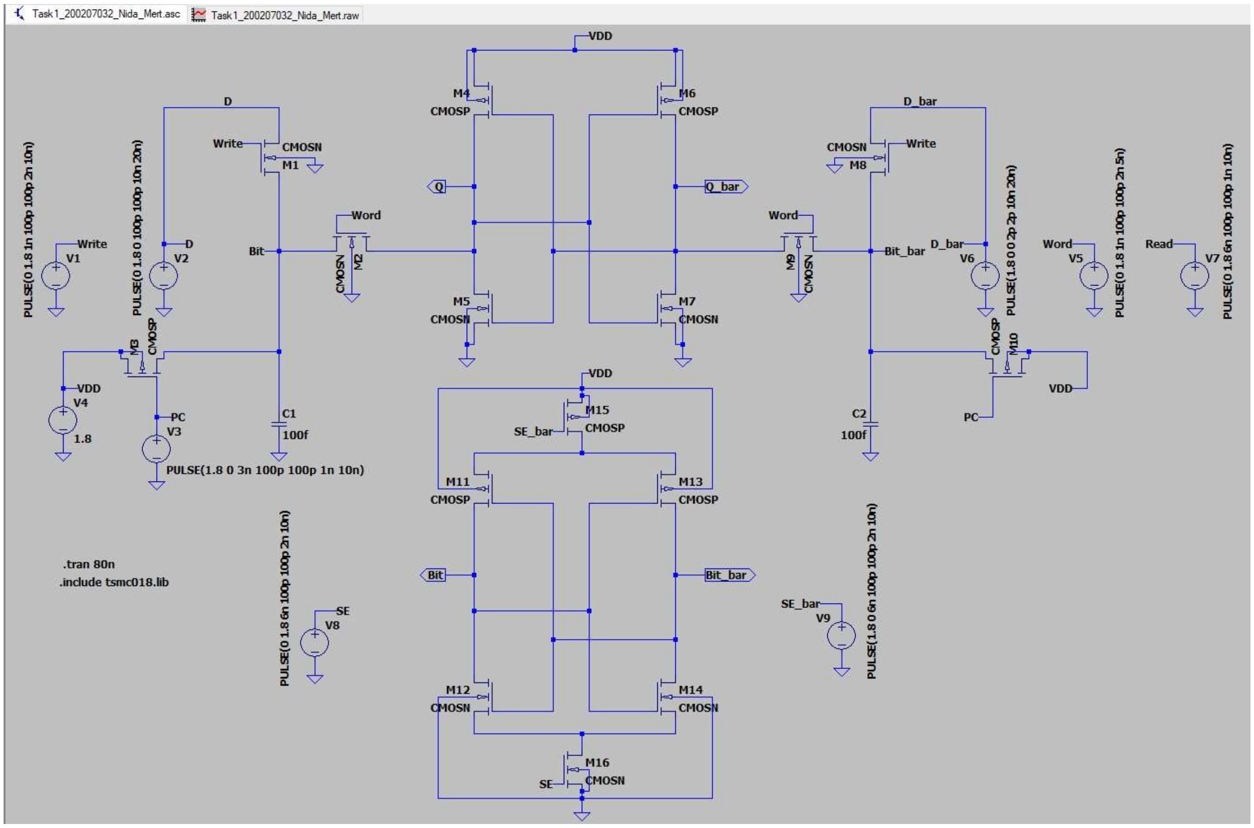
Design and Simulation of an 8x8 SRAM Array

Task1: Regenerate the 1-bit 6T SRAM cell integrated with Sense Amplifier, pre,charge and write drive circuitry which is described in the link below.

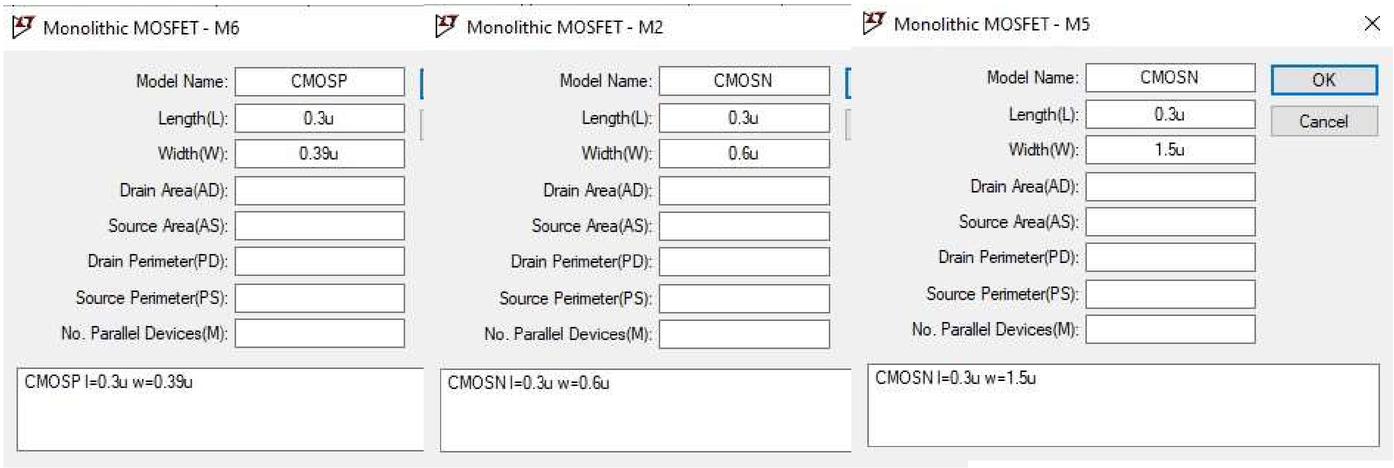

Task1_200207032_Nida_Mert.asc
Task1_200207032_Nida_Mert.raw
VDD
M4
CMOSP
M6
CMOSP
D
Write
CMOSN
M1
D_bar
CMOSN
M8
Write
Q
Q_bar
Word
Word
Bit
M2
CMOSN
M9
CMOSN
Bit_bar
D_bar
V6
Write
V1
PULSE(0 1.8 1n 100p 100p 2n 10n)
D
V2
PULSE(0 1.8 0 100p 100p 10n 20n)
PULSE(1.8 0 0 2p 2p 10n 20n)
Word
V5
PULSE(0 1.8 1n 100p 100p 2n 5n)
Read
V7
PULSE(0 1.8 6n 100p 100p 1n 10n)
M5
CMOSN
M7
CMOSN
M3
CMOSP
M10
CMOSP
VDD
V4
1.8
PC
V3
PULSE(1.8 0 3n 100p 100p 1n 10n)
C1
100f
C2
100f
PC
VDD
VDD
M15
SE_bar
CMOSP
M11
CMOSP
M13
CMOSP
Bit
Bit_bar
SE
V8
PULSE(0 1.8 6n 100p 100p 2n 10n)
SE_bar
V9
PULSE(1.8 0 6n 100p 100p 2n 10n)
M12
CMOSN
M14
CMOSN
M16
SE
CMOSN

```
.tran 80n
.include tsmc018.lib
```

Figure 1: Circuit for 1-bit 6T SRAM cell with Sense amplifer, pre-charge and write driver

As shown in Figure 1, the circuit for a 1-bit 6T SRAM cell with sense amplifier, pre-charge, and write driver (implemented as demonstrated in the assignment video) has been constructed. The sources and simulation parameters used in the circuit are the same with the video, and the transistor lengths and widths used in the circuit are explained in Figure 2-3 below.

During the transistor sizing phase, it was observed that the weak, medium, and strong configurations shown in the video led to the most reliable results across the tests. Therefore, the values used for read/write and hold stability are as follows:



Monolithic MOSFET - M6

Model Name: CMOSP
Length(L): 0.3u
Width(W): 0.39u
Drain Area(AD):
Source Area(AS):
Drain Perimeter(PD):
Source Perimeter(PS):
No. Parallel Devices(M):

CMOSP l=0.3u w=0.39u

Monolithic MOSFET - M2

Model Name: CMOSN
Length(L): 0.3u
Width(W): 0.6u
Drain Area(AD):
Source Area(AS):
Drain Perimeter(PD):
Source Perimeter(PS):
No. Parallel Devices(M):

CMOSN l=0.3u w=0.6u

Monolithic MOSFET - M5

Model Name: CMOSN
Length(L): 0.3u
Width(W): 1.5u
Drain Area(AD):
Source Area(AS):
Drain Perimeter(PD):
Source Perimeter(PS):
No. Parallel Devices(M):

CMOSN l=0.3u w=1.5u

Figure 2: Transistor sizing

First, the lengths of all transistors in the cell structure were set to 0.3 µm. The widths of the pull-up PMOS transistors M4 and M6 were chosen to be the weakest among all transistors, while M2 and M9 (pull-down NMOS and access transistor, respectively) were sized as medium strength. Lastly, M5 and M7 (also pull-down NMOS and access transistor) were given the strongest width values. Figure 2 shows the selected values for one example from each of the three sizing pairs. Figure 3 presents a snapshot of the configuration used in the video.

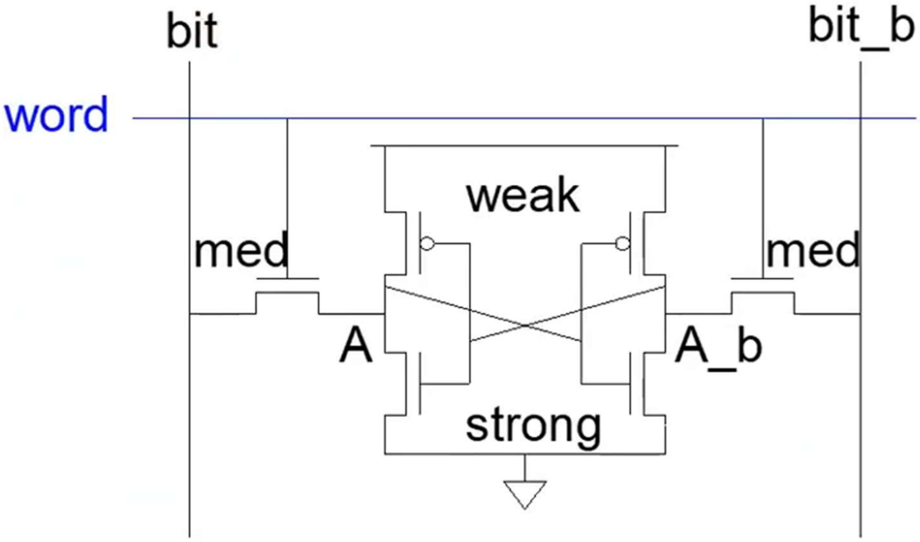


Figure 3: Configuration which is used in the Task 1 video

In this configuration, write stability was achieved by ensuring M2 >> M4, while read stability was ensured by setting M5 >> M2.

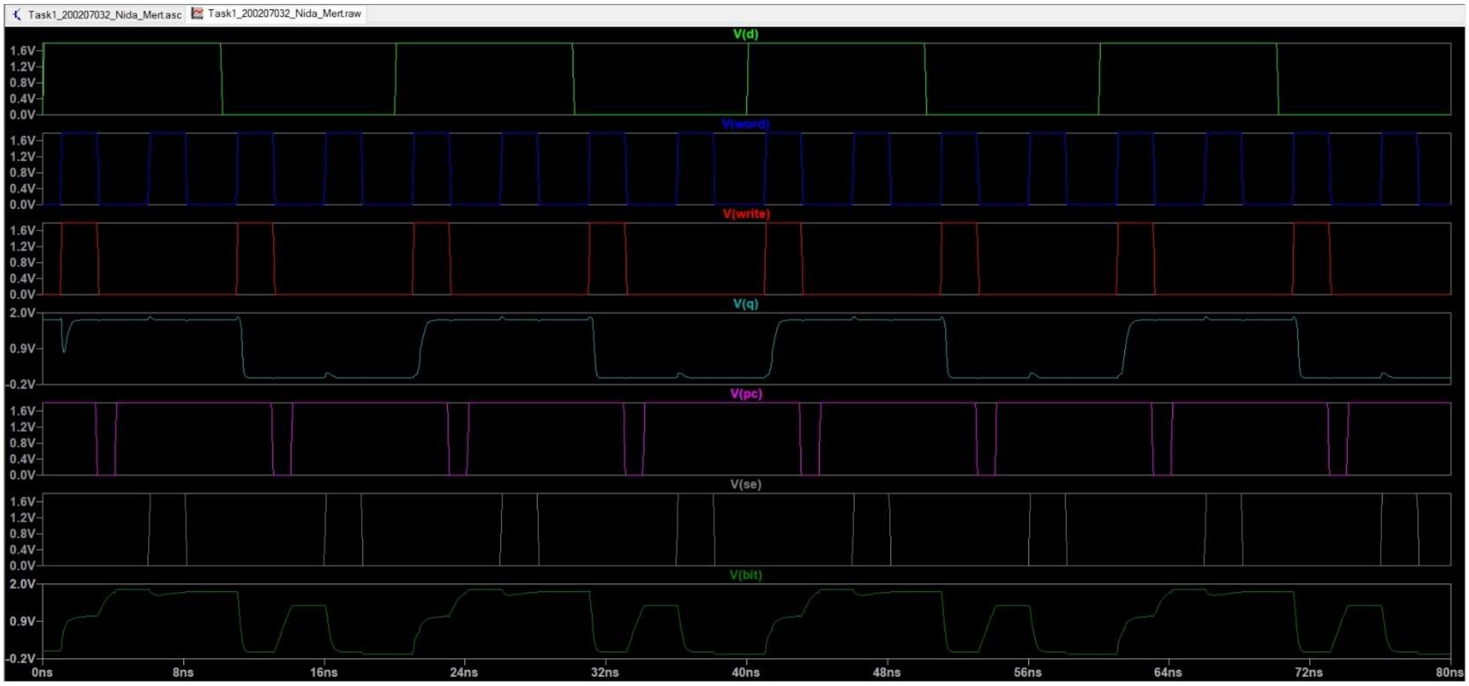


Figure 4: Simulation results for Task1

The simulation results align well with the SRAM read/write operation principles outlined in the referenced lesson. In the write phase, the write, word, and d signals are asserted simultaneously, allowing the input data to be written into the SRAM cell through the bitlines. This is reflected in the q output signal, which updates according to the input data. Prior to the read operation, the pc (precharge) signal is activated, bringing both bitlines to a VDD. Following this, the se (sense enable) signal triggers the sense amplifier, allowing it to detect and amplify the small voltage difference developed on the bitline due to the stored data. The bit signal demonstrates this behavior, showing clear transitions that correlate with the expected readout. The stability of the q output during read access further confirms the robustness of the sizing strategy applied in the design, ensuring the cell maintains its state. Overall, the waveform transitions and timing sequences confirm that the circuit behaves as expected, consistent with the theoretical structure presented in the SRAM design video.

