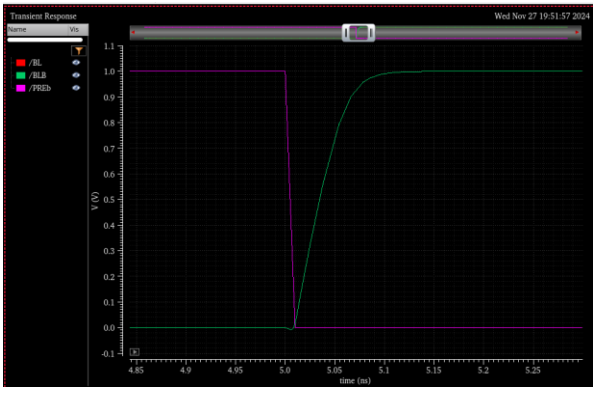
To design the full system SRAM memory system, peripheral circuits were first built.

1. BL Precharger

The schematic for Bitline Precharger and its testbench is shown below:



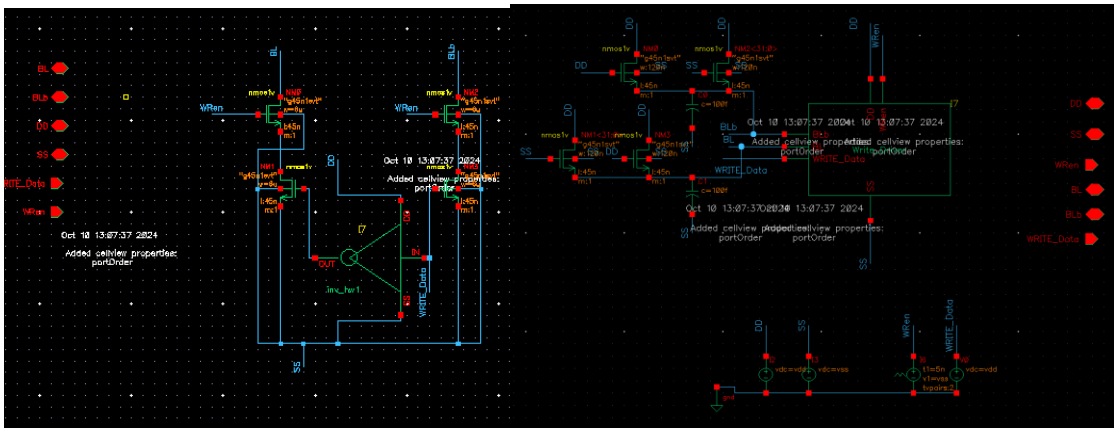
To account for bitline's capacitance due to the long wires, and the effect of 32 transistors connected to each bitline, a relatively large capacitance of 100fF is included, with transistors sized similarly to those in the SRAM. Initially, with all the PMOS transistors to have the same size of 120nm, the precharge time exceeded 0.2ns, which did not meet our required specification. After switching the PMOS pull up transistor's size to 10um, I was able to make the pull up delay less than 0.2ns. The pre simulation result is as follows.



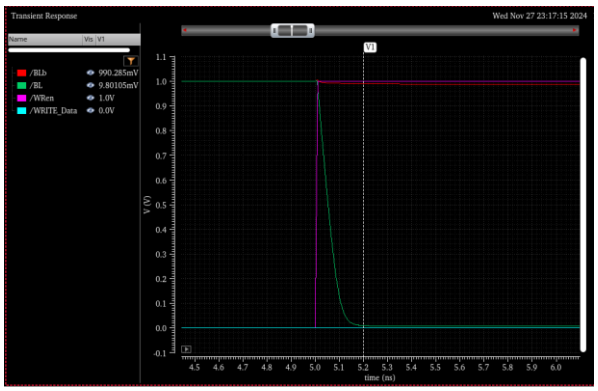
We can observe BL and BLB are precharged to VDD well before 5.2ns, which means the rising time of the signal is less than 200ps.

2. Write Driver

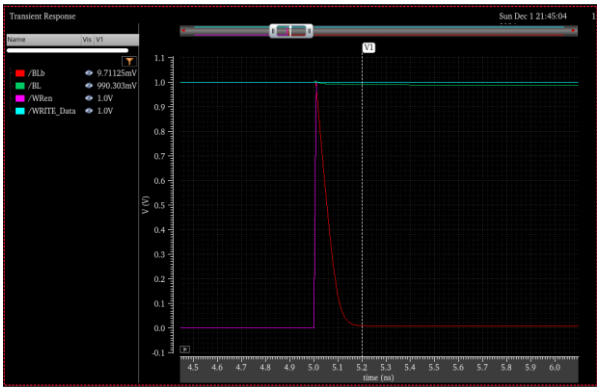
The schematic for Write Driver and its testbench is shown below:



Similarly, to consider bitline's capacitance due to long wire, and also the effect of 32 transistors connected to each bitline, capacitance is included as relatively large 100fF, and transistors are included with the same size as that of SRAM's. In order to make the pull down delay less than 0.2ns, NMOS transistors of size 6um were used. The pre simulation result is as follows.



<Write Data = 0>

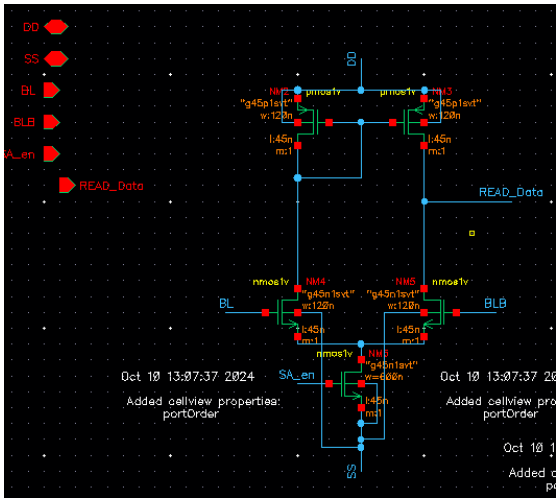


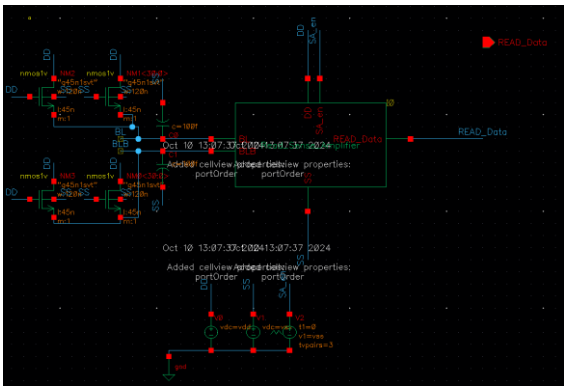
<Write Data = 1>

The simulation results show that when WRen is 1, writing data 0 causes BL to drop to 0 while BLB stays 1. Conversely writing data 1 keeps BL at 1 and BLB drops to 0 with transitions occurring in less than 200ps.

