



EE457: Digital IC Design

Fall Semester 2019

Final Report Cover Sheet

Due: Monday 3:30PM – 5:45PM, 12/16/2019

PROJECT TITLE: 1K SRAM

Group Name: Team Hydras

Group Leader: MD MUYIN

Partner Name: SHAMIM BABUL

You Check for completion here	Topics	GRADES
Required	Section1: Executive Summary	
✓	Section 2: Introduction and Background	/5
✓	Section 3: Electric Circuit Schematics	/10
✓	Section 4: Detailed Electric Layouts	/25
✓	Section 5: IRSIM Logic Simulations and Measurements for Layout and Schematic (<u>must provide comparisons between the two</u>)	/10
✓	Section 6: LTSPICE code and parasitic extractions with calculation analysis. Put only samples of code.	/15
✓	Section 7: Measurements in LTSPICE for delays for Layout and Schematic (<u>must provide comparisons between the two</u>)	/15
✓	Section 8: Measurements of <u>power, delay, chip area, timing, number of transistors</u> for the layout. (If you are using TG, static or dynamic CMOS, compare here.)	/10
Required	Section 9: Conclusion and References	
Required	Presentation (Graded by students)	/10
	TOTAL	/100

Instructor Comments:

Table of Contents

Section 1: Executive Summary	3
Section 2: Introduction and Background	3
Section 3: Electric Circuit Schematics	5
Section 4: Detailed Electric Layouts	16
Section 5: IRSIM Logic Simulations and Measurements for Layout and Schematic with Comparison	29
Section 6: LTSPICE code and parasitic extractions with calculation analysis...	42
Section 7: Measurements in LTSPICE for delays for Layout and Schematic with comparison	45
Section 8: Measurements of power, delay, chip area, timing, number of transistors for the layout.	54
Section 9: Conclusion and References	56

Section1: Executive Summary

The objective of this project was to learn and design 1K SRAM (Static Random-Access Memory) using conventional CMOS gates. In 1K SRAM we have used 6 transistor memory cells. 1K SRAM contains 1024 cells of 6T SRAM, a total of 6,144 transistors. To store and read data in SRAM we have used decoder, mux/demux and sense amplifier. Decoder was made using Inverter and AND gates. First, we have created a 2 to 4 decoder then used that decoder to design a 5 to 32 decoder with enable that connects to the word lines of the 1K SRAM cell. For the bit line and bit_b line we have used mux/ demux. It was designed by using a 2 to 4 decoder connected to an 8 NMOS all together connected to bit lines and bit_b lines. Output of mux/demux is connected to the write drive and sense amplifier. Lastly, we have used pre charge to charge the bit lines and bit_b lines that is needed before the read operation.

To simulate the design, we have used the software called Electric and IRSIM. IRSIM gives us a complete overview of how the waveforms defer for schematic design and layout design. On the other hand, LTSPICE gives us the propagation delay of the schematic and layout of the circuit.

Section 2: Introduction and Background

Introduction:

Static Random-Access Memory (SRAM) is a type of RAM that holds data in a static form as long as the memory has power. The SRAM was invented at Fairchild Semiconductor in 1964 by an electrical engineer named John Schmidt. It has some key differences than the regular DRAM.

SRAM is a memory chip that is faster and requires less power than the dynamic memory, but it is volatile since it losses all the memory when there is no power connected. Though it provides a faster service for the users it is very expensive compared to DRAM. To design the 1K SRAM we have used 6 transistor memory cell, 2 to 4 decoder to make a 5 to 32 decoder, mux/demux and a sense amplifier. Below are some block diagrams of the designs we have used to create the 1K SRAM. 6T SRAM is designed by connecting to inverters facing each other along with two NMOS for bit and bit_b lines.

6T SRAM block diagram:

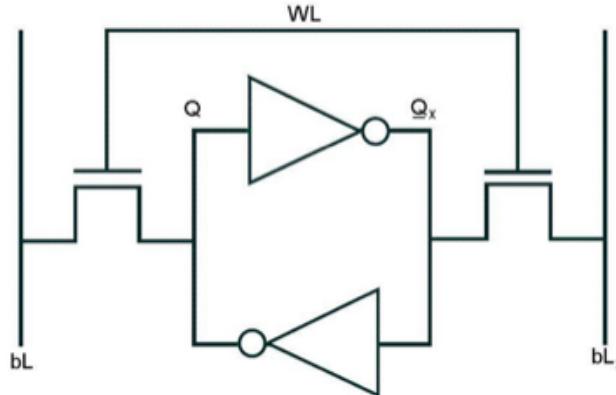


Figure 1: 6T SRAM block diagram

The 6-transistor cell is created using this block diagram. Below is a image of the SRAM cell we have designed.

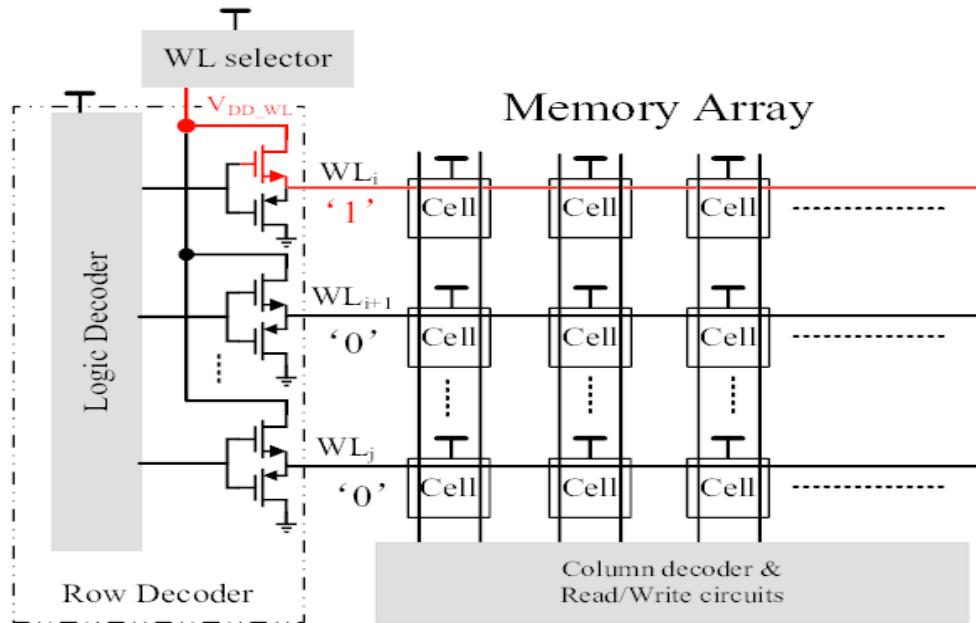


Figure 2: 1K SRAM block diagram

Background: We started our project by learning about the SRAM first. Since this is the first time both me and my partner are designing a SRAM, we had to do a lot of research to get enough idea about how SRAM works and what circuits we need to complete a full working design. We have included some references at the end of the report that we used for our research. After doing enough

research I have started building my circuit piece by piece. We needed to do a lot of trial and error. We had to take a lot of help from our classmates and Professor Kim to design the SRAM.

Section 3: Electric Circuit Schematics

1K SRAM is a combined design of 6T memory cells, 2 to 4 decoder, 5 to 32 decoder, mux/demux and sense amplifier. We have included all the schematic designs one by one below.

6T SRAM Schematic:

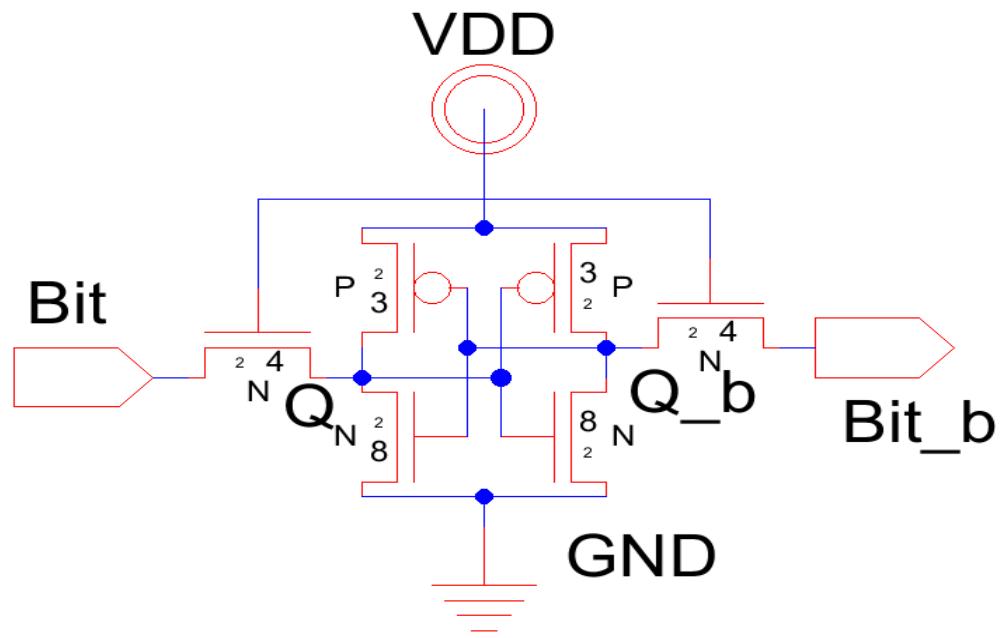


Figure 3: 6T SRAM Memory cell

```
=====
45
Checking schematic cell '6TSRAM{sch}'
  No errors found
  0 errors and 0 warnings found (took 0.004 secs)
=====
46=====
```