Task 2: Building the 8x8 SRAM Array and integrate with Decoder and SAs

- **Task 2.1. Get 1-bit 6T SRAM cell with PC and WD from Task 1 and build a Symbol.**

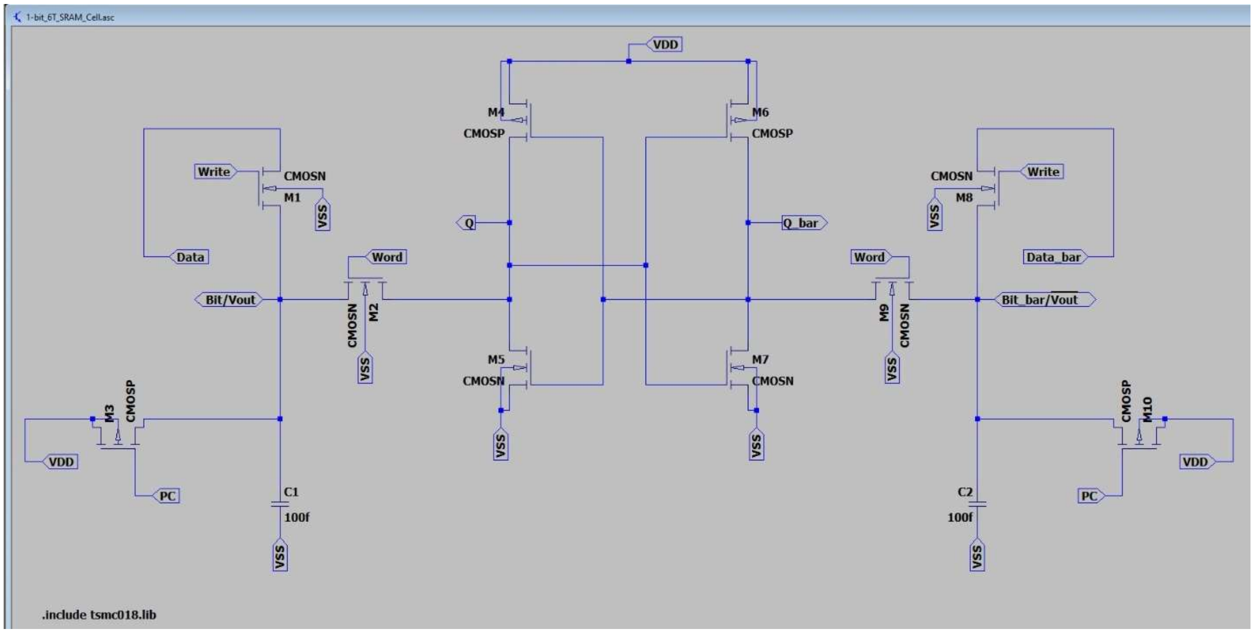


Figure 5: 1-Bit 6T SRAM Cell circuit

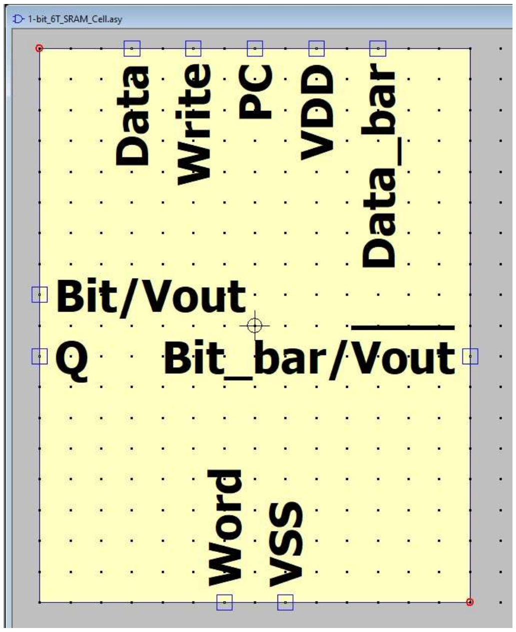


Figure 6: 1-Bit 6T SRAM Cell Symbol

- **Task 2.2. Get SA from Task 1 and Build a Symbol.**

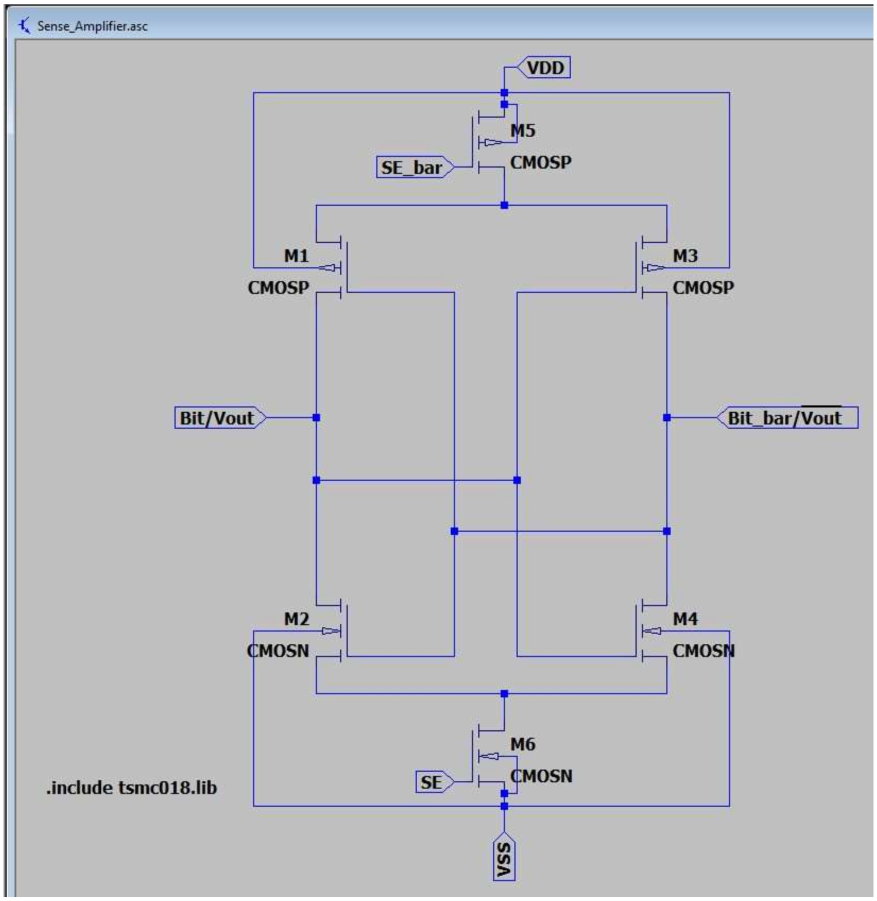


Figure 7: Sense Amplifier Circuit

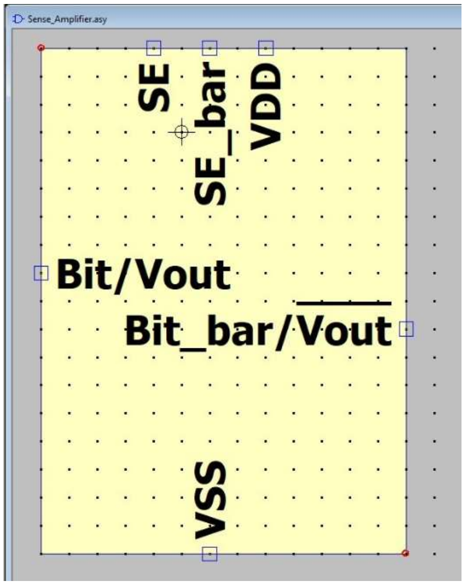
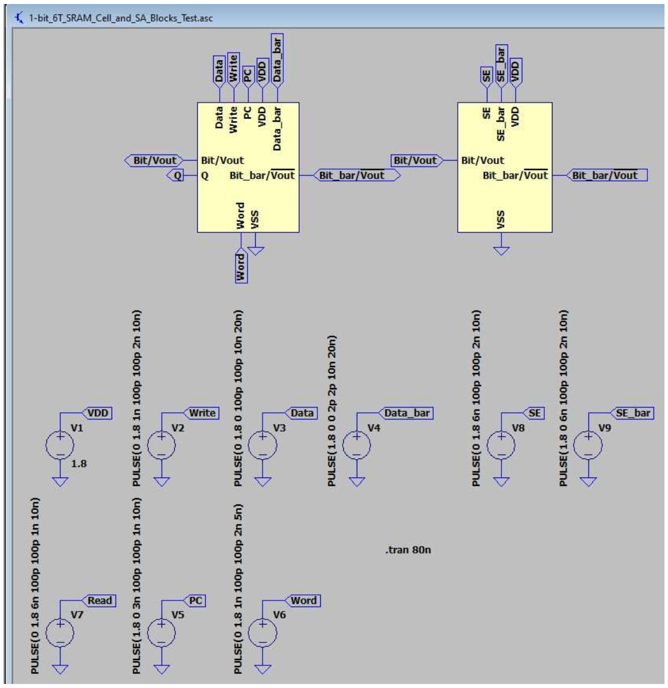


Figure 8: Sense Amplifier Symbol



1-bit_6T_SRAM_Cell_and_SA_Blocks_Test.asc

V1 1.8

V2 PULSE(0 1.8 1n 100p 100p 2n 10n)

V3 PULSE(0 1.8 0 100p 100p 10n 20n)

V4 PULSE(1.8 0 0 2p 2p 10n 20n)

V8 PULSE(0 1.8 6n 100p 100p 2n 10n)

V9 PULSE(1.8 0 6n 100p 100p 2n 10n)

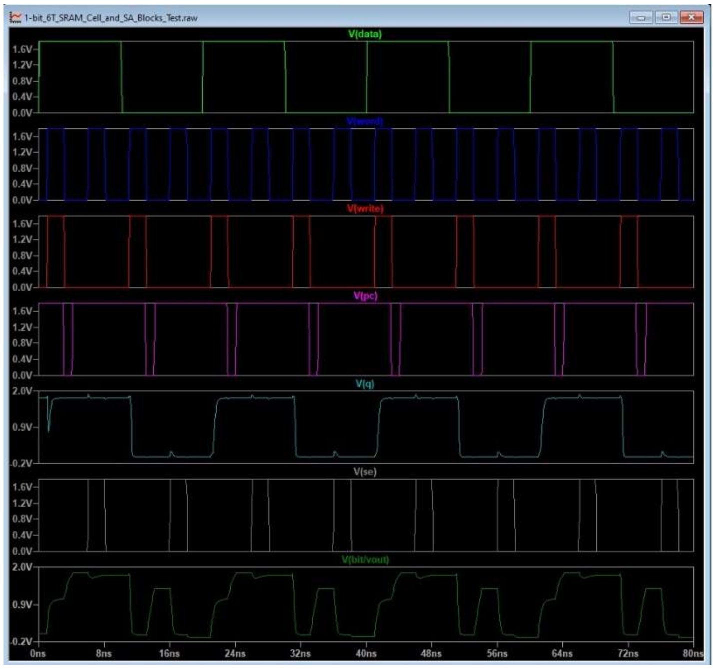
V7 PULSE(0 1.8 6n 100p 100p 1n 10n)

V5 PULSE(1.8 0 3n 100p 100p 1n 10n)

V6 PULSE(0 1.8 1n 100p 100p 2n 5n)

.tran 80n

Figure 9: 1-bit 6T SRAM Cell and Sense Amplifier Symbols’ Test Circuit

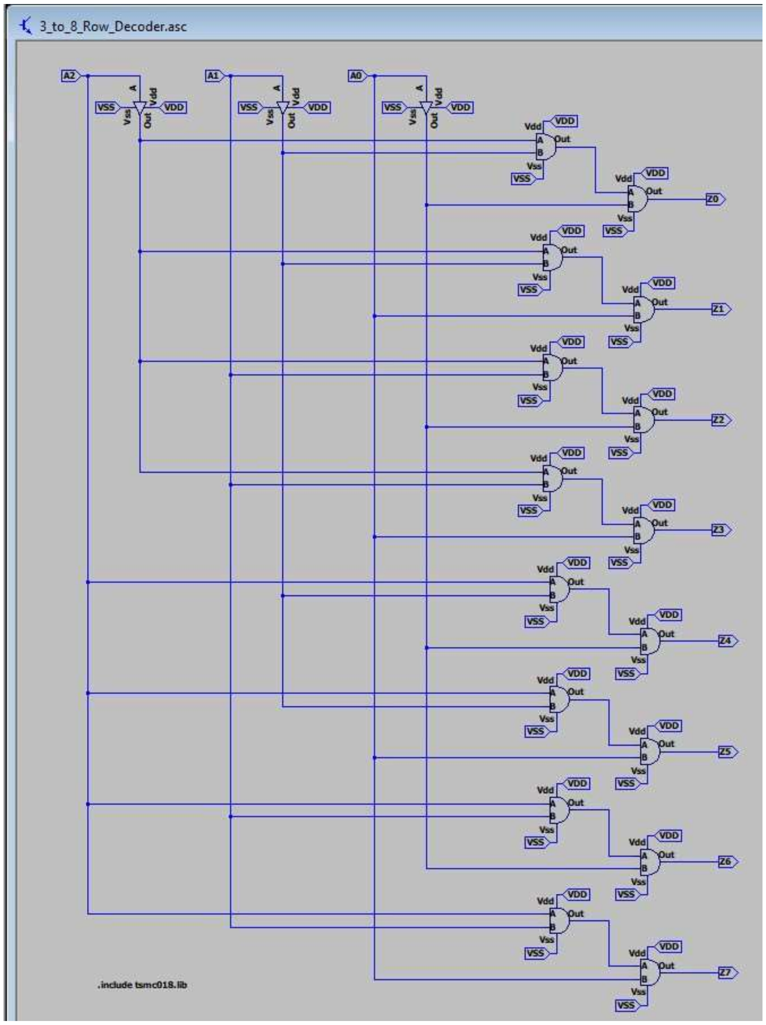


1-bit_6T_SRAM_Cell_and_SA_Blocks_Test.raw

Figure 10: 1-bit 6T SRAM Cell and Sense Amplifier Symbols’ Test Circuit’s Simulation Results

A separate test circuit was created to verify the 1-bit 6T SRAM Cell and Sense Amplifier symbols. The timing settings were taken from Task 1, and the goal was to observe the same simulation results as in Task 1. Accordingly, it was confirmed that everything in the constructed circuit functioned as expected and that there were no errors in the symbols.