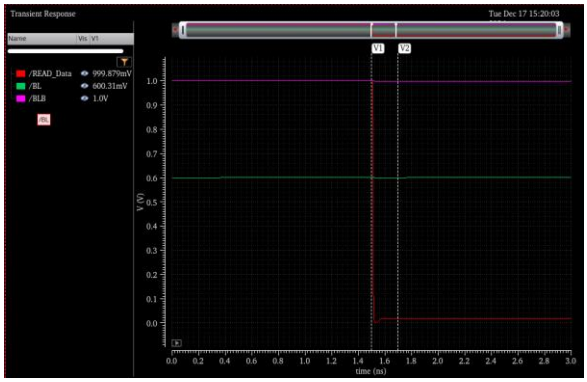
3. Read Sense Amplifier

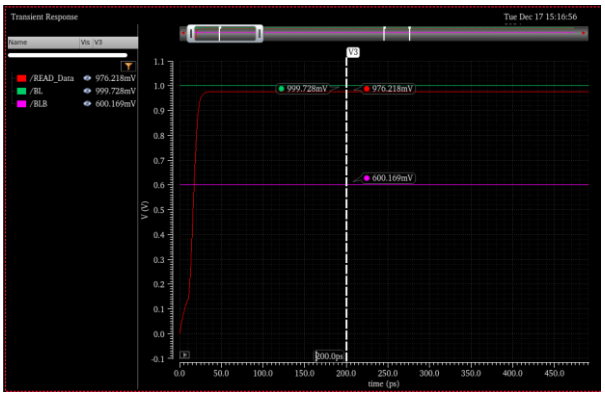
The schematic for Read Sense Amplifier and its testbench is shown below: The testbench includes the effect of transistors and bitline capacitance connected to the bitline. For Read Sense Amplifier, I only had to increase the width of the NMOS that has the signal SA_en as an input, because the differential amplifier quickly amplifies the voltage difference of BL and BLB. Considering the pull down delay of less than 0.2ns, the width of the SA_en input NMOS was set as 600nm.





The pre simulation results are shown below:



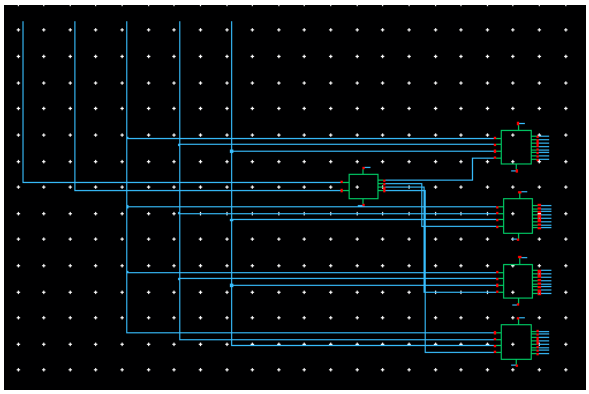


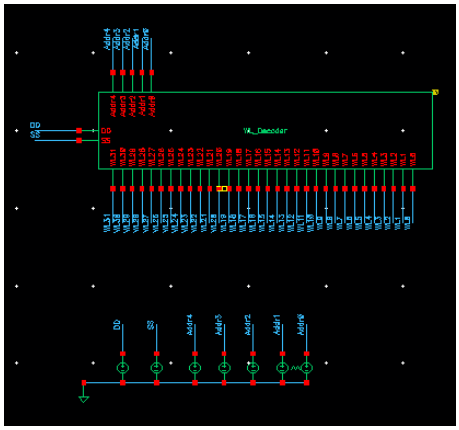
<Read Data = 0> <Read Data = 1>

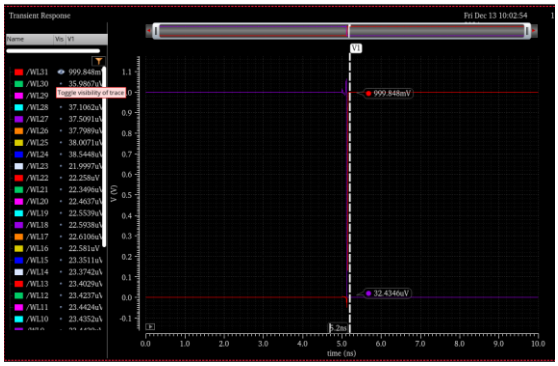
For reading 1, the SA_en signal was enabled at 0.01ns, because READ_DATA node floats when the SA_en signal is disabled, causing deviations from the expected value due to noise when its off. Additionally, BL and BLb are set as 1V or 0.6V, assuming sufficient time has passed when sensing the BL and BLb. The simulation results show that the Read Sense Amplifier successfully amplifies the voltage difference between bitline and bitline_bar, within 0.2ns.

4. WL Decoder

WL Decoder was constructed with 1 2:4 decoder, and 4 3:8 decoder with enable. From the 5 bit address address[4:0], address[4] and address[3] was used as an input for 2:4 decoder to make an enable for the 4 3:8 decoders.