Figure 4: DRC check for 6T SRAM

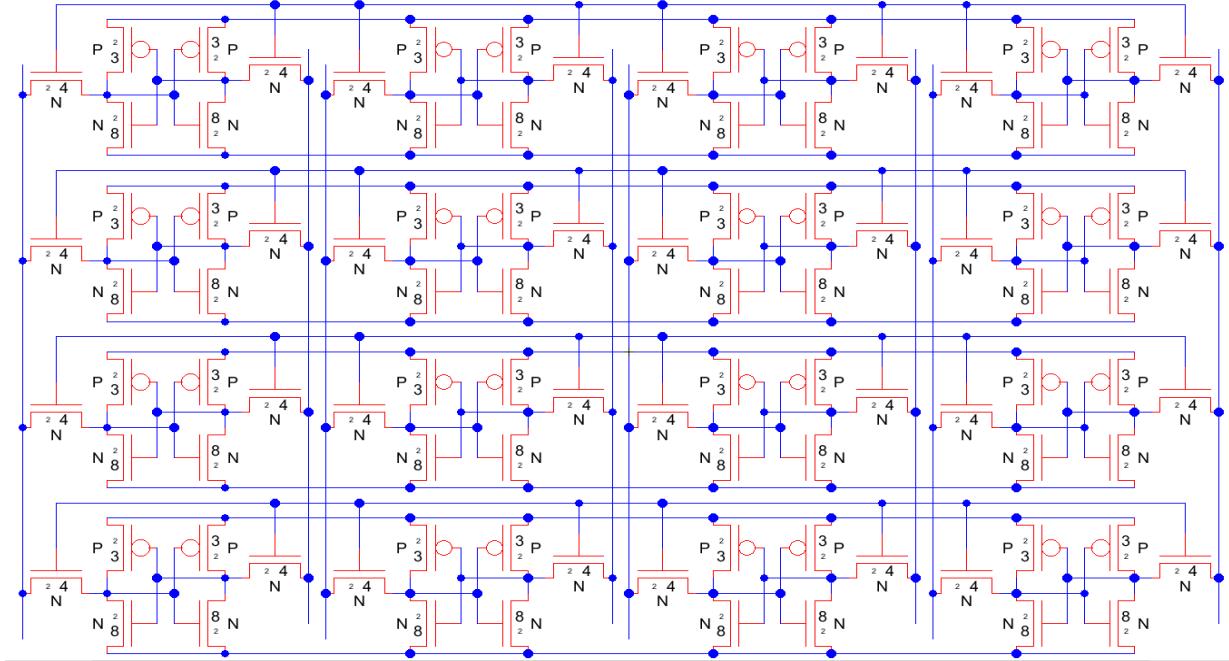


Figure 5 : 16 cells of SRAM schematic unlabeled



Figure 6 : 1024 (1K) cells of SRAM schematic unlabeled

2 to 4 Decoder:

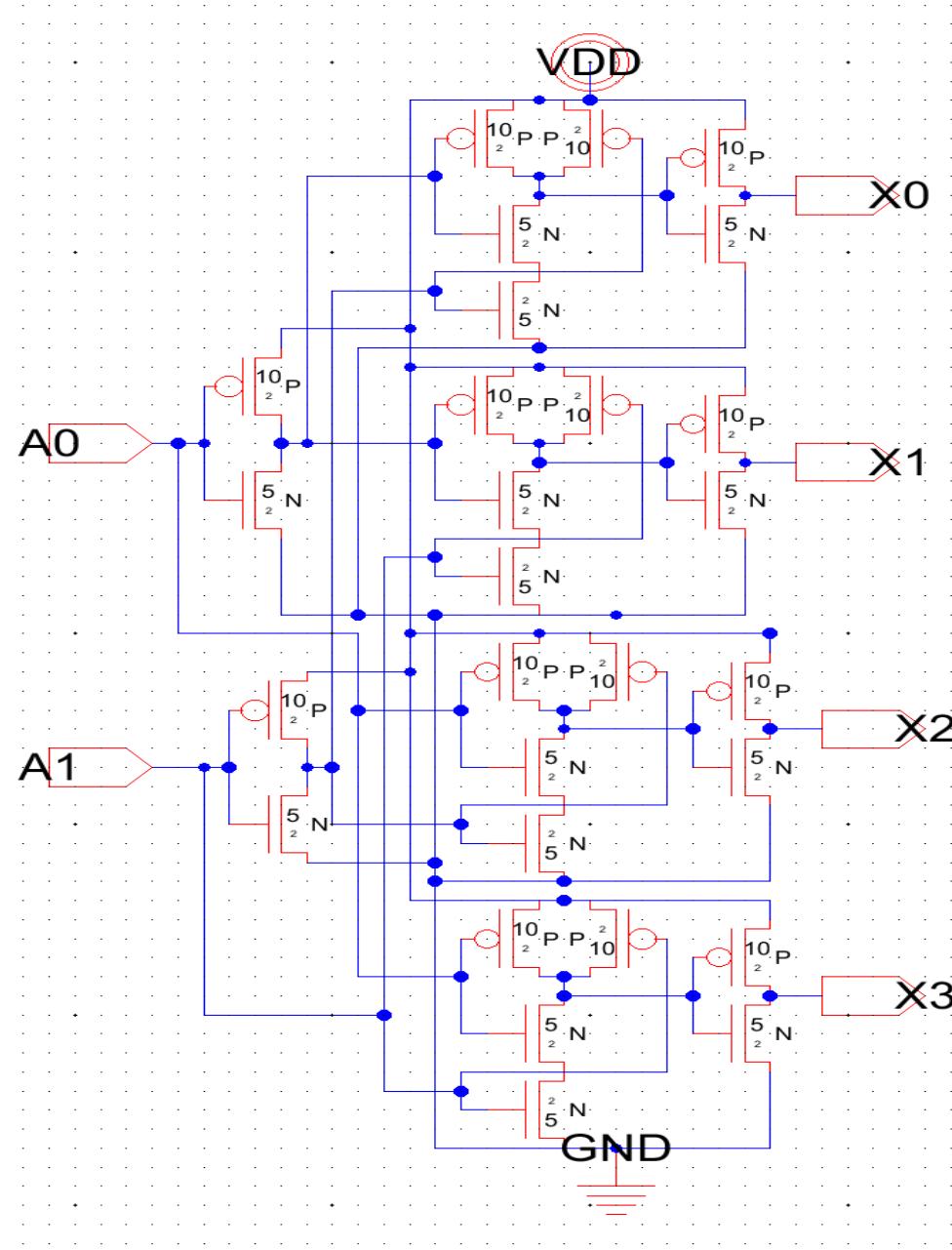


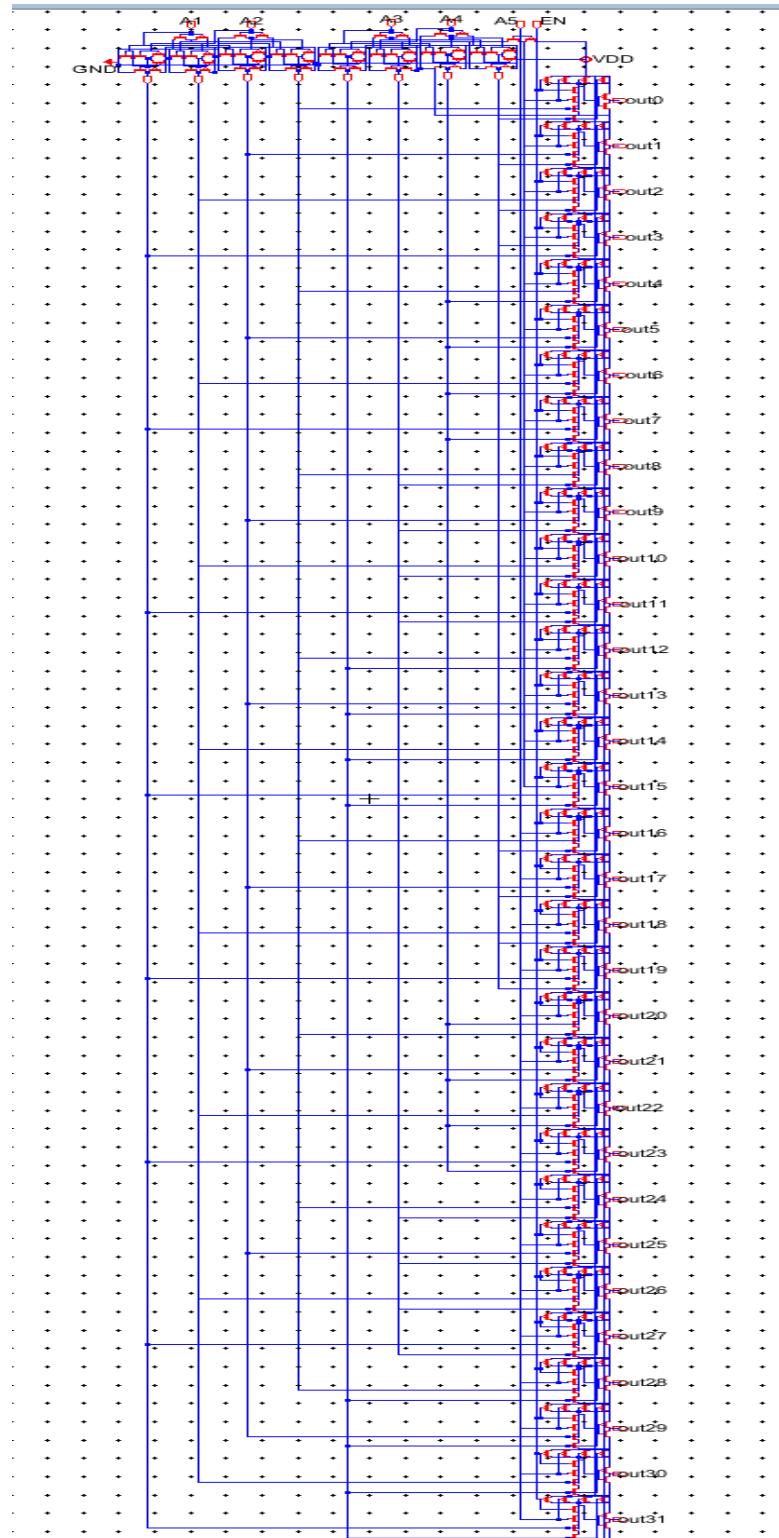
Figure 7: 2 to 4 Decoder schematic

```

Checking schematic cell '2to4decoder{sch}'
No errors found
0 errors and 0 warnings found (took 0.013 secs)

