EE577A Phase 1 Lab Report

In Memory Computing using Vector Bitwise Boolean Operations in 10T-SRAM Cell Utilizing Multi-Vt Structure For Power Optimization

Group Name: 4TSRAM

Group Members:
Ziqiao Fan
Zeyu Xie
Chunxiao Lin
Mustafa Altay Karamuftuoglu

November 10, 2020

Objective

For the future, we are planning to use multi-Vt design for the SRAM. Additionally, we also assign an inverter to the pull down network in order to decrease the leakage current through the NMOS.

Project Plan

For the phase 2, we are planning to apply different voltages to the body of NMOS and PMOS transistors in the 10T SRAM cell. Additionally, for the grounding line, we plan to use a big inverter circuit that will drive in order to decrease the leakage current on the non-activated SRAM cells by increasing the source voltage to VDD. The disadvantage of this is the pull-down section depends on the strength of NMOS in the big inverter. Therefore, we will adjust its size and try to increase the pull-down speed. For overall area constraint, we plan to keep all the expected layouts as compact as possible and this will help further capacitance decrement coming from extracted layout parasitic capacitances. Additionally, we kept the size of pull-down transistor relatively large in 10T SRAM cell; however, we plan to decrease its size since we do non-disturbing read operation on 10T SRAM structure.

Phase One Design

1. 10T-SRAM

Description

Use the 10T structure and the multi-Vt technology. The Vt of NMOS can be adjusted by the Vbn and Vbp. The corresponding layout to the schematic below has 2.788 um^2 (0.87um x 3.205um). For the layout, up to 3 metals are used for all interconnection and signal declarations. For SRAM 2x2, the overall area is 9.161 um^2 (1.48um x 6.19um). If we didn't overlap the areas, the 4 SRAM cells will take 11.152 um^2. Therefore, we have saved approximately 18% of the area due to capability of layout design. This ratio will increase if we try to aim bigger sized SRAM structures. The current sizes of BL read, BLB read, BL access, BLB access, Q pull-down, Q_bar pull-down, Q pull-up and Q_bar pull-up are 150nm, 150nm, 200nm, 200nm, 330nm, 330nm, 120nm and 120nm, respectively. We plan to decrease the sizes of Q and Q_bar pull-down NMOS transistors to 120nm since it will not affect the read operation. As a result, SRAM cell area will decrease.

Since the simulation takes a lot of time and memory space, we initially tested the structure with a single column SRAM design. By doing this, we can observe decoder rows, multiple row selection on in-memory computing, small signal sensing and large signal sensing while confirming all the structures' functionality.

For the single cell, we simulated write 1-> read 1 -> write 0 -> read 0 -> write 1 sequence. In addition to that, we included the current coming from the VDD source in the waveform in order to calculate the capacitances. We assumed that the order of capacitances for wordline and bitline will be approximately the same. Therefore, for the simulations, we use the reference of WL capacitance value (71a F). For the WL, the capacitance will be 63 x 2 x WL-cap. The value 2 is coming from Write BL and Write BLB access transistors and 63 is from non-included SRAM cells since the total number of expected SRAM numbers will be 64 in one row.

When we are in normal mode, we only use one decoder on the left hand side; however, when we are in memory computing mode, we need to use both of the decoders. Therefore, since multiple WL access is required, the results of the decoders are passed through the logic OR gate in order to obtain multiple activated WLs. The related figures for single SRAM cell and one column structure are shown from Figure 1 to 16.