- **Task 2.3. Design the 3-to-8 Row Decoder and build a Symbol.**



```
.include tsmc018.lib
```

Figure 11: 3-to-8 Row Decoder Circuit

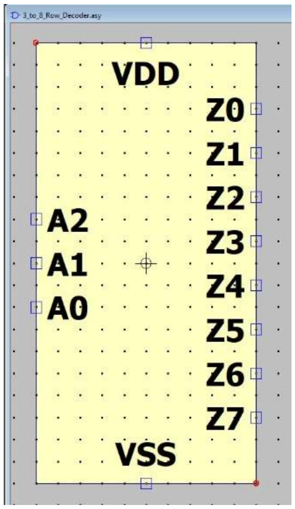
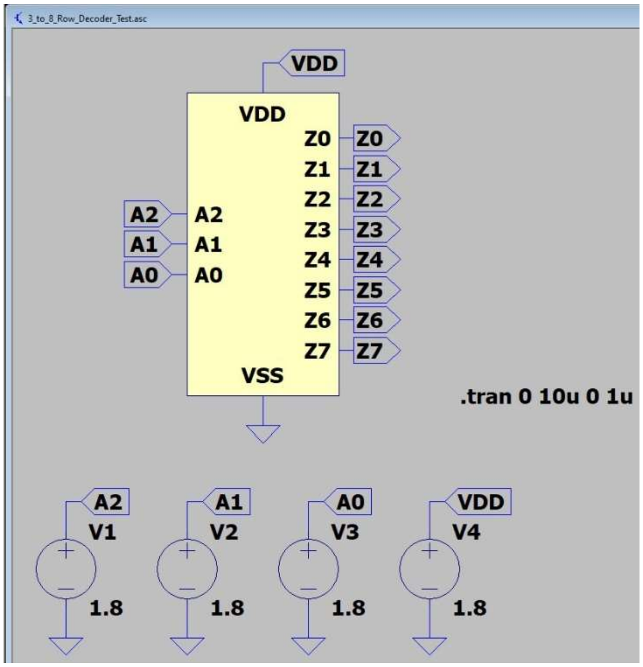


Figure 12: 3-to-8 Row Decoder Symbol



```
.tran 0 10u 0 1u
```

Figure 13: 3-to-8 Row Decoder Test Circuit for verification

This circuit was built to verify the correct operation of a 3-to-8 row decoder. The inputs of the decoder are labeled A2, A1, and A0, and are driven by voltage sources V1, V2, and V3 respectively, each providing 1.8 V or 0V. According to the logic of a standard 3-input 8-output decoder, only one of the outputs (Z0 through Z7) is expected to be active (high) for each unique combination of input signals. For instance, when A2A1A0 = 000, only Z0 should be high, and when A2A1A0 = 111, only Z7 should be high. The decoder block is powered through the VDD pin, and the entire circuit is simulated in the time domain using a .tran command for 10 µs.

The simulation results of the test circuit are presented on the following pages, with a total of 8 cases. In addition, the simulation results were verified by referring to the truth table of the 3-to-8 row decoder, and it was observed that all outcomes were correct.