```

Figure 8: DRC check for 2 to 4 Decoder Schematic

5 to 32 Decoder:**Figure 9:** 5 to 32 Decoder Schematic

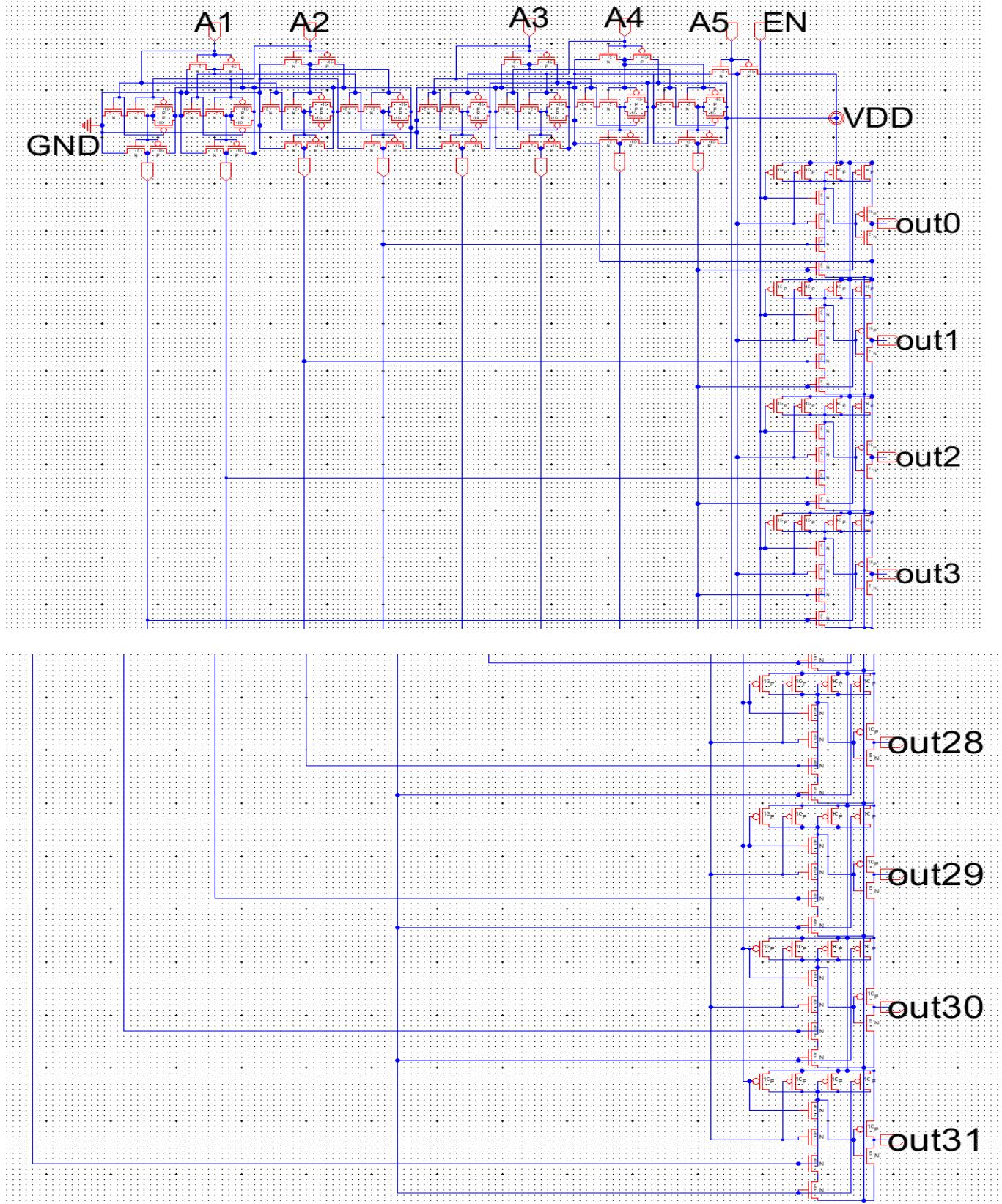


Figure 10: Zoom in 5 to 32 Decoder Schematic top 4 and bottom 4 output

```
Checking schematic cell 'SRAM_Fin-(1):5to32{sch}'  
    No errors found  
0 errors and 0 warnings found (took 0.039 secs)
```

Figure 11: DRC error check of 5 to 32 Decoder Schematic

Sense Amplifier:

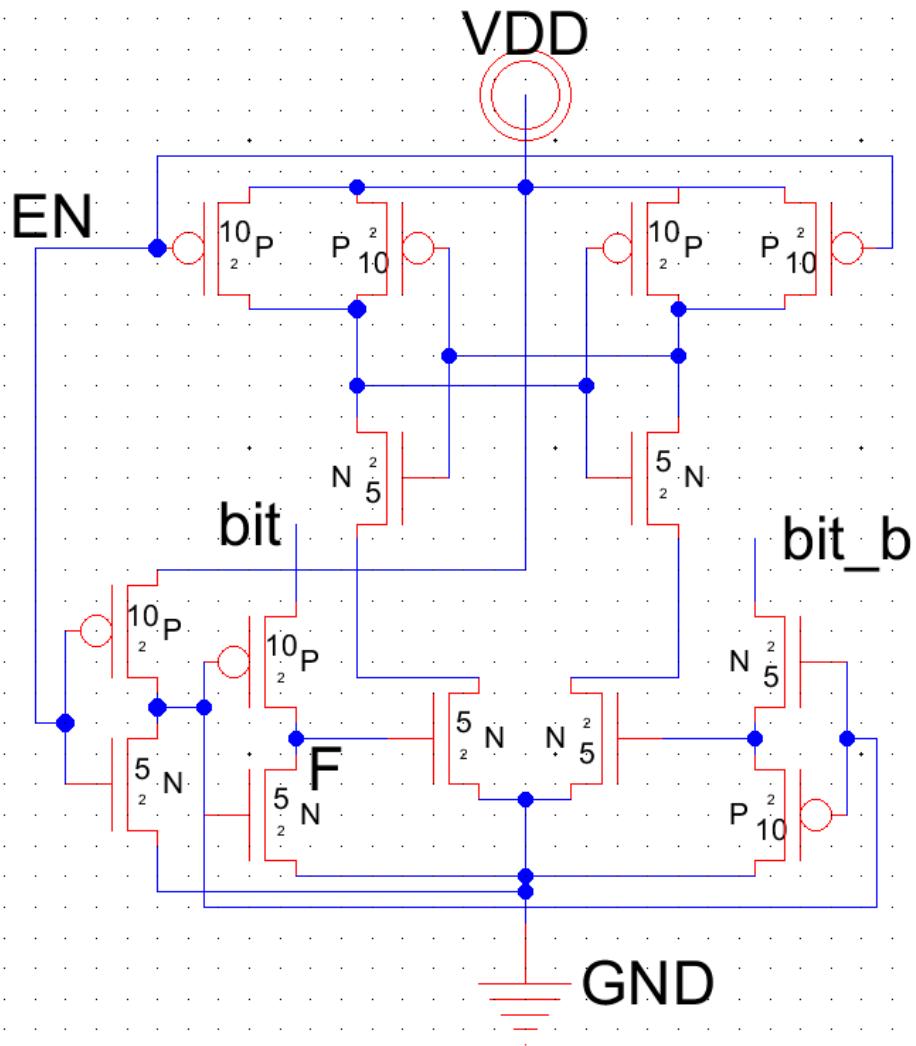


Figure 12: Sense Amplifier Schematic

```
=====
Checking schematic cell 'saas{sch}'
  No errors found
0 errors and 0 warnings found (took 0.009 secs)
```