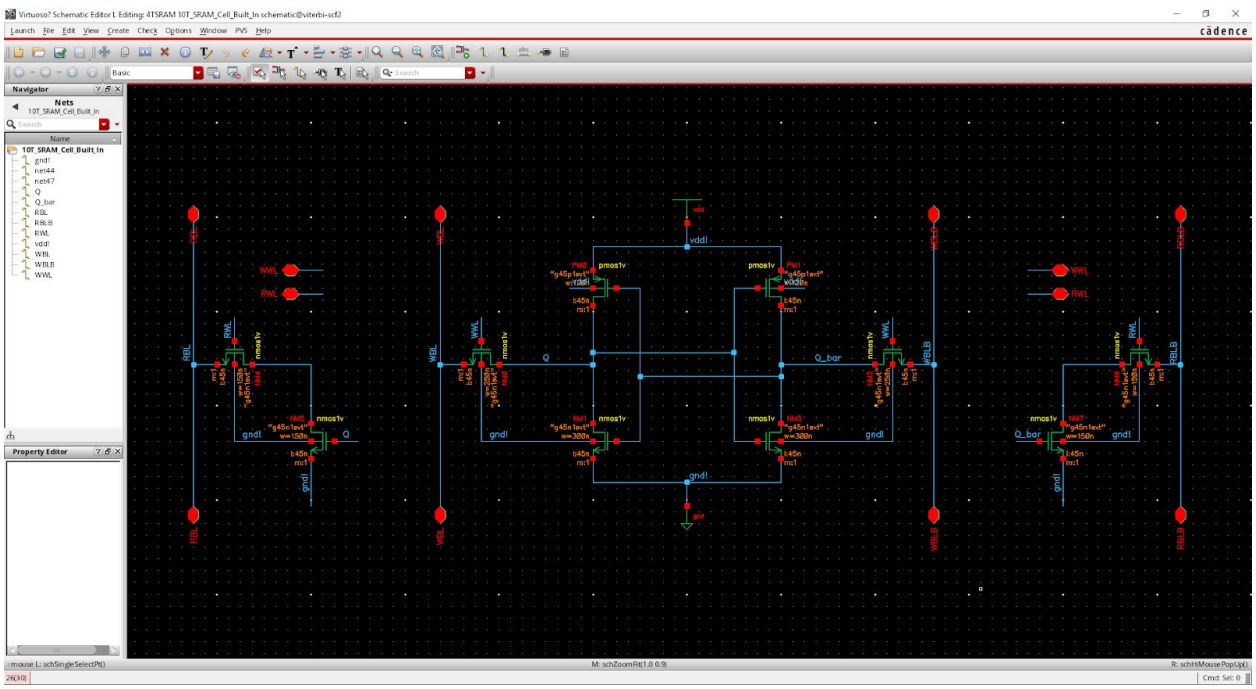

Figure 1: 10T SRAM single cell schematic

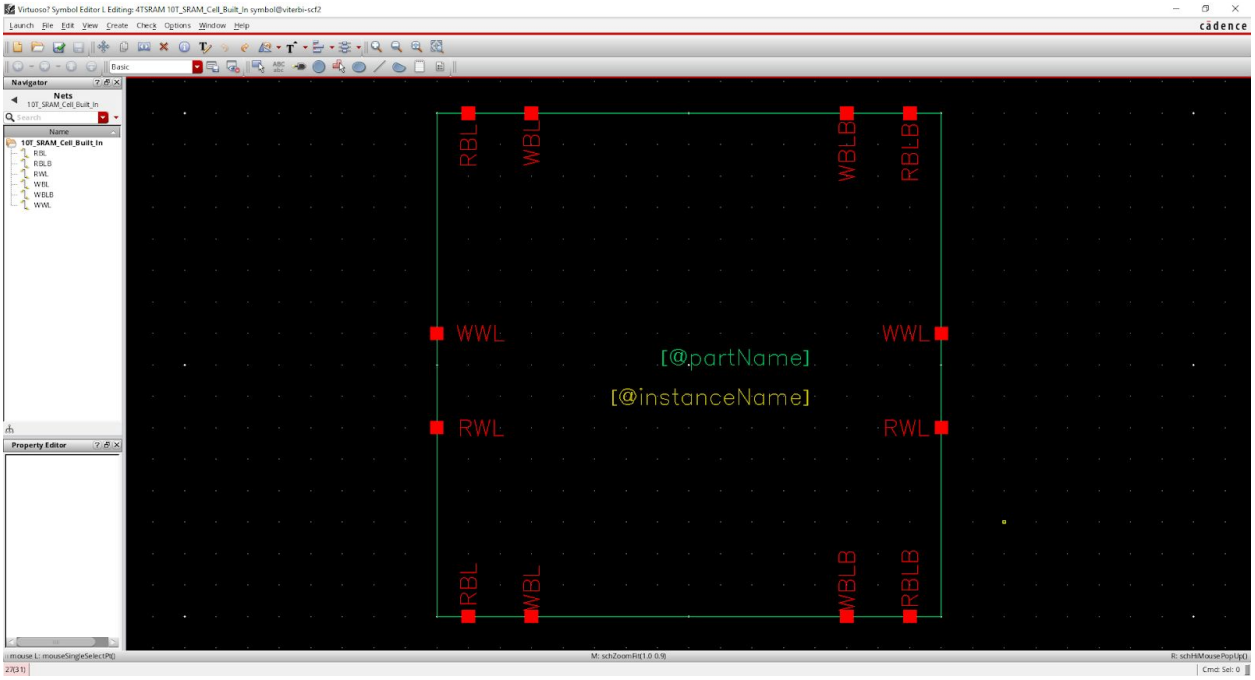

Figure 2: 10T SRAM single cell symbol

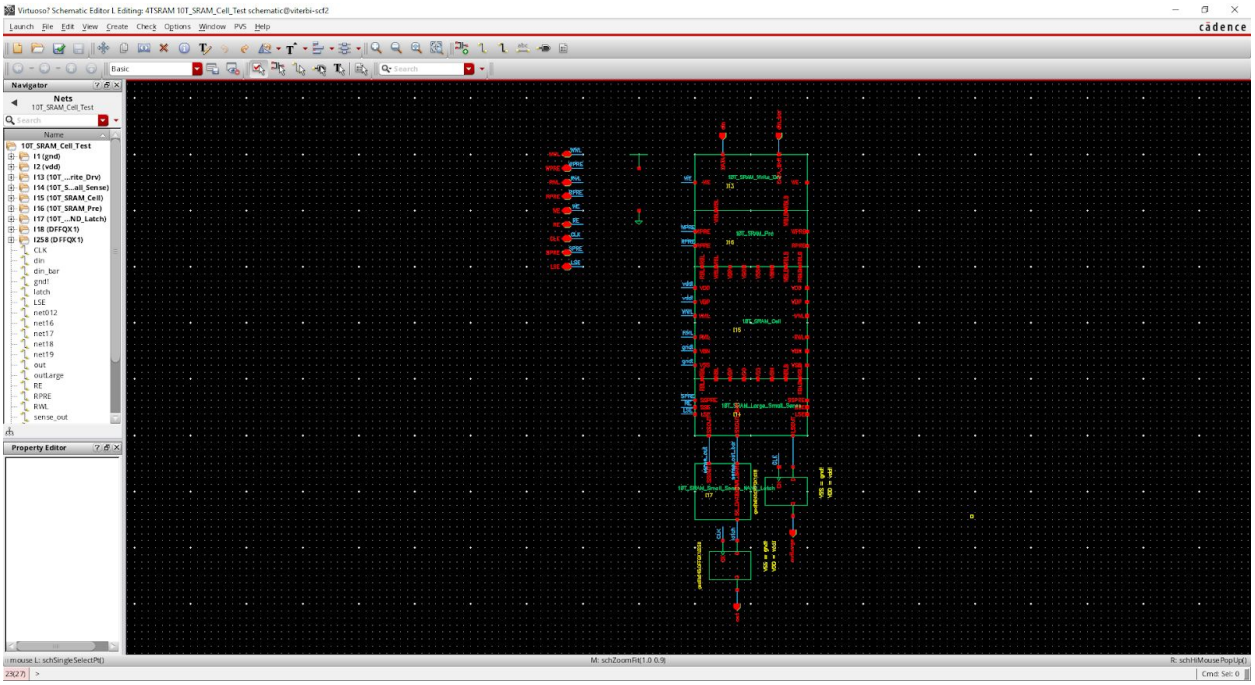

Figure 3: 10T SRAM single cell test schematic

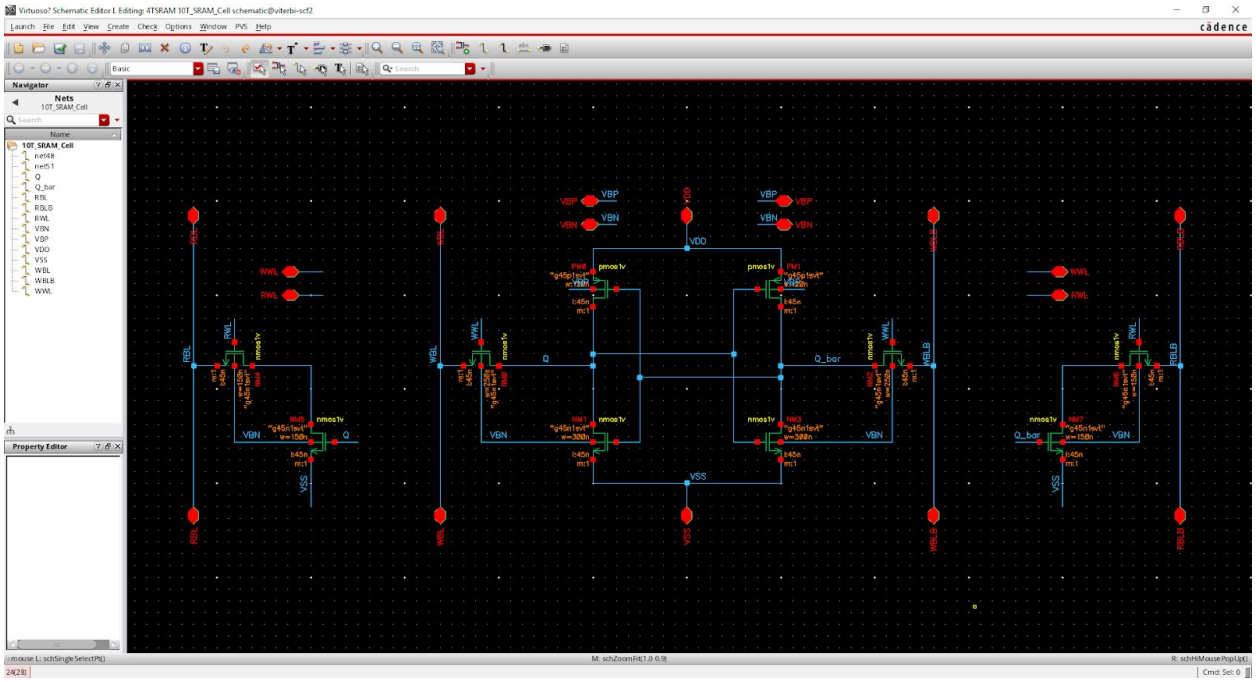

Figure 4: 10T-SRAM single cell body bias schematic

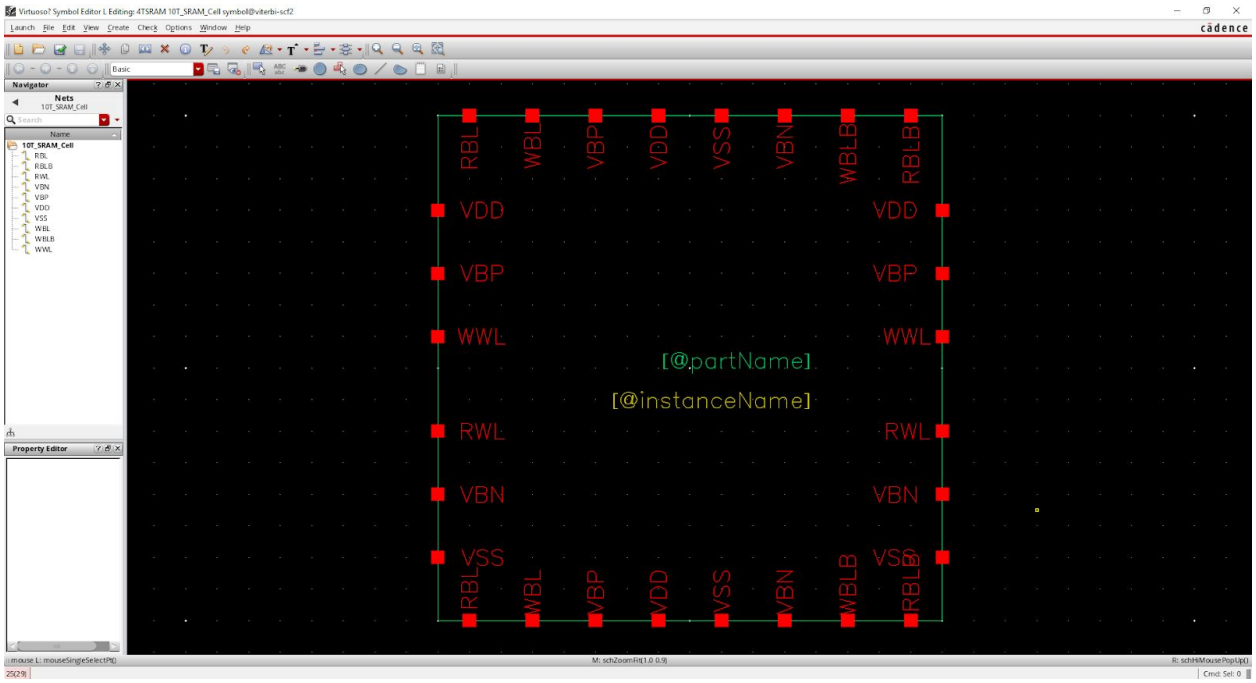

Figure 5: 10T-SRAM single cell body bias symbol

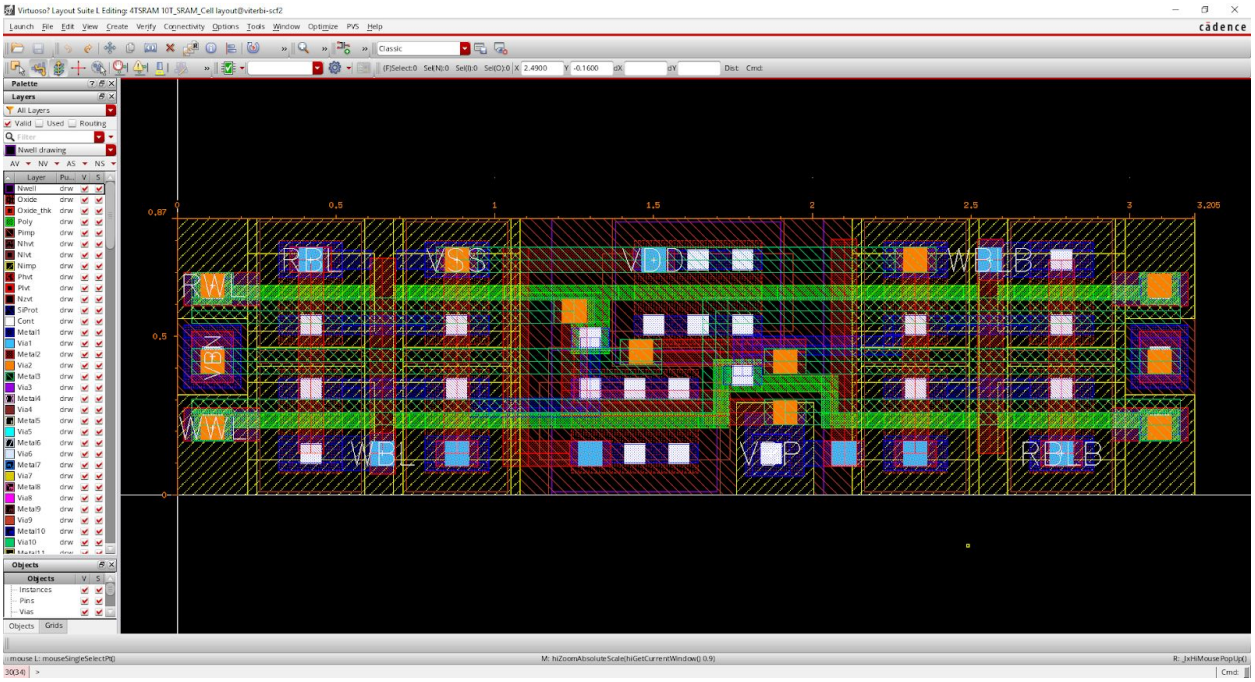

Figure 6: 10T-SRAM single cell body bias layout

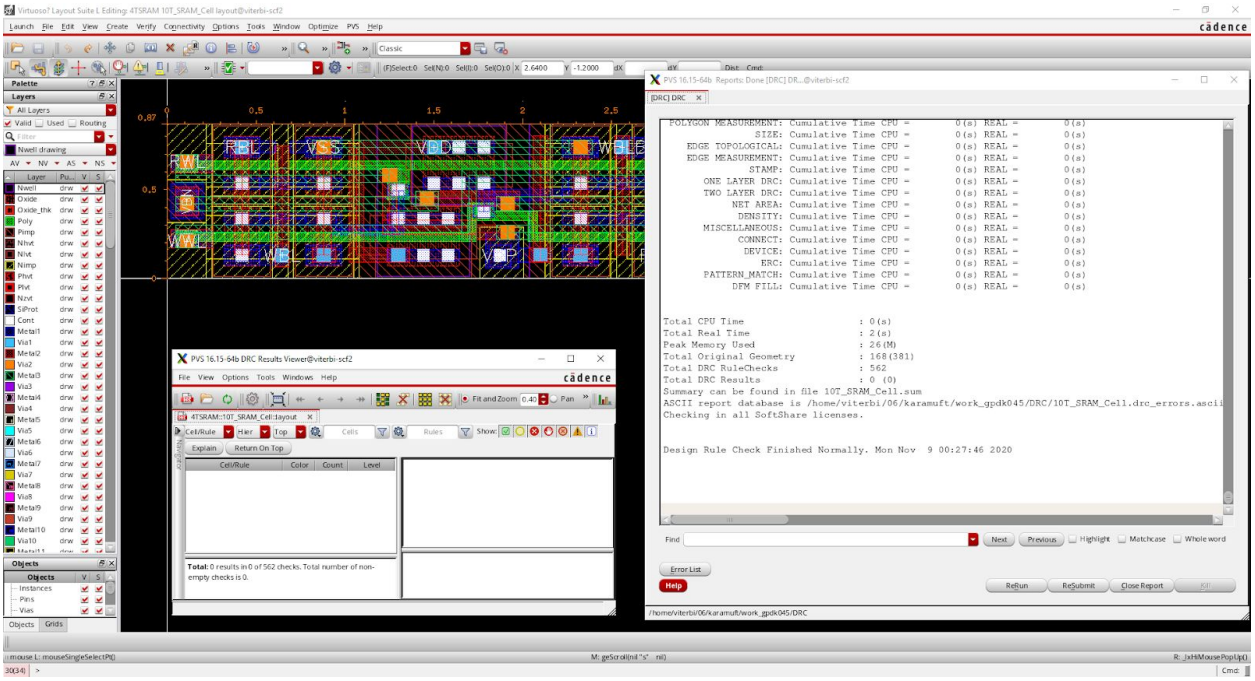

Figure 7: 10T-SRAM single cell body bias DRC

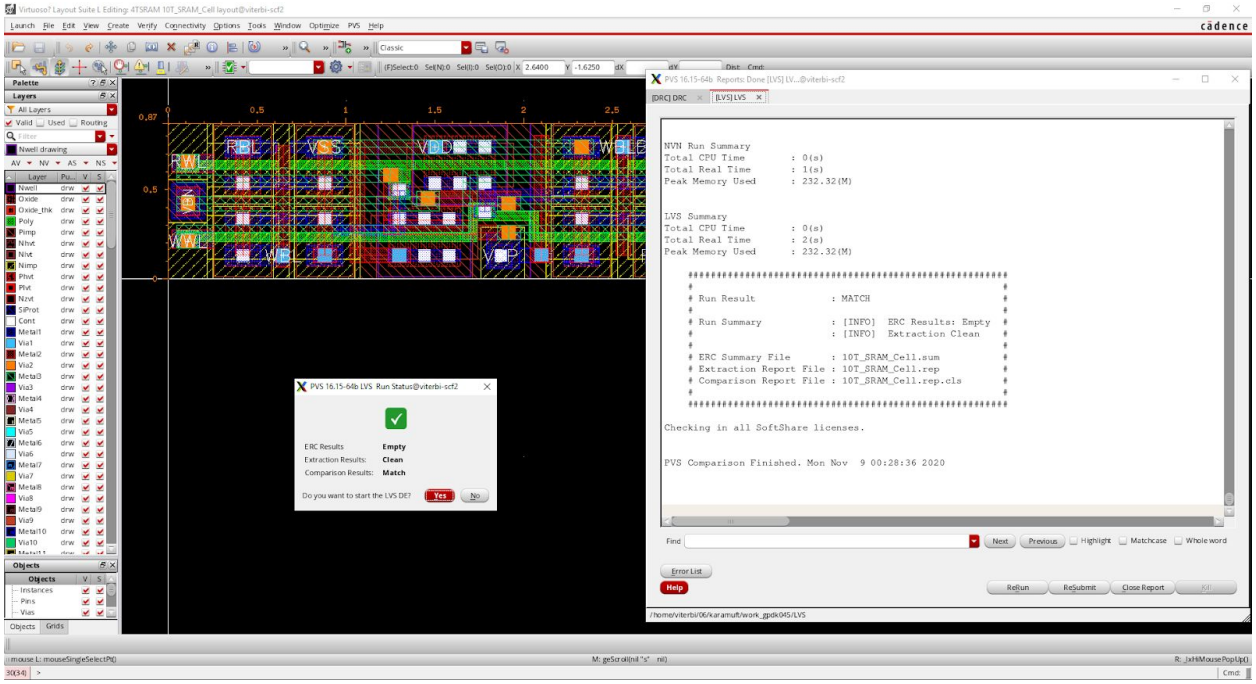

Figure 8: 10T-SRAM single cell body bias LVS

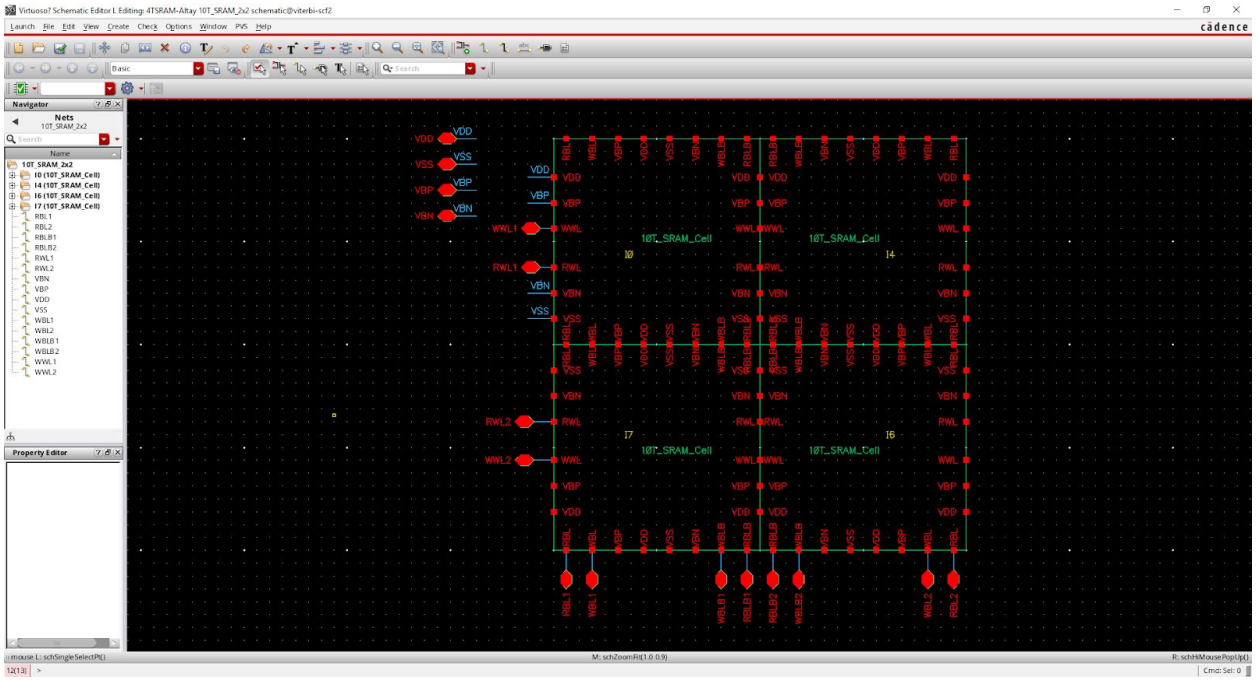

Figure 9: 2x2 SRAM schematic

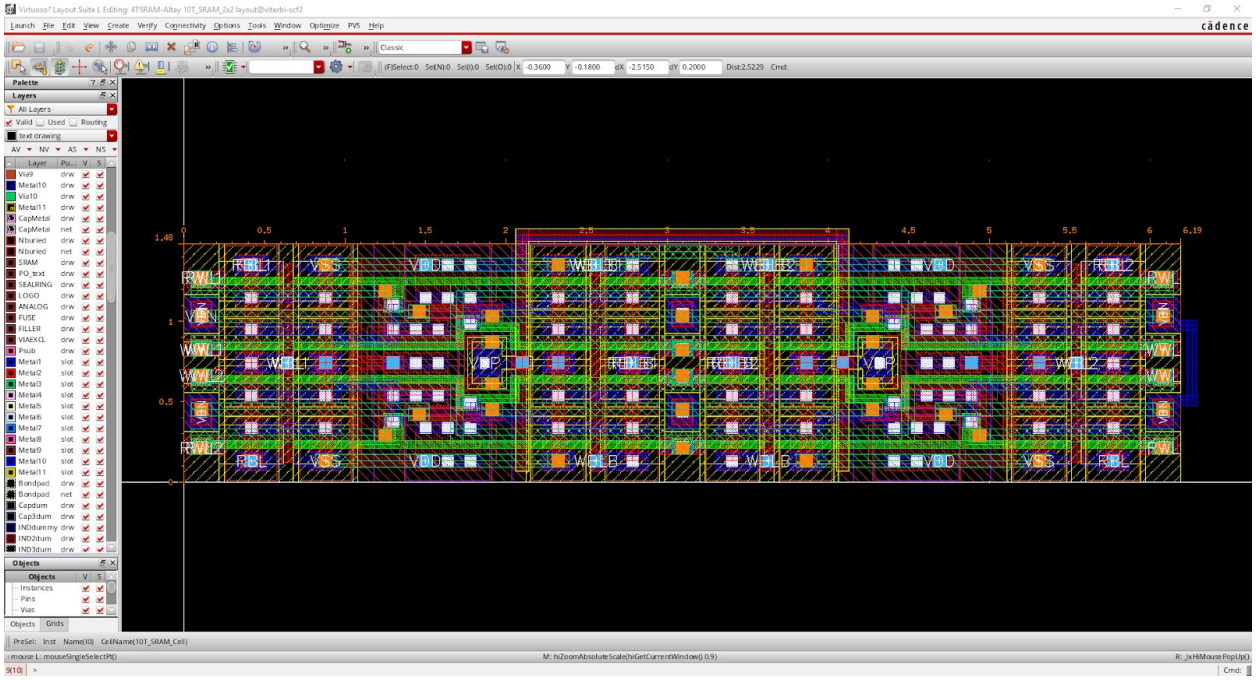

Figure 10: 2x2 SRAM layout

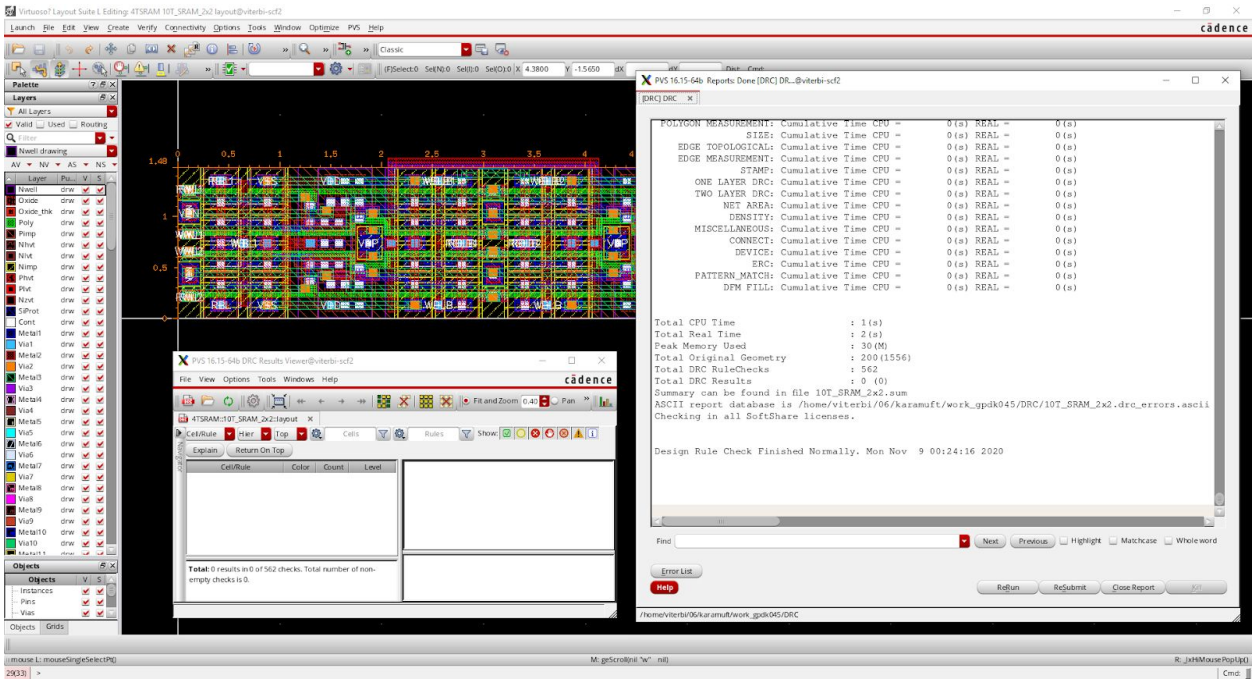

Figure 11: 2x2 SRAM DRC

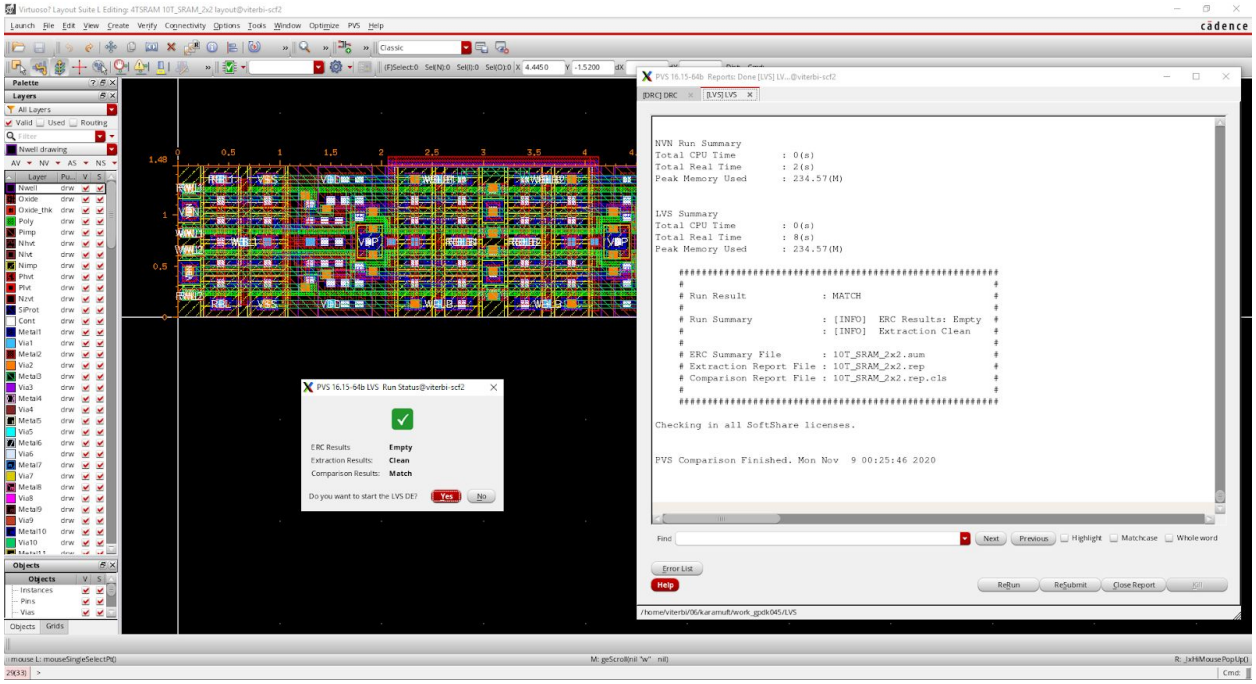

Figure 12: 2x2 SRAM LVS

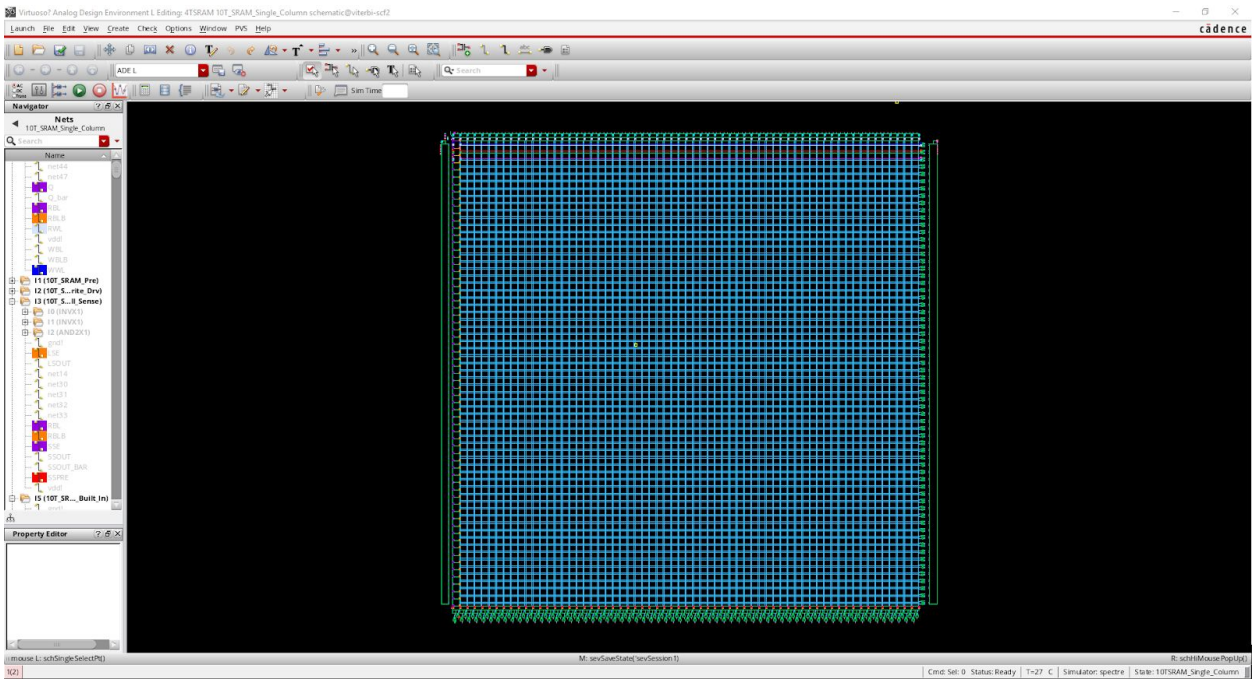

Figure 13: Single column schematic overview

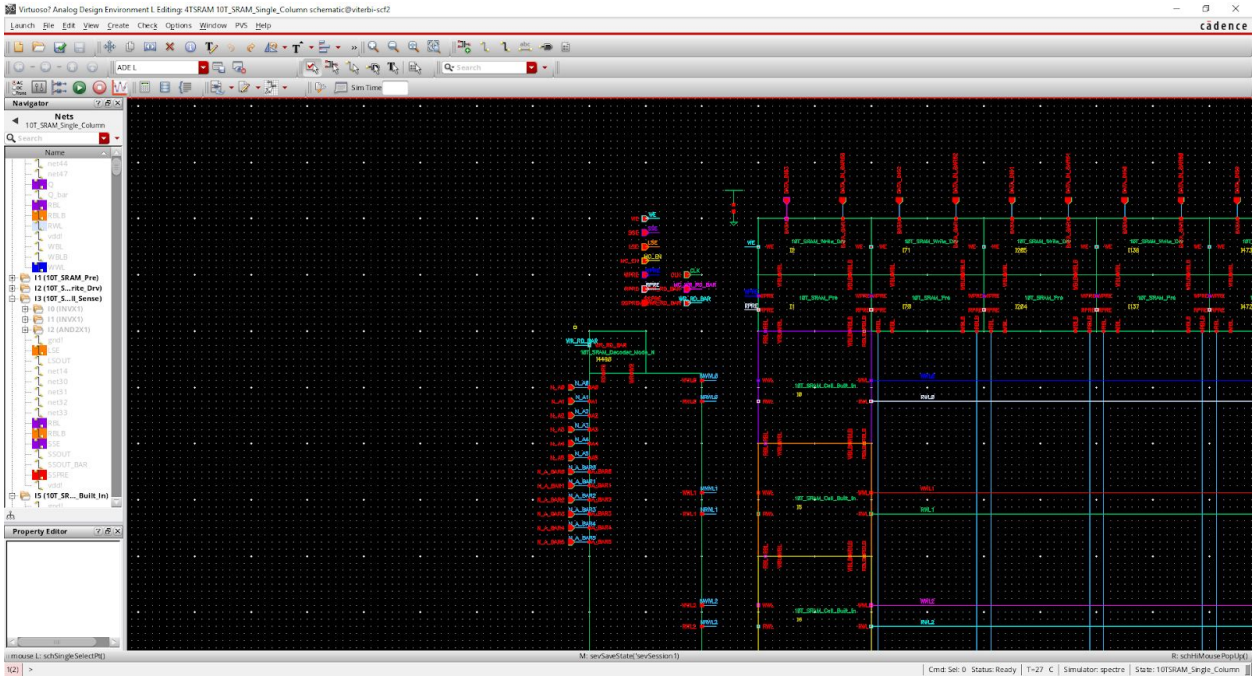

Figure 14: Single column schematic zoomed-1

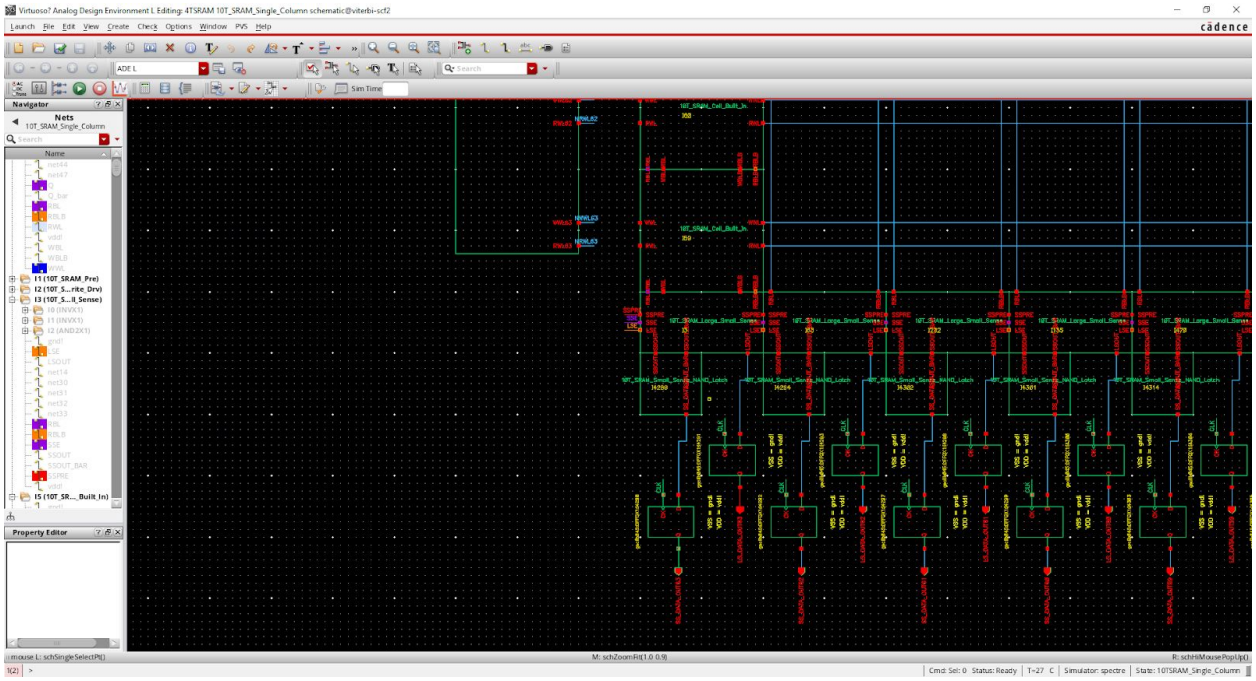

Figure 15: Single column schematic zoomed-2

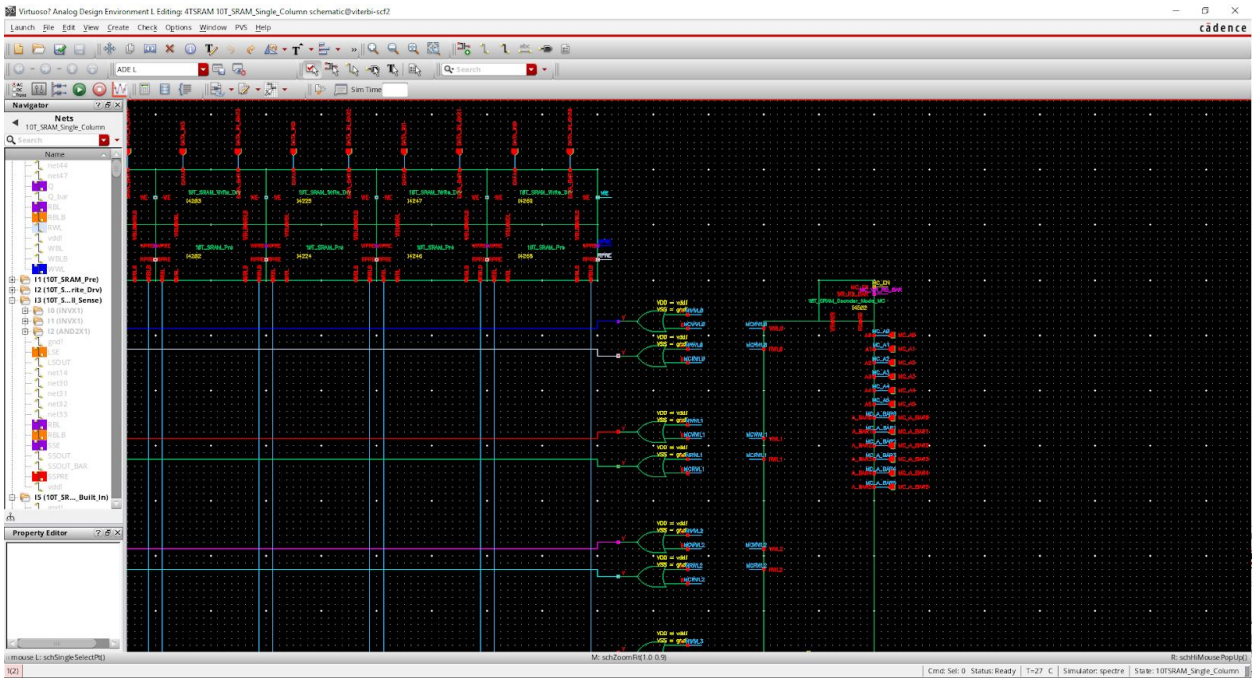

Figure 16: Single column schematic zoomed-3

Simulation

In the initial part, we tried to simulate write 1 -> read 1 -> write 0 -> read 0 -> write 1 operations in order to observe the functionality of circuits for controlling SRAM cells. Additionally, as expected, the maximum voltage drop on the Bitline and Bitline_Bar is 97mV and 52mV. These values satisfy the requirements. To achieve this operation, we enable the write driver circuit and assign the data and data_bar values into the write bitline and write bitline_bar. Since the write wordline transistor is activated, these values manipulate the internal state of cross-coupled inverters. After write operation, we enable the read transistors by making read wordline logic as 1. If the stored logic is 1, then the BL value will be pulled down to the ground. Therefore, in the NAND latch design, unlike the previous lab, we need to observe the output on the other side of NAND outputs.