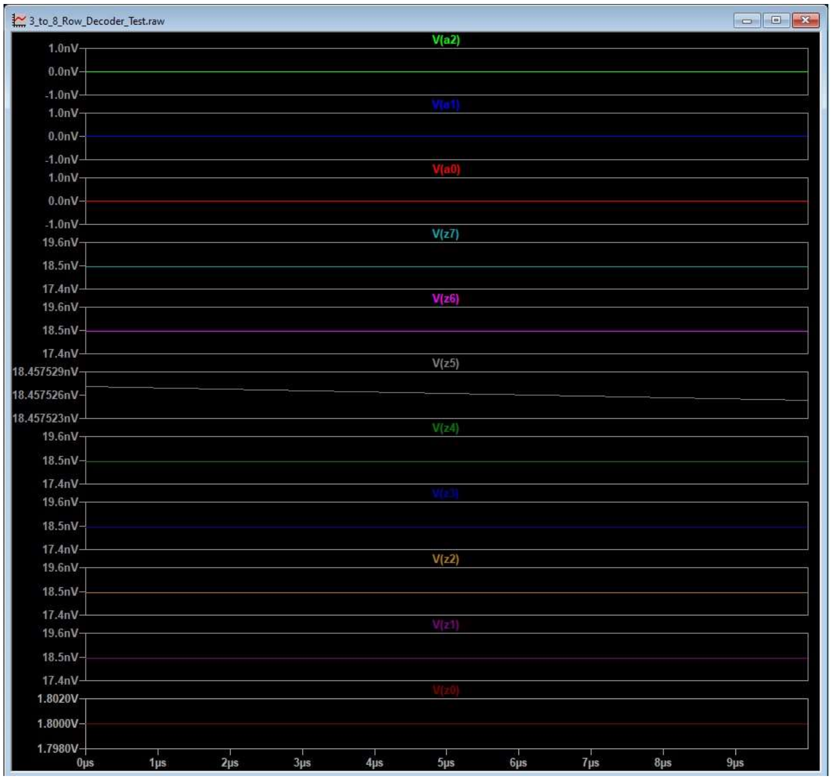


Figure 14: 3-to-8 Row Decoder (000) Result

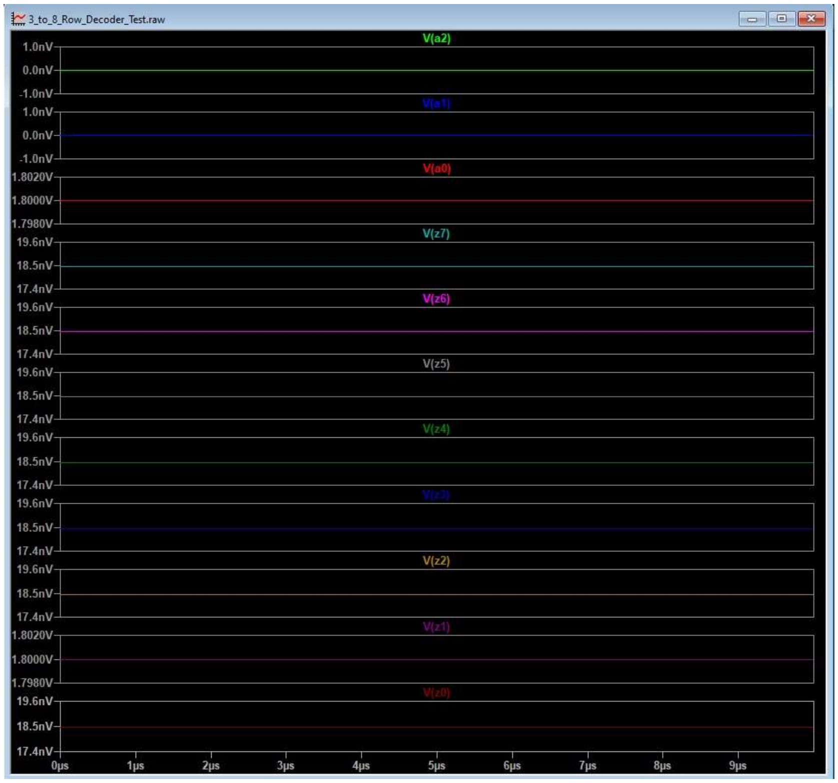


Figure 15: 3-to-8 Row Decoder (001) Result

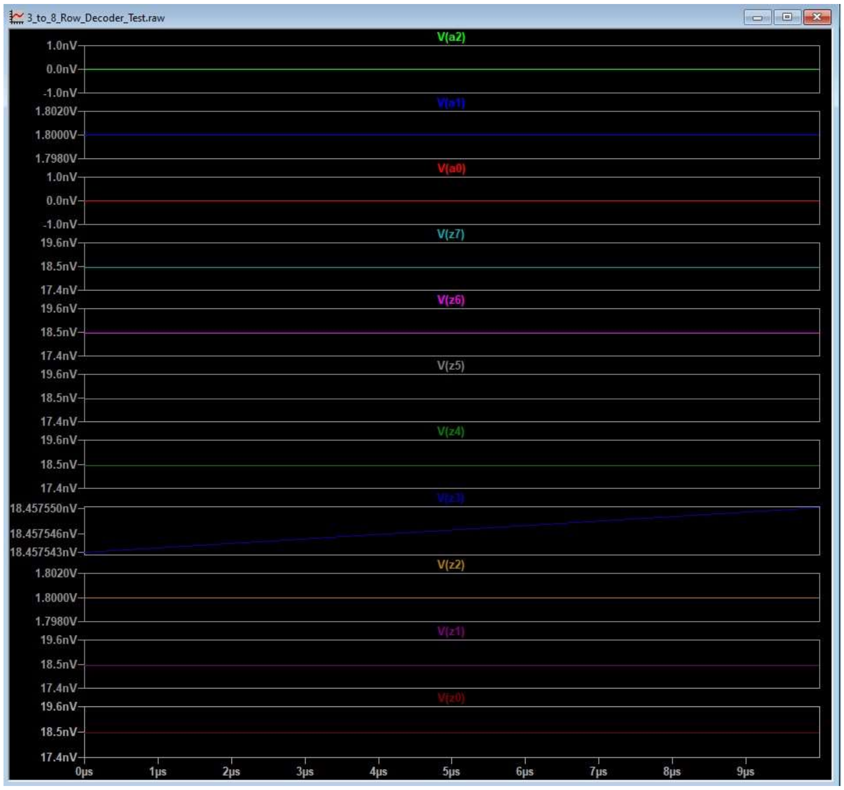


Figure 16: 3-to-8 Row Decoder (010) Result

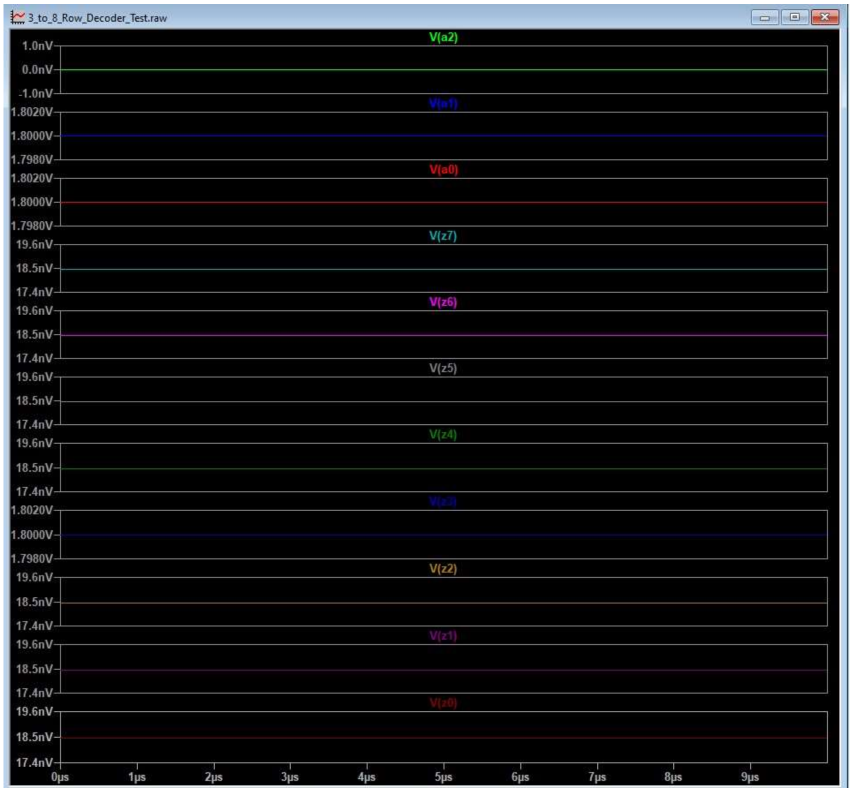


Figure 17: 3-to-8 Row Decoder (011) Result

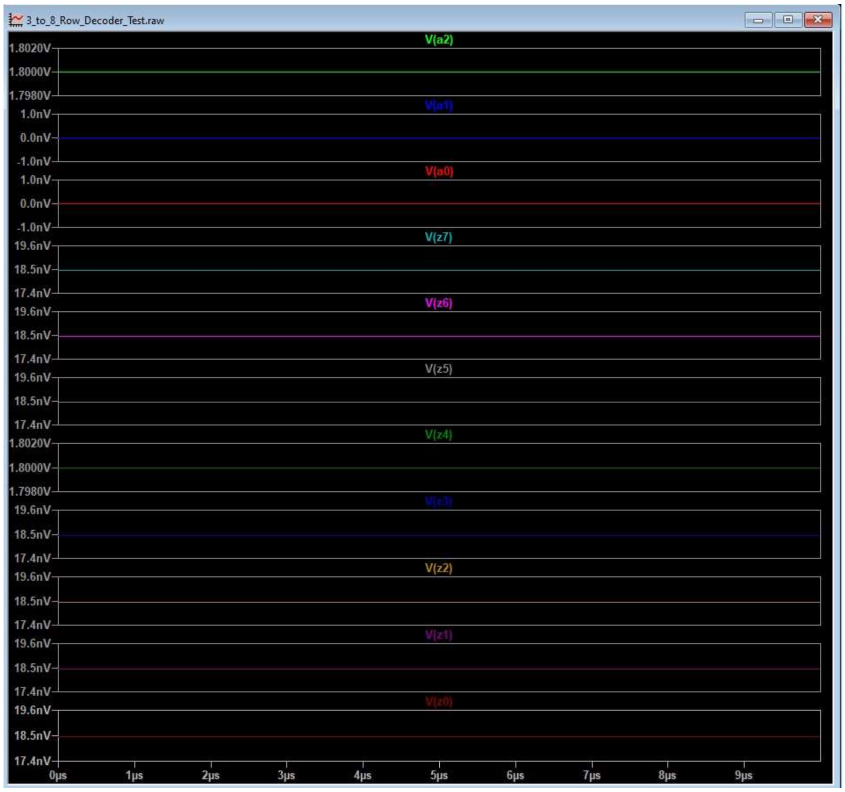


Figure 18: 3-to-8 Row Decoder (100) Result

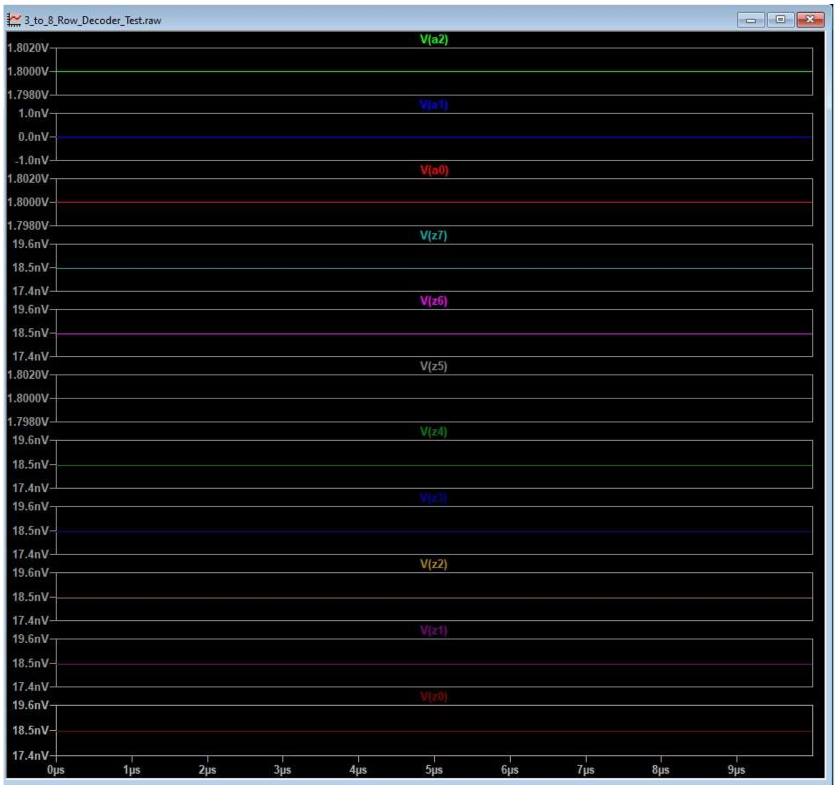


Figure 19: 3-to-8 Row Decoder (101) Result

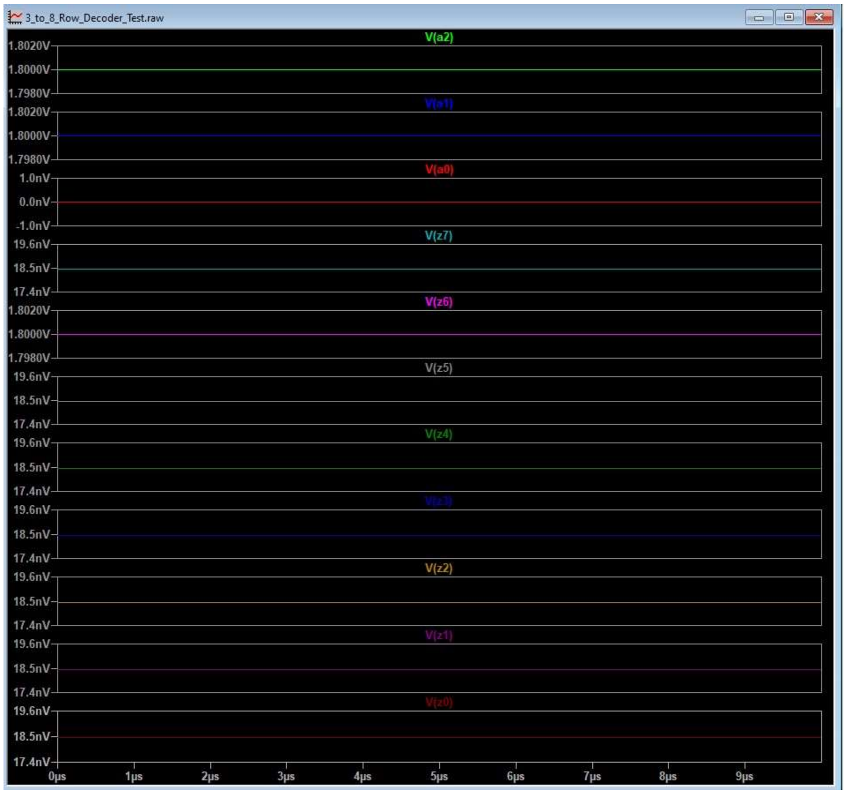


Figure 20: 3-to-8 Row Decoder (110) Result

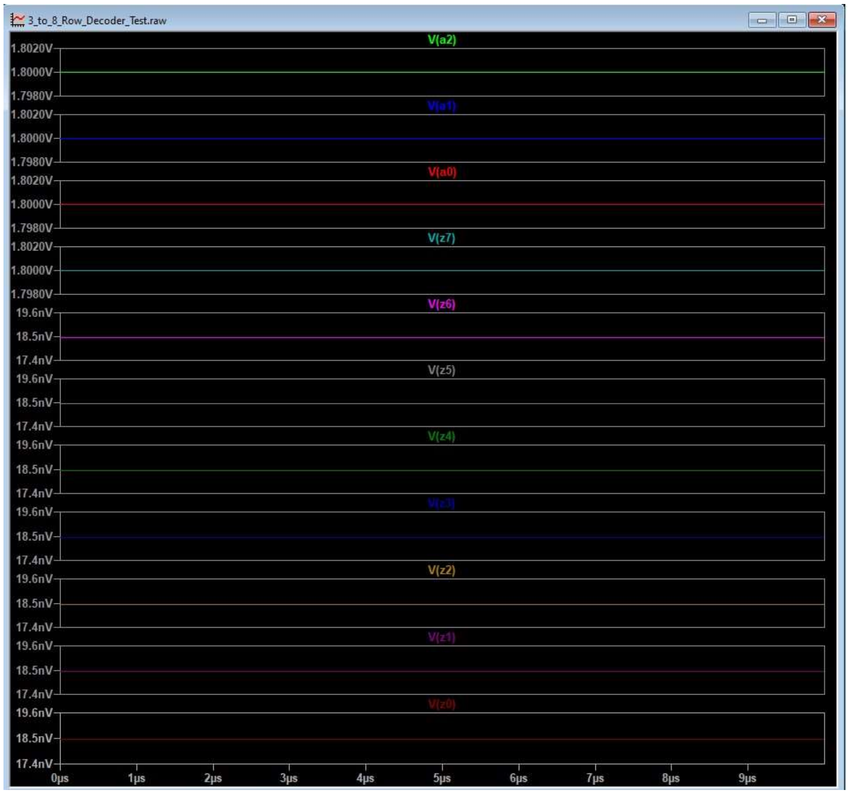


Figure 21: 3-to-8 Row Decoder (111) Result

- **Task 2.4. - 2.5. Construct the Full 8x8 Array / Integrate Column Circuitry**

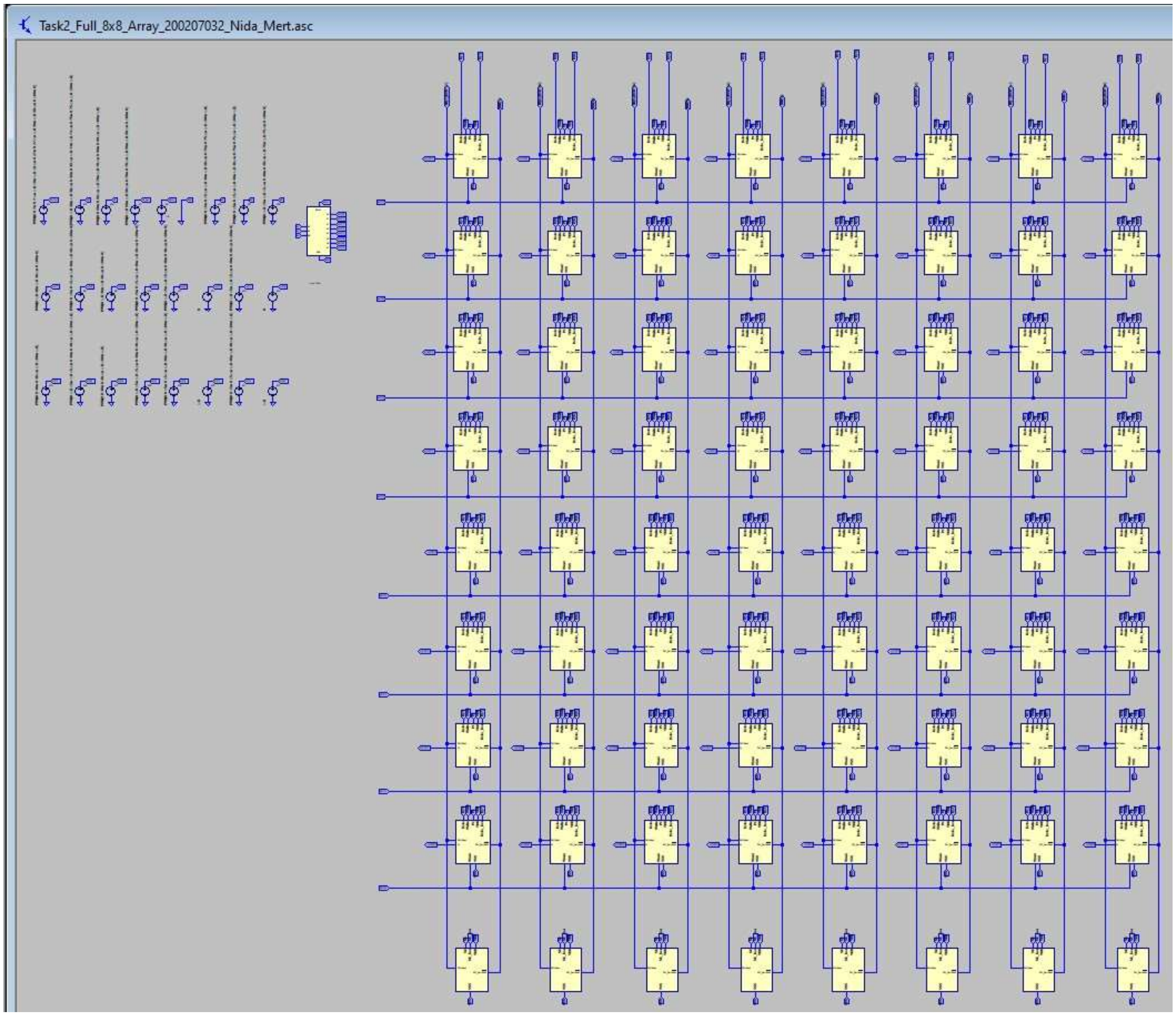


Figure 22: Task 2 Full 8x8 Array with All Connections

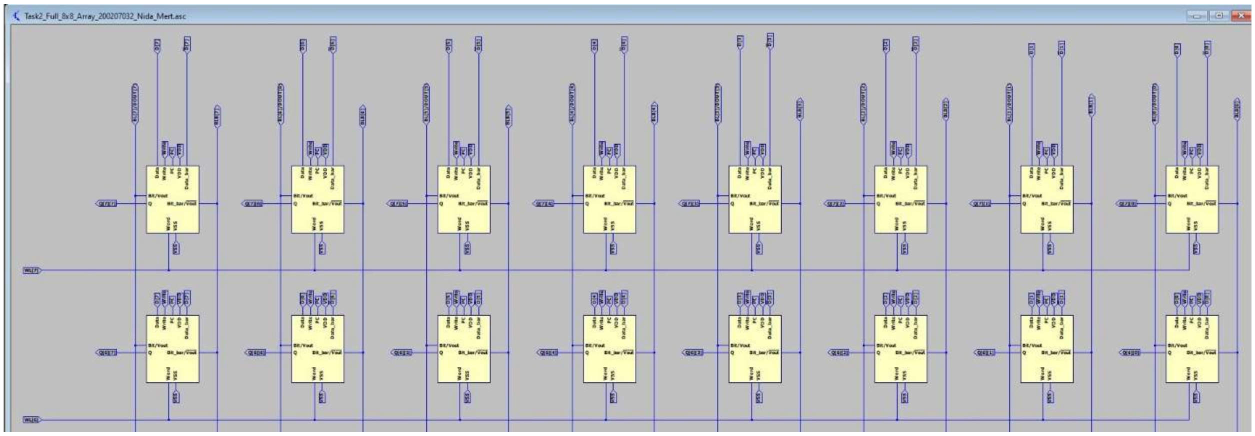


Figure 23: Full 8x8 Array’s top 2 lines zoomed screenshot

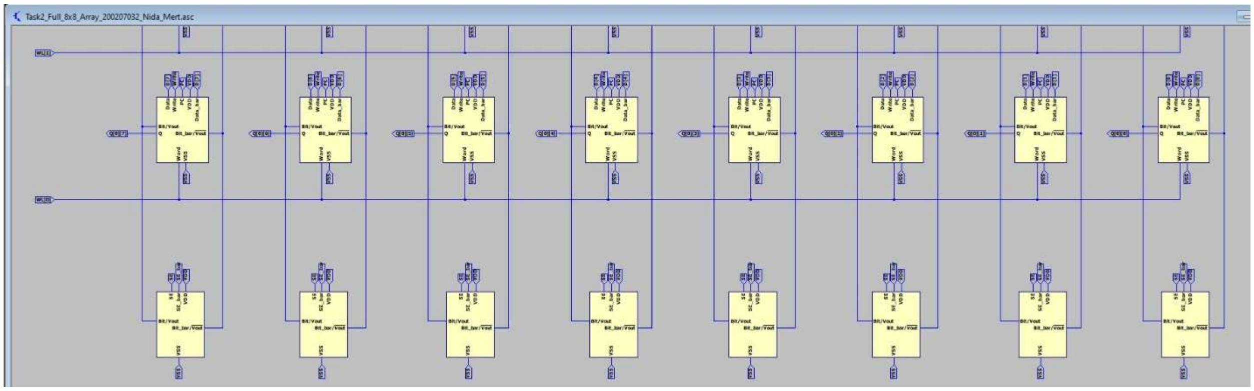


Figure 24: Full 8x8 Array’s bottom 2 lines zoomed screenshot

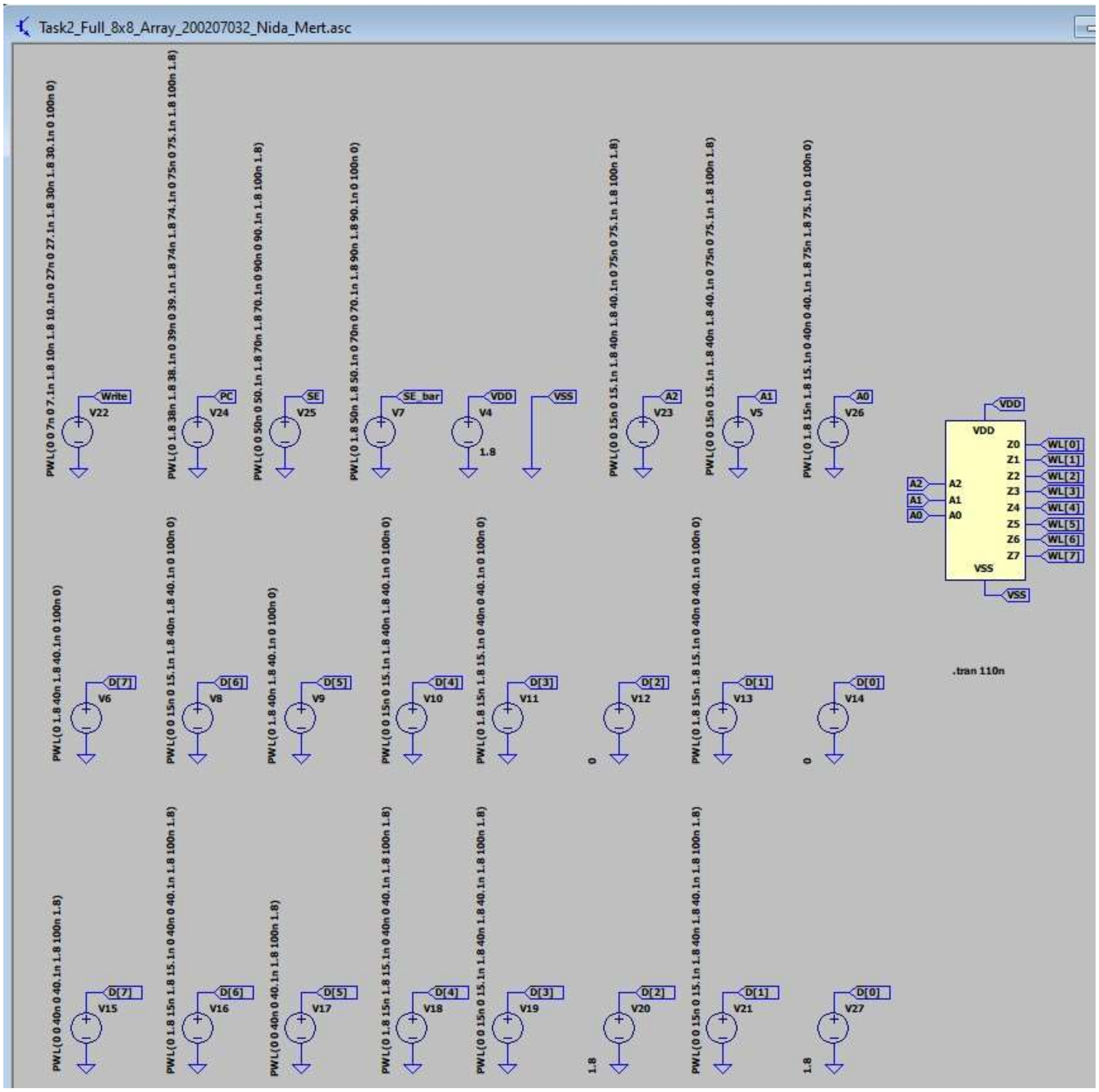

Task2_Full_8x8_Array_200207032_Nida_Mert.asc
PWL(0 0 7n 0 7.1n 1.8 10n 1.8 10.1n 0 27n 0 27.1n 1.8 30n 1.8 30.1n 0 100n 0)
Write
V22
PWL(0 1.8 38n 1.8 38.1n 0 39n 0 39.1n 1.8 74n 1.8 74.1n 0 75n 0 75.1n 1.8 100n 1.8)
PC
V24
PWL(0 0 50n 0 50.1n 1.8 70n 1.8 70.1n 0 90n 0 90.1n 1.8 100n 1.8)
SE
V25
PWL(0 1.8 50n 1.8 50.1n 0 70n 0 70.1n 1.8 90n 1.8 90.1n 0 100n 0)
SE_bar
V7
VDD
V4
1.8
VSS
PWL(0 0 15n 0 15.1n 1.8 40n 1.8 40.1n 0 75n 0 75.1n 1.8 100n 1.8)
A2
V23
PWL(0 0 15n 0 15.1n 1.8 40n 1.8 40.1n 0 75n 0 75.1n 1.8 100n 1.8)
A1
V5
PWL(0 1.8 15n 1.8 15.1n 0 40n 0 40.1n 1.8 75n 1.8 75.1n 0 100n 0)
A0
V26
VDD
VDD
A2
A1
A0
A2
A1
A0
Z0
Z1
Z2
Z3
Z4
Z5
Z6
Z7
WL[0]
WL[1]
WL[2]
WL[3]
WL[4]
WL[5]
WL[6]
WL[7]
VSS
VSS
PWL(0 1.8 40n 1.8 40.1n 0 100n 0)
D[7]
V6
PWL(0 0 15n 0 15.1n 1.8 40n 1.8 40.1n 0 100n 0)
D[6]
V8
PWL(0 1.8 40n 1.8 40.1n 0 100n 0)
D[5]
V9
PWL(0 0 15n 0 15.1n 1.8 40n 1.8 40.1n 0 100n 0)