Figure 13: DRC error check for Sense Amplifier

Write Driver Schematic:

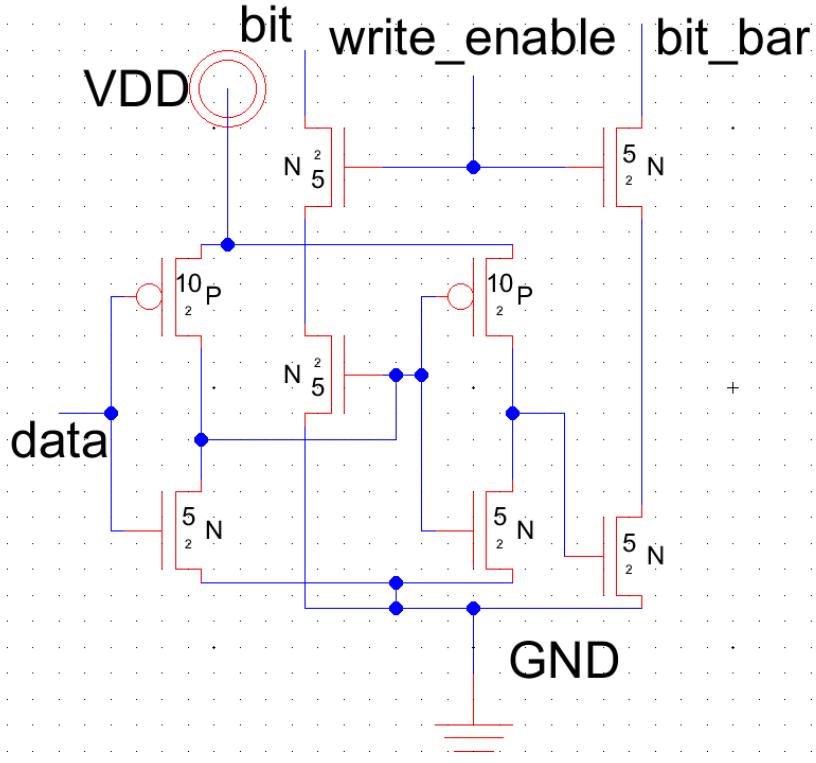
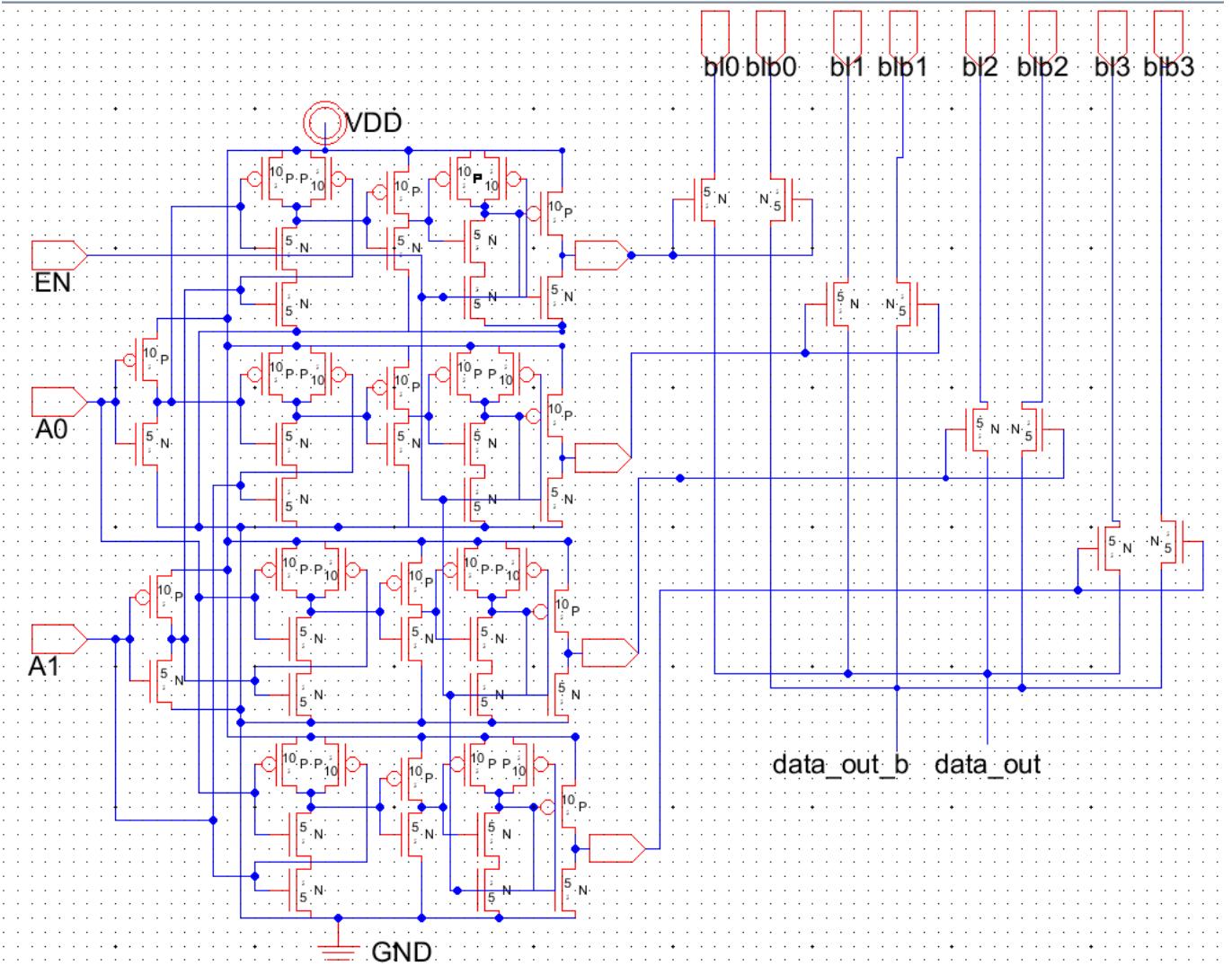


Figure 14: Write driver schematic

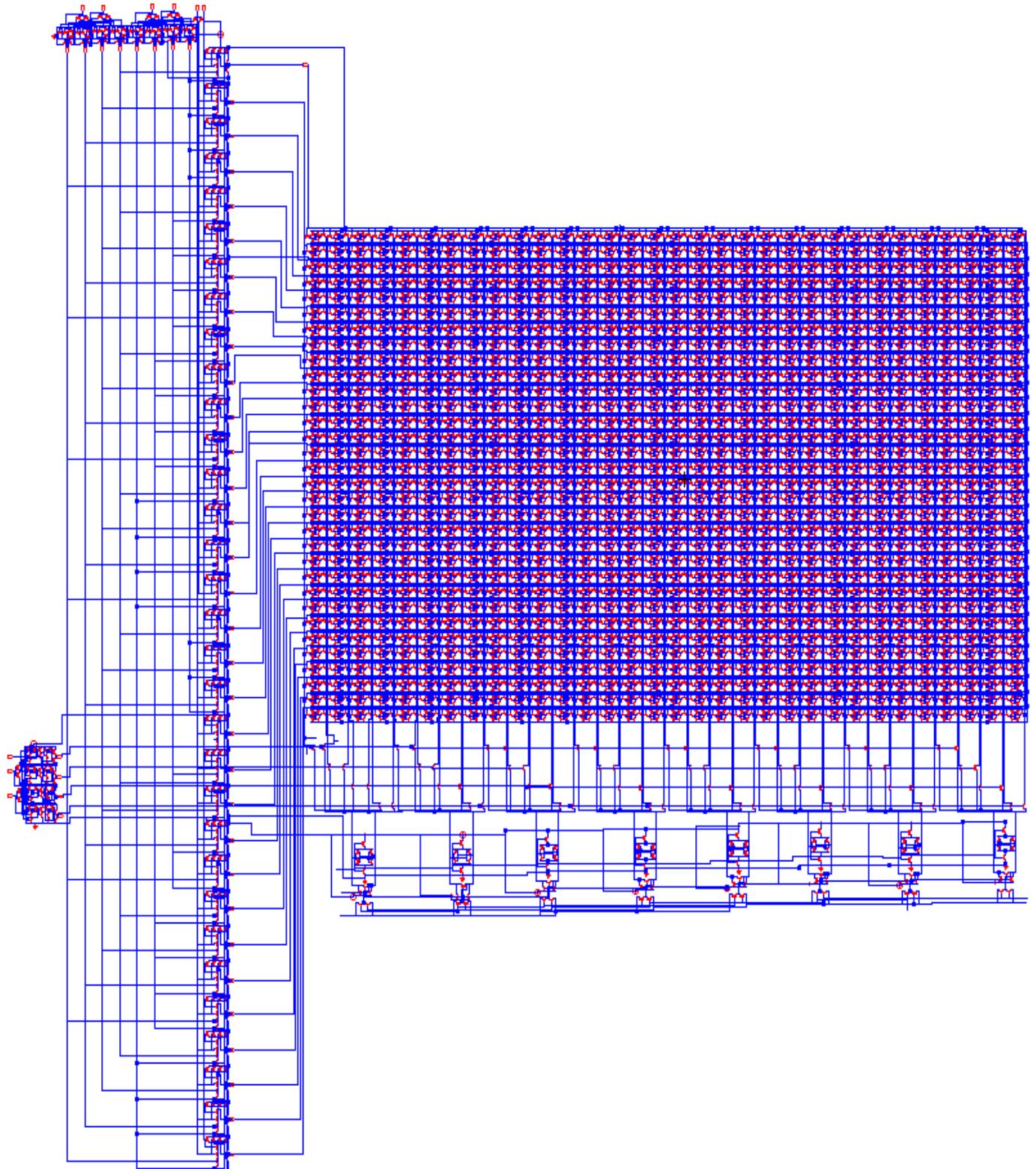
```
=====
Checking schematic cell 'write_driver{sch}'
  No errors found
0 errors and 0 warnings found (took 0.003 secs)
```