For the simulation of a single column circuit, we accessed the WL-0, WL-1 and WL-2. After writing 1, 0 and 1 logic values into the corresponding SRAM cells, we tried to read with the same order. Additionally, we performed the in-memory computing by activating WL-0 and WL-1 in the first additional cycle and WL-0 and WL-2 in the second additional cycle. Since the logic values are 1-0 and 1-1, the output of in-memory computation (XOR logic) will be 1 and 0, respectively. Related figures are shown from Figure 17 to 21.

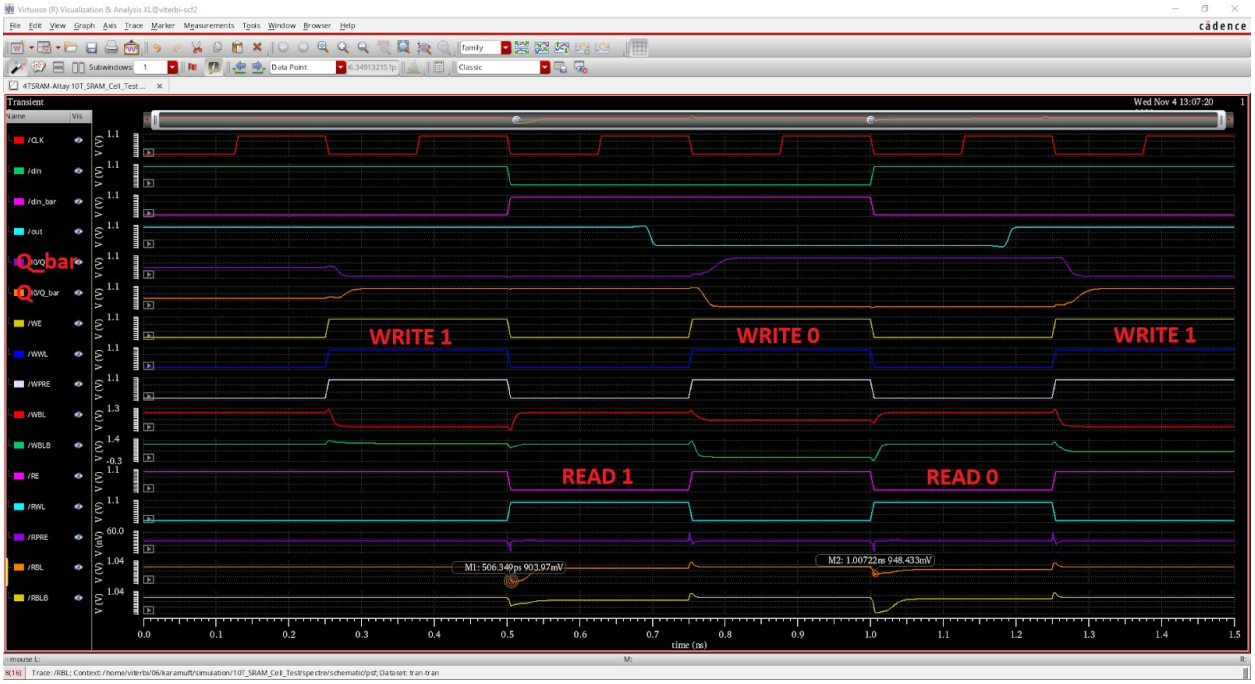

Figure 17: SRAM cell waveform

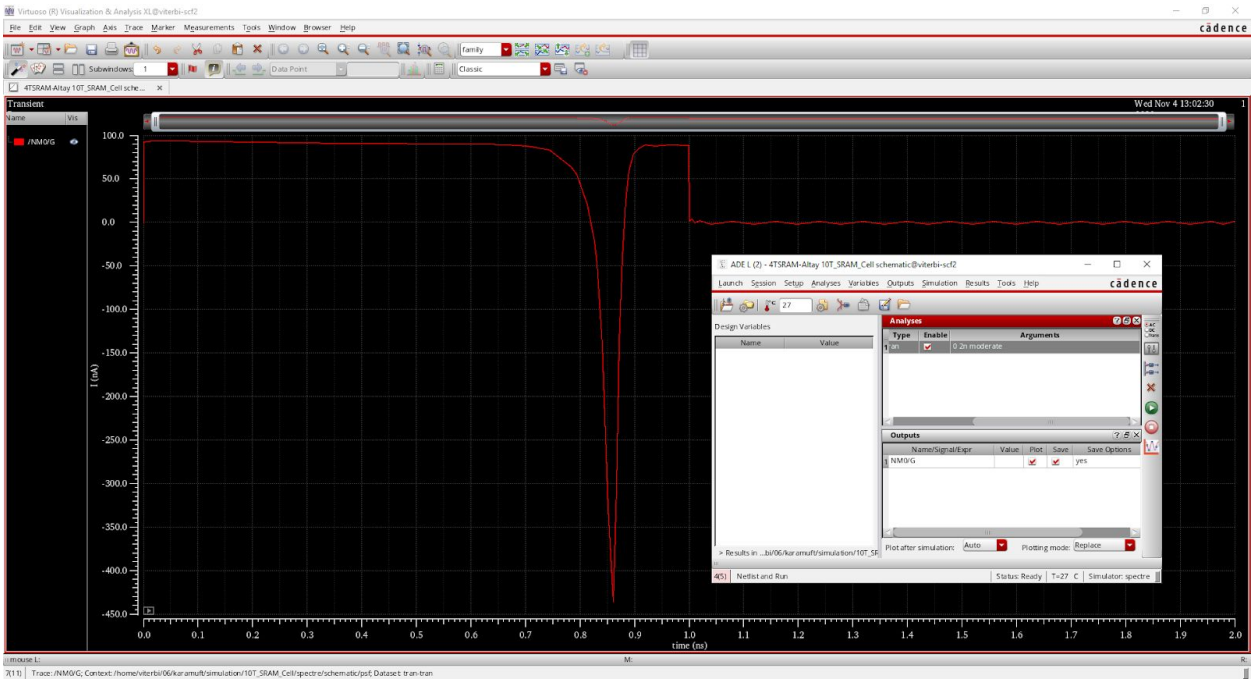

Figure 18: SRAM cell capacitance simulation

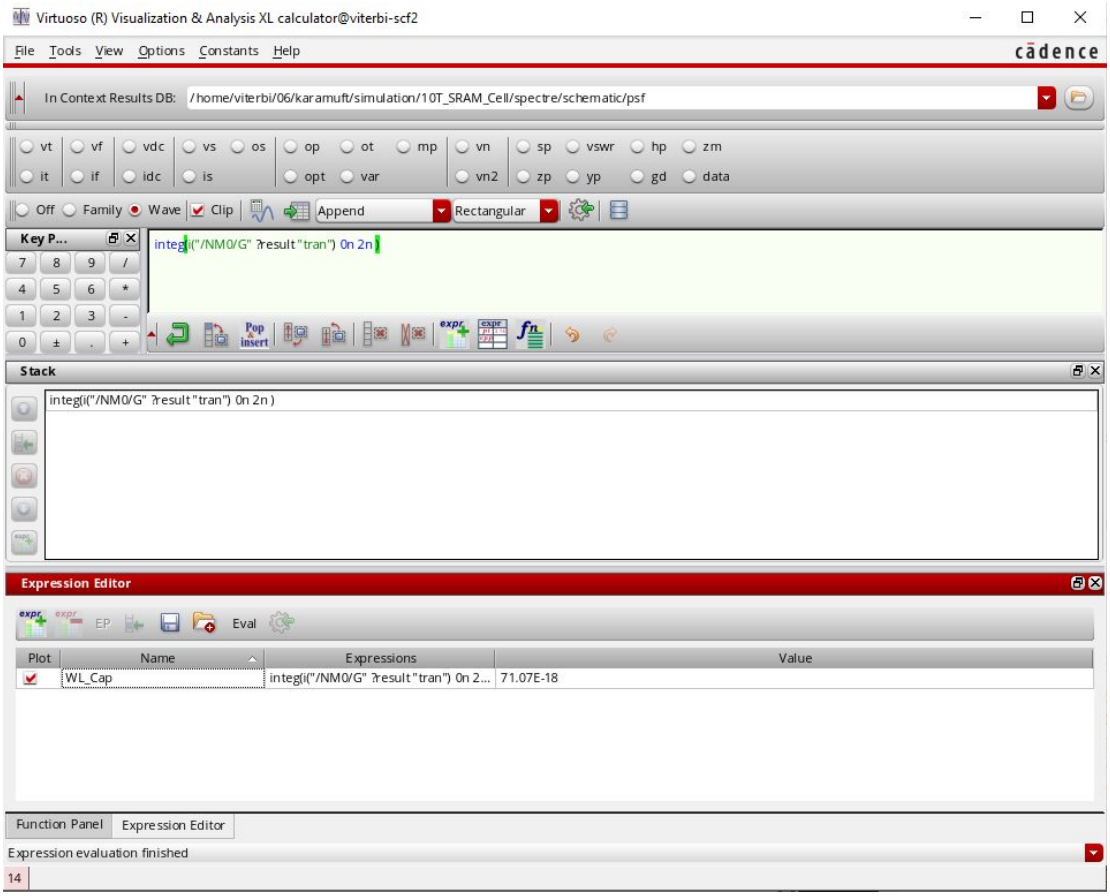

Figure 19: Word line capacitance

Figure 20: Single column waveform (w/o small sense precharge)

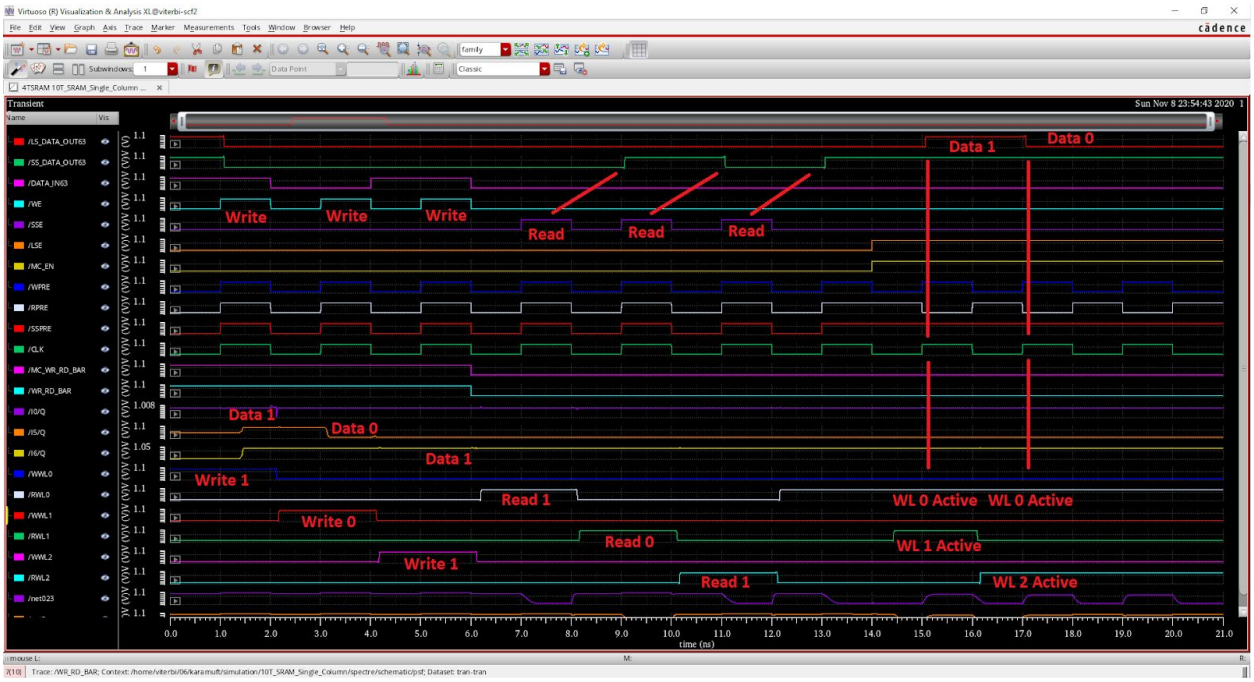

Figure 21: Single column waveform (with small sense precharge and labels)

2. Write driver circuit

Description

In order to achieve correct write operation on the SRAM columns, each column has its own write driver circuitry to pull down the write BL and write BLB. The sizes of these NMOS transistors determine the pull-down speed of the BL and BLB voltages. The related pictures are shown in Figures 22-25.

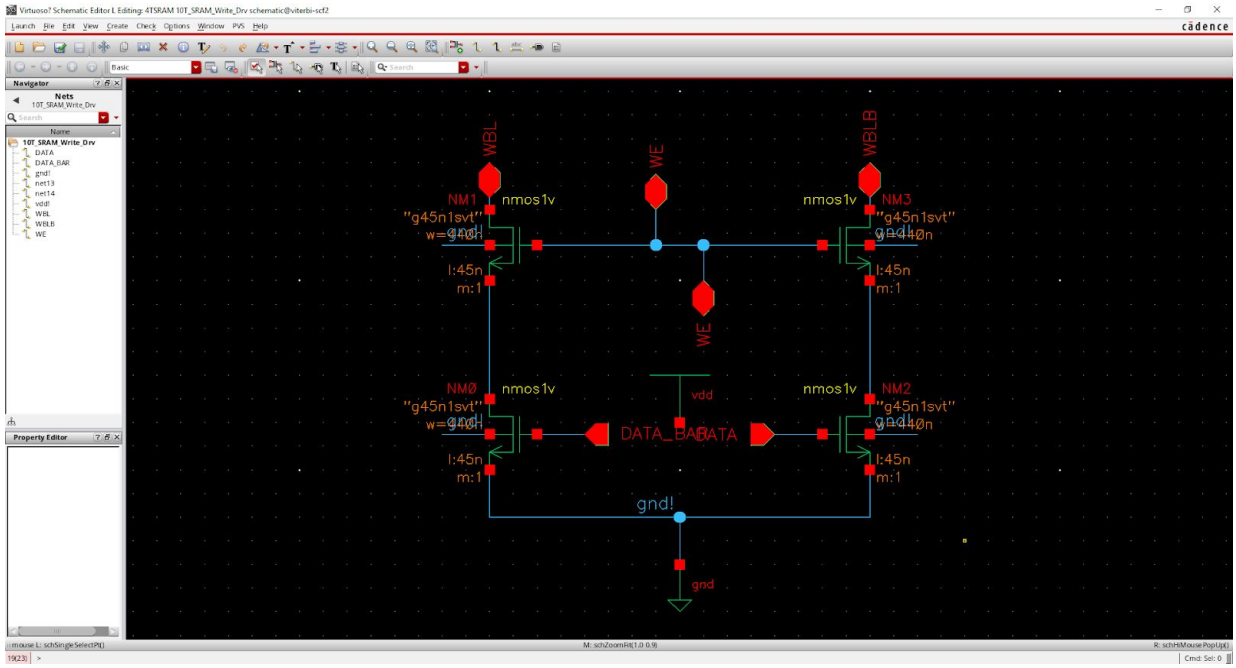

Figure 22: Single write driver cell schematic

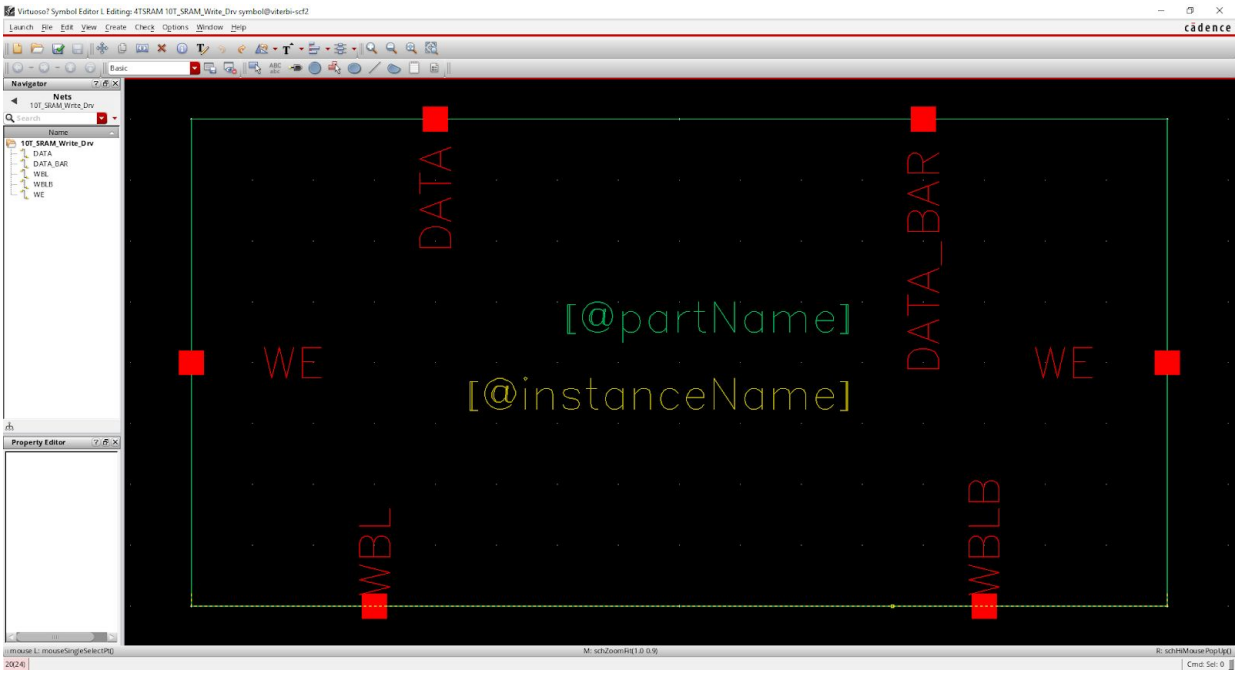

Figure 23: Single write driver cell symbol

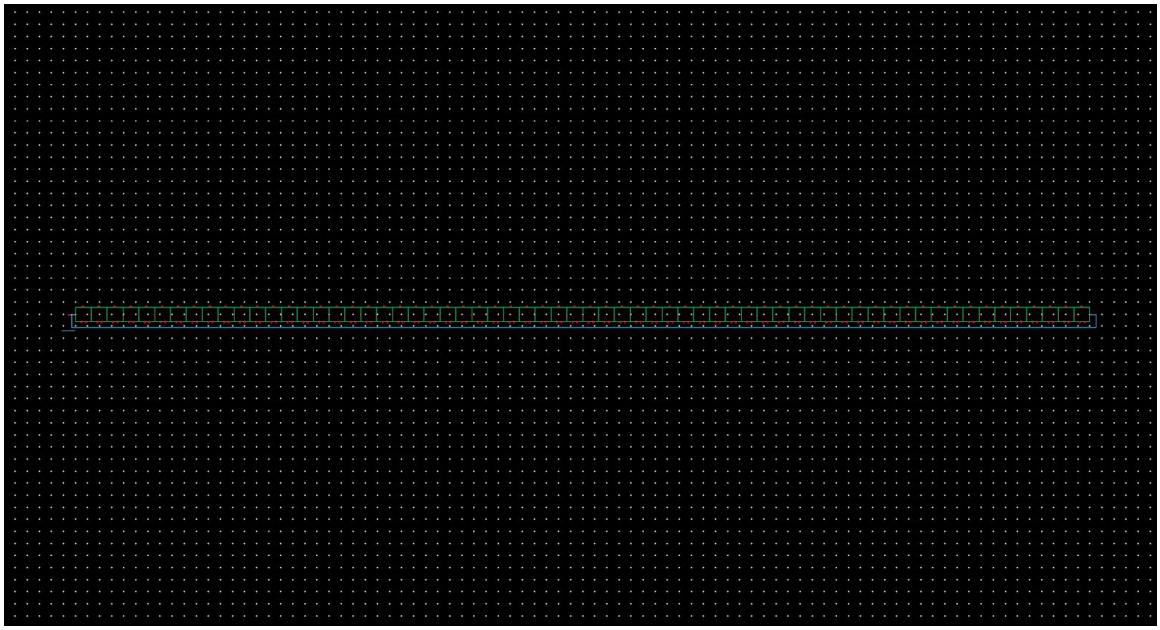

Figure 24: 64x Write driver circuit schematic

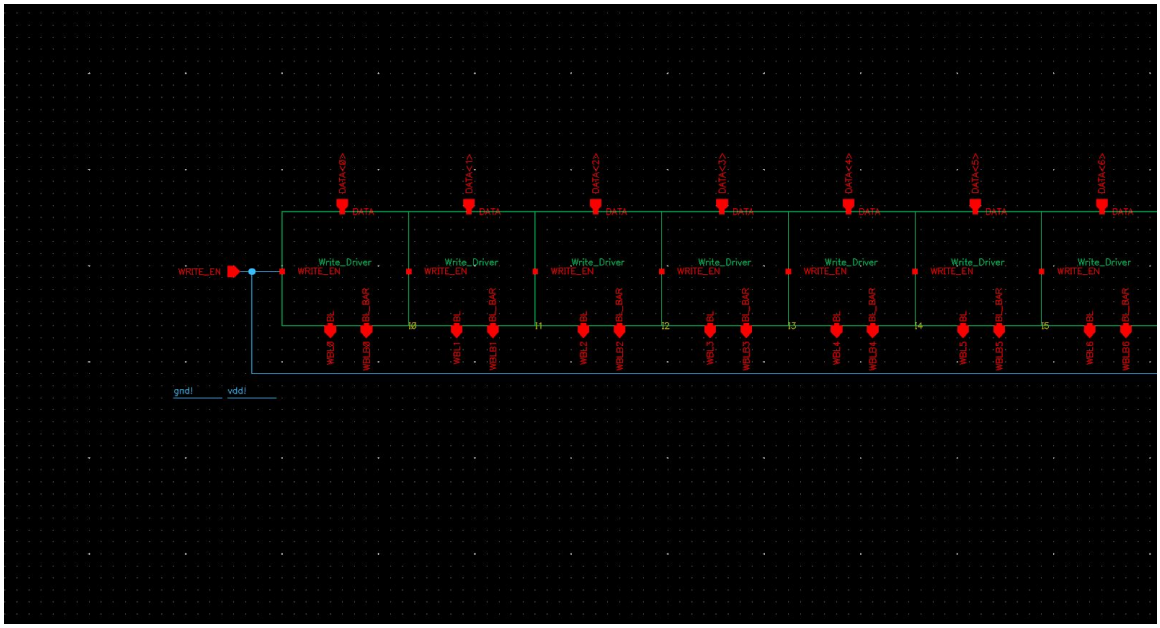

Figure 25: 64x Write driver circuit schematic zoomed version

3. Precharge Circuit

Description

There are two precharge circuits and they are used to precharge the bitline and bitline_bar of write and read operations. The sizes of these transistors are assigned as 330nm.