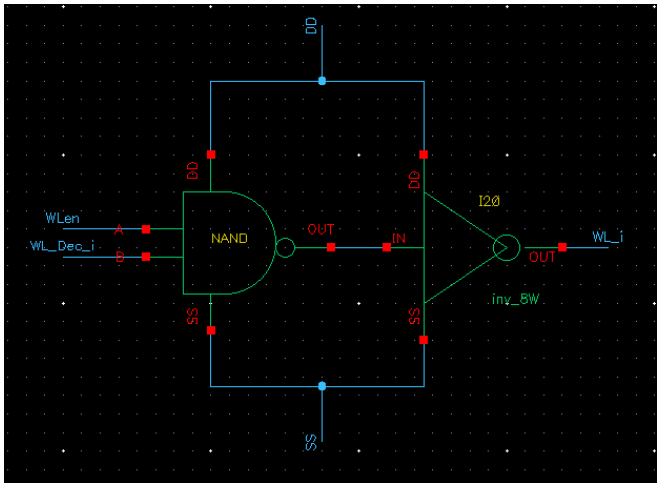


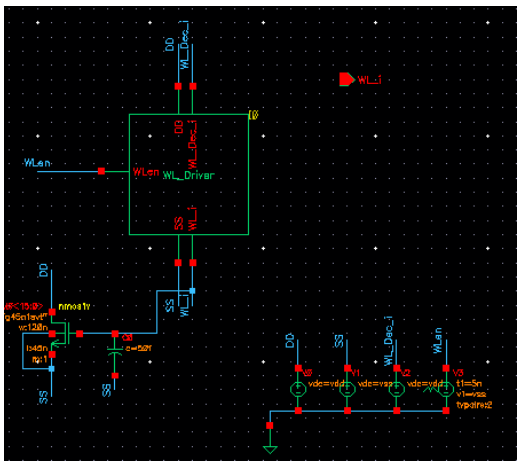


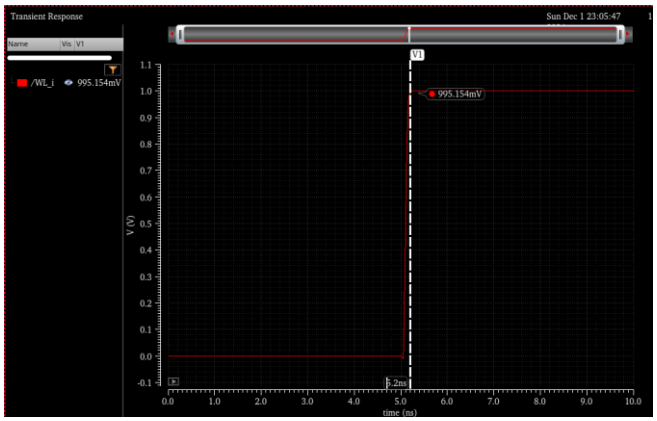
Switching the addresses (address[4] to address[0]) from 0 to 1, the simulation result shows that WL0 is changed from 1 to 0, and WL31 is changed from 0 to 1 in less than 0.2ns, which satisfies our specification.

5. WL Driver

In order to make the pull up delay less than 0.2ns, I had to make the inverter that comes after the NAND gate larger than the NAND gate's size. Initially using an unit inverter, it was not possible to meet our required specification. Considering the word line capacitances and the transistors, inverter's size was selected to be 8 times larger than the unit inverter because it was the smallest inverter size that could satisfy our pull up delay requirement. The schematic for Write Driver and its testbench is shown below:



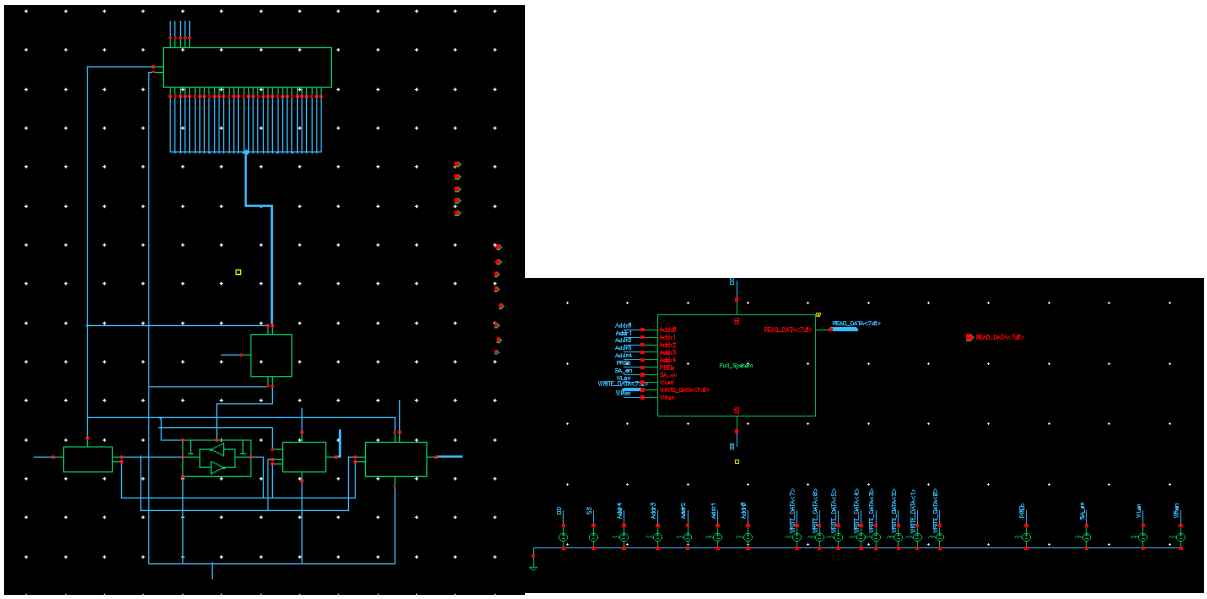




The simulation result shows that when WLen is enabled, it makes the selected word line(WL_i) on in less than 0.2ns.