D[4]
V10
PWL(0 1.8 15n 1.8 15.1n 0 40n 0 40.1n 0 100n 0)
D[3]
V11
0
D[2]
V12
PWL(0 1.8 15n 1.8 15.1n 0 40n 0 40.1n 0 100n 0)
D[1]
V13
0
D[0]
V14
.tran 110n
PWL(0 0 40n 0 40.1n 1.8 100n 1.8)
D[7]
V15
PWL(0 1.8 15n 1.8 15.1n 0 40n 0 40.1n 1.8 100n 1.8)
D[6]
V16
PWL(0 0 40n 0 40.1n 1.8 100n 1.8)
D[5]
V17
PWL(0 1.8 15n 1.8 15.1n 0 40n 0 40.1n 1.8 100n 1.8)
D[4]
V18
PWL(0 0 15n 0 15.1n 1.8 40n 1.8 40.1n 1.8 100n 1.8)
D[3]
V19
1.8
D[2]
V20
PWL(0 0 15n 0 15.1n 1.8 40n 1.8 40.1n 1.8 100n 1.8)
D[1]
V21
1.8
D[0]
V27

Figure 25: Sources and Row Decoder zoomed screenshot

The full 8x8 SRAM array circuit was constructed as shown in Figure 22. Due to the large size of the circuit, certain regions were zoomed in and captured separately as screenshots from Figure 23 to Figure 25, in order to provide a clearer view of the components and their interconnections. The circuit was designed in accordance with the structure specified in Tasks 2.4 and 2.5. The sources for D[i] (Data), _D[i] (Data_bar), Write, PC (Pre-Charge), SE (Sense Amplifier Enable), _SE, A2, A1, and A0 were configured using PWL in order to realize the simulation defined in Task 2.6. The simulation results will be examined in the following section.

- **Task 2.6. System Simulation**

The configuration of the sources to enable writing the values 10101010 and 11110000 to addresses 1 and 6 respectively, and subsequently reading them in order, (Figure 26 to 29).

Test 1: (Write data 10101010 to address 001)

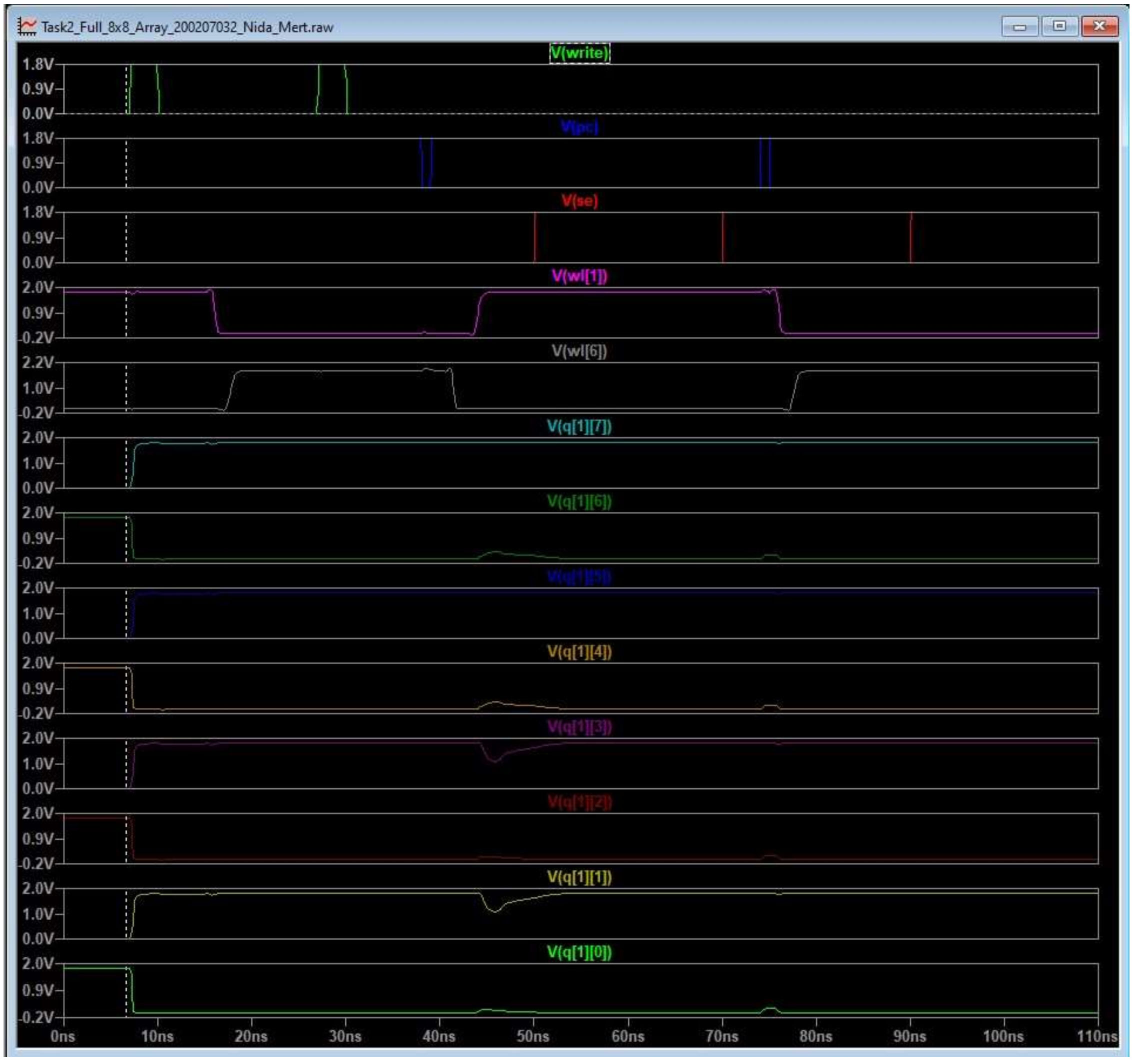


Figure 26: Write Data 10101010 to address 001

The simulation result for the "Write data 10101010 to address 001" test is shown in Figure 26. The simulation waveform includes, in order: Write, PC, SE, WL[1], WL[6], and Q[1][7] to Q[1][0] signals. The point at which the first write operation occurs is marked with a cursor. From that point until the end of the simulation, the value written to address 1 remained unchanged, indicating that the first write operation was successfully completed.