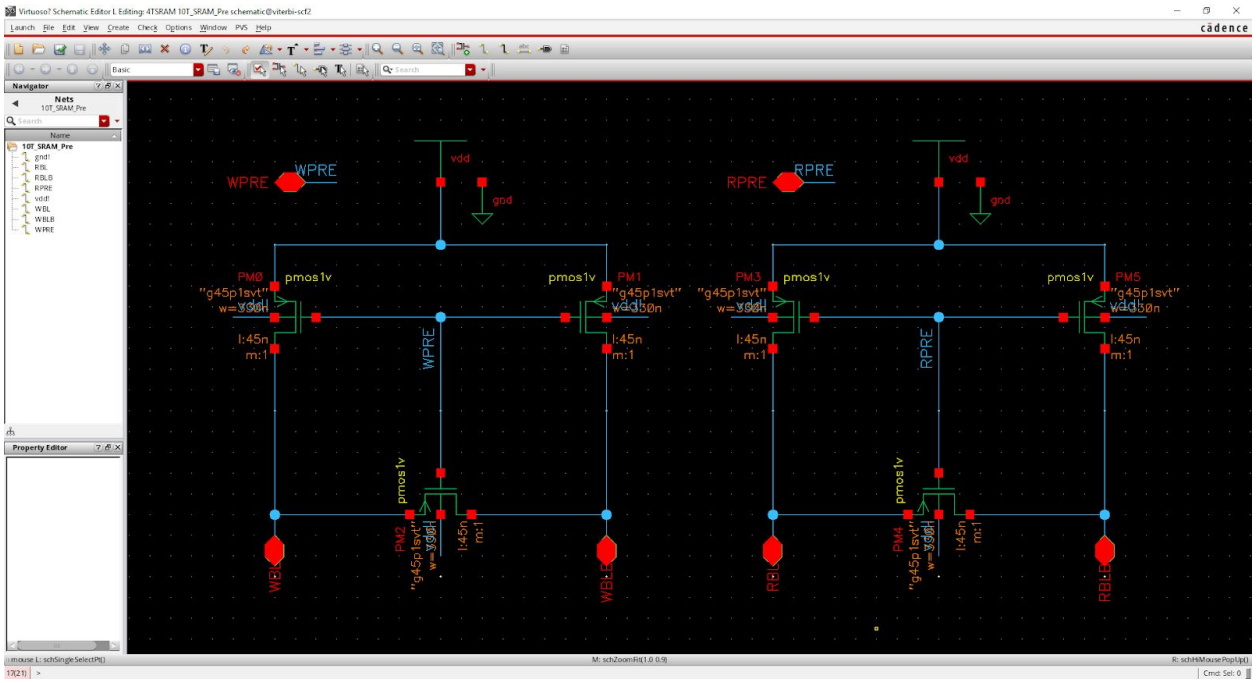

Figure 26: Single write read precharge circuit schematic

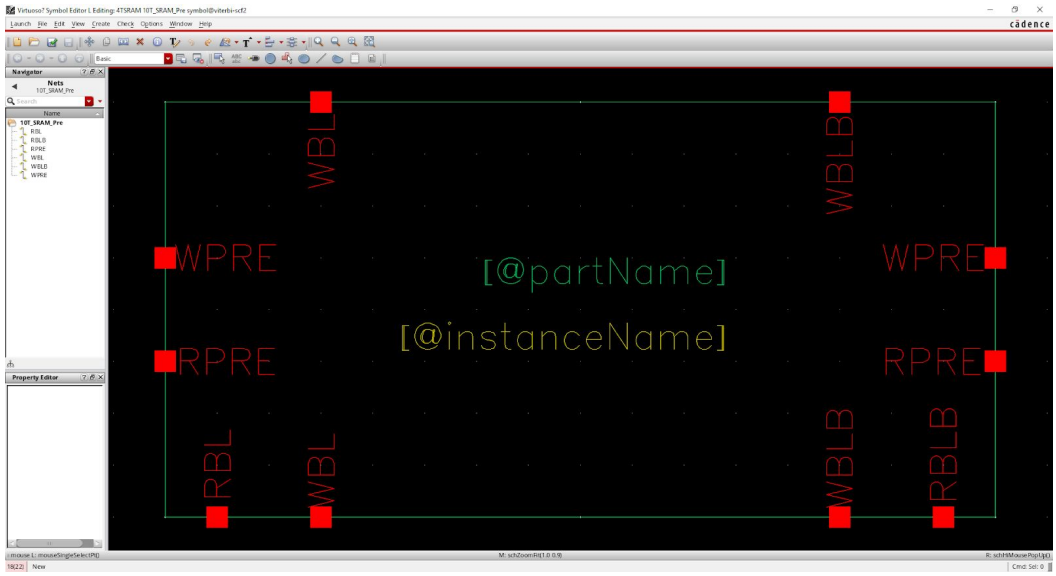

Figure 27: Single write read precharge circuit symbol

4. Large & small signal sensing circuit

Description

The large sensing circuit uses two inverters and one NAND gate to do the XOR calculation. Additionally, we used pass transistors to enable and disable the computation result. This feature is desired but for the sake of simplicity, it is not required for the project since we are allowed to separate the results of two sense amplifiers. We can simply ignore the other result while we expect from one of them.

As stated above in the SRAM part, the output of NAND structure is observed from the left hand side rather than the right hand side like we did in the previous lab. The related pictures are shown in Figures 28-31.

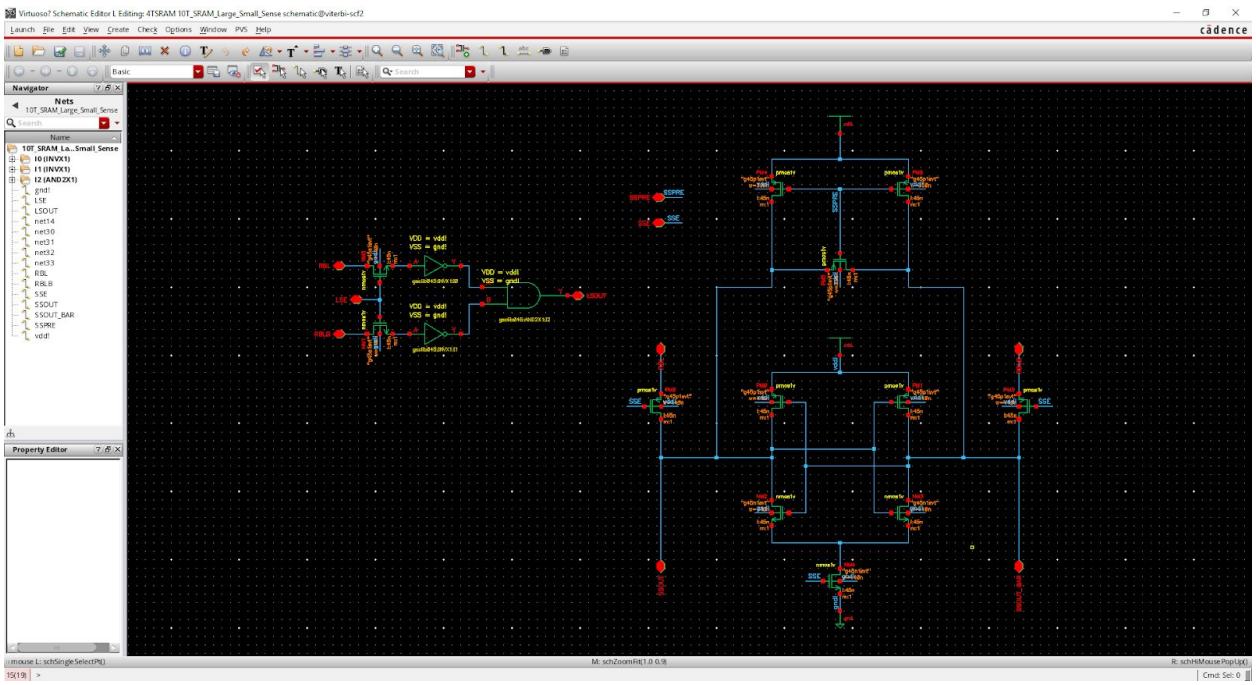

Figure 28: Large & small sense amplifier schematic

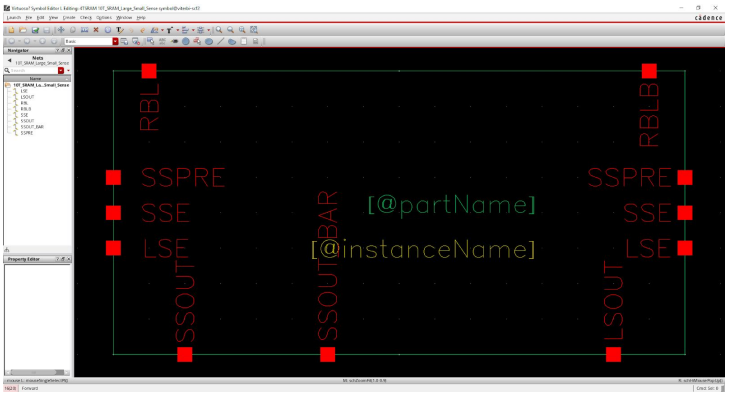

Figure 29: Large & small sense amplifier symbol

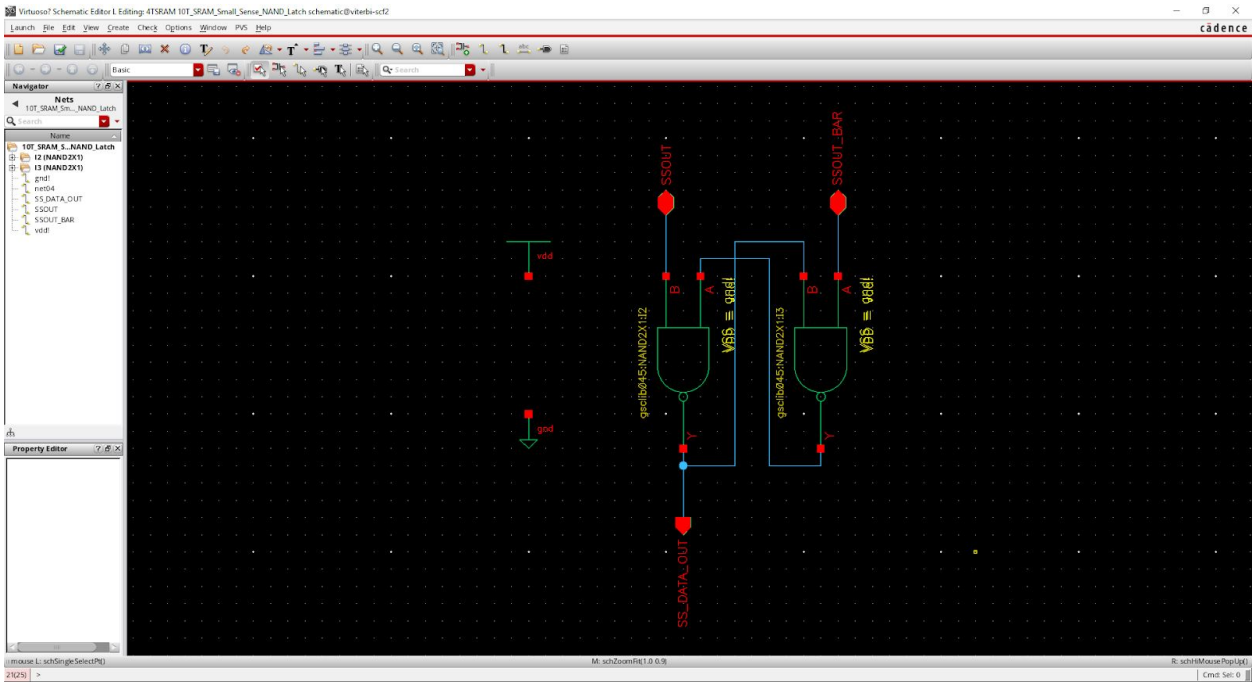

Figure 30: Small sense NAND gates latch schematic

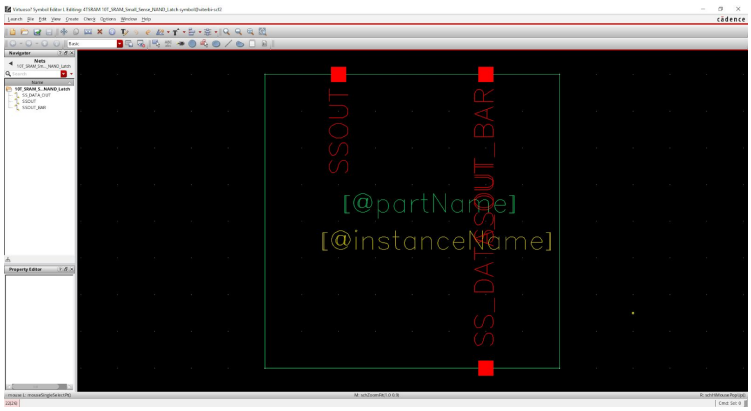

Figure 31: Small sense NAND gates latch symbol

Simulation

For the observation of small sense operation, we provide 10 sequential write operations and 10 sequential read operations. The initial start of the write operation digit and reading bits are depending on the USC ID. For our team, we have used one of the IDs (1066806391). This means the writing order is 1-2-3-4-5-6-7-8-9-0 and the read order is 1-0-6-6-8-0-6-3-9-1. Due to the nature of DFF, the correct output values will be observed after one clock cycle. We have confirmed that the small sense operation for 64x64 SRAM is correct which can be confirmed from the figures below.

Figure 32: Small sense amplifier data in (1)

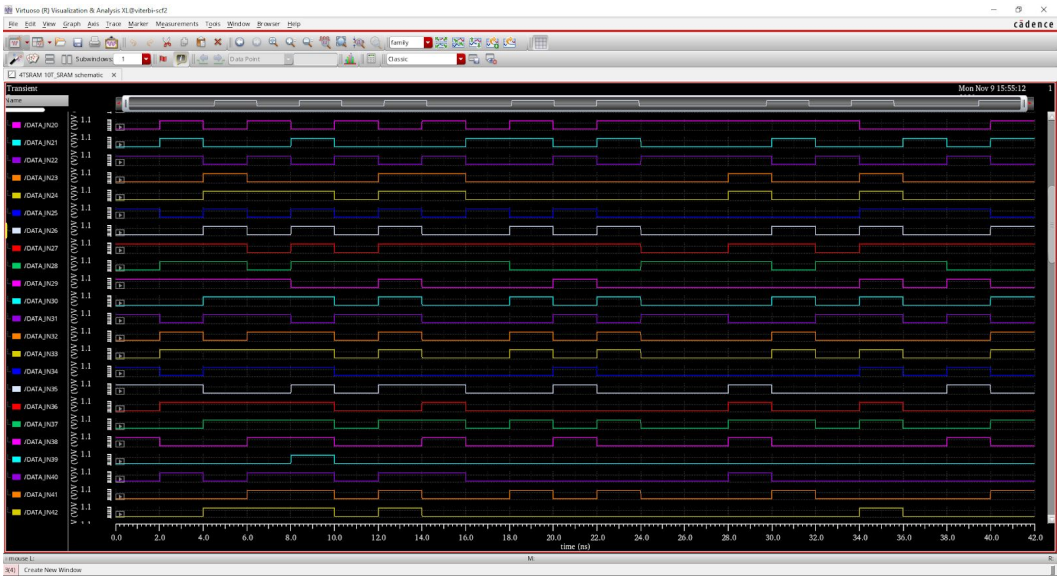

Figure 33: Small sense amplifier data in (2)

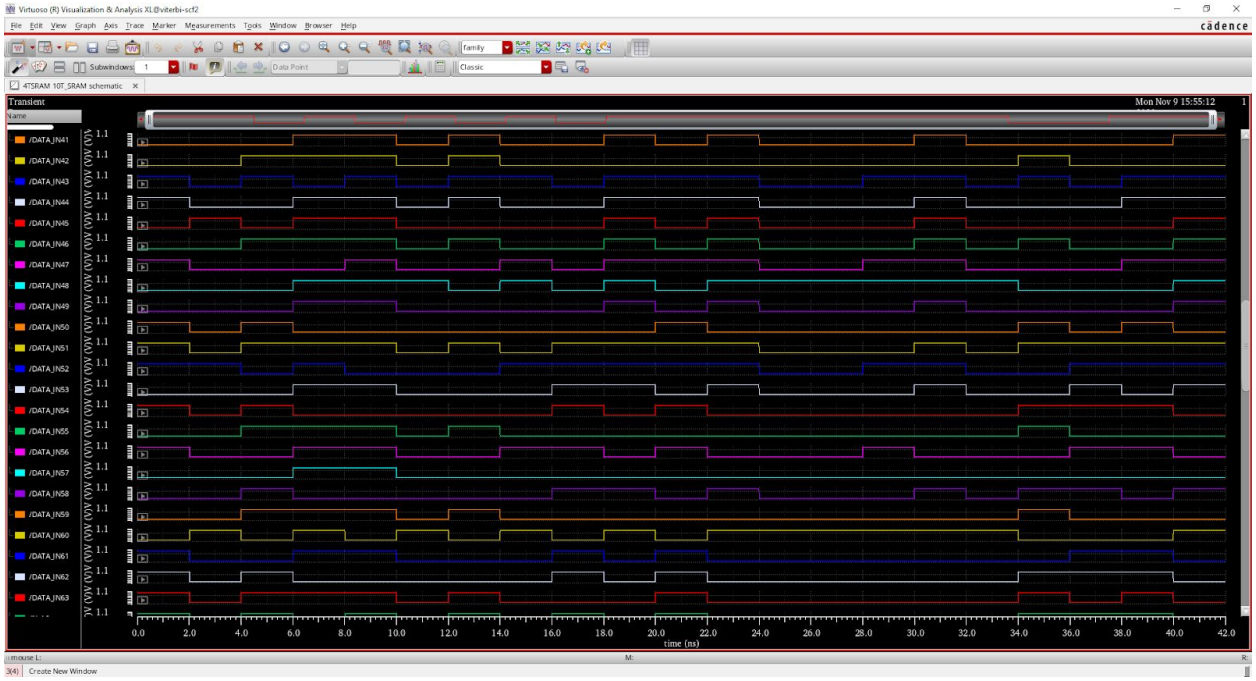

Figure 34: Small sense amplifier data in (3)