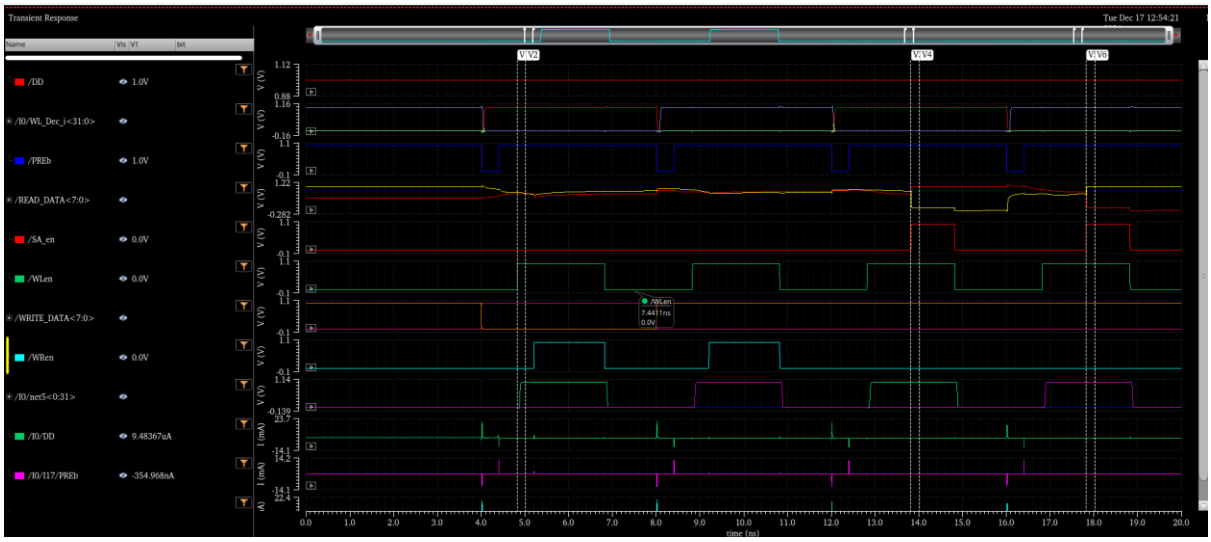
6. Full System

Merging all the peripheral circuits(8 BL precharger, 1 WL decoder, 32 WL Driver, 8 Write Driver, and 8 Read Sense Amplifier) and the SRAM cell array, the fully merged system's schematic and its testbench is as below:



To describe the testbench for full system, the simulation was conducted for 20ns, with considering 1 clock cycle as 4ns. Signals were written starting at 4ns. Data 10101010 was written to address 11111 at 4ns, and data 01010101 was written to address 00000 at 8ns. At the next clock cycle starting from 12ns, address 11111 was looked up to read the data, and at time 16ns address 00000 was looked up to read the data. The word lines were enabled(WLen was enabled) from 4.82ns to 6.82ns in the first clock cycle, 8.82ns to 10.82ns in the second clock cycle, 12.82ns to 14.82ns in the third clock cycle, and 16.82ns to 18.82ns in the fourth clock cycle. The Read Sense Amplifier was enabled(SA_en was enabled) from 13.82ns to 14.82ns in the first read cycle(12ns to 16ns) and 17.82ns to 18.82ns in the second read cycle(16ns to 20ns). The Write Driver was

enabled(WRen was enabled) from 5.2ns to 6.82ns in the first write cycle(4ns to 8ns), and from 9.2ns to 10.82ns in the second write cycle(8ns to 12ns). The Bitline Prechager was enabled(Preb being low) from 4.01ns to 4.4ns in the first clock cycle, 8.01 to 8.4ns in the second clock cycle, 12.01ns to 12.4ns in the third clock cycle, and 16.01ns to 16.4ns in the fourth clock cycle. The results are as follows.



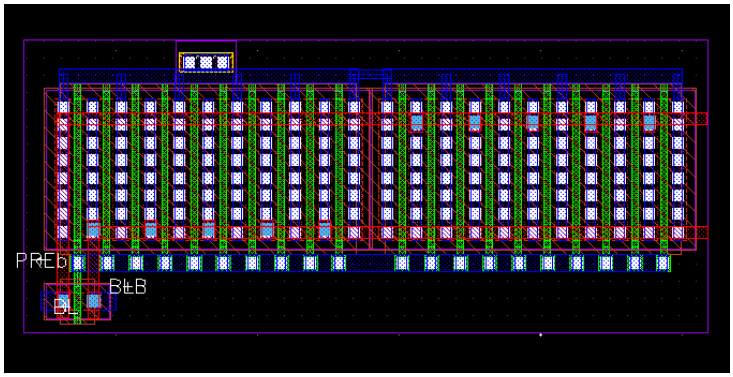
We can observe that the full system successfully writes and reads the data within 0.2ns of rising time and falling time, which meets our required specifications. Also, the word lines indicated as /I0/net5<0:31> shows transition within 0.2ns after WLen is activated, which meets our specification.

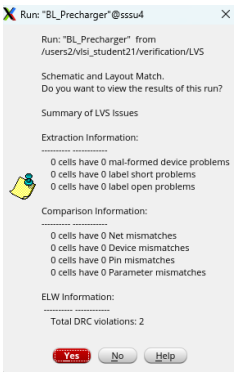
Part II. Layout

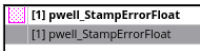
Similar to building the schematic of the full system SRAM memory system, the layout of the peripheral circuits was first made.

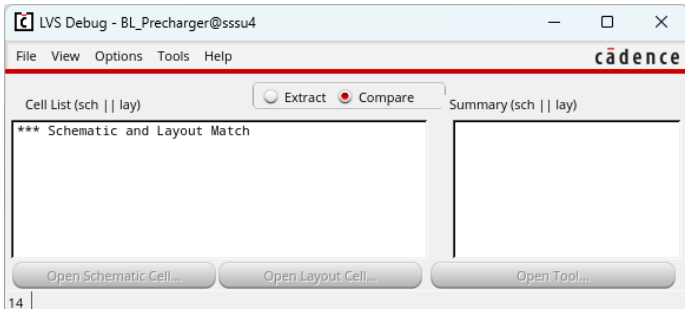
1. BL Precharger

The layout and the DRC, LVS results of the BL precharger are as below:



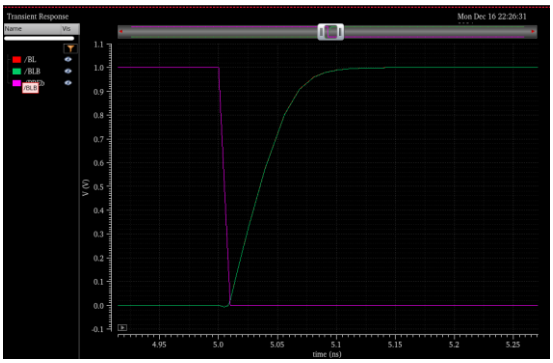






The layout matched with the designed schematic, but there was an DRC error due to the absence of the Metal – P substrate via which fixes the potential of the p substrate to the ground. However, this not an issue when merging all the peripheral circuits, because the other circuits fixes the p-substrate potential to ground, preventing any DRC errors later.