Test 2: (Write data 11110000 to address 110)

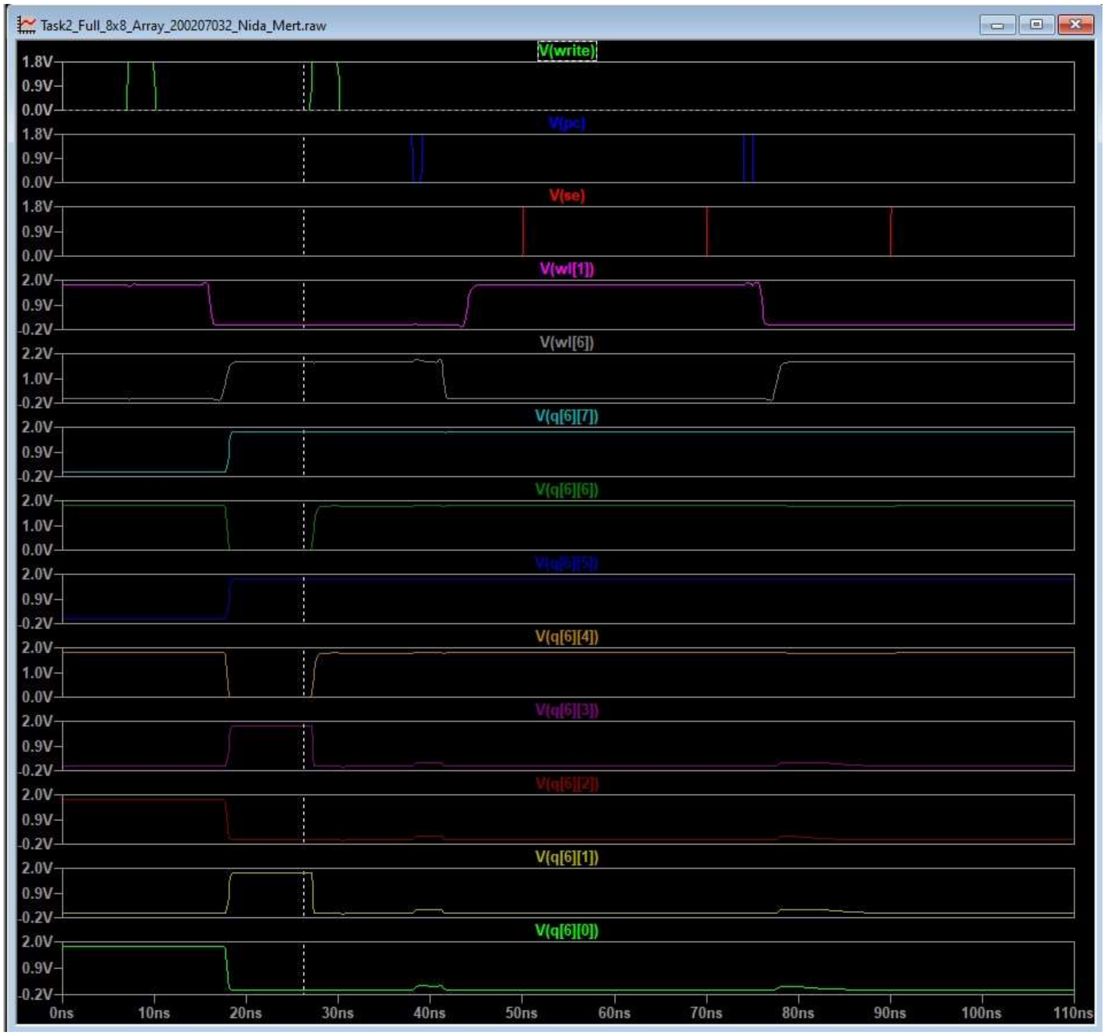


Figure 27: Write data 11110000 to address 110

The simulation result for the " Write data 11110000 to address 110" test is shown in Figure 27. The simulation waveform includes, in order: Write, PC, SE, WL[1], WL[6], and Q[6][7] to Q[6][0] signals. The point at which the second write operation occurs is marked with a cursor. From that point until the end of the simulation, the value written to address 6 remained unchanged, indicating that the second write operation was successfully completed.

Test 3: (Read data from address 001 and verify the output is 10101010)

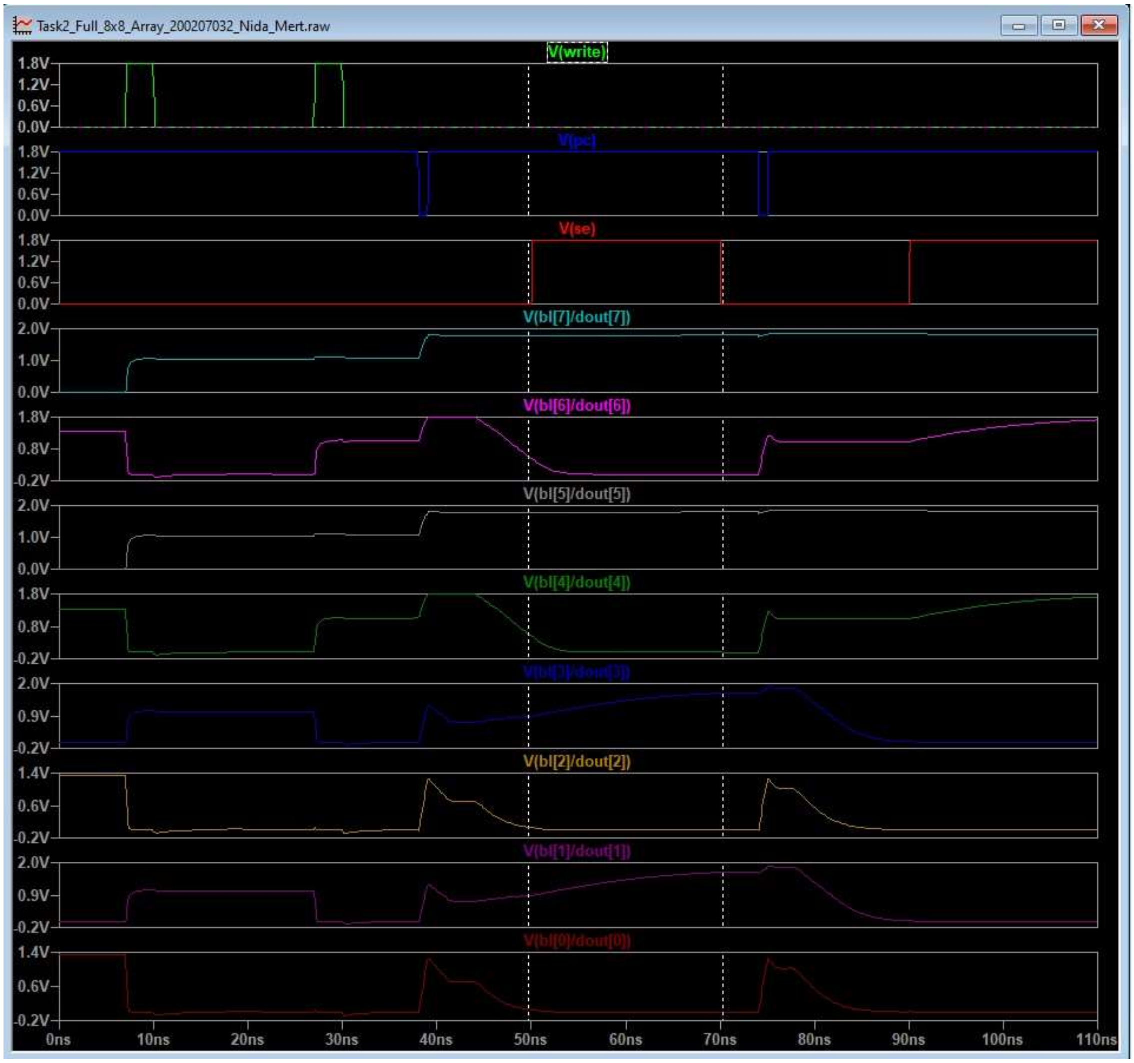


Figure 28: Read data from address 001 and verify the output is 10101010

The simulation result for the "Read data from address 001 and verify the output is 10101010" test is shown in Figure 28. The simulation waveform includes, in order: Write, PC, SE, and BL[7]/DOUT[7] to BL[0]/DOUT[0] signals. The read operation was performed during the interval when the first time SE signal was active, which is marked by two cursors. When examining the output of BL[7]/DOUT[7] to BL[0]/DOUT[0] between the two cursors, it is observed that the value 10101010 (written to address 1 in the first test step) was correctly read.

Test 4: (Read data from address 110 and verify the output is 11110000)

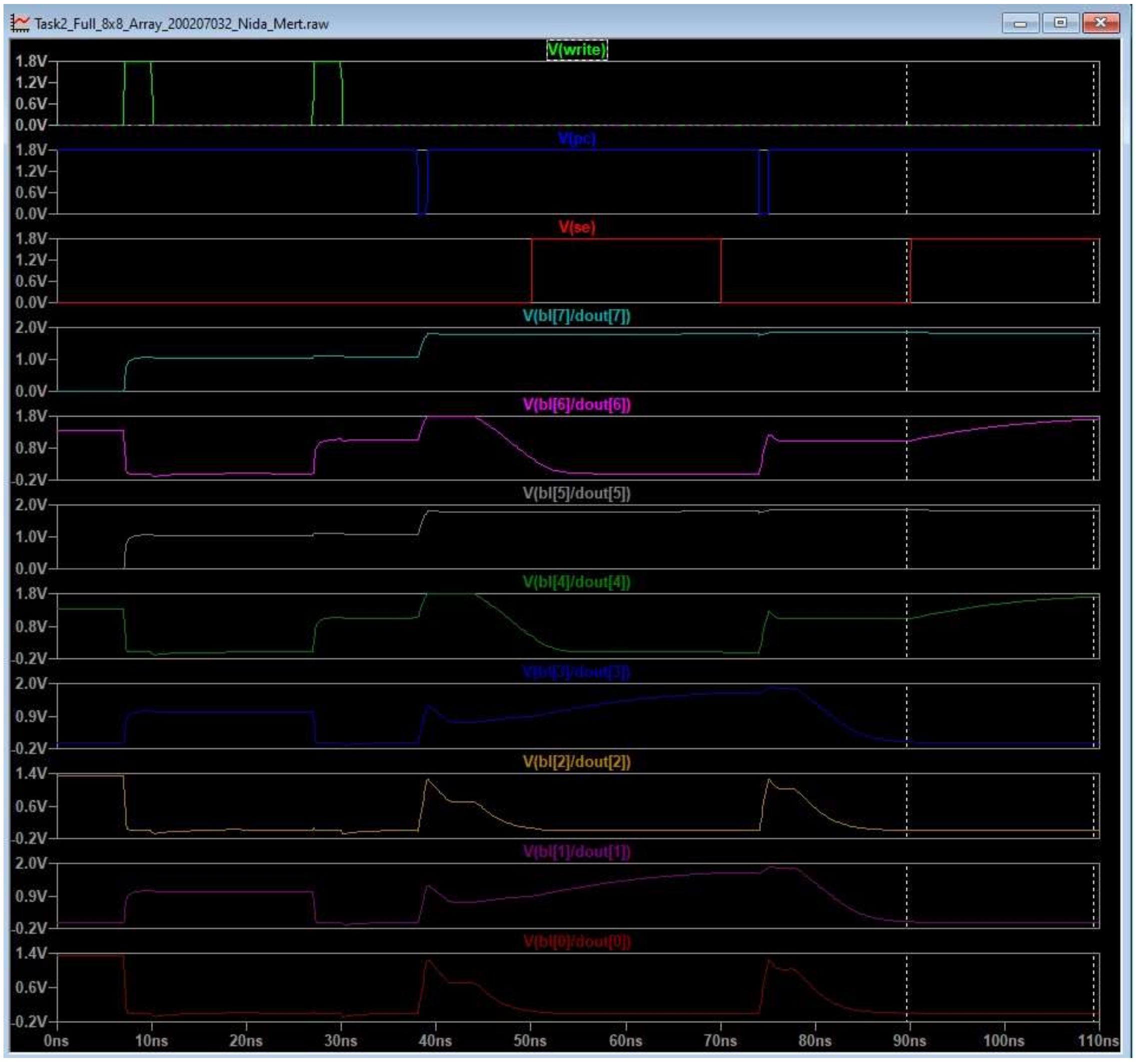


Figure 29: Read data from address 110 and verify the output is 11110000

The simulation result for the " Read data from address 110 and verify the output is 11110000" test is shown in Figure 29. The simulation waveform includes, in order: Write, PC, SE, and BL[7]/DOUT[7] to BL[0]/DOUT[0] signals. The read operation was performed during the interval when the second time SE signal was active, which is marked by two cursors. When examining the output of BL[7]/DOUT[7] to BL[0]/DOUT[0] between the two cursors, it is observed that the value 11110000 (written to address 6 in the second test step) was correctly read.