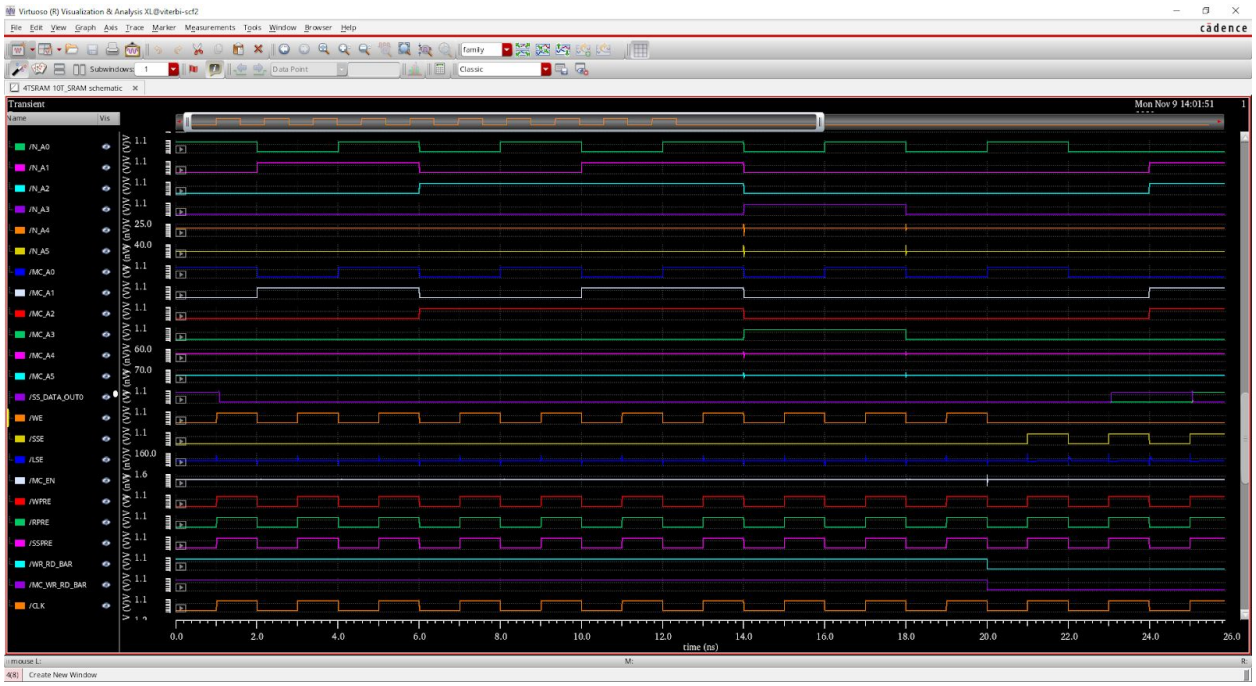

Figure 35: Small sense control waveform

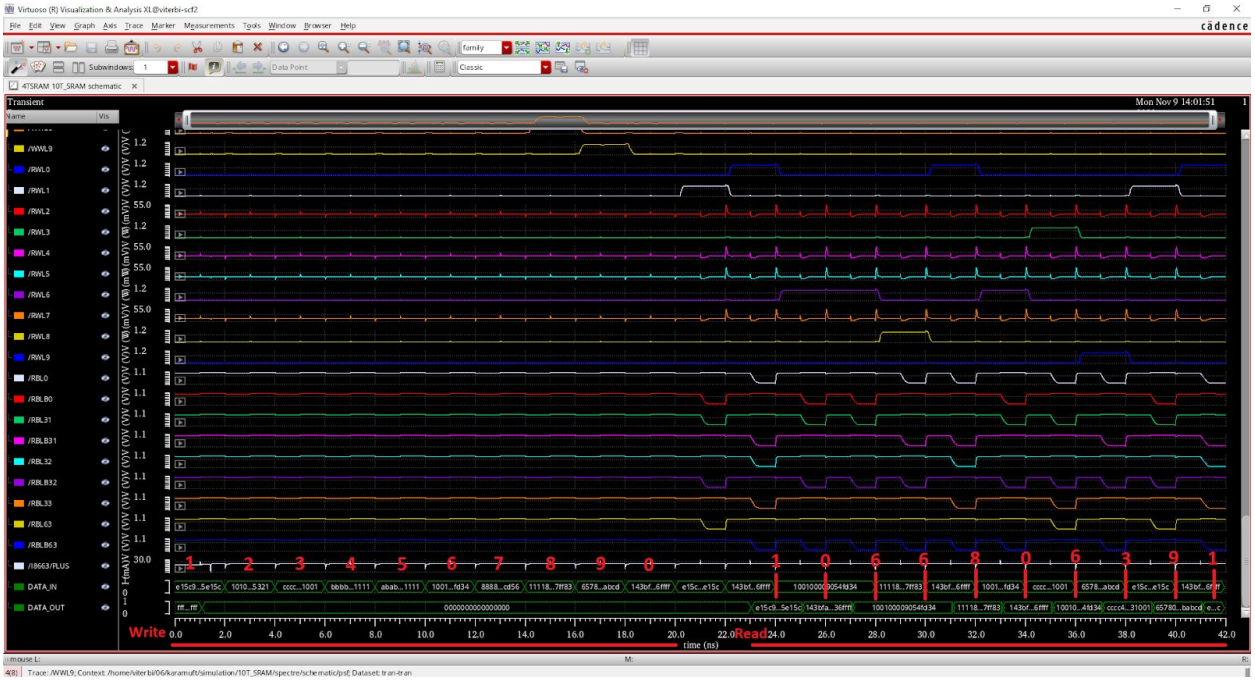

Figure 36: Small sense input and output waveform

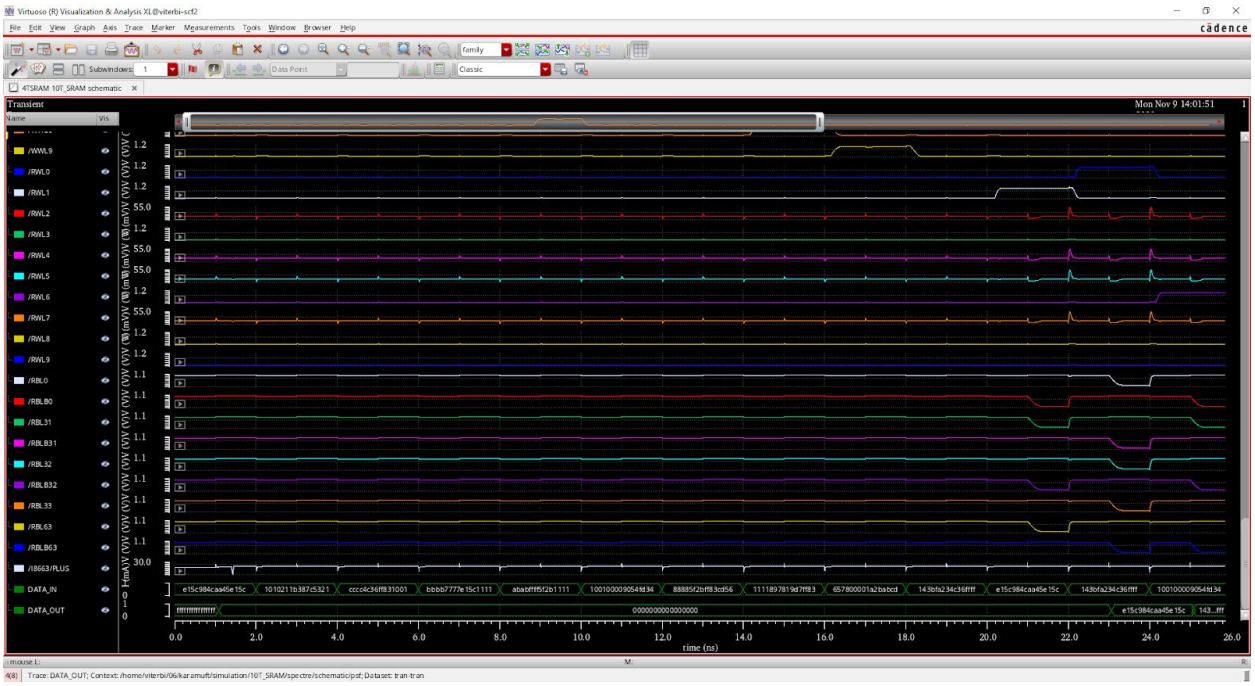

Figure 37: Small sense input zoomed waveform

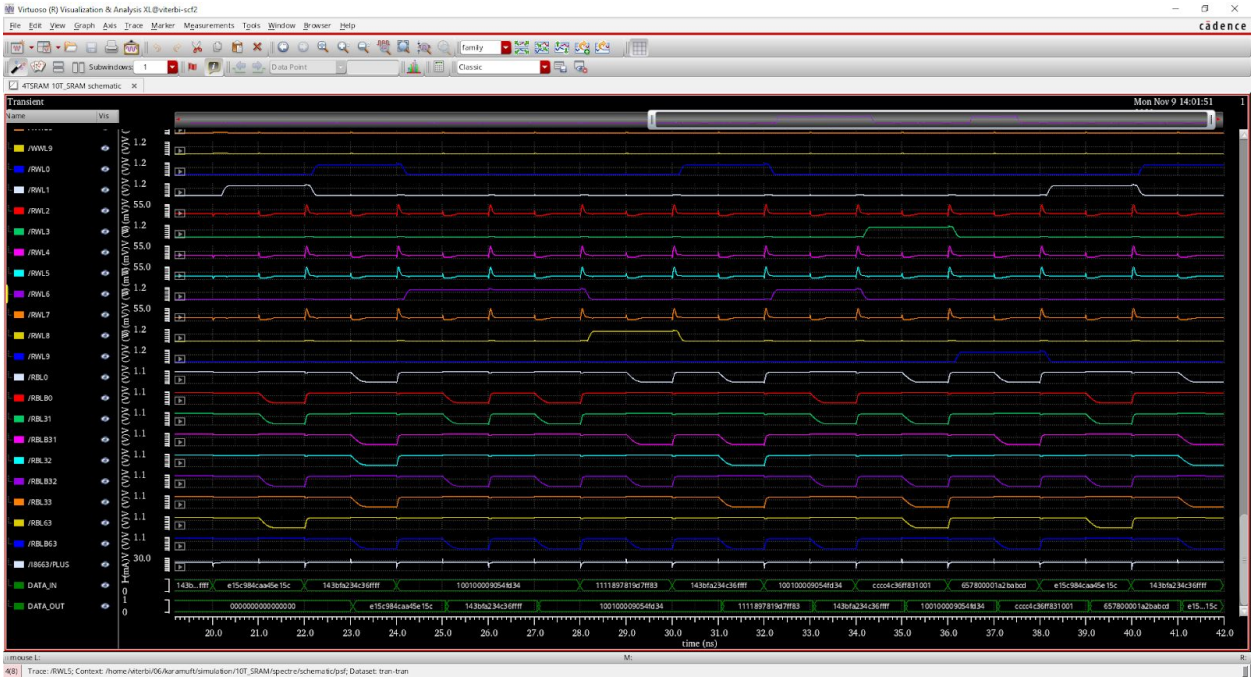

Figure 38: Small sense output zoomed waveform

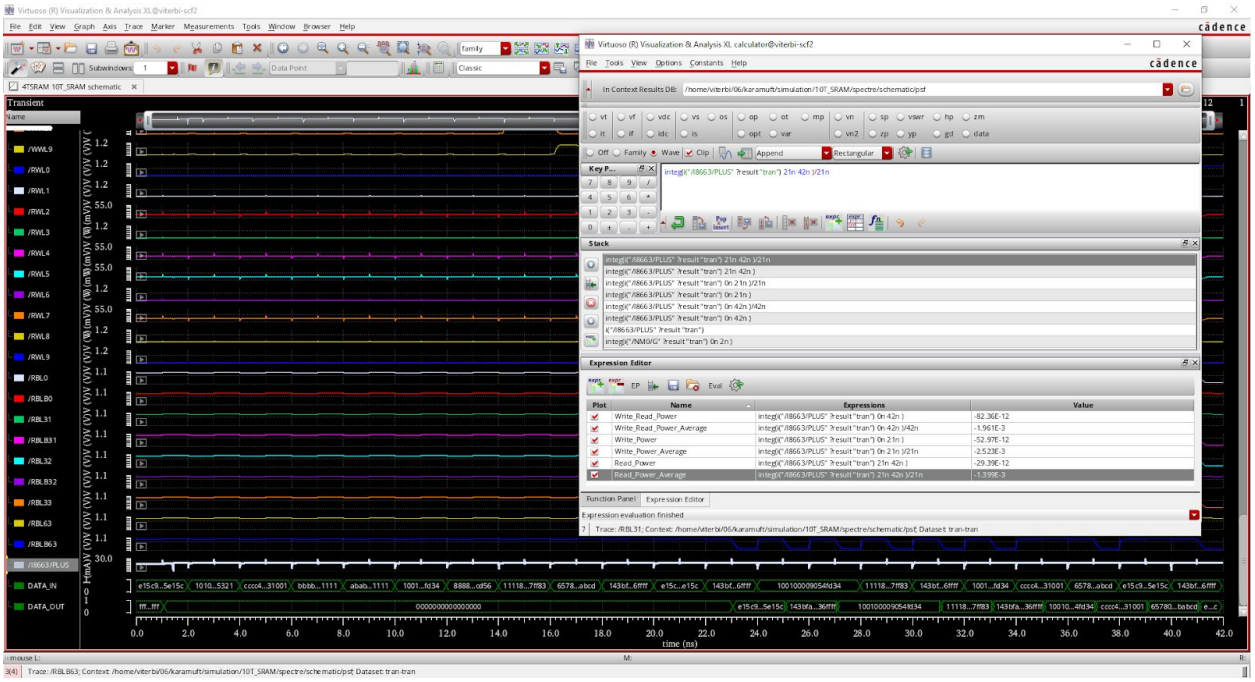

Figure 39: Small sense power calculation

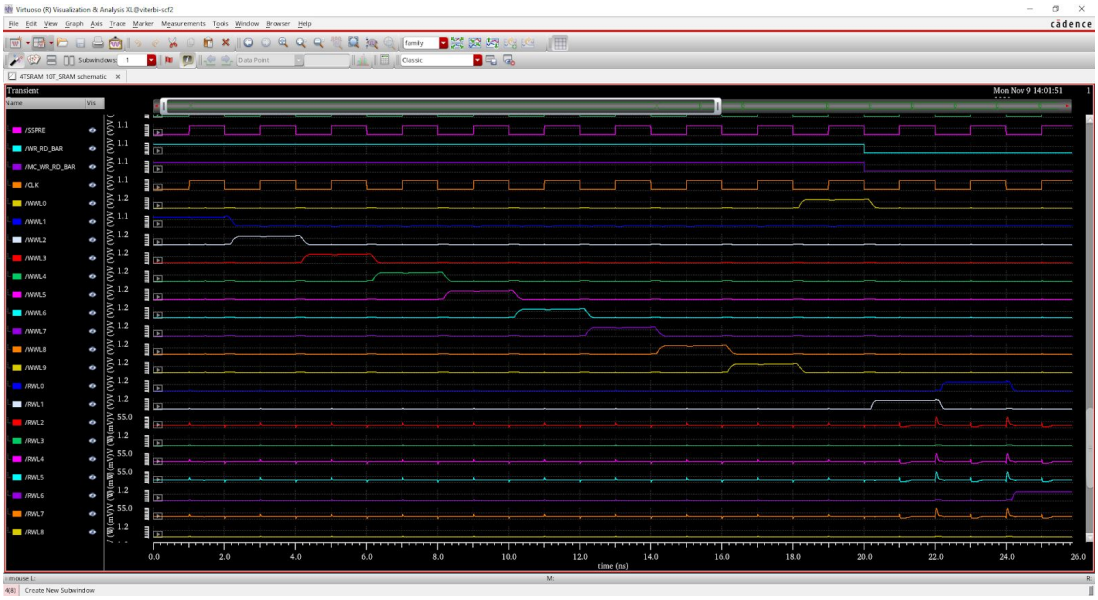

Figure 40: Small sense wordline waveform

For the observation of large sense operation, we provide 4 sequential write operations and 3 sequential in-memory computations. The write operations are required for the initialization of the SRAM cells. We selected the rows 1, 2, 3 and 4 for the writing addresses. For the in-memory computation, we proceed to do XOR logic with 3 pairs (Row 1-2, Row 2-3, Row 3-4). The correct operations are confirmed and the results can be observed within the figures below. For the comparison, please see the Figures 43 and 79 which correspond to the golden result. In Figure 43, the in-memory compute delay is expected to have a similar timing as in the small sense which is approximately 63ps.

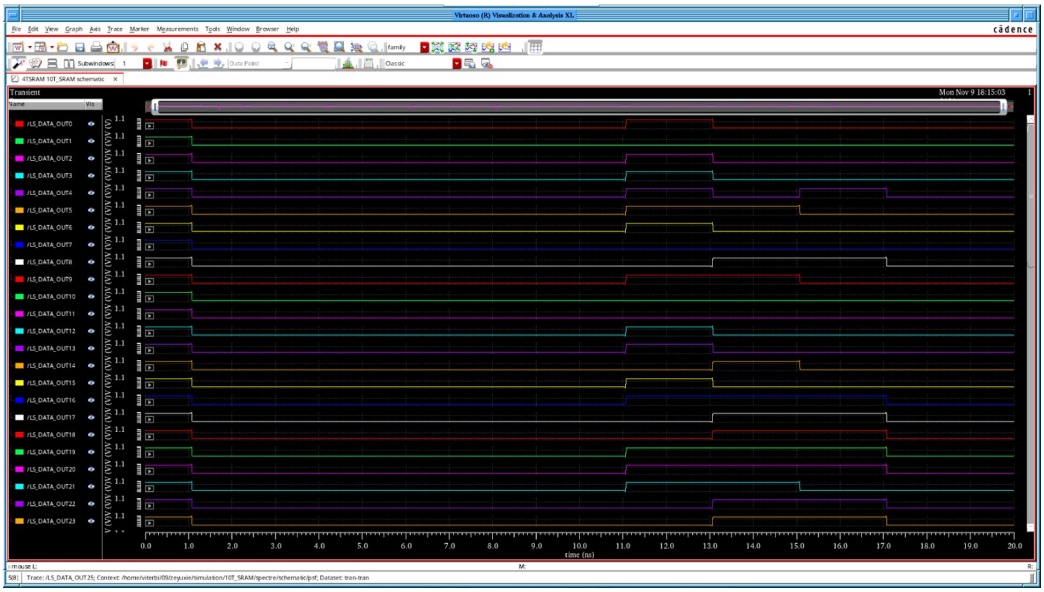

Figure 41: Large sense amplifier data out (1)

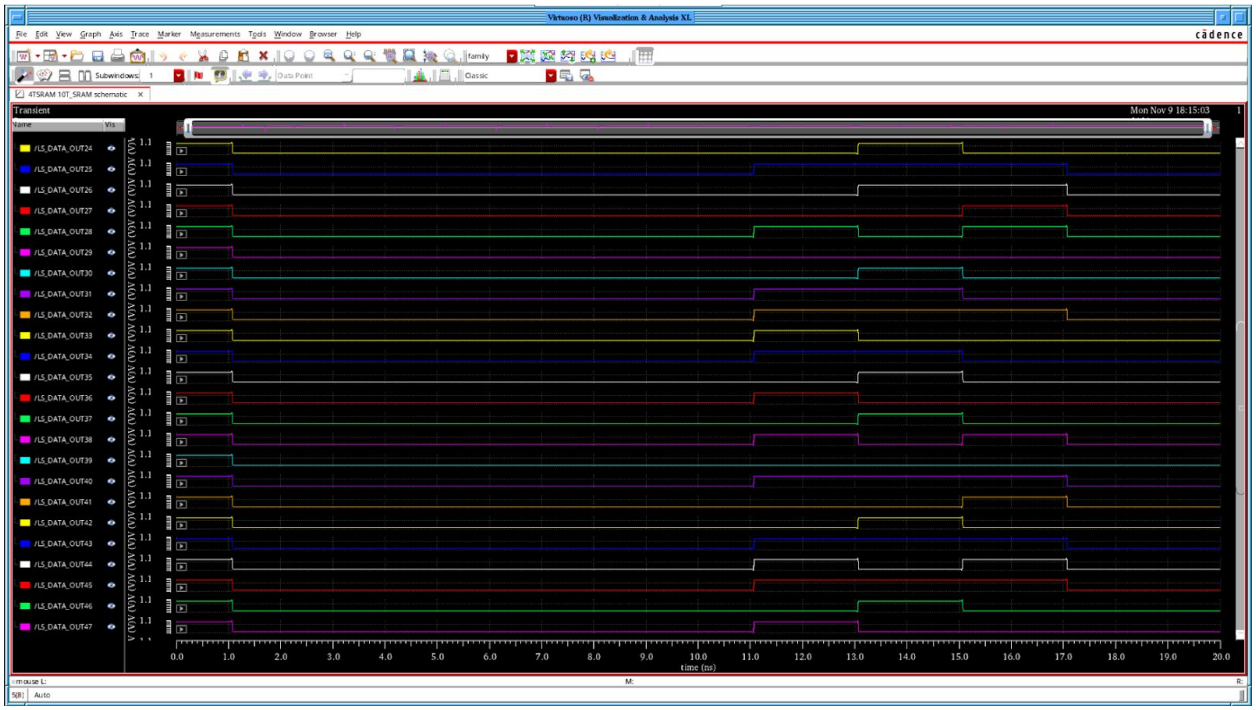

Figure 42: Large sense amplifier data out (2)

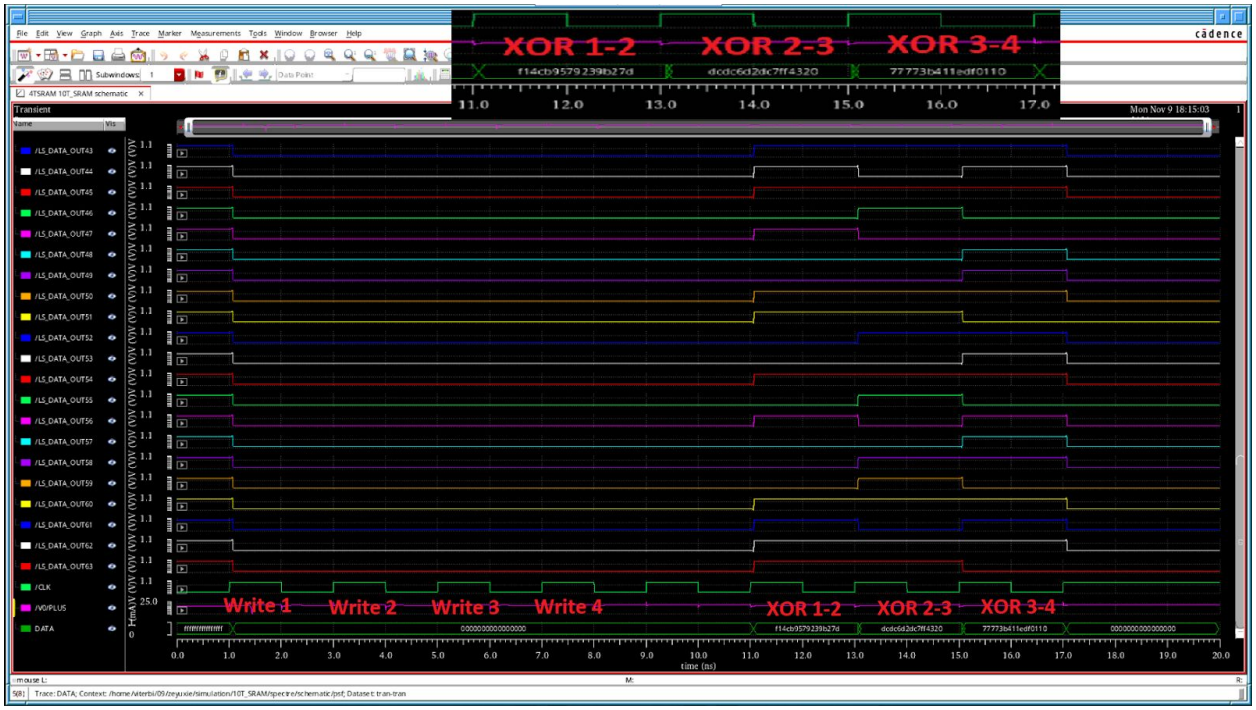

Figure 43: Large sense amplifier data out (3)

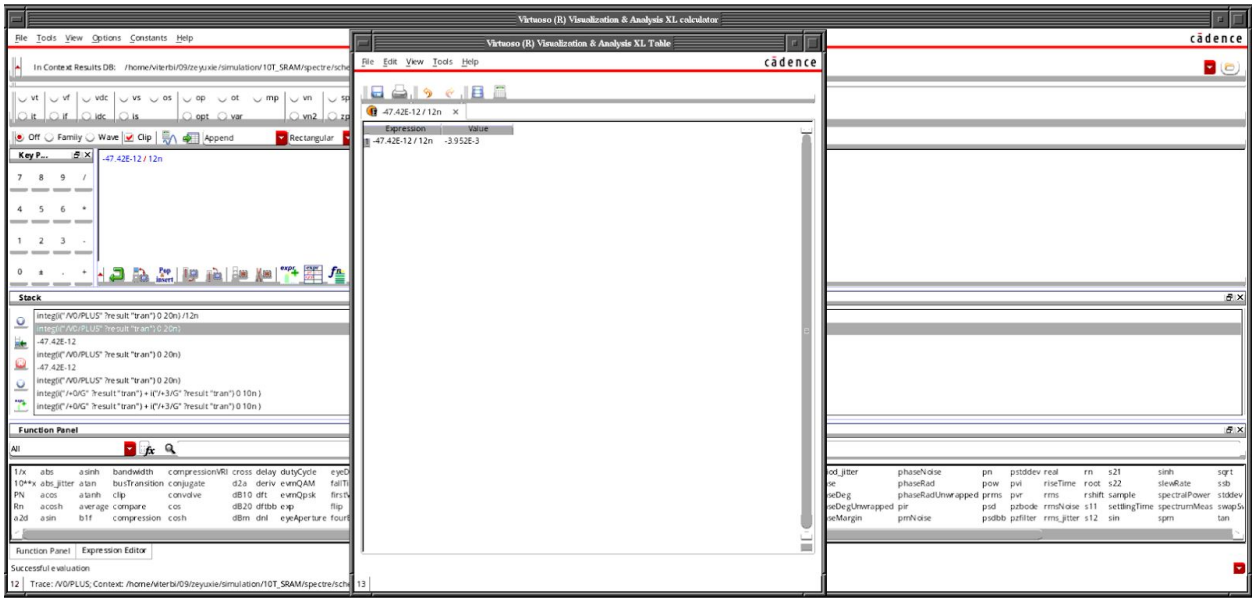

Figure 44: Large Sense Average power calculation

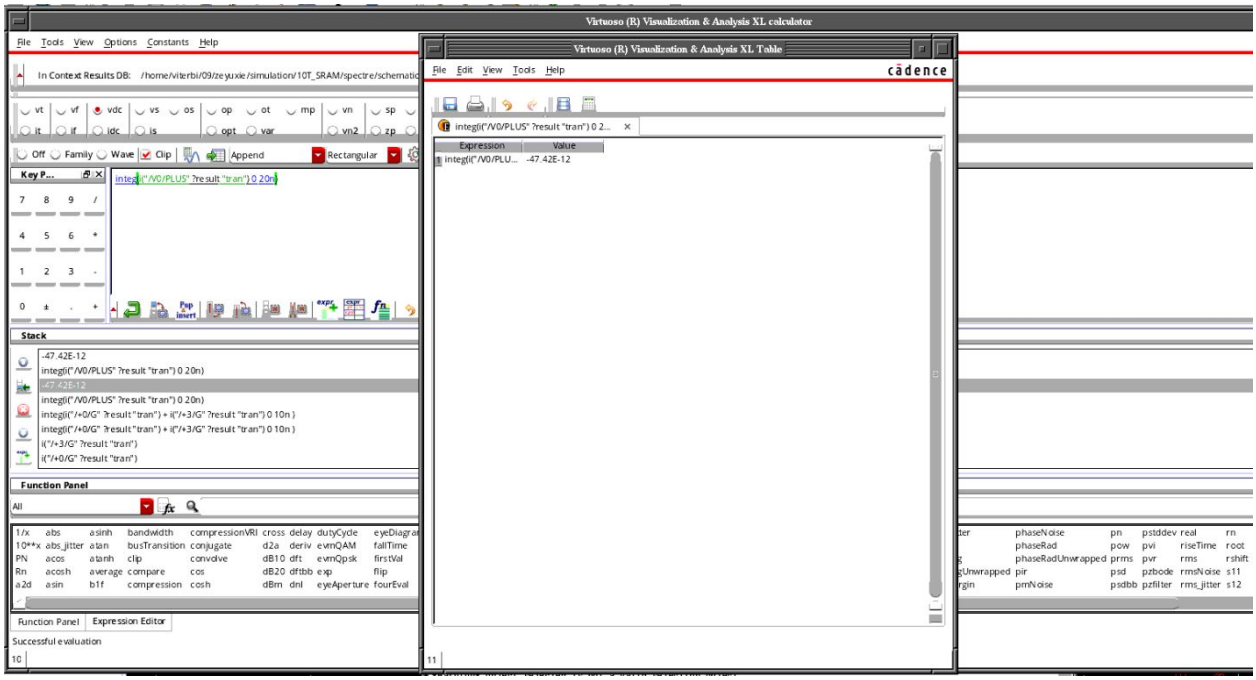

Figure 45: Large Sense Power calculation

5. Row Decoder

Description

After calculating the capacitance of sram, which is around 50E-18m, and doing logic effort, the best stage is 6 or 7, the decoder delay decreased from 376ps to around 200ps. For the modes of normal and memory computing, we designed small circuits for mode decision. Memory mode circuit has two NAND gates and one inverter to achieve only wordline selection during the memory computing. The normal mode circuit has only two buffers and one inverter which makes it relatively simple. This decoder controller is always active since the decoder on the left hand side will be activated in both normal and in-memory computation.