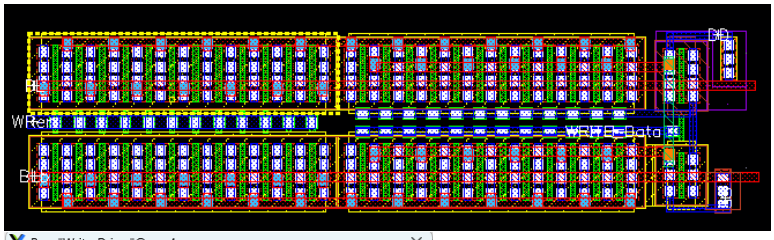
After extracting the R,C values of the circuit, post simulation was conducted. Its results are as follows:

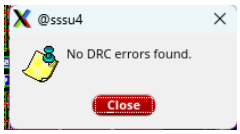


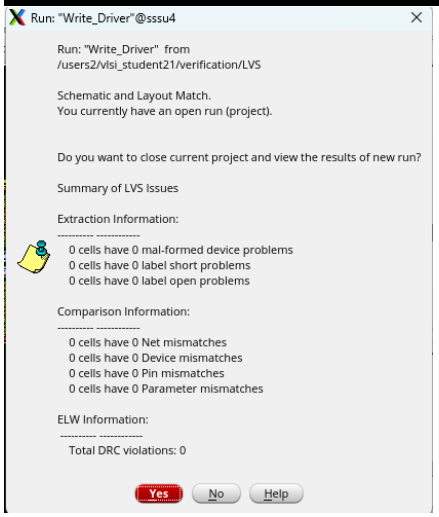
From the result, we can observe that the pull up delay is still less than 0.2ns, which satisfies our specifications.

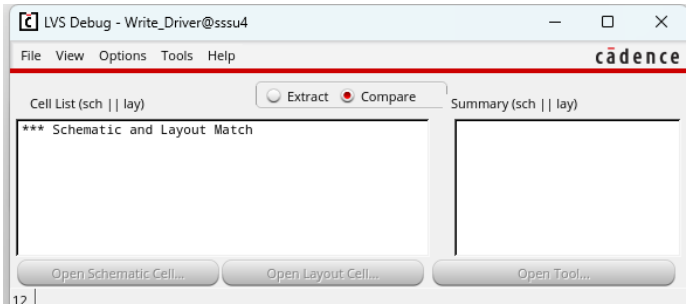
2. Write Driver

The layout and the DRC, LVS results of the Write Driver are as below:



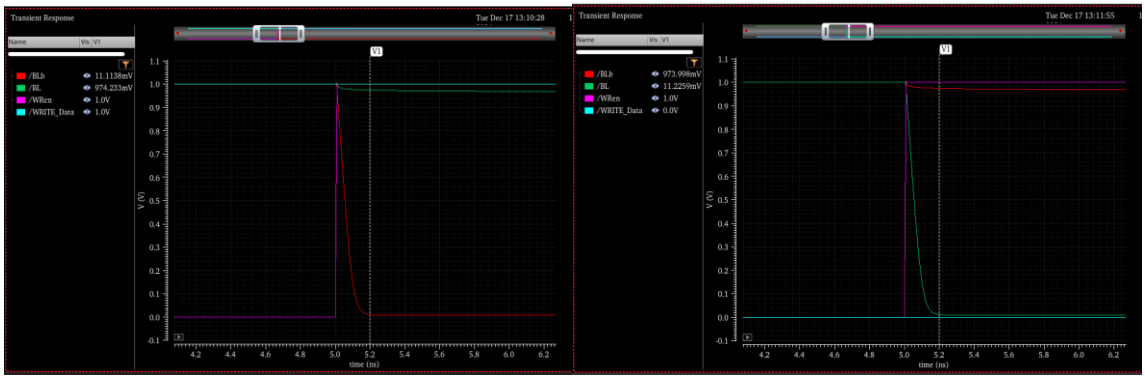






The layout matched our schematic design, without any DRC problems.

After extracting the R,C values of the circuit, post simulation was conducted. Its results are as follows:

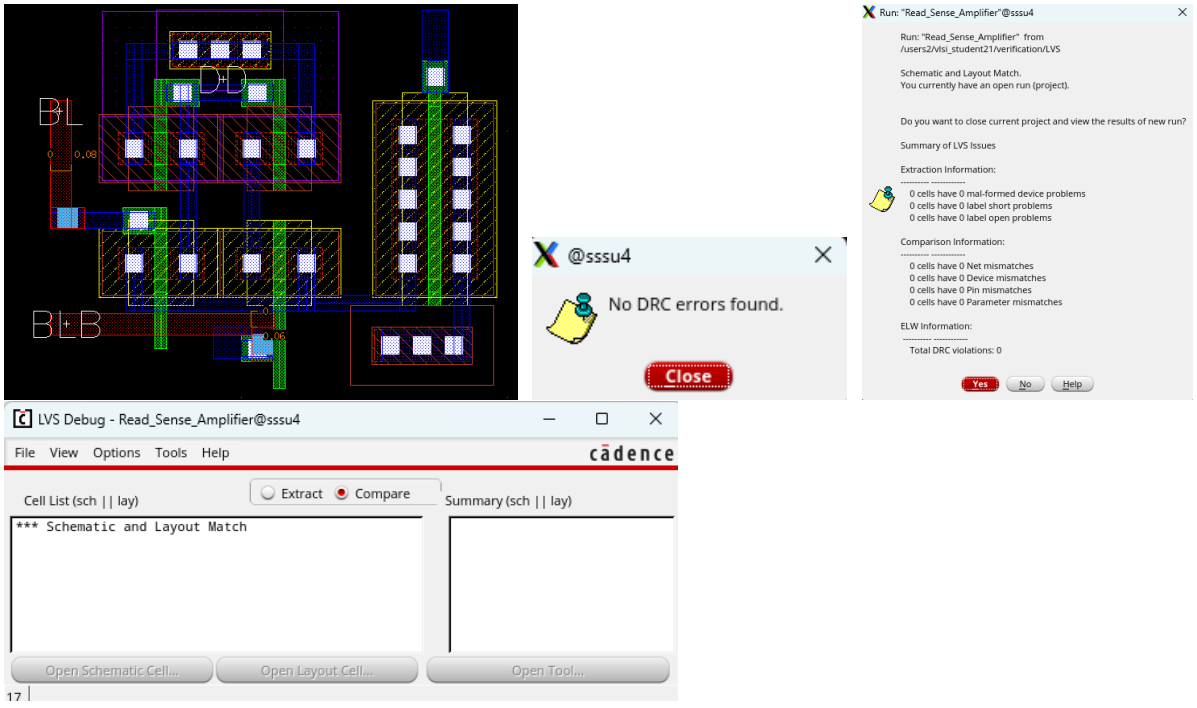


< Write Data = 1> <Write Data = 0>

From the result, we can observe that the pull up delay is still less than 0.2ns, which satisfies our specifications. However, we can observe the drop in BL's voltage, which drops to 0.973V. This error made by the voltage drop was negligible considering the noise margin.