Task 3: Full Array Read/Write Cycle Simulation

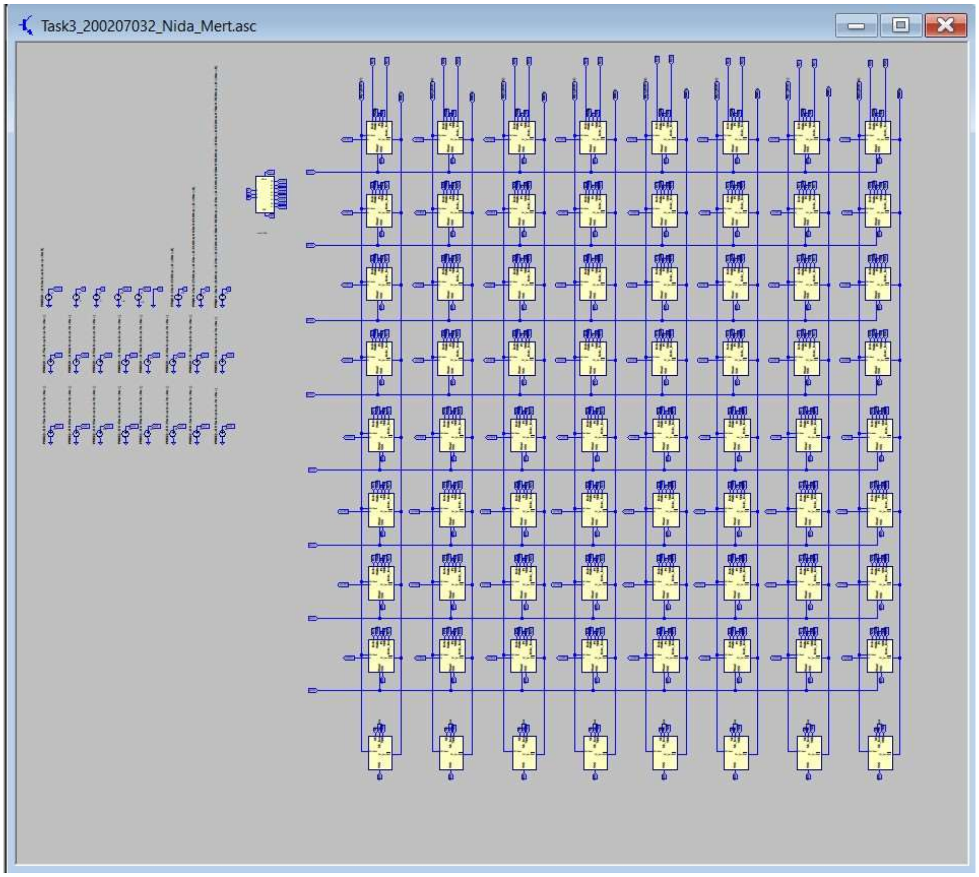


Figure 30: Task 3 Full 8x8 SRAM Array Circuit

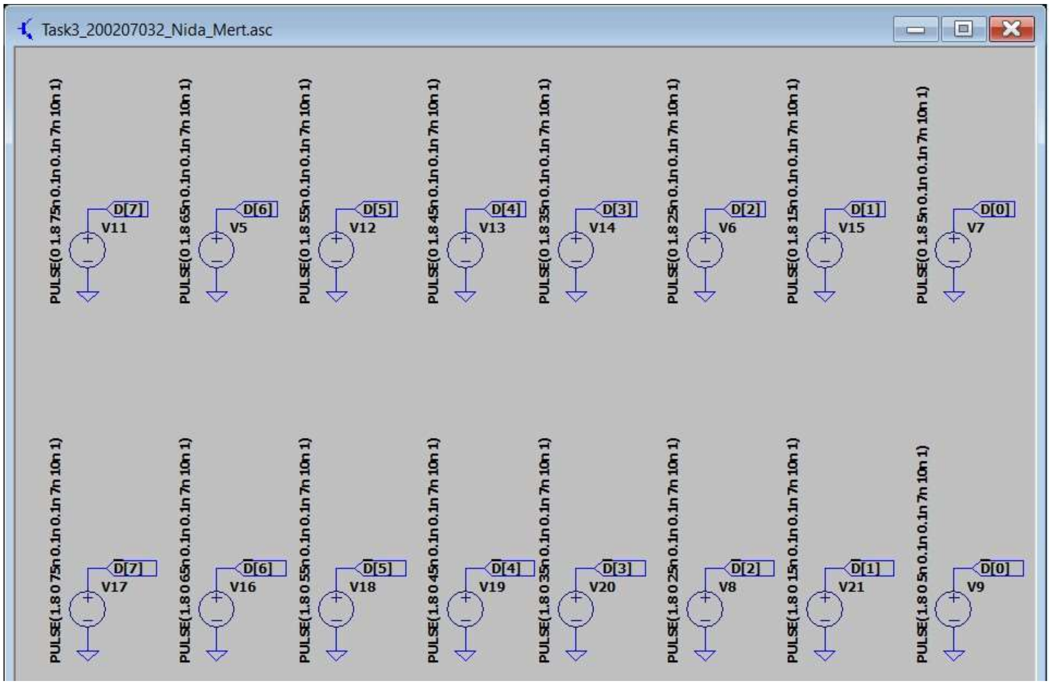
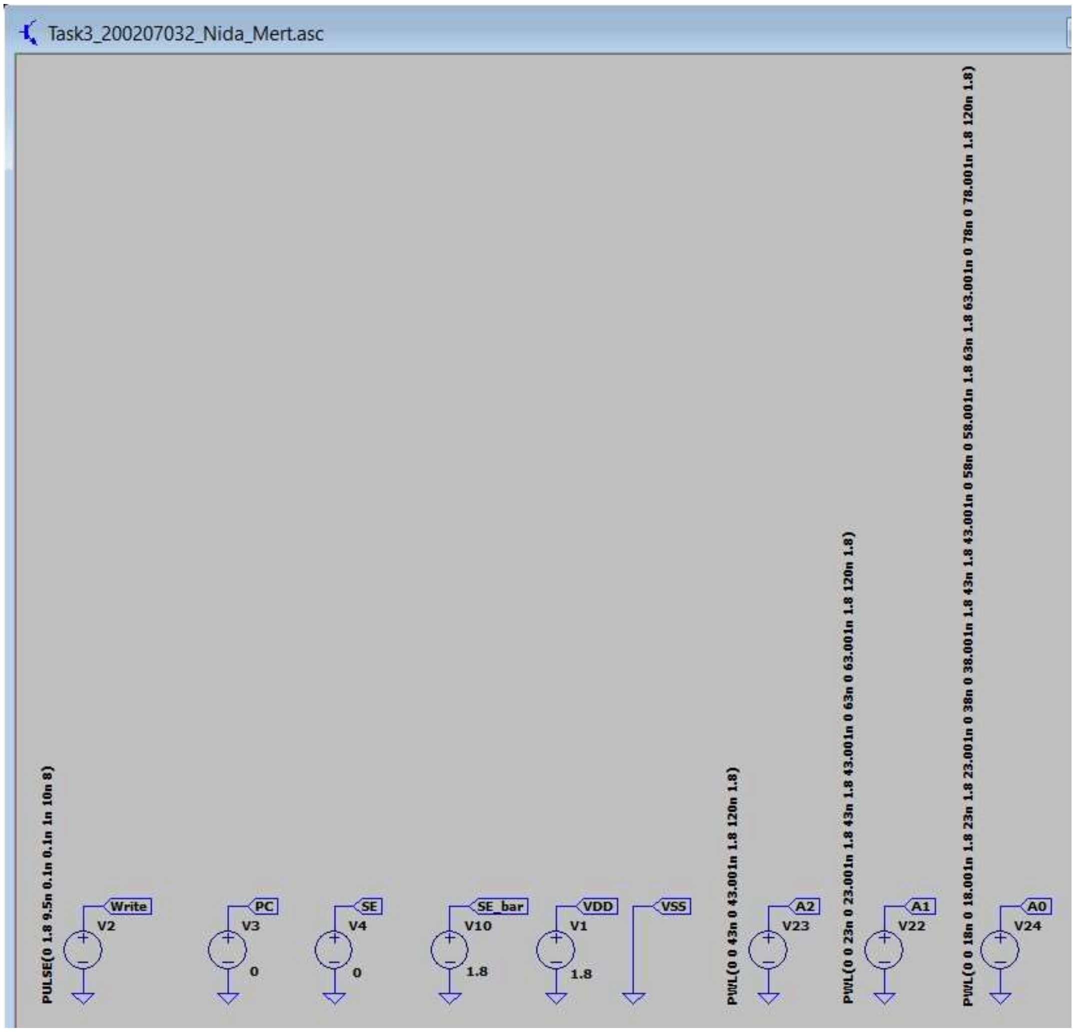

Task3_200207032_Nida_Mert.asc
PULSE(0 1.8 75n 0.1n 0.1n 7n 10n 1)
D[7]
V11
PULSE(0 1.8 65n 0.1n 0.1n 7n 10n 1)
D[6]
V5
PULSE(0 1.8 55n 0.1n 0.1n 7n 10n 1)
D[5]
V12
PULSE(0 1.8 45n 0.1n 0.1n 7n 10n 1)
D[4]
V13
PULSE(0 1.8 35n 0.1n 0.1n 7n 10n 1)
D[3]
V14
PULSE(0 1.8 25n 0.1n 0.1n 7n 10n 1)
D[2]
V6
PULSE(0 1.8 15n 0.1n 0.1n 7n 10n 1)
D[1]
V15
PULSE(0 1.8 5n 0.1n 0.1n 7n 10n 1)
D[0]
V7
PULSE(1.8 0 75n 0.1n 0.1n 7n 10n 1)
$\overline{D}[7]$
V17
PULSE(1.8 0 65n 0.1n 0.1n 7n 10n 1)
$\overline{D}[6]$
V16
PULSE(1.8 0 55n 0.1n 0.1n 7n 10n 1)
$\overline{D}[5]$
V18
PULSE(1.8 0 45n 0.1n 0.1n 7n 10n 1)
$\overline{D}[4]$
V19
PULSE(1.8 0 35n 0.1n 0.1n 7n 10n 1)
$\overline{D}[3]$
V20
PULSE(1.8 0 25n 0.1n 0.1n 7n 10n 1)
$\overline{D}[2]$
V8
PULSE(1.8 0 15n 0.1n 0.1n 7n 10n 1)
$\overline{D}[1]$
V21
PULSE(1.8 0 5n 0.1n 0.1n 7n 10n 1)
$\overline{D}[0]$
V9

Figure 31: Data(D[i]) and Data_bar(_D[i]) Signals Configuration


Task3_200207032_Nida_Mert.asc

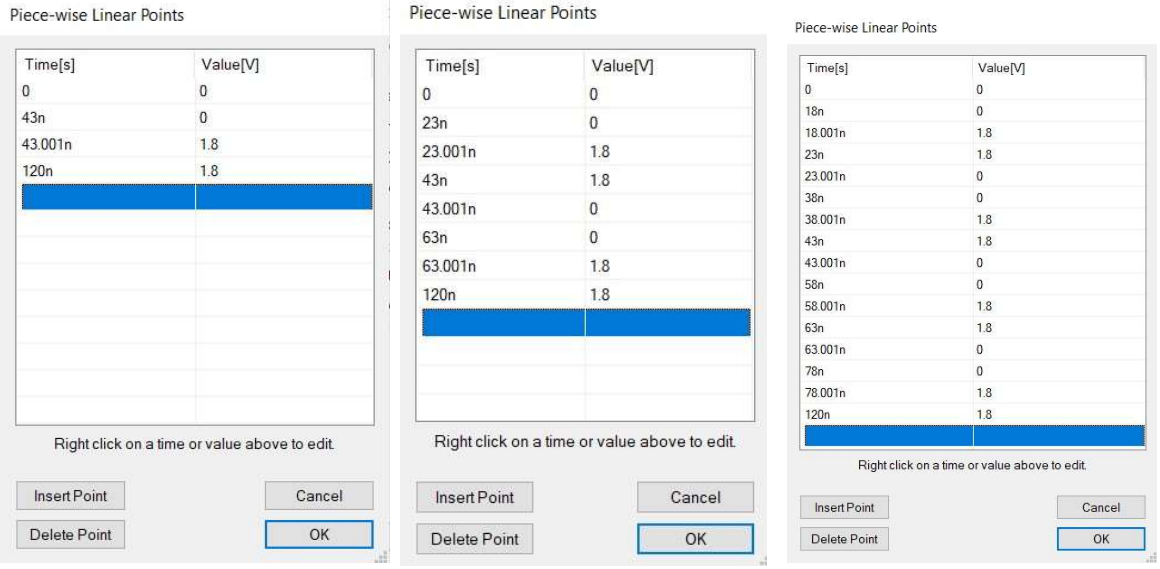
PULSE(0 1.8 9.5n 0.1n 0.1n 1n 10n 8)

PWL(0 0 43n 0 43.001n 1.8 120n 1.8)

PWL(0 0 23n 0 23.001n 1.8 43n 1.8 43.001n 0 63n 0 63.001n 1.8 120n 1.8)

PWL(0 0 18n 0 18.001n 1.8 23n 1.8 23.001n 0 38n 0 38.001n 1.8 43n 1.8 43.001n 0 58n 0 58.001n 1.8 63n 1.8 63.001n 0 78n 0 78.001n 1.8 120n 1.8)

Figure 32: A0-to-A2 input signals, Write signal and other signal configurations


Piece-wise Linear Points

Time[s]	Value[V]
0	0
43n	0
43.001n	1.8
120n	1.8

Right click on a time or value above to edit.

Piece-wise Linear Points

Time[s]	Value[V]
0	0
23n	0
23.001n	1.8
43n	1.8
43.001n	0
63n	0
63.001n	1.8
120n	1.8

Right click on a time or value above to edit.

Piece-wise Linear Points

Time[s]	Value[V]
0	0
18n	0
18.001n	1.8
23n	1.8
23.001n	0
38n	0
38.001n	1.8
43n	1.8
43.001n	0
58n	0
58.001n	1.8
63n	1.8
63.001n	0
78n	0
78.001n	1.8
120n	1.8

Right click on a time or value above to edit.

Figure 33: PWL values for A2, A1 and A0 input signals

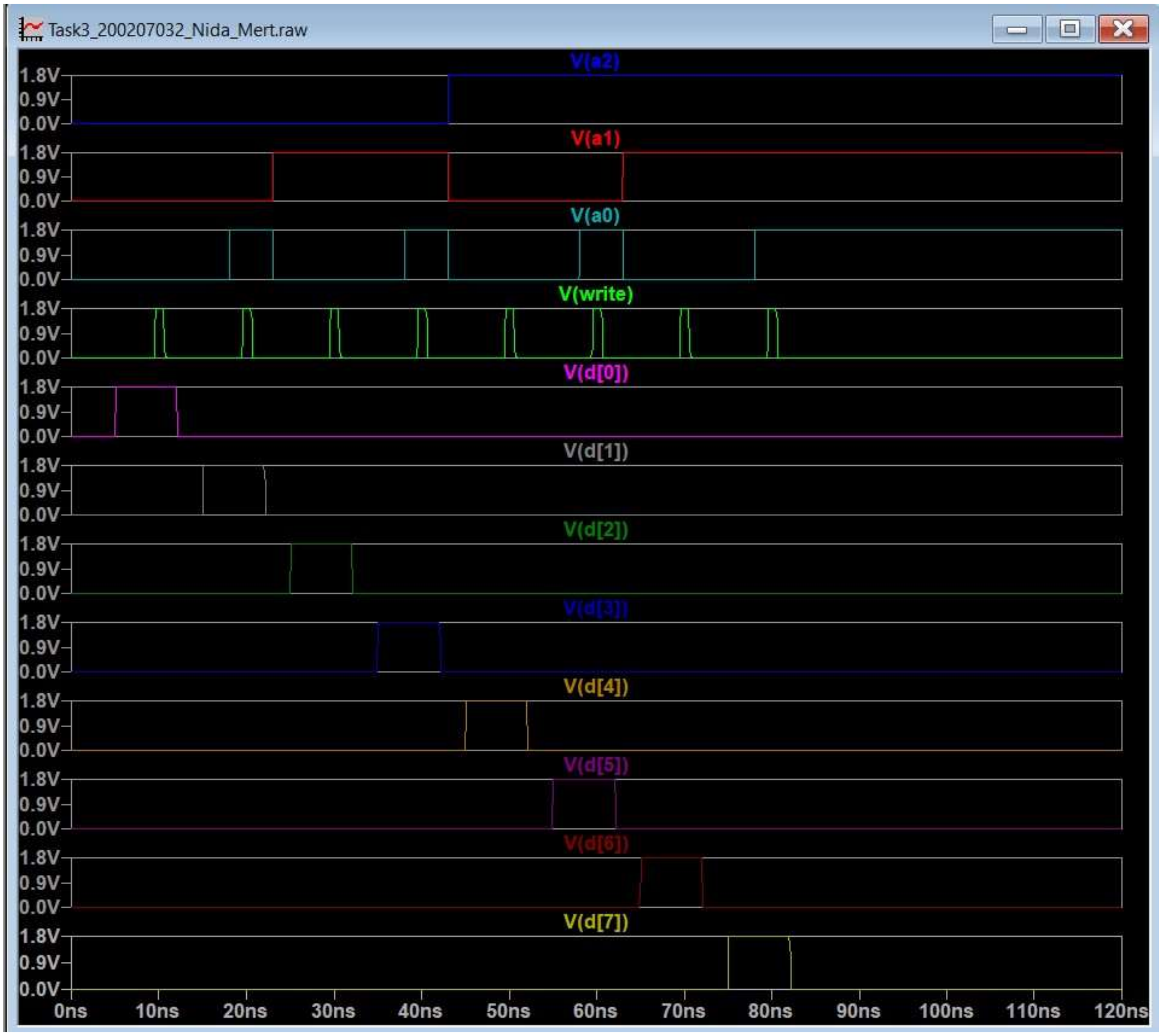


Figure 34: A2-to-A0 input signals, Write signal, Data(D[i]) Signals Simulation Result

Firstly, the simulation output obtained based on the values applied to the A2-to-A0 inputs, Write, Data (D[i]), and Data_bar (_D[i]) sources is shown in Figure 31 to 33. When examining the intersections of the A2, A1, and A0 signals with each Write signal pulse, it can be observed that the address signals were generated correctly. The Write signal was triggered every 10 ns up to 80 ns. After that, the simulation was extended to 120 ns in accordance with the assignment requirements, and no further write operations occurred beyond 80 ns, as the Write signal ends at that point. And it can be confirmed that everything is functioning in accordance with the specified requirements.