Figure 15: DRC error check for write driver schematic

Mux/Demux Schematic:**Figure 15:** Mux/Demux Schematic

```
=====186=====
Checking schematic cell 'mux{sch}'
  No errors found
0 errors and 0 warnings found (took 0.034 secs)
```

Figure 16: DRC error check for Mux/Demux schematic

1K SRAM Schematic:**Figure 17: 1K SRAM Schematic**

Zoom in view of 1K SRAM Schematic:

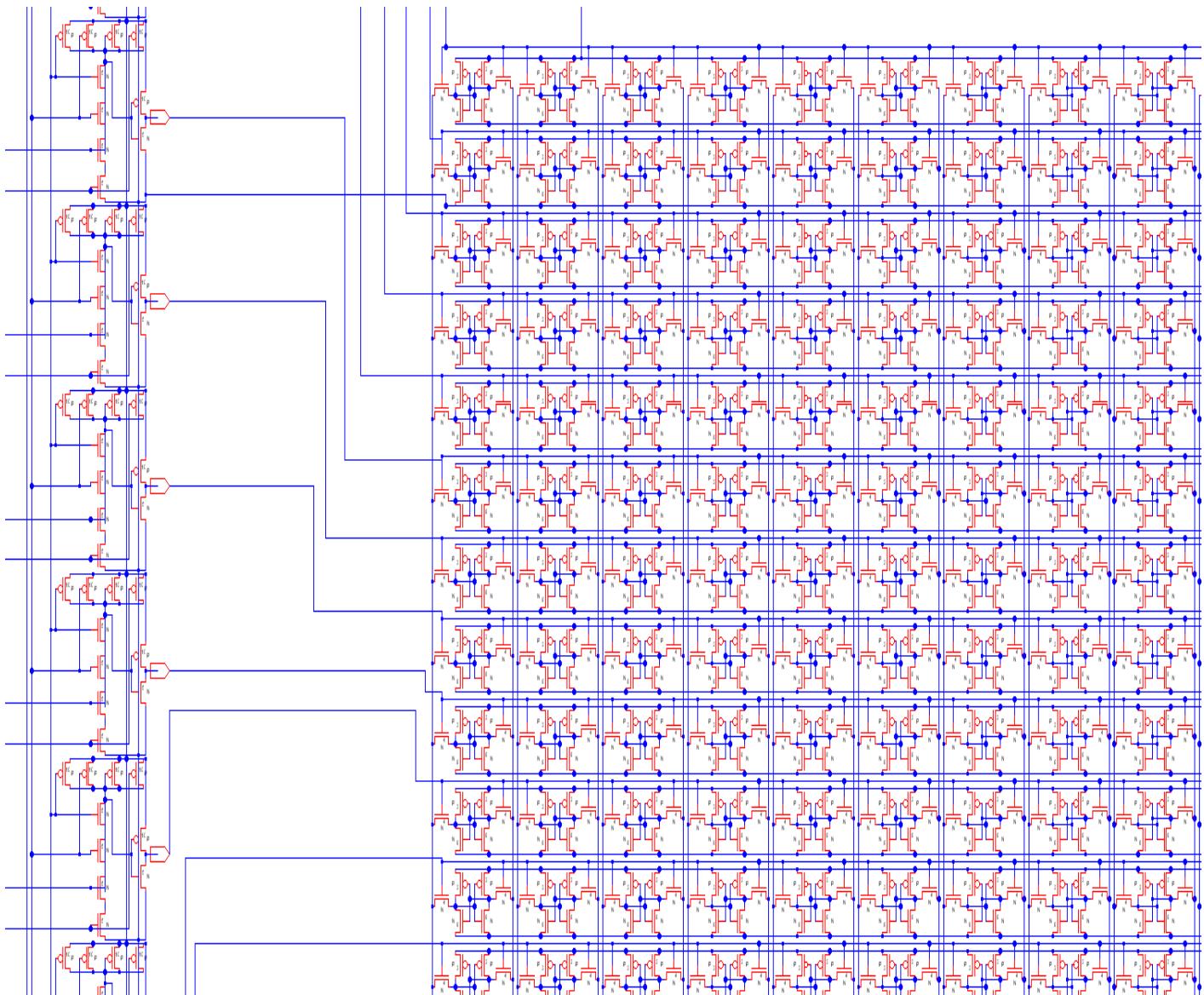


Figure 18: Zoom in view of 1K SRAM, 5 to 32 decoder connected with the word lines.

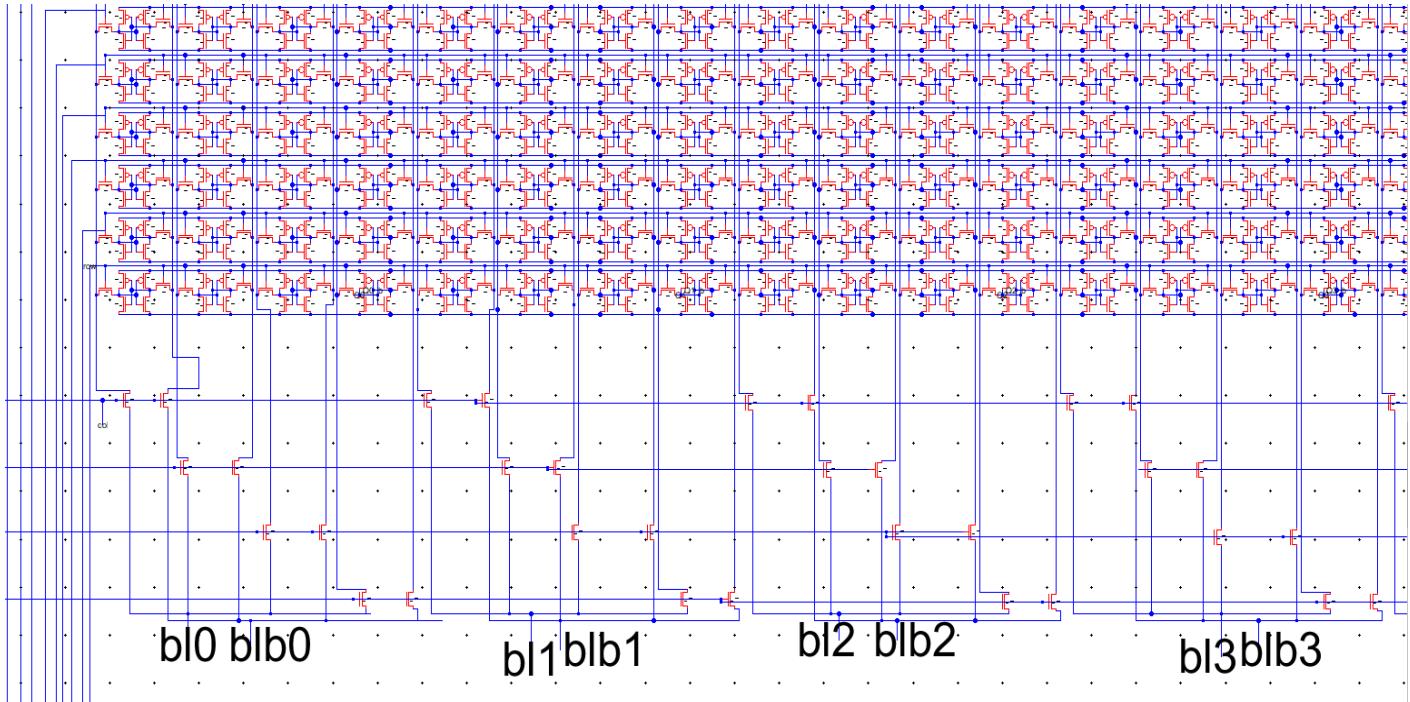


Figure 19: Mux/Demux Connected to the bit lines

```
Checking schematic cell 'lktest;l{sch}'  
No errors found  
0 errors and 0 warnings found (took 0.464 secs)
```

Figure 20: DRC error check for 1K SRAM Schematic

Section 4: Detailed Electric Layouts

We have included all the layout designs one by one below.