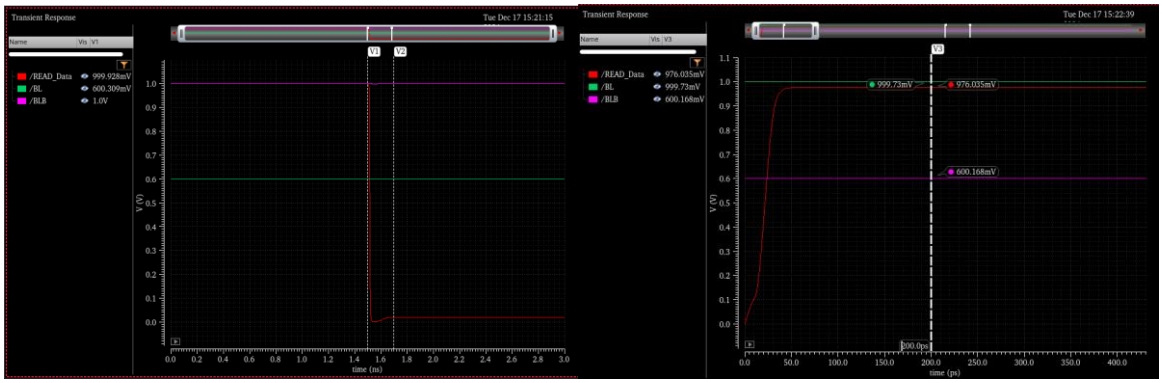
3. Read Sense Amplifier

The layout and the DRC, LVS results of the Read Sense Amplifier are as below:



The layout matched our schematic design, without any DRC problems.

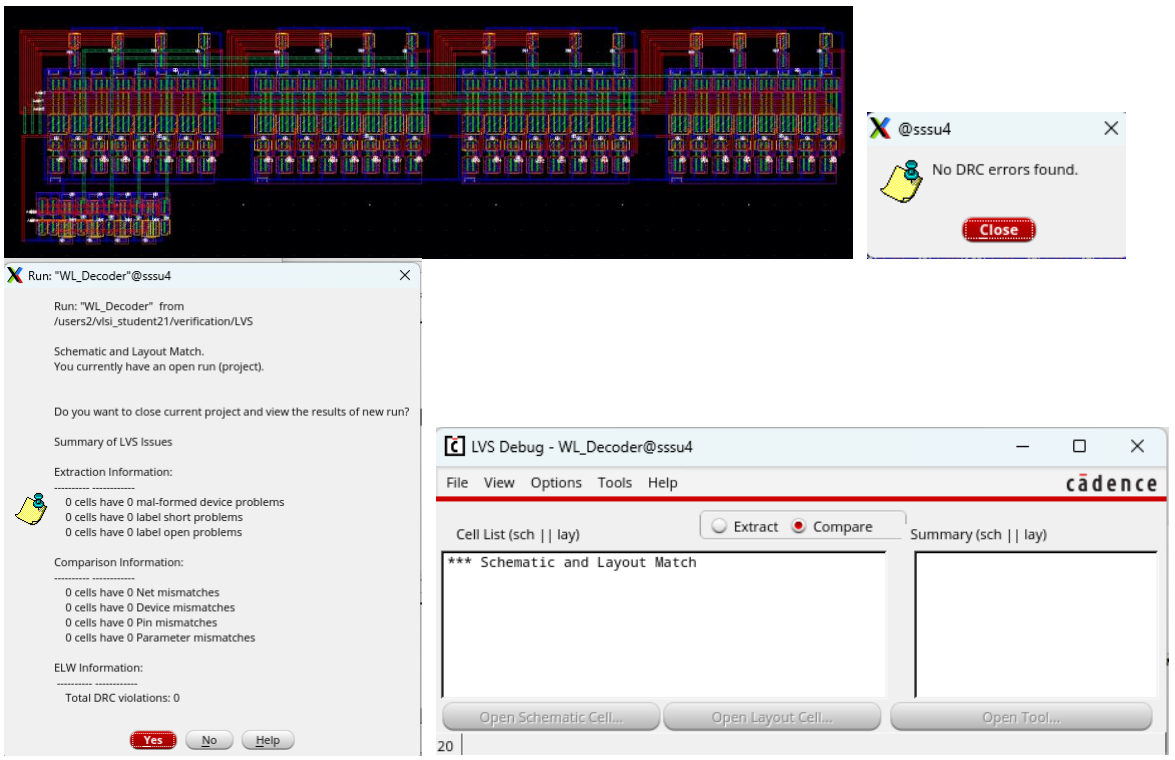
After extracting the R,C values of the circuit, post simulation was conducted. Its results are as follows:



From the result, we can observe that the pull up delay and pull down delay are still less than 0.2ns, which satisfies our specifications.

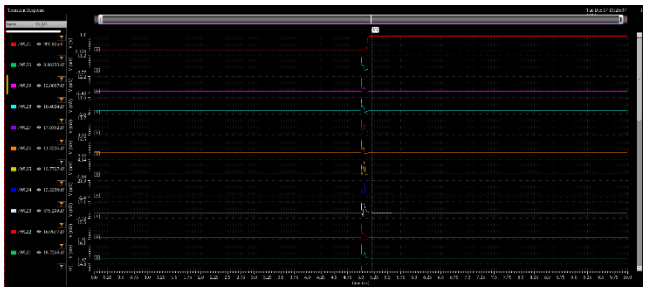
4. WL Decoder

The layout and the DRC, LVS results of the WL Decoder are as below: The same four layout instances are the 3 to 8 decoders, which are enabled by the output of the 2 to 4 decoder placed beneath the four 3 to 8 decoders.



The reason why 2 to 4 decoder is at the bottom of the 4 3 to 8 decoders is because I have considered the arrangement of other peripheral circuits, and in order to fill out the

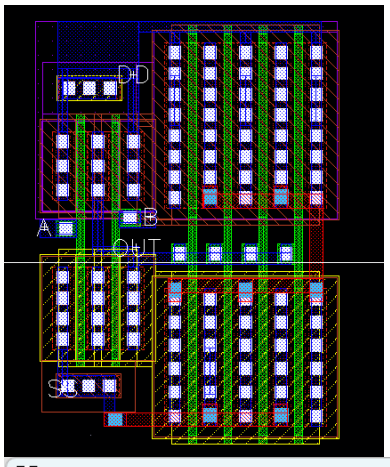
empty spaces and to not make the system's area larger, the 2 to 4 decoder was placed in the following position.