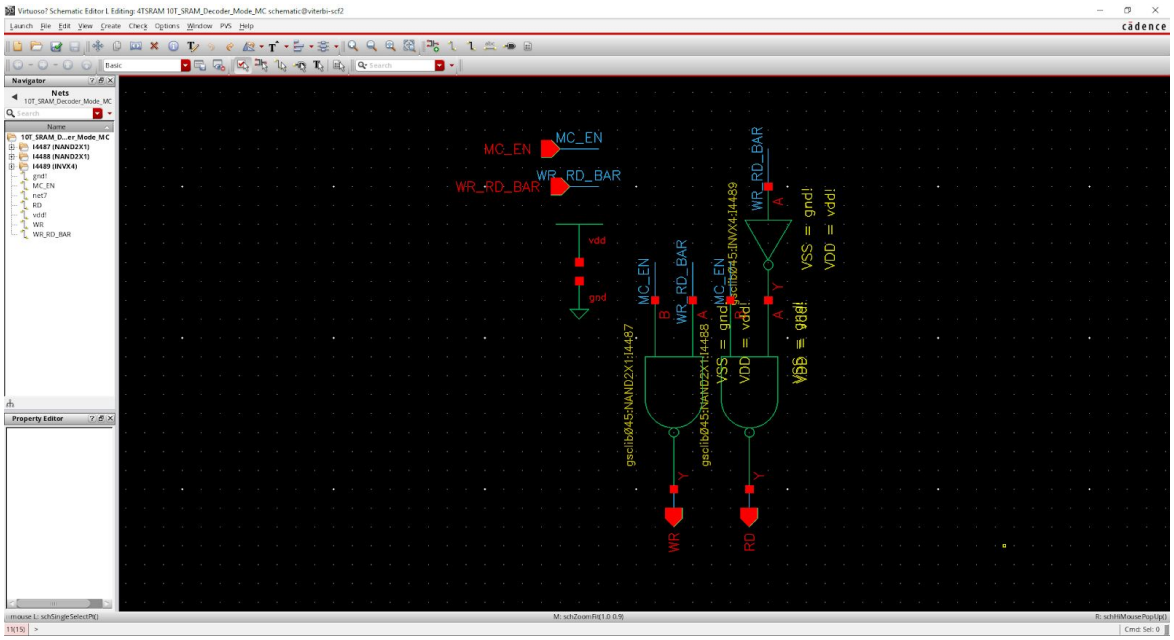

Figure 46: Decoder mode: Memory schematic

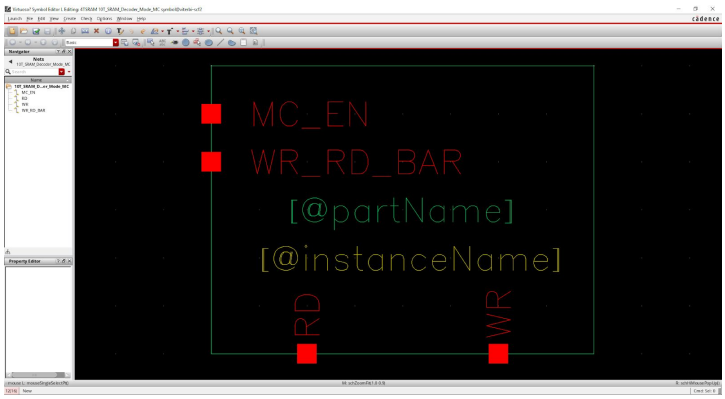

Figure 47: Decoder mode: Memory symbol

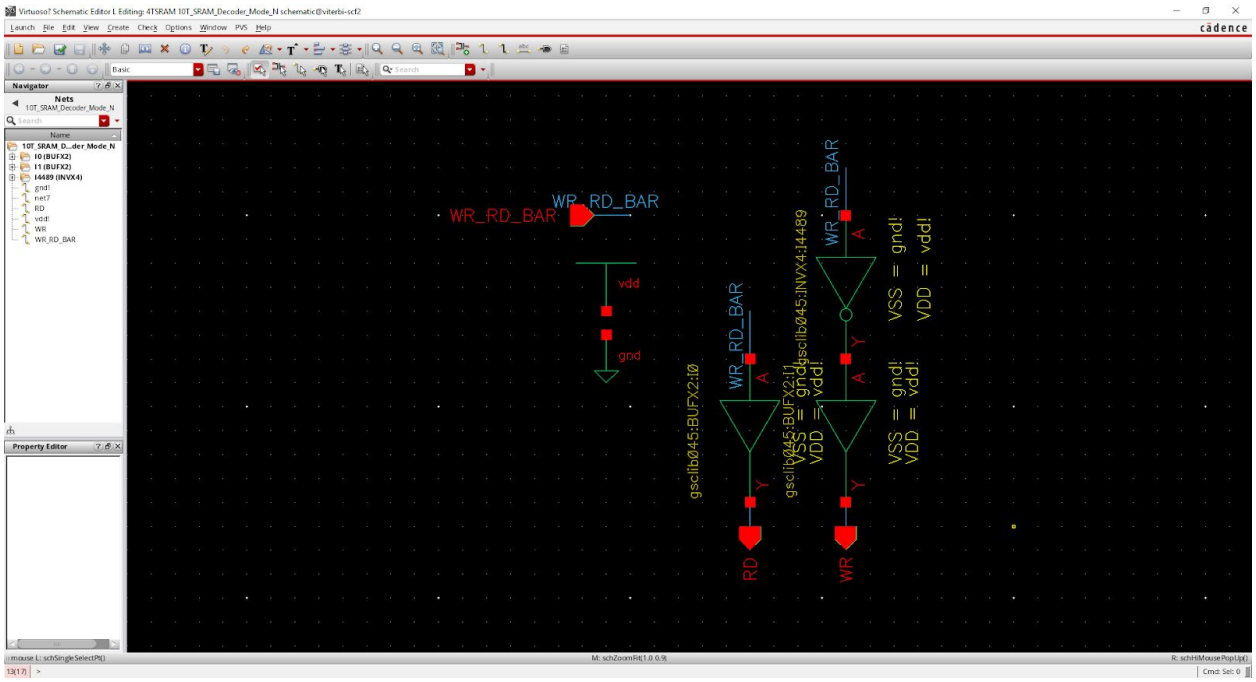

Figure 48: Decoder mode: Normal Schematic

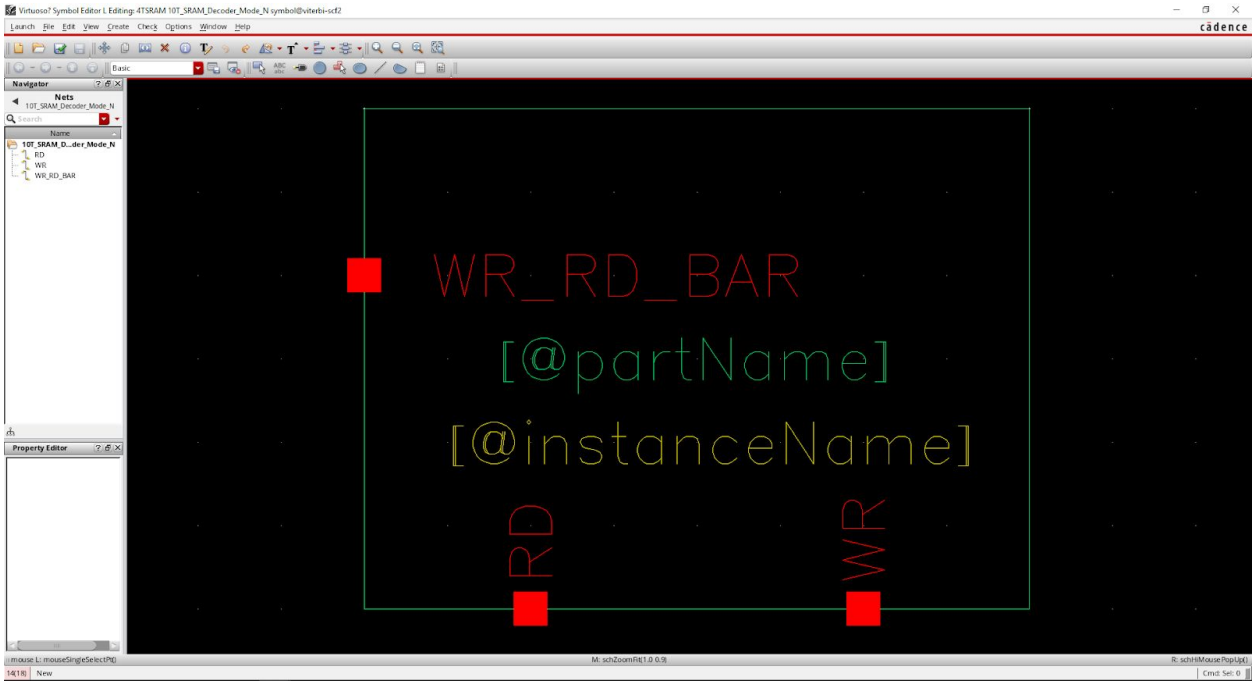

Figure 49: Decoder mode: Normal symbol

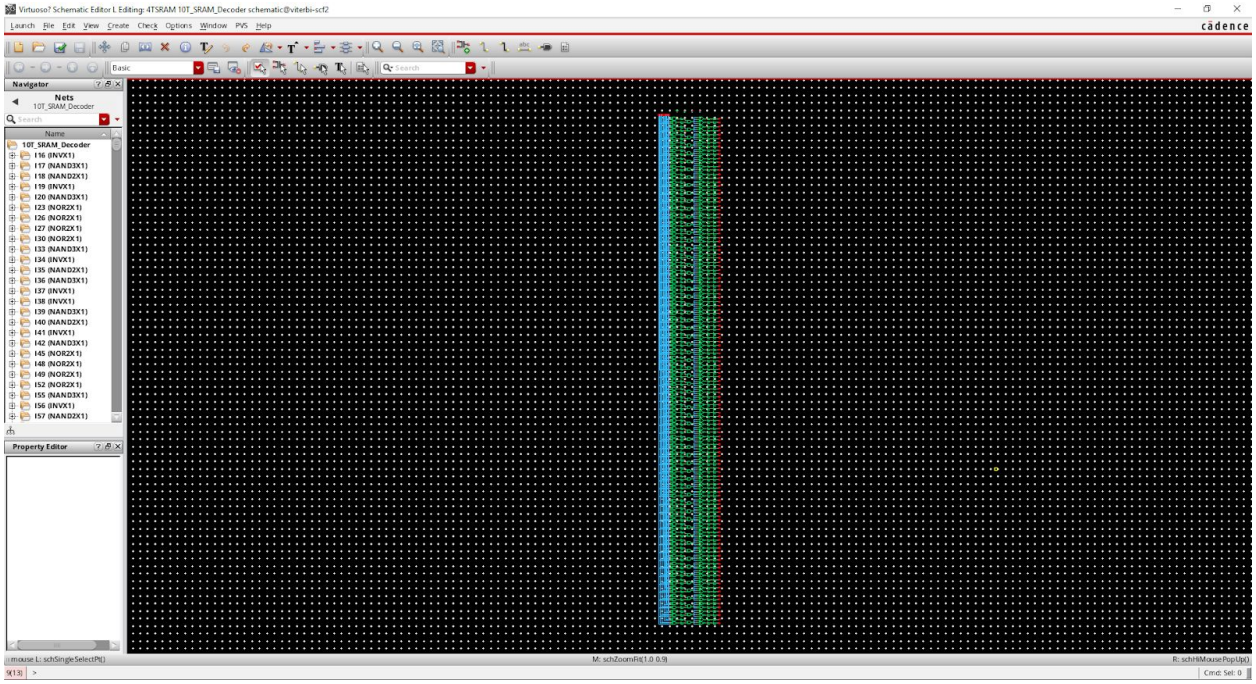

Figure 50: Decoder schematic overview

Figure 51: Decoder schematic zoomed

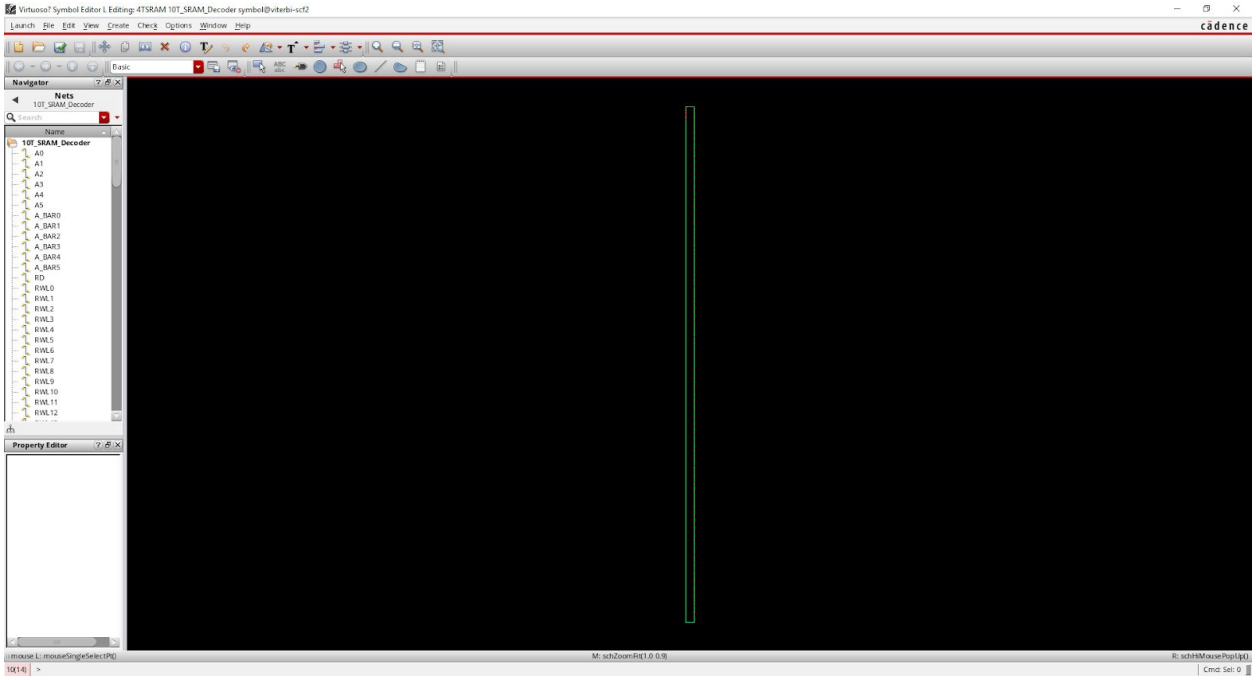

Figure 52: Decoder symbol

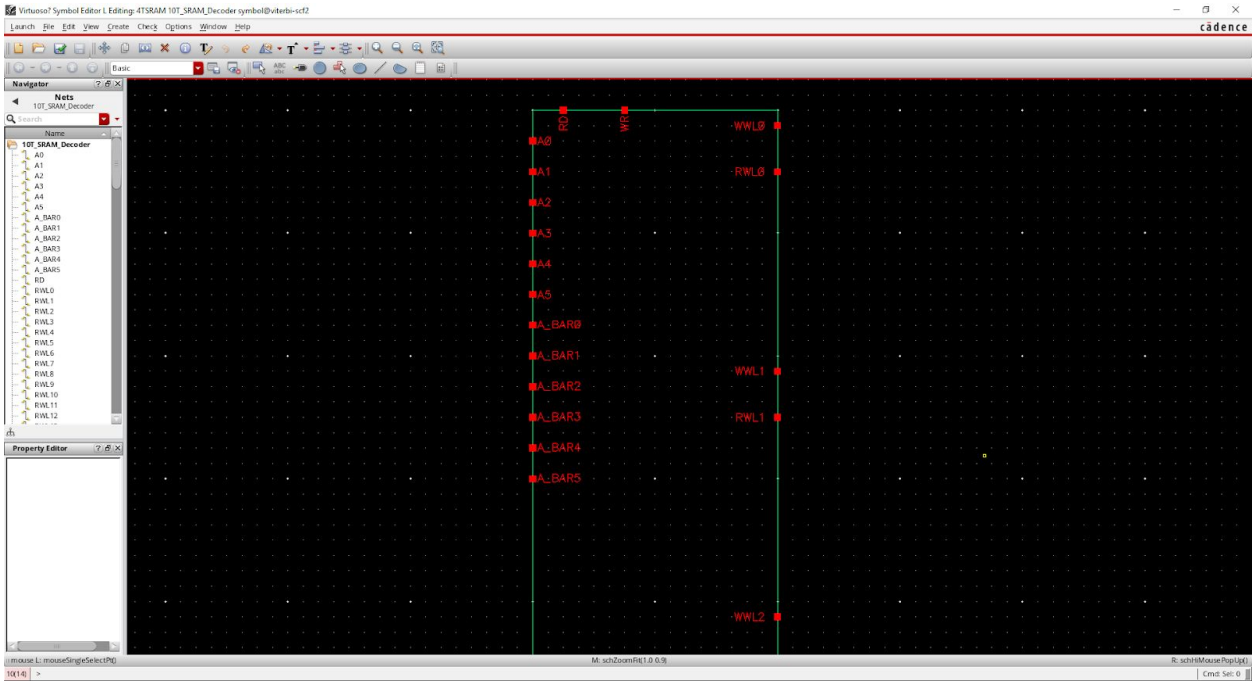

Figure 53: Decoder symbol zoomed

Simulation

After using the logical effort technique, one buffer to the wordlines to get the best number of stages. As a result, the decoder delay is observed about 200ps after optimizing.

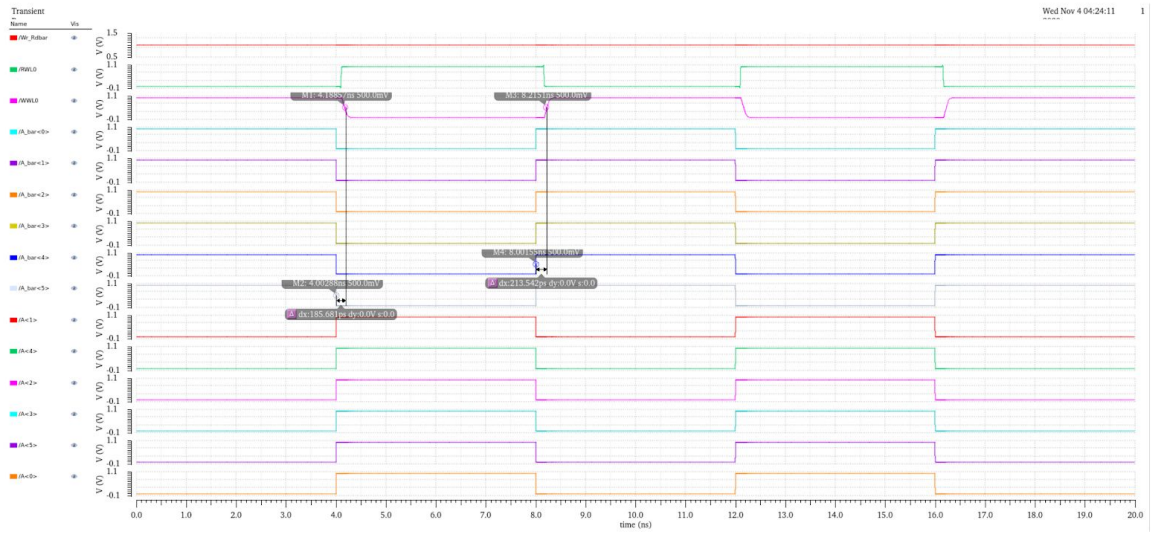

Figure 54: Decoder timing test

6. DFF

Description

Since we don't have any multiplexer structure, we have placed DFF to large sensing and small sensing parts separately. With every rising edge of the clock pulse, we sample the data coming from the sense amplifiers.