6T SRAM Layout:

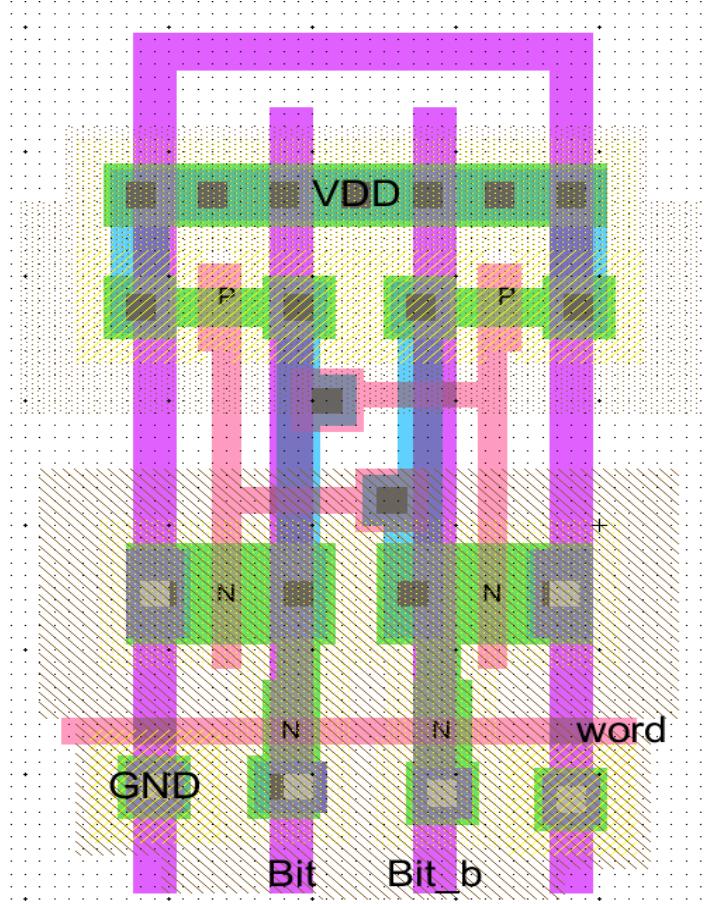


Figure 21: 6T SRAM Layout

```
=====
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.0 secs)
Found 14 networks
0 errors and 0 warnings found (took 0.001 secs)
=====
Checking Wells and Substrates in 'SRAM-(1):6TSRAM(lay)' ...
    Geometry collection found 22 well pieces, took 0.001 secs
    Geometry analysis used 4 threads and took 0.002 secs
NetValues propagation took 0.0 secs
Checking short circuits in 4 well contacts
    Additional analysis took 0.0 secs
No Well errors found (took 0.003 secs)
```

Figure 22: DRC, Well check for 6T SRAM layout

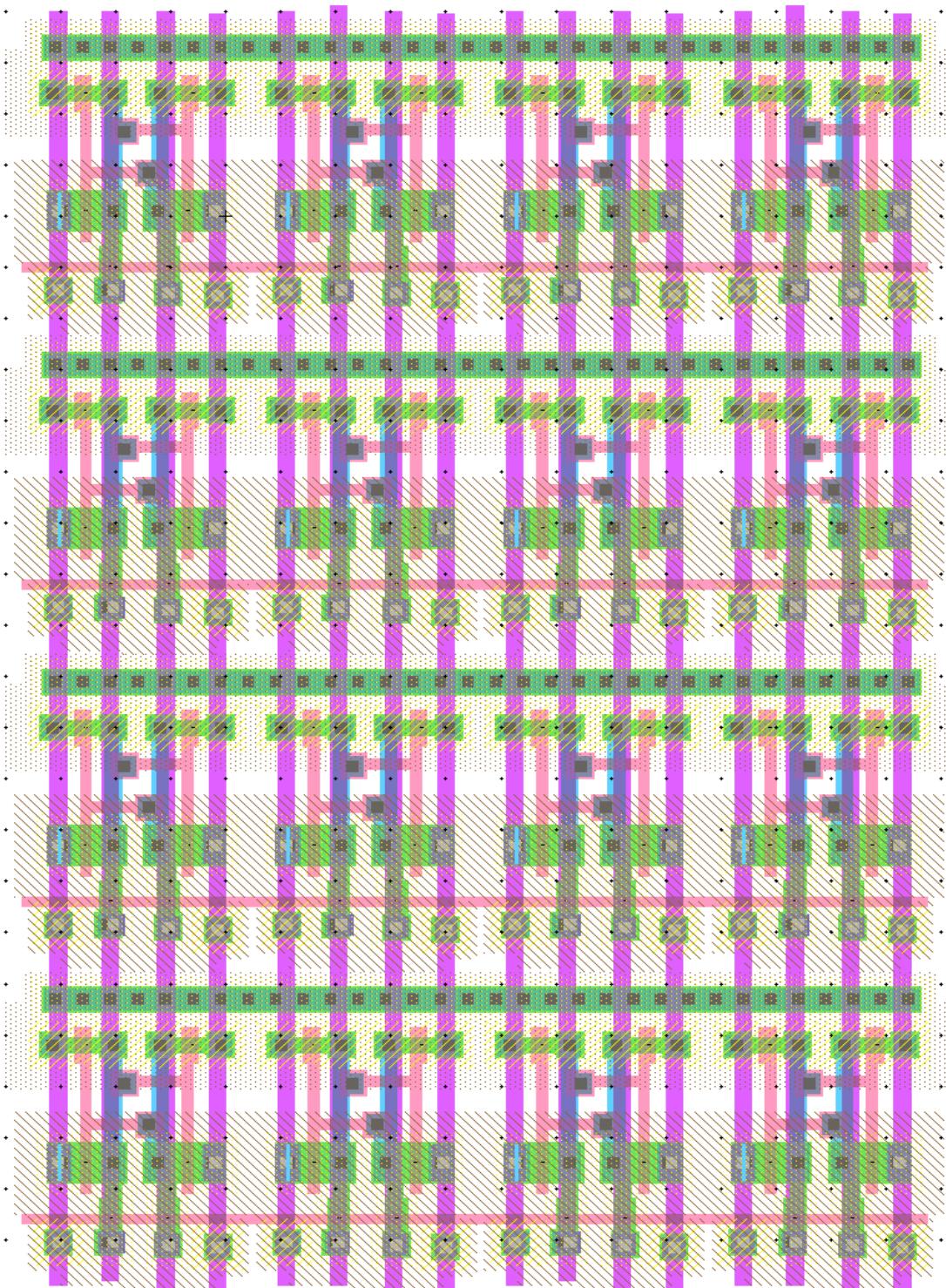


Figure 23: 16 cells of SRAM layout unlabeled

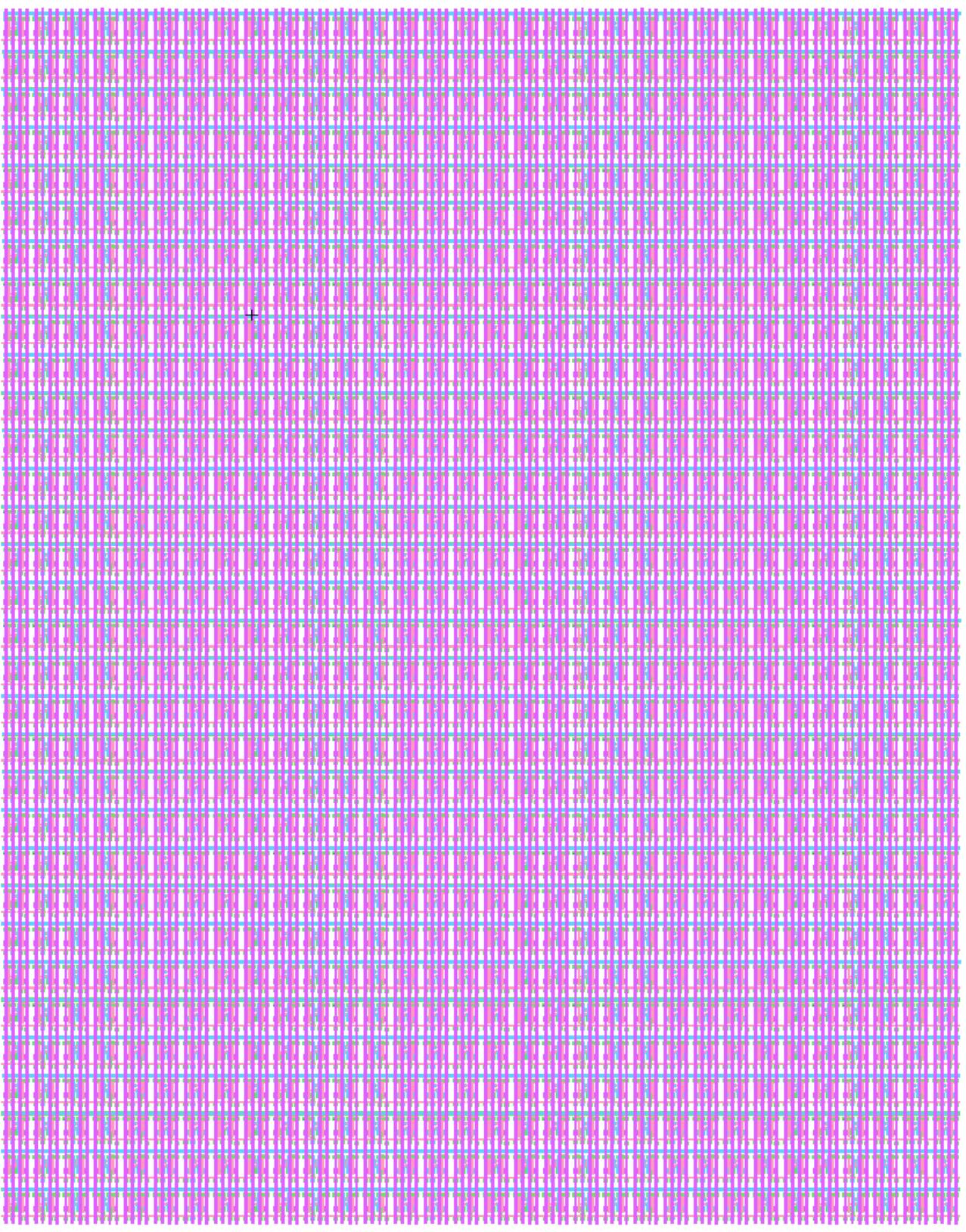


Figure 24: 1024 (1K) cells of SRAM layout unlabeled

2 to 4 Decoder:

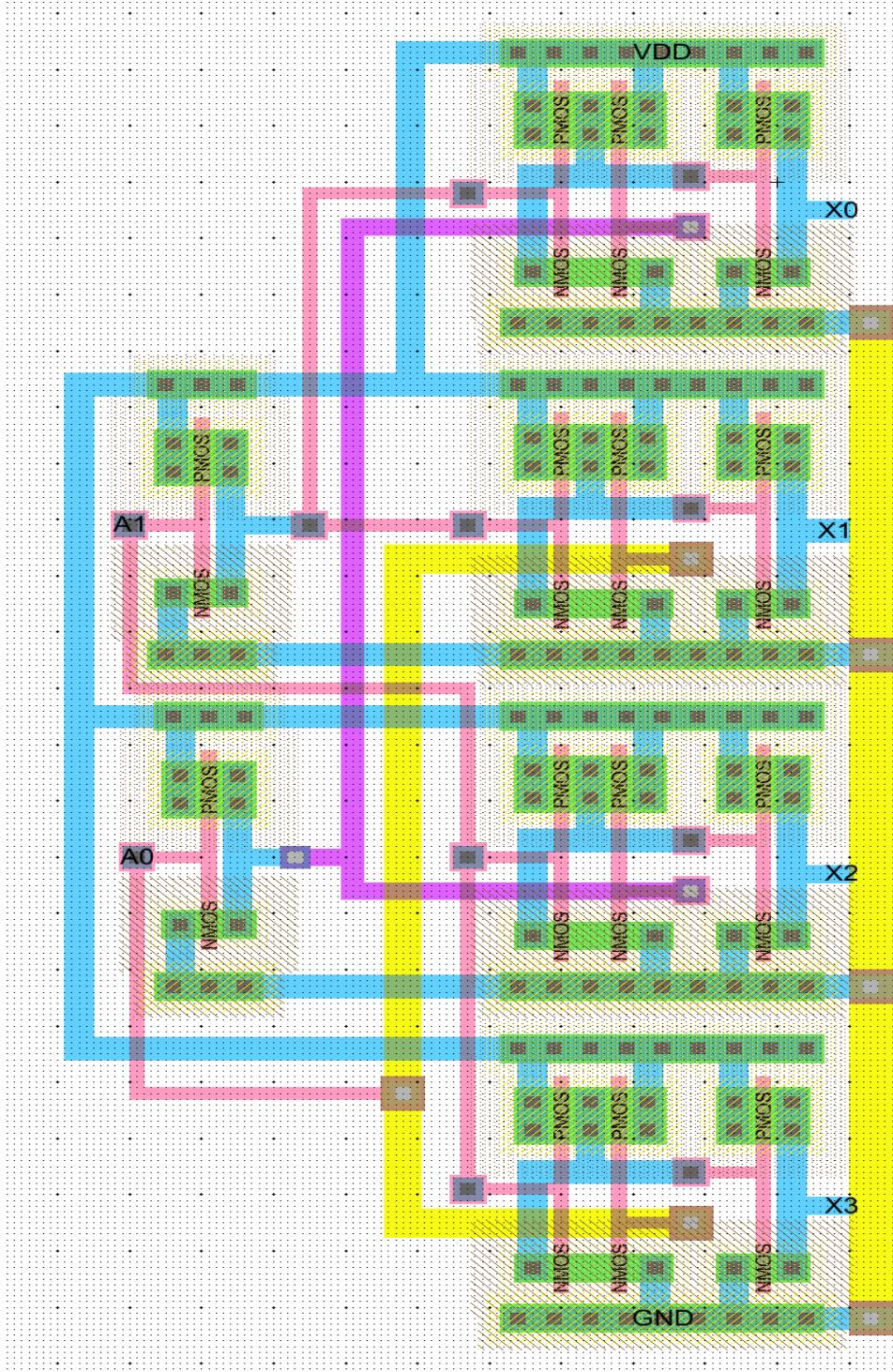


Figure 25: 2 to 4 Decoder layout

```

-----  

Running DRC with area bit on, extension bit on, Mosis bit  

Checking again hierarchy .... (0.001 secs)  

Found 47 networks  

0 errors and 0 warnings found (took 0.006 secs)  

=====41=====  

Checking Wells and Substrates in 'SRAM-(1)-(1):2to4decoder{lay}' ...  

    Geometry collection found 92 well pieces, took 0.008 secs  

    Geometry analysis used 4 threads and took 0.009 secs  

NetValues propagation took 0.001 secs  

Checking short circuits in 12 well contacts  

    Additional analysis took 0.0 secs  

No Well errors found (took 0.019 secs)  

=====42=====  

Hierarchical NCC every cell in the design: cell '2to4decoder{sch}' cell '2to4decoder{lay}'  

Comparing: SRAM-(1)-(1):2to4decoder{sch} with: SRAM-(1)-(1):2to4decoder{lay}  

    exports match, topologies match, sizes match in 0.05 seconds.  

Summary for all cells: exports match, topologies match, sizes match  

NCC command completed in: 0.066 seconds.

```

Figure 26: DRC, Well and NCC check for 2 to 4 layout**5 to 32 Decoder:**

```

=====95=====  

Running DRC with area bit on, extension bit on, Mosis bit  

Checking again hierarchy .... (0.001 secs)  

Found 576 networks  

0 errors and 0 warnings found (took 0.01 secs)  

=====96=====  

Checking Wells and Substrates in 'SRAM_Fin-(1):5to32{lay}' ...  

    Geometry collection found 1024 well pieces, took 0.015 secs  

    Geometry analysis used 4 threads and took 0.004 secs  

NetValues propagation took 0.001 secs  

Checking short circuits in 90 well contacts  

    Additional analysis took 0.001 secs  

No Well errors found (took 0.021 secs)