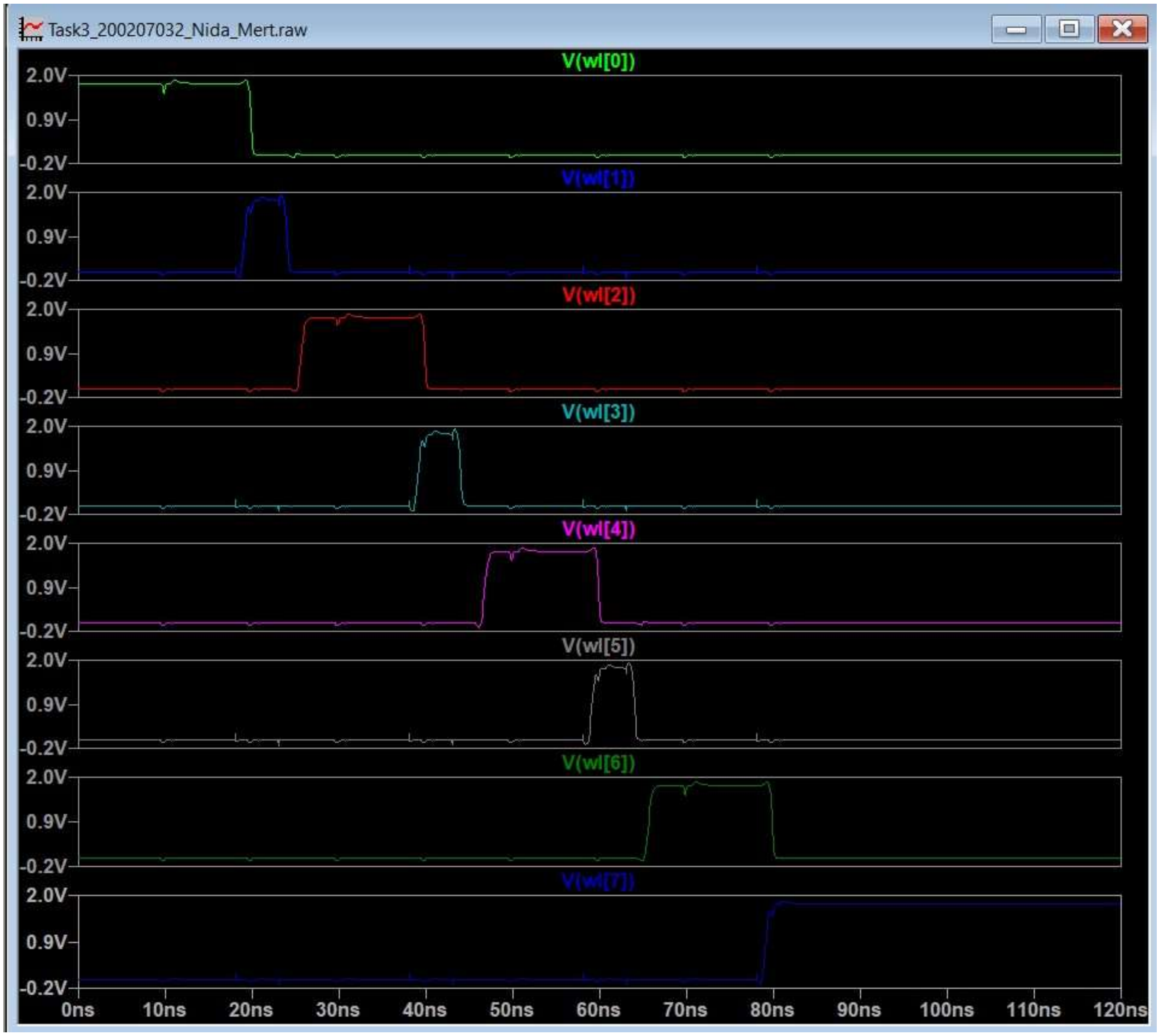


Figure 35: Word Line (WL[i]) Signals Simulation Results

According to the changes in the A2, A1, and A0 input signals, the active word line changed every 10 ns up to 80 ns. This is because, as specified in the assignment requirements, data is written to the next address every 10 ns. The resulting word line signals from this operation are shown in the simulation output in Figure 35, and it can be confirmed that everything is functioning in accordance with the specified requirements.

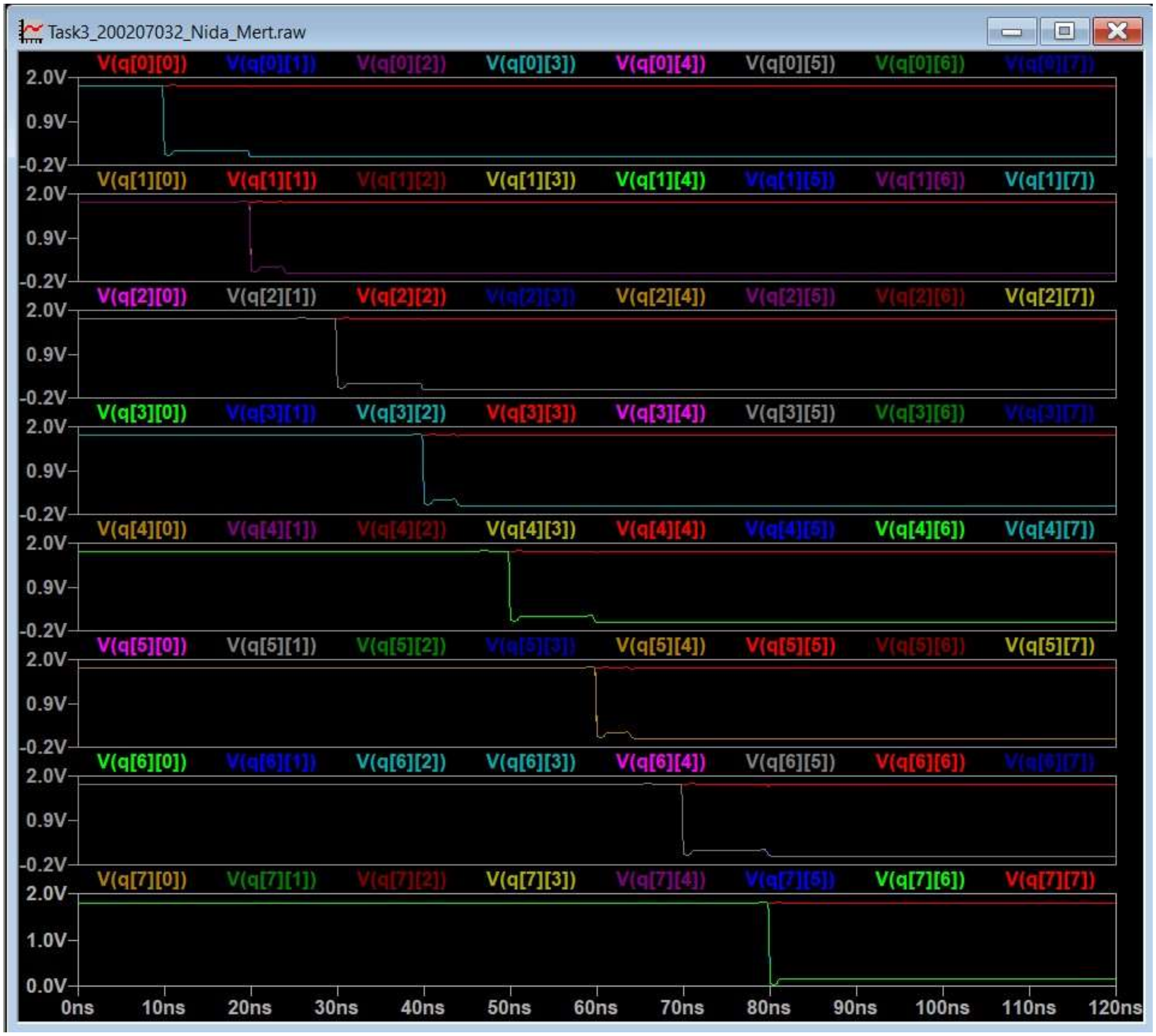


Figure 36: Q0-to-Q63 output signals simulation result

Time (@)	Write Data (MSB-left)	Adress
10ns	00000001	Row0 (000)
20ns	00000010	Row1 (001)
30ns	00000100	Row2 (010)
40ns	00001000	Row3 (011)
50ns	00010000	Row4 (100)
60ns	00100000	Row5 (101)
70ns	01000000	Row6 (110)
80ns	10000000	Row7 (111)

Figure 37: Final Project Testbench Table

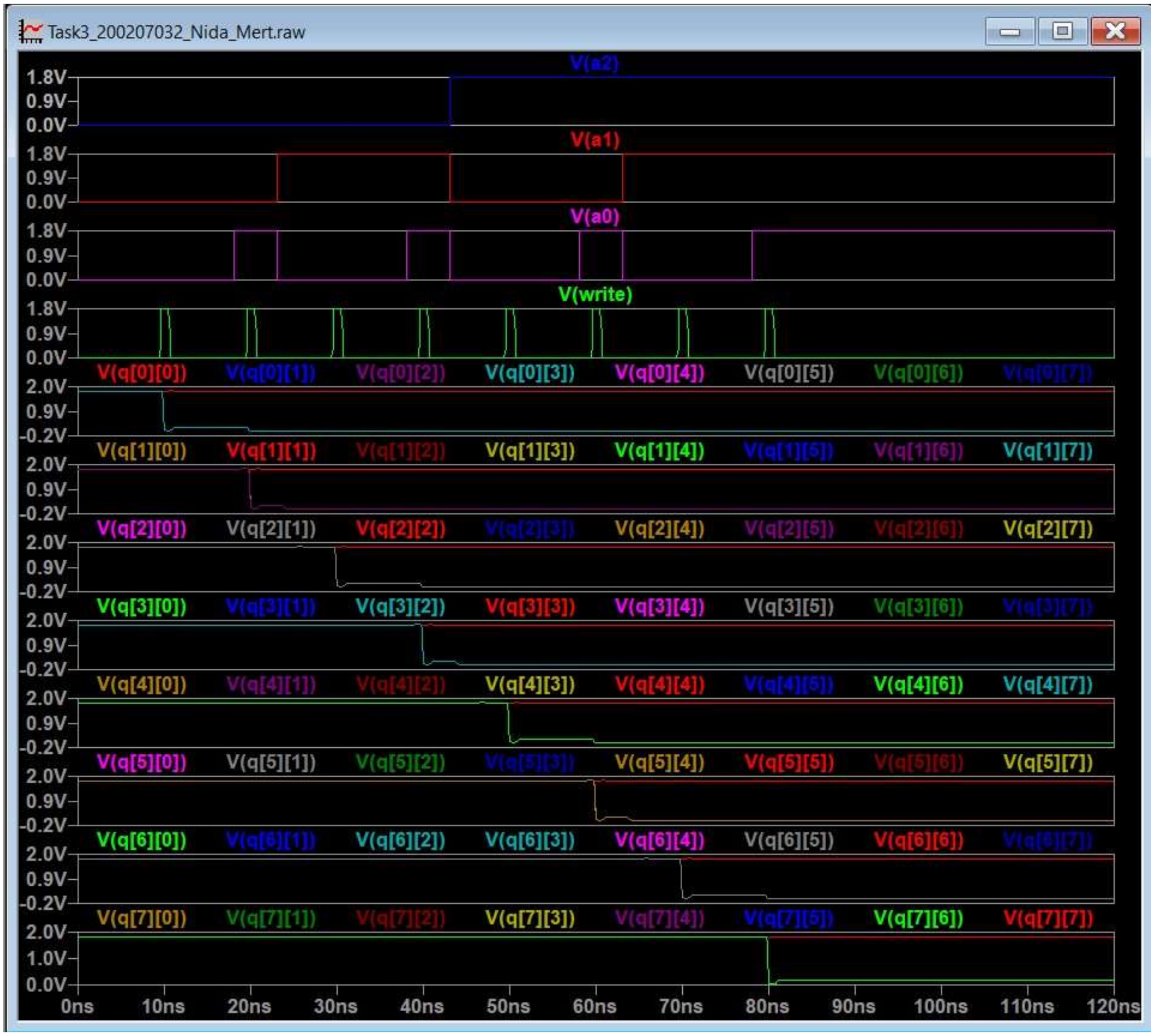


Figure 38: Q0-to-Q63 output signals with A2, A1, A0 and Write input signals simulation result

All Q signals were arranged as specified in the assignment requirements. As a result, the values required to be written to addresses from 000 to 111 were successfully written to each row in sequence. In other words, in each row, only the corresponding bit was set to 1 while all others remained 0. In the simulation, each write operation began at intervals of 10 ns, starting from 10 ns up to 80 ns. Before each write, the Q outputs held undefined (garbage) values. Once the write operation occurred, these undefined values were replaced with the correct data.

Consequently, after verifying the written values, it was observed that the simulation functioned successfully and the testbench was correctly implemented.