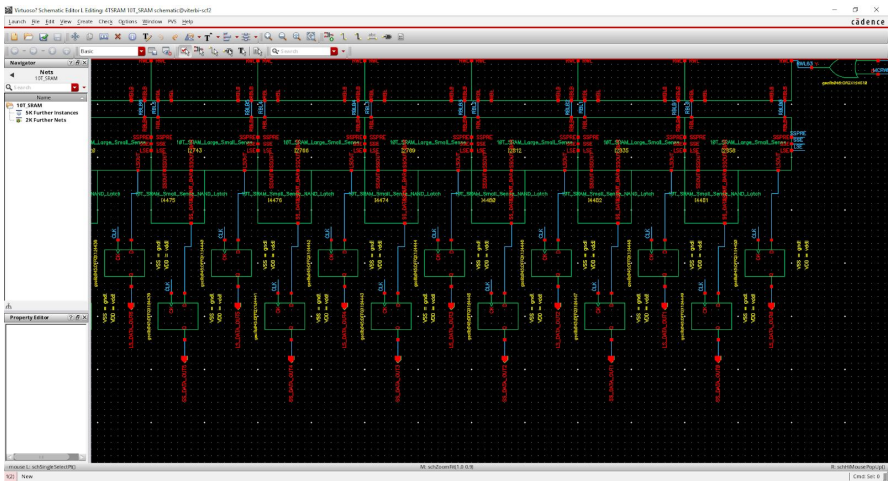

Figure 55: DFF placements

7. Overall 64x64 SRAM schematic

Description

The SRAM structure consists of cross-coupled inverters which are used to hold the state in CMOS. This static storage in a powered cell does not require a refresh operation. To write a new state, it is required to force the nodes into the opposite state. The writing operation is achieved by enabling the write wordline and it is connected to the gate of access transistors. The read operation is achieved by peripheral transistors that are placed in a way that reading will not affect the data. Overall, we have 10 transistors (4 for the two inverters, 2 for the access, and 4 for BL and BLB read lines). In the project, we are asked to design a 1K bit (64x64) SRAM circuit.

In normal mode, we activate the decoder on the left hand side given in the figure below. In in-memory computing, we activate both of the decoders and by using logic OR gate, we achieve multiple wordline access on SRAM structure. Since small sense and large sense circuits have their own enable signals, we can connect the read BL and BLB paths to the corresponding sense circuits.

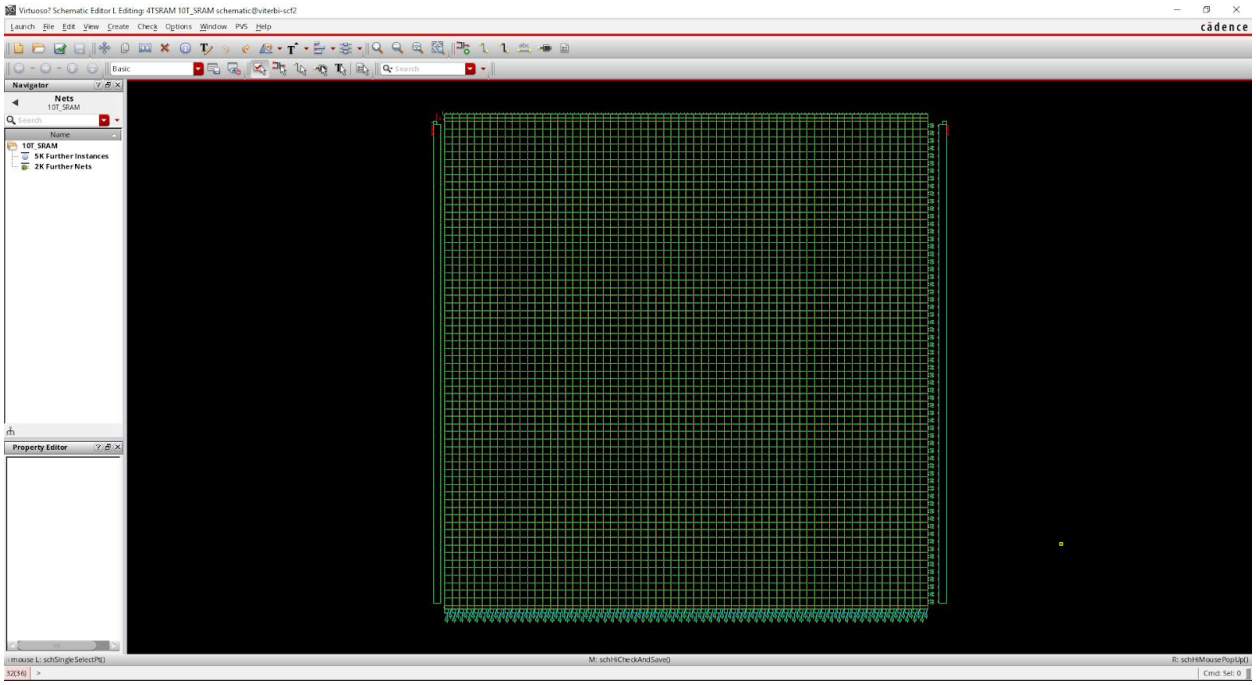

Figure 56: 10T SRAM 64x64 Schematic

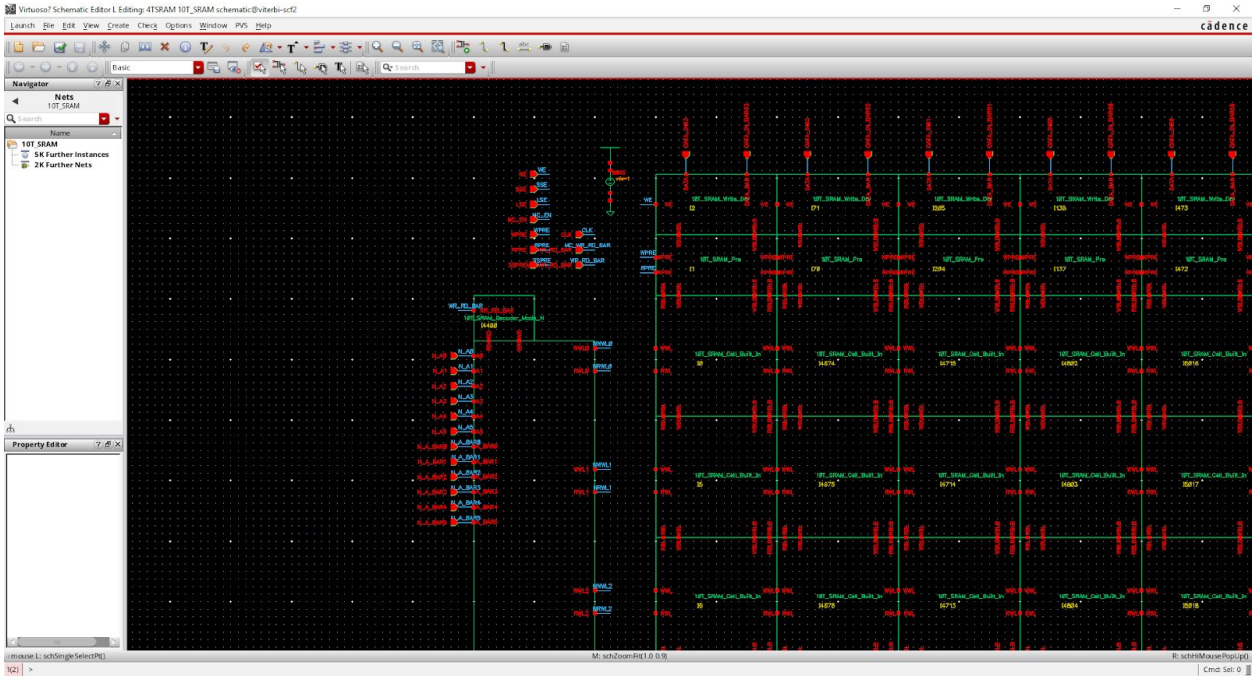

Figure 57: 10T SRAM 64x64 Schematic Zoomed

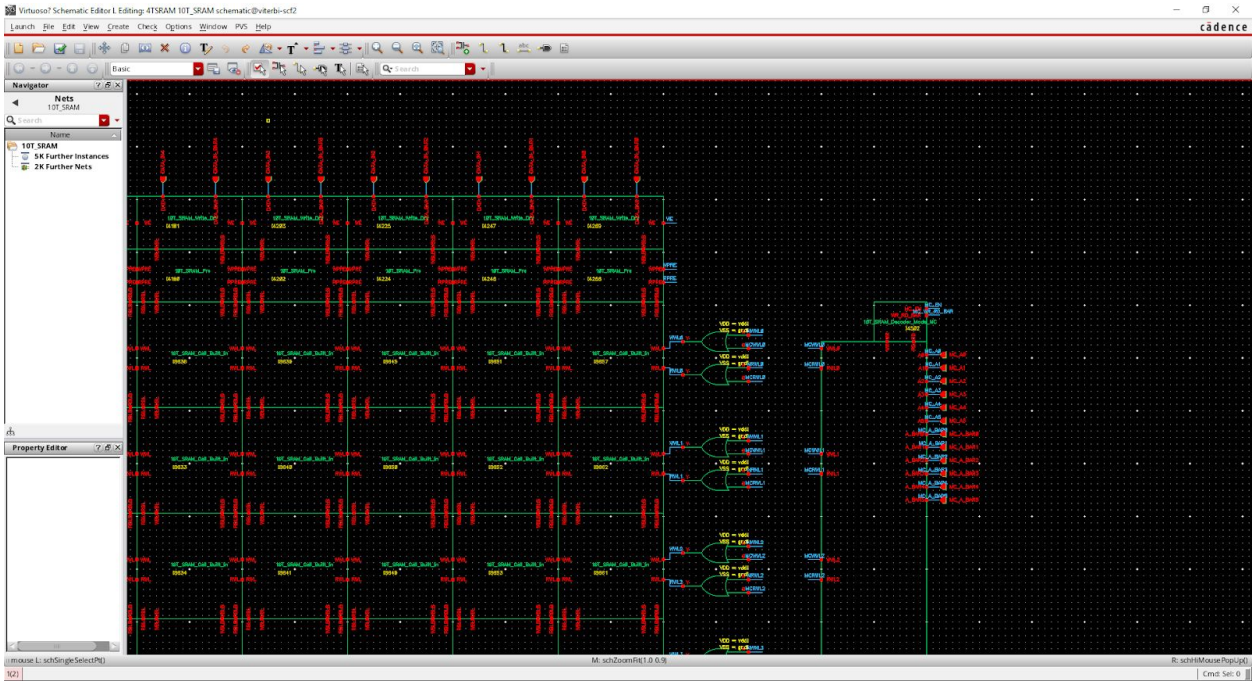

Figure 58: 10T SRAM 64x64 Schematic Zoomed-1

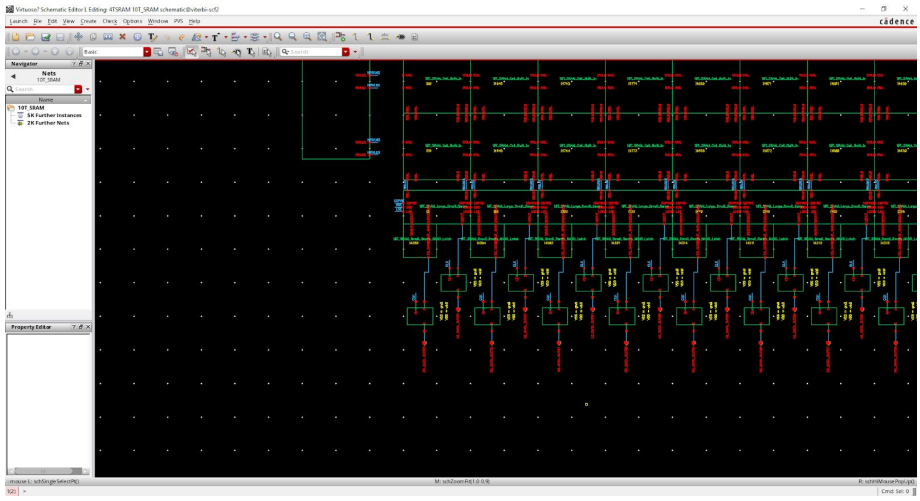

Figure 59: 10T SRAM 64x64 Schematic Zoomed-2

Simulation

In order to acquire the timings for write and read operations, we have performed write 1 -> read 1 -> write 0 -> read 0 -> write 1 operations on 64x64 SRAM structure. For the observation, we plotted the wordline 0 and 1 in addition to the bitline & bitline_bar 0 and 1. This means our observation cells are row 0-1 and column 0-1. The correct operations are achieved. The timing of the precharge circuit is skipped since we provided the details of voltage drops (52mV and 97mV) on BL-BLB in the first section. Therefore, the results of BL-BLB voltages drop to zero since there is no precharge during the read operation. This does not affect the result but it affects the timing for the next read operation. Additionally, we placed the markers to 900mV points on BL-BLB to get accurate differences.

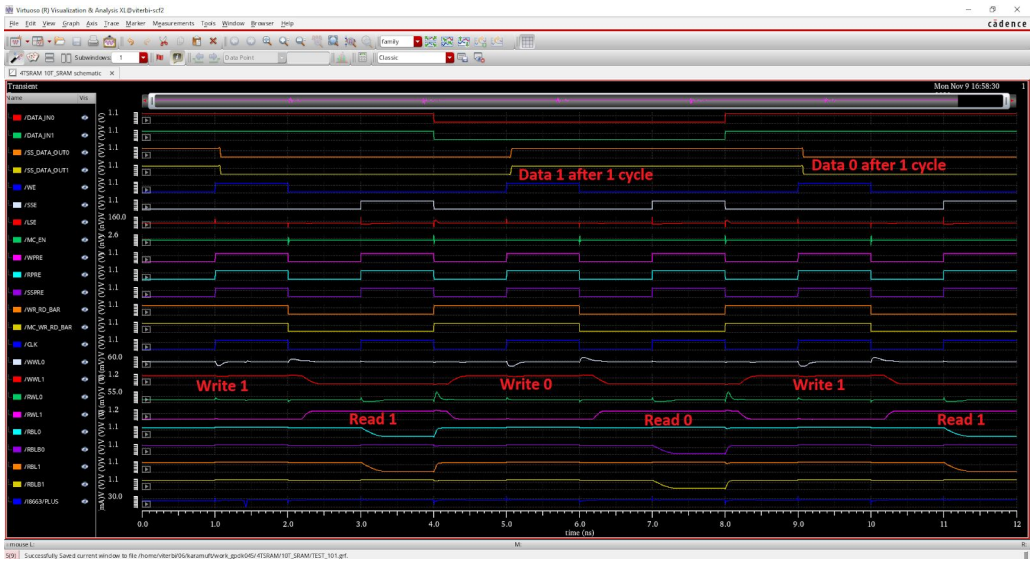

Figure 60: 64x64 SRAM write read operations waveform (W1=>R1=>W0=>R0=>W1)

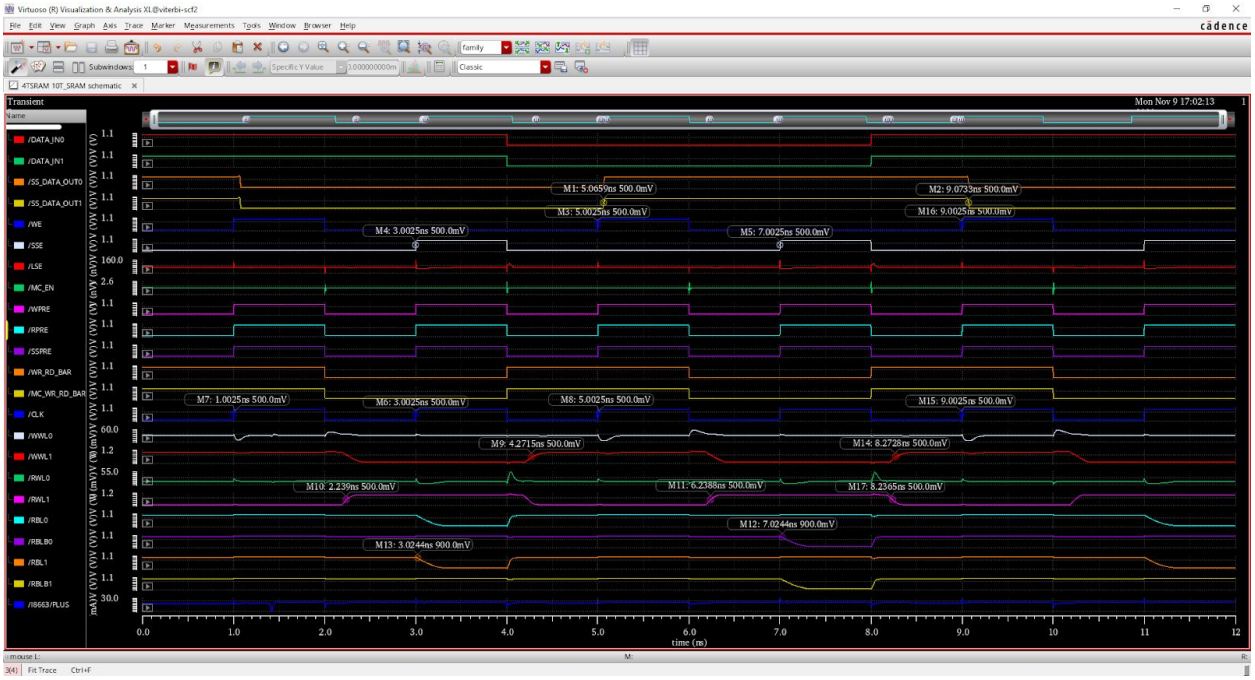

Figure 61: 64x64 SRAM write read operations timing labels (W1=>R1=>W0=>R0=>W1)

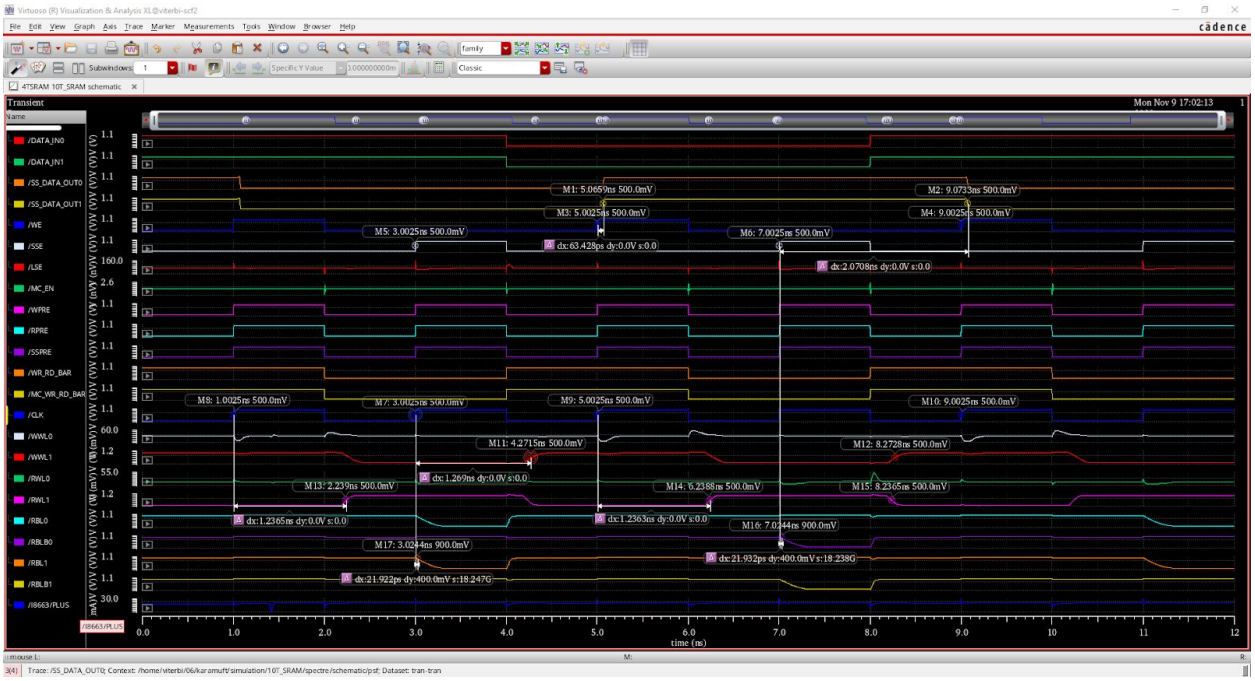

Figure 62: 64x64 SRAM write read operations timing difference (W1=>R1=>W0=>R0=>W1)

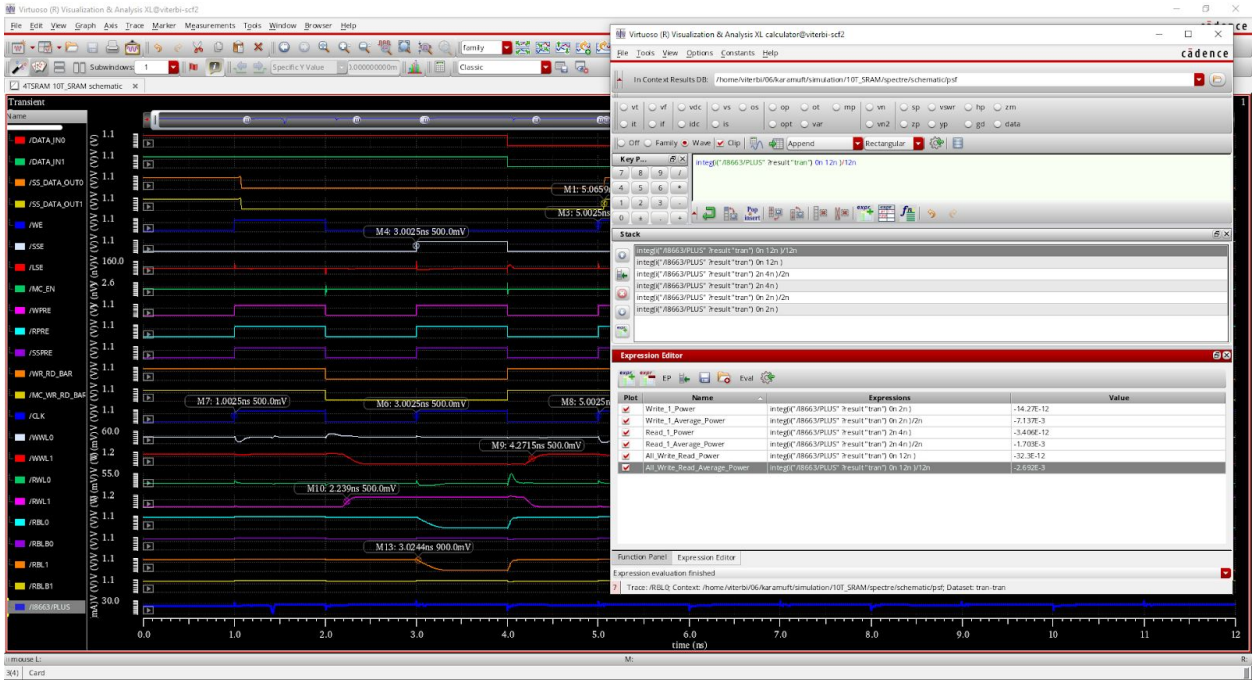

Figure 63: 64x64 SRAM Write-Read power calculation

As shown in Figure 54, decoder delay is approximately 236.5ps in 64x64 SRAM structure. According to Figure 62, due to our pattern timings, the difference appears as 1.2365ns and it includes an extra 1ns. Since pattern timing doesn't follow the standard ways, the bitline delay appears to be 800ps. However, this value should be smaller than the one shown in Figure 62. Due to memory issues on the server side and segmentation faults during the simulations, we only saved one simulation result. Sense amplifier delay obtained from Figure 62 is observed as 21.932ps. Total read delay is achieved as 63.428ps.