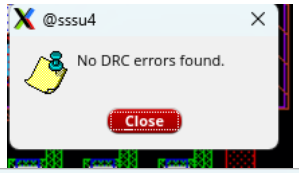


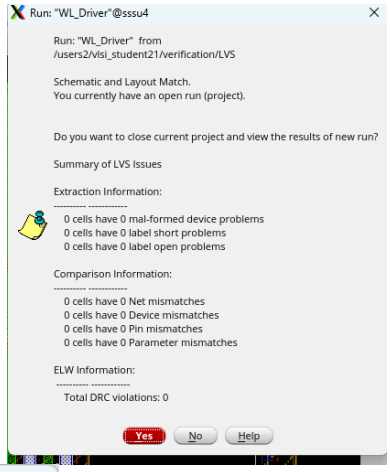
The layout matched our schematic design, without any DRC problems. The post simulation result shows that the decoder selects one word line within 0.2ns, which meets our specification.

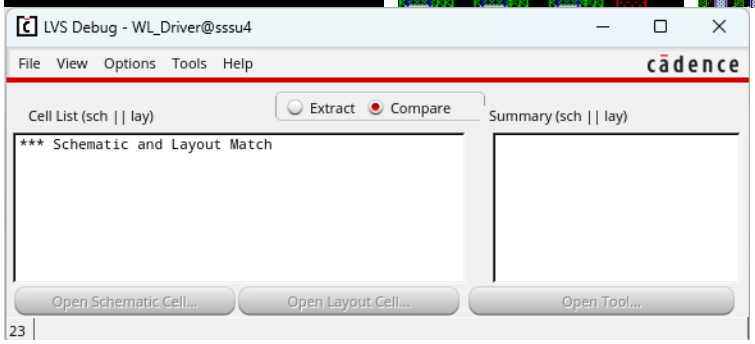
5. WL Driver

The layout and the DRC, LVS results of the WL Driver are as below:



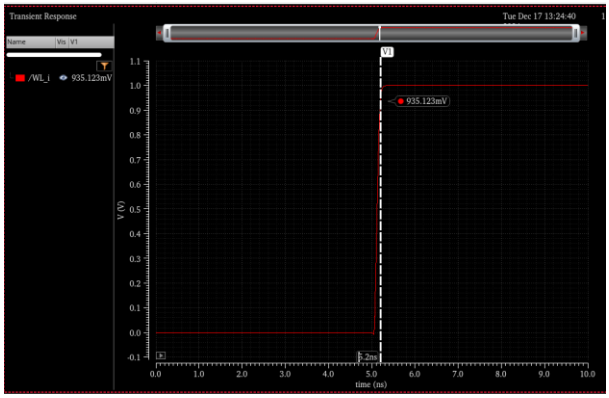






Input A for the 2 input NAND gate is the wordline, and input B for the NAND gate is the WLen signal. The layout matched our schematic design, without any DRC problems.

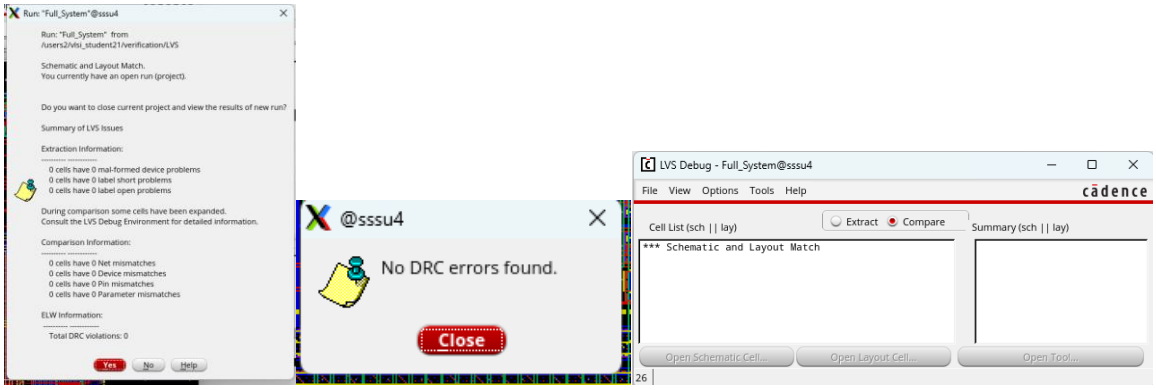
The post simulation result shows that after WLen is enabled, the WL pulls up to

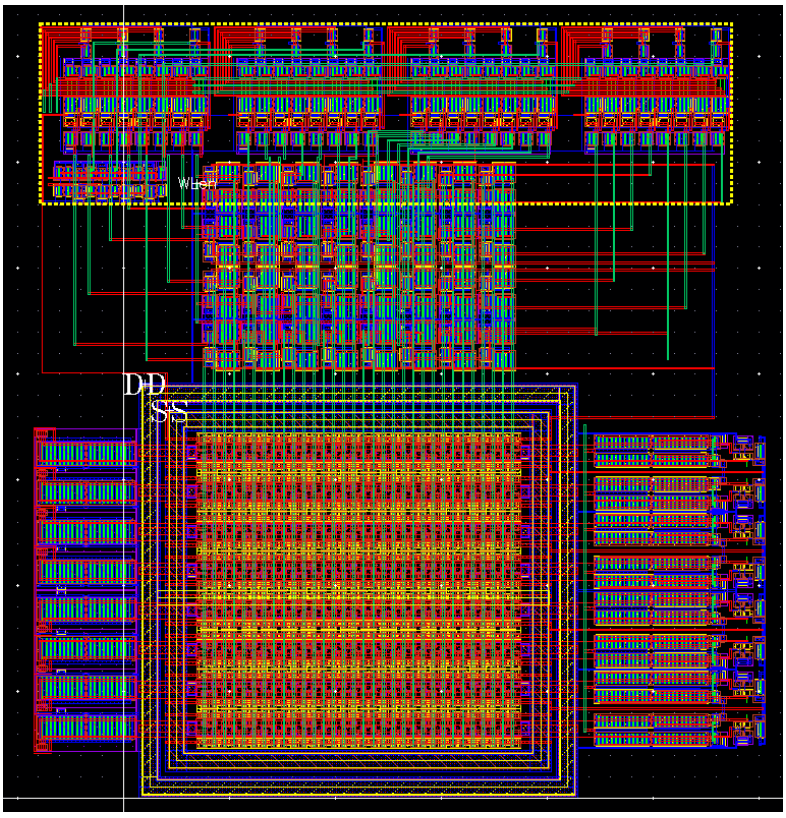


0.935V in 0.2ns, which satisfies our specification.