Pins

We have divided the related control inputs and the DFF outputs of normal and in-memory read operations. The related information is given in Table 1.

Table 1: Summary of all the I/O signals

Input Pins		
Decoder Inputs	N_A<0:5>	N_A_BAR<0:5>
Memory Computation Decoder Inputs	MC_A<0:5>	MC_A_BAR<0:5>
Input Data	DATA_IN<0:63>	DATA_IN_BAR<0:63>
In-Memory Enable	MC_EN	1 to activate
In Memory-Write Read Bar	MC_WR_RD_BAR	1 to Write, 0 to Read
Write Read Bar	WR_RD_BAR	1 to Write, 0 to Read
Read Precharge	RPRE	0 to activate
Write Precharge	WPRE	0 to activate
Sense Amplifier Precharge	SSPRE	0 to activate
Write Enable	WE	1 to activate
Read Enable	SSE	1 to perform
VDD	vdd!	1
VSS	gnd!	0
VBN	gnd!	0
VBP	vdd!	1
Clock	CLK	0-1
Output Pins		
Output Data-Small Sense	SS_DATA_OUT<0:63>	
Output Data-Large Sense	LS_DATA_OUT<0:63>	

Related Vector files

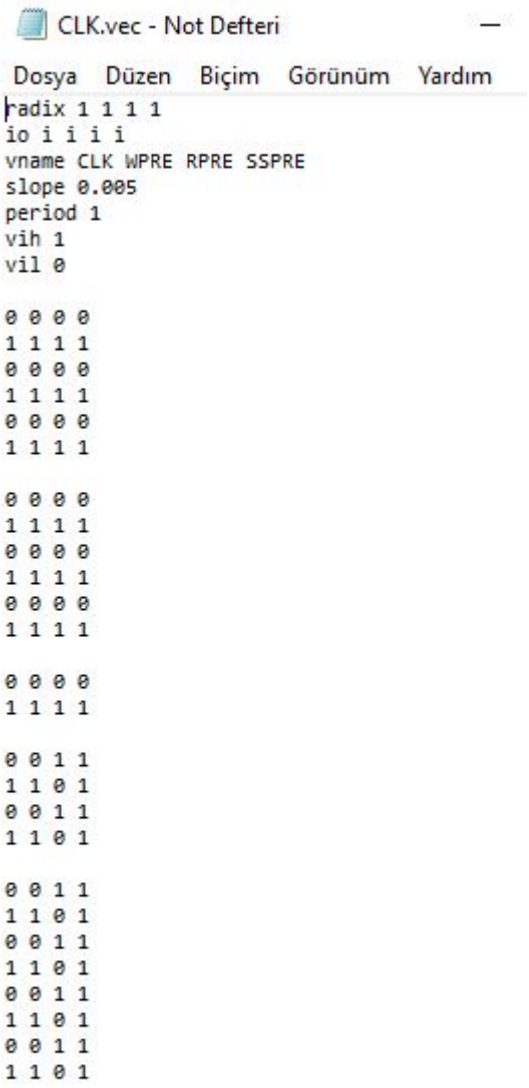

Figure 64: Clock vector file for single column

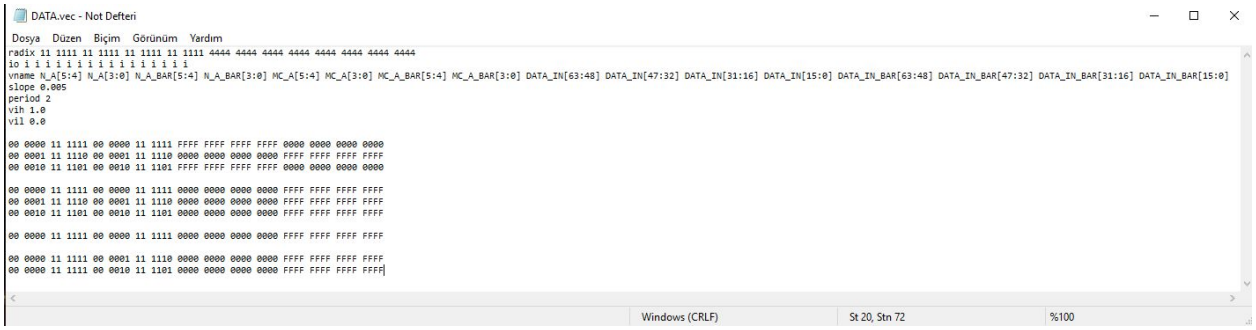

Figure 65: Data vector file for single column

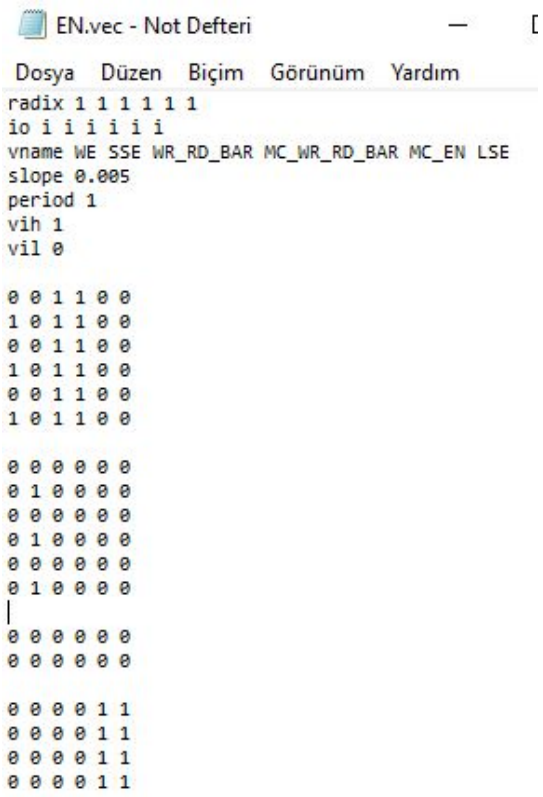

Figure 66: Single column enable vector file

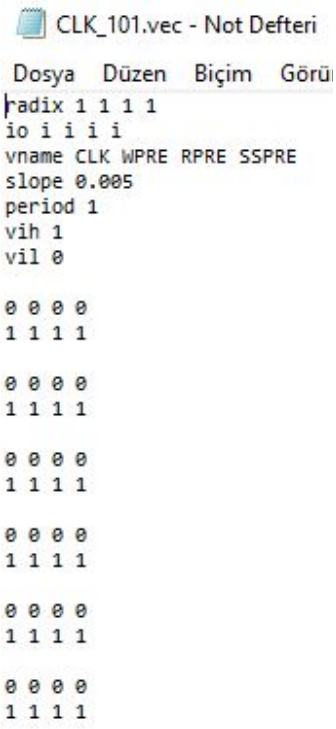

Figure 67: SRAM 64x64 Write1-Read1-Write0-Read0-Write1 clock vector file

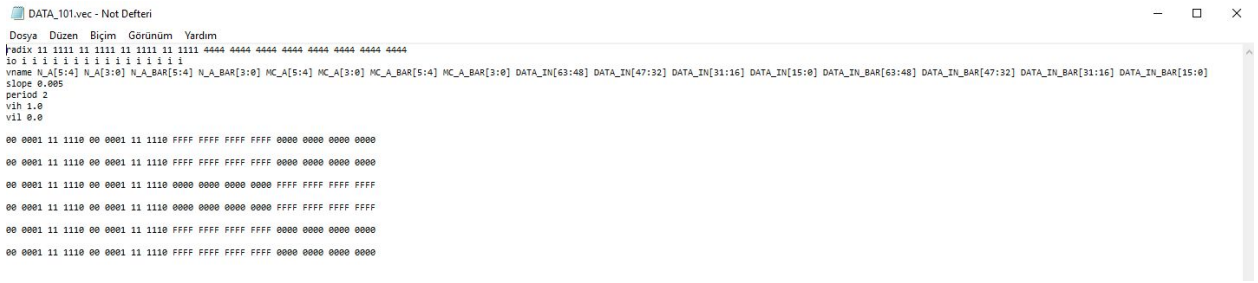

Figure 68: SRAM 64x64 Write1-Read1-Write0-Read0-Write1 data vector file

EN_101.vec - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

```
radix 1 1 1 1 1 1
io i i i i i i
vname WE SSE WR_RD_BAR MC_WR_RD_BAR MC_EN LSE
slope 0.005
period 1
vih 1
vil 0

0 0 1 1 0 0
1 0 1 1 0 0

0 0 0 0 0 0
0 1 0 0 0 0

0 0 1 1 0 0
1 0 1 1 0 0

0 0 0 0 0 0
0 1 0 0 0 0

0 0 1 1 0 0
1 0 1 1 0 0

0 0 0 0 0 0
0 1 0 0 0 0
```

Figure 69: SRAM 64x64 Write1-Read1-Write0-Read0-Write1 enable vector file

CLK_MEM.vec - Not Defteri

Dosya Düzen Biçim Görür

```
radix 1 1 1 1
io i i i i
vname CLK WPRE RPRE SSPRE
slope 0.005
period 1
vih 1
vil 0

0 0 0 0
1 1 1 1
0 0 0 0
1 1 1 1
0 0 0 0
1 1 1 1
0 0 0 0
1 1 1 1

0 0 0 0
1 1 1 1

0 0 1 1
1 1 0 1
0 0 1 1
1 1 0 1
0 0 1 1
1 1 0 1

0 0 0 0
1 1 1 1
```

Figure 70: SRAM 64x64 large sense clock vector file

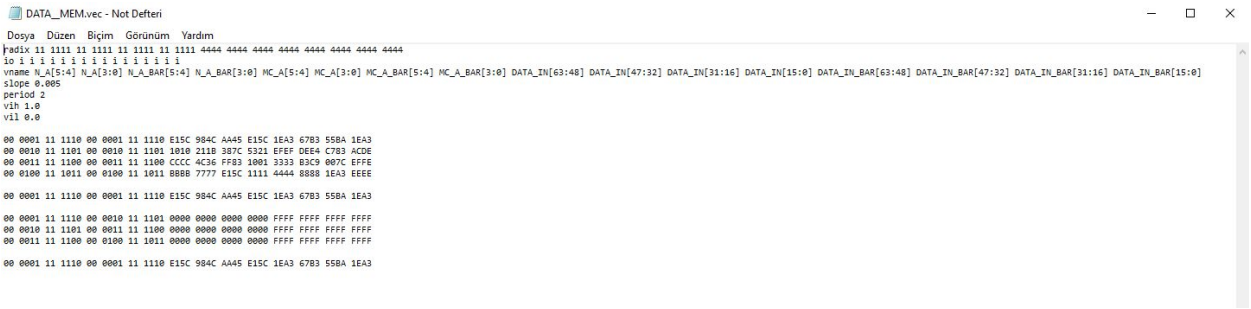

Figure 71: SRAM 64x64 large sense data vector file

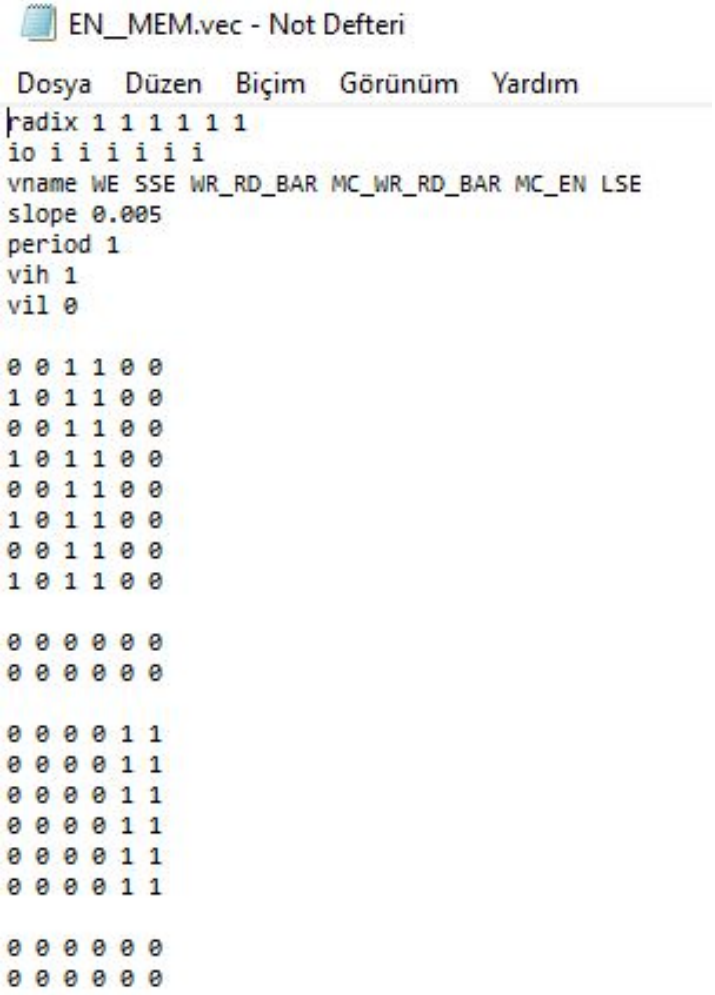

Figure 72: SRAM 64x64 large sense enable vector file

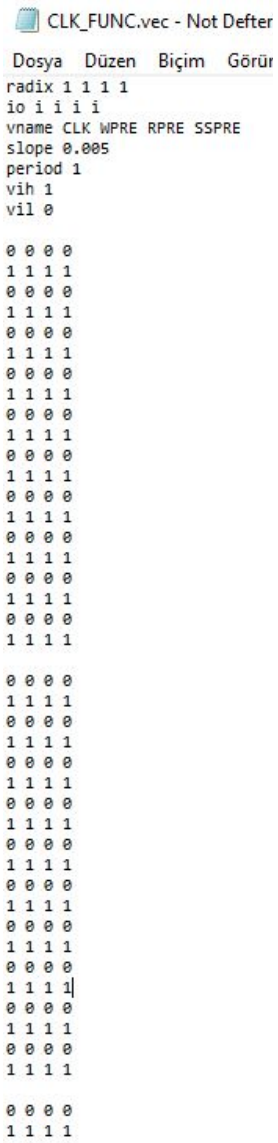

Figure 73: SRAM 64x64 small sense clock vector file

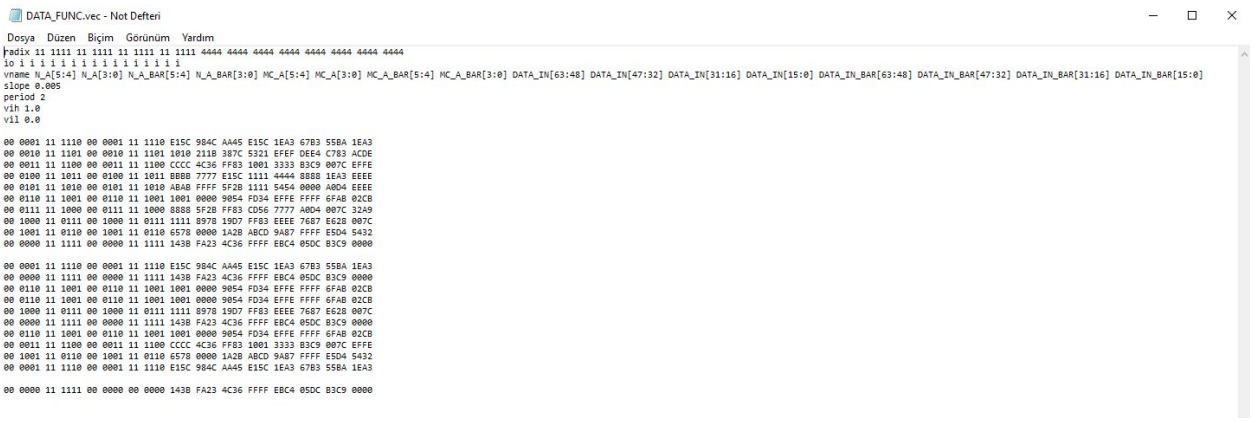

Figure 74: SRAM 64x64 small sense data vector file

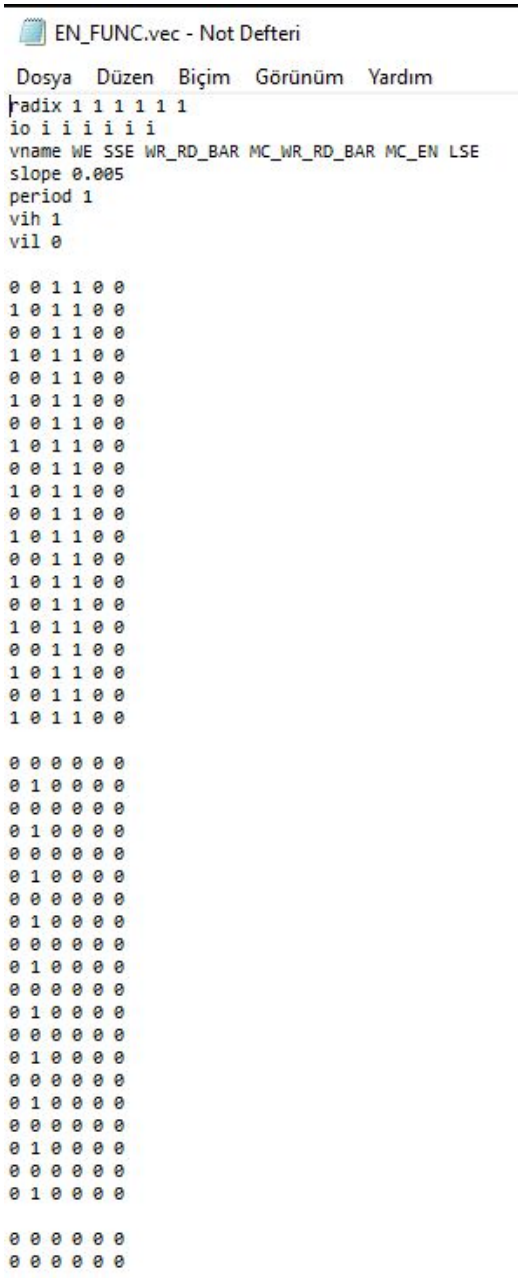

Figure 75: SRAM 64x64 small sense enable vector file

Python Scripts' Results

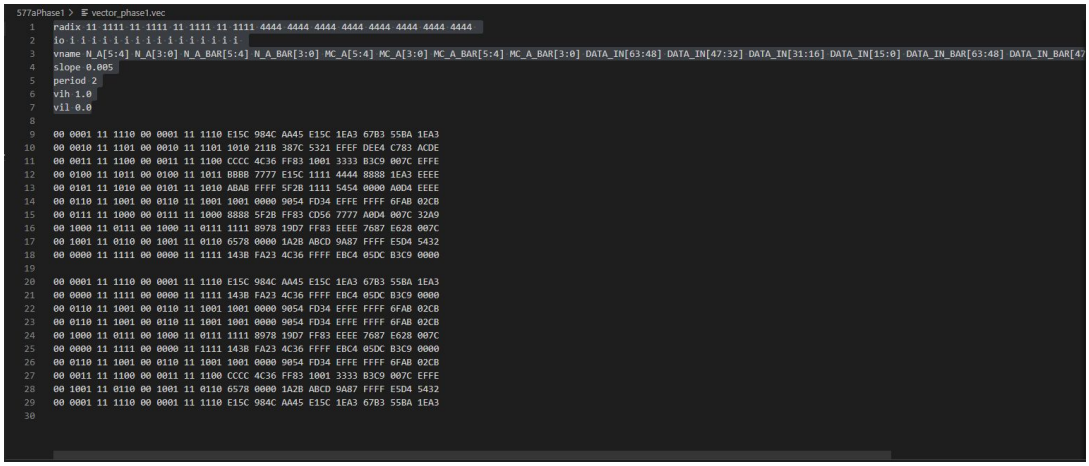

Figure 76: Vector file for normal read operation from the script

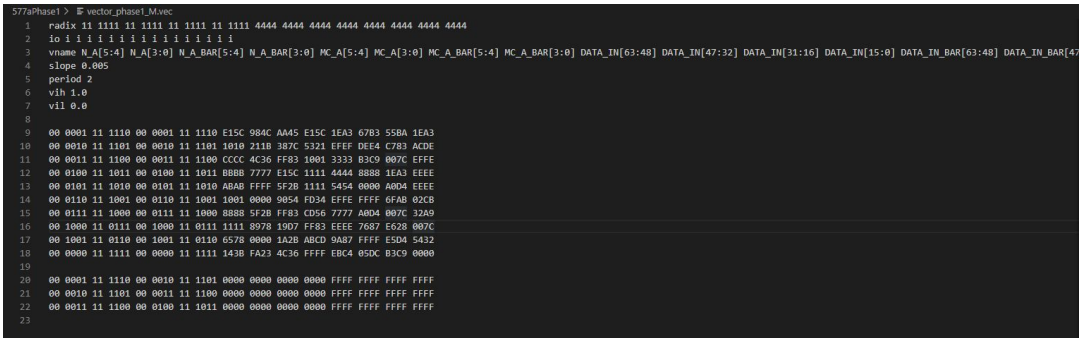

Figure 77: Vector file for in memory operation from the script

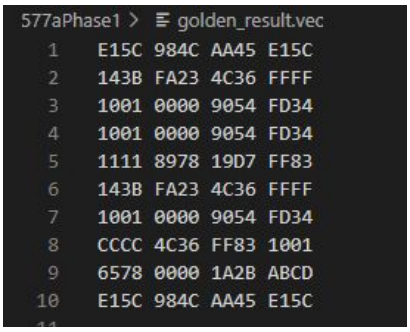

Figure 78: The Golden result for Normal Read Operation

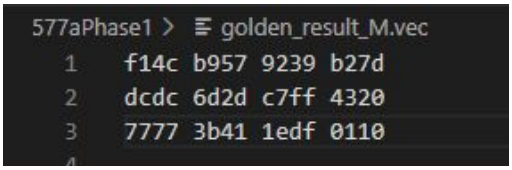

Figure 79: The Golden result for In Memory Operation

Specs

10T-SRAM

Sizing

BL read:	**150nm**
BLB read:	**150nm**
BL access:	**200nm**
BLB access:	**200nm**
Q pull-down:	**330nm**
Q_bar pull-down:	**330nm**
Q pull-up:	**120nm**
Q_bar pull-up:	**120nm**

Layout

Layout Area:	**2.788 um^2 (0.87um x 3.205um)**

Write Driver

Sizing

All NMOS:	**440nm**

Precharge Circuits

Sizing

All PMOS:	**330nm**

Large Sense Amplifier

Sizing

Inverters:	**x1**
AND gates:	**x1**
Two NMOS:	**440nm**

Small Sense Amplifier

Sizing

Precharge PMOS:	**330nm**
Small Sense PMOS:	**440nm**
Small Sense NMOS:	**440nm**
Latch NAND gates:	**x1**

Decoder Modes

Sizing

Inverters:	**x4**
NAND gates:	**x1**
Buffer:	**x2**

Row Decoder

Sizing

NAND gates:	**x1**
Inverters:	**x1**
NOR gates:	**x1**

DFF

Sizing

DFFQ：	**x1**