6. Full system

As we completed all the layouts for the peripheral circuits, all the peripheral circuits(8 BL precharger, 1 WL decoder, 32 WL Driver, 8 Write Driver, and 8 Read Sense Amplifier) and the SRAM cell array were merged to build a full system SRAM memory system layout. To describe the full system's layout, the 32x8 SRAM cell array is placed in the center. On the left, 8 Bitline Prechargers are connected to BL and BLbs each. The WL Decoder and WL Driver are placed above the cell array. To match the width of the WLs with the WL Drivers, 32 WL Drivers are placed as a 4x8 array, each column connecting to 4 WLs. On the right of the cell array, 8 Write Drivers and 8 Read Sense Amplifiers are placed, connected to each of the BL and BLbs. The layout, DRC, LVS results are as below.





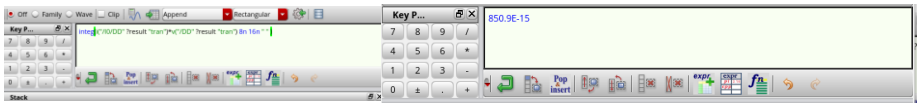
The layout matched our schematic design, without any DRC problems.

The post simulation result is as above. Even considering all the R,C delay made after the layout, the SRAM still managed to operate properly, reading the Data stored within 0.2ns of rising and falling time. Also, internal signals such as word line indicated as /I0/net5<0:31> transitions within 0.2ns, which meets our specification.

III. Analysis

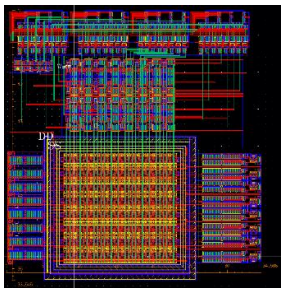
1. Measuring FoM

To evaluate the Figure of Merit, I measured the average power consumed over 2 clock cycle that includes both writing and reading. Specifically, from time 8ns to 16ns, which conducts writing in 8ns to 12ns and reading in 12ns to 16ns. Since the total power consumed over 8ns to 16ns was calculated as 850.9×10^{-15}, the average power consumed during the time interval can be calculated as $850.9 \times 10^{-15} \times \frac{1}{8ns} = 106.3625 \times 10^{-6}\text{W}$.



Also, using the ruler, the total area of the full system SRAM is measured. From the image, the observed dimensions are width 34.586um and height 36.545um. The total area is then calculated as $34.856\mu m \times 36.545\mu m = 1273.8 \times (\mu m)^2$

Therefore, the figure of merit FoM is calculated as $FoM = 250MHz \times \frac{1}{106.3625 \times 10^{-6}W} \times \frac{1}{1273.8\ (\mu m)^2} = 1.845 \times 10^9/\text{J} \cdot (\mu\text{m})^2$



To maximize the FoM, the layout was redesigned to be as compact as possible. Initially, the peripheral circuits especially the WL Driver and BL precharger did not align properly with the word lines and bit lines. This resulted in wasted area, lowering the FoM. By rearranging and carefully aligning the circuits to the given word lines and bit lines, the layout area was significantly reduced, which improved the FoM.

2. Functionality and Simulation conditions

The simulation condition as mentioned earlier, the simulation was conducted for 20ns, with considering 1 clock cycle as 4ns. Signals were written starting at 4ns. Data 10101010 was written to address 11111 at 4ns, and data 01010101 was written to address 00000 at 8ns. At the next clock cycle starting from 12ns, address 11111 was looked up to read the data, and at time 16ns address 00000 was looked up to read the data.

The word lines were enabled(WLen was enabled) from 4.82ns to 6.82ns in the first clock cycle, 8.82ns to 10.82ns in the second clock cycle, 12.82ns to 14.82ns in the third clock cycle, and 16.82ns to 18.82ns in the fourth clock cycle. The Read Sense Amplifier was enabled(SA_en was enabled) from 13.82ns to 14.82ns in the first read cycle(12ns to 16ns) and 17.82ns to 18.82ns in the second read cycle(16ns to 20ns). The Write Driver was enabled(WRen was enabled) from 5.2ns to 6.82ns in the first write cycle(4ns to 8ns), and from 9.2ns to 10.82ns in the second write cycle(8ns to 12ns). The Bitline Prechager was enabled(Preb being low) from 4.01ns to 4.4ns in the first clock cycle, 8.01 to 8.4ns in the second clock cycle, 12.01ns to 12.4ns in the third clock cycle, and 16.01ns to 16.4ns in the fourth clock cycle.

With this condition, we were able to observe correct functionalities of both Write Operation and Read Operation within 0.2ns of transition time and 250MHz clock frequency(4ns clock cycle), meeting the design specification.

For Write operation, the input address is decoded by the WL Decoder. The BL precharger then precharges all the bitlines to VDD. When WLen is enabled, the decoded word line is activated. Then, the Write Driver drives the data into the bitlines, enabled by WRen.

For Read operation, the SRAM system goes through the same process until word line is being enabled. Then, the SA_en signal is enabled, activating Read Sense Amplifier to read the data stored at the selected address.

3. Limitations

While the SRAM system met the required specifications, the following limitations were observed.

From the full system layout design, the empty spaces between the WL decoder and the BL precharger, Write Driver, Read Sense Amplifier were noticable. If it were possible to figure out how to fill up the open space by rearranging the layout design, the FoM will be calculated larger. Despite efforts to reduce the empty space, it was not possible to reduce beyond the current design.

Also, for the routing of layout, the routing used up to metal 3 in the layout. However, the metals were not laid out as the rule suggests, as the layout used metal 2 as a vertical line and metal 3 as a horizontal line. If the layout's routing followed the recommended rules, the parasitic capacitances and resistances will be smaller than current design, making a faster SRAM system.

4. Conclusion

The SRAM system that I have designed for the project showed correct functionality of read and write of the SRAM memory with operation frequency of 250Mhz, within 0.2ns of transition time. While the layout was optimized to reduce area therefore to improve FoM, further improvements are possible such as reducing the empty spaces and routing following the recommended way are possible.