```

Figure 27: DRC, Well check for 5 to 32 layout

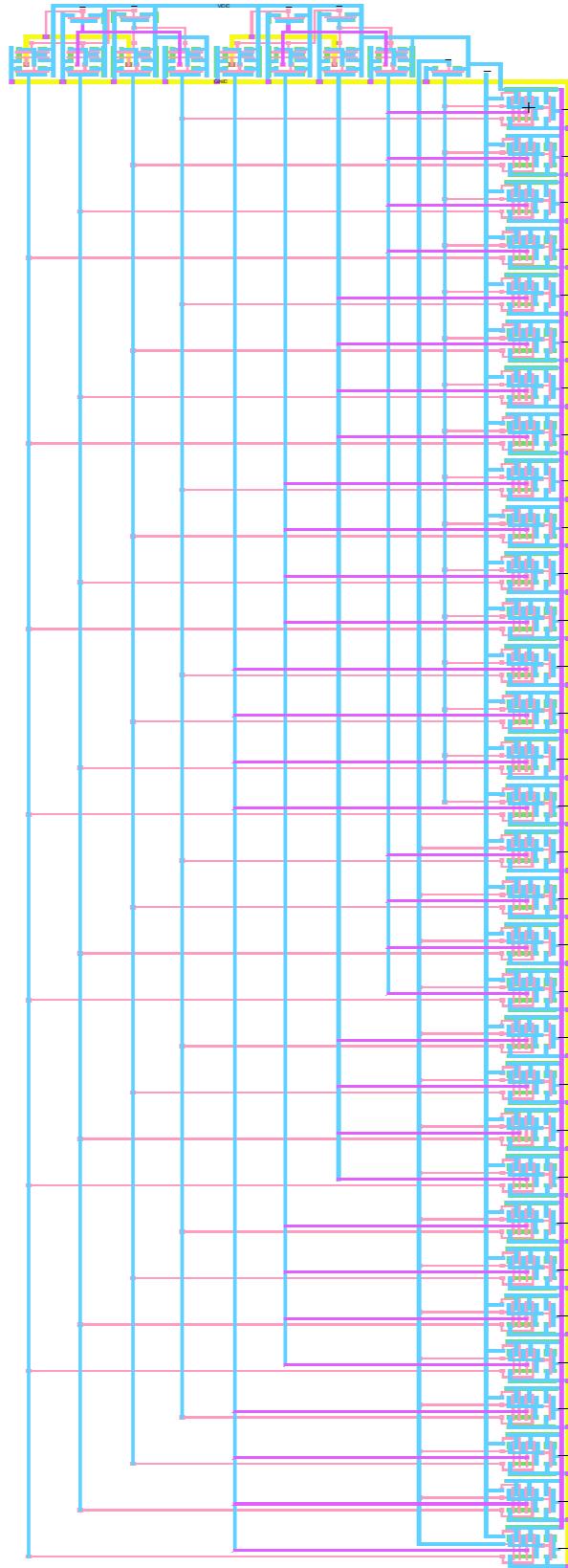


Figure 28: 5 to 32 Decoder Layout

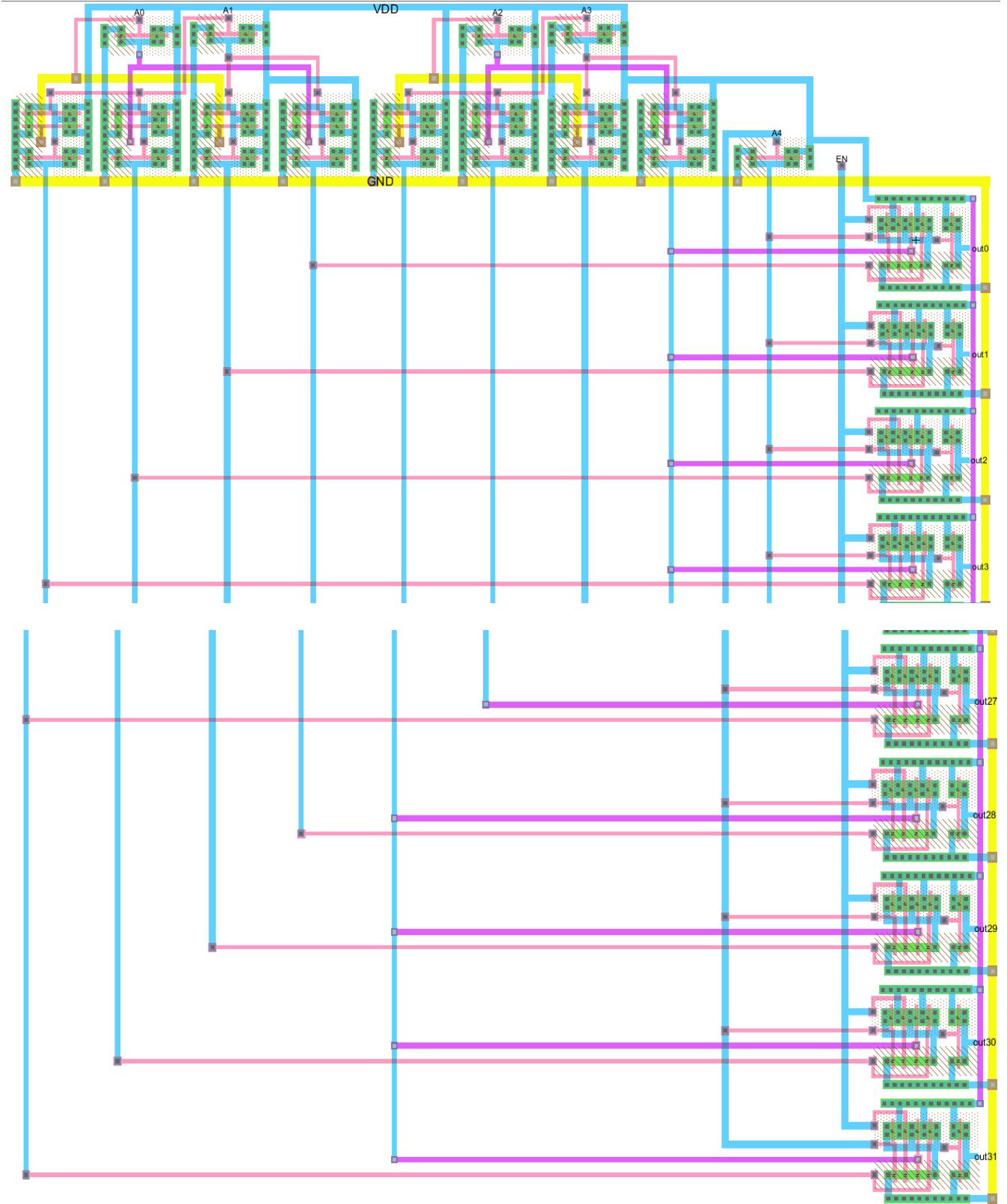


Figure 29: Zoom in 5 to 32 Decoder Layout top 4 and bottom 4 output

Sense Amplifier:

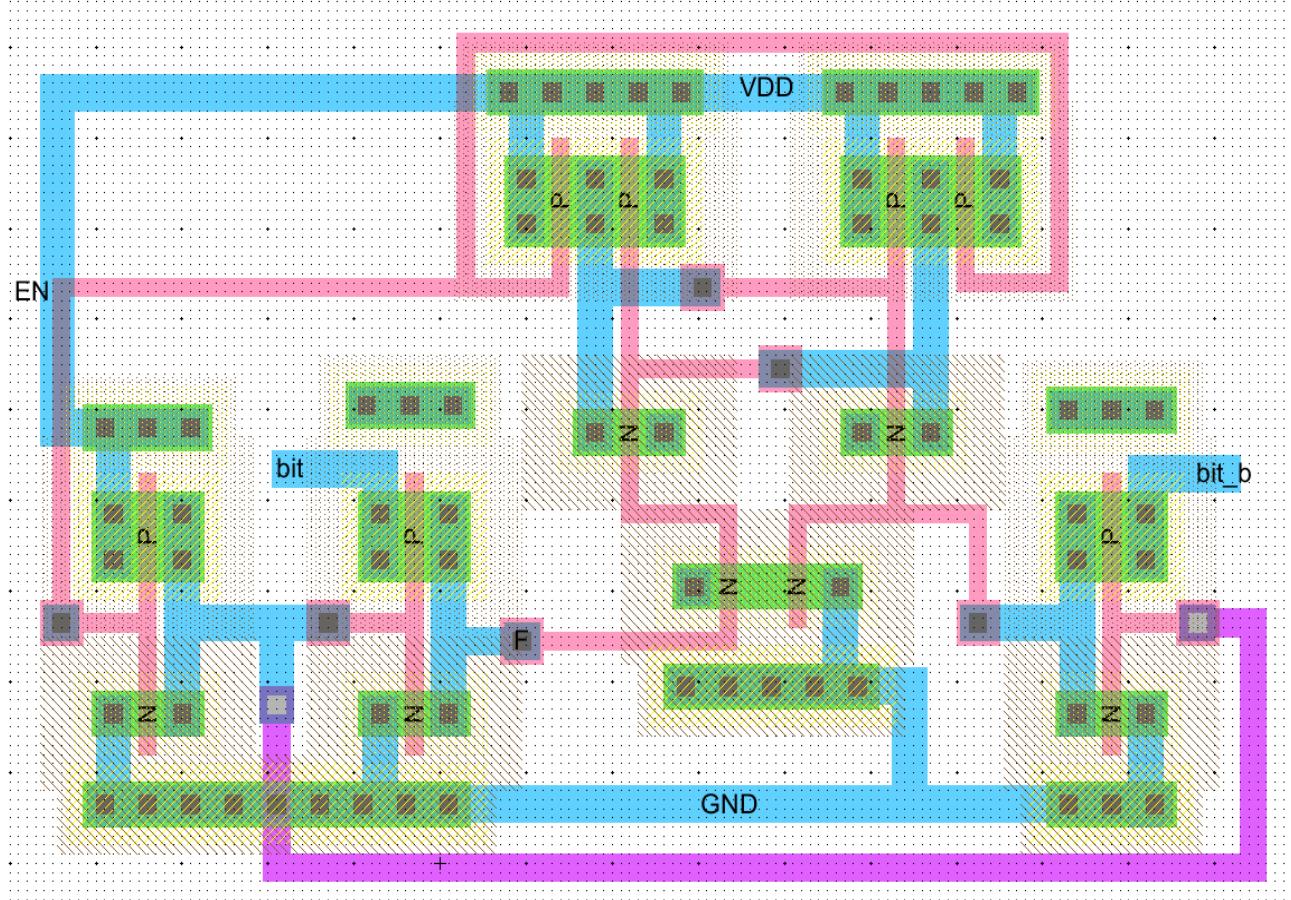


Figure 30: Sense Amplifier Layout

```

-----104-----
Checking Wells and Substrates in 'SRAM-(1)-(1):saas{lay}' ...
    Geometry collection found 48 well pieces, took 0.0 secs
    Geometry analysis used 4 threads and took 0.009 secs
    NetValues propagation took 0.0 secs|
    Checking short circuits in 8 well contacts
        Additional analysis took 0.0 secs
    No Well errors found (took 0.01 secs)

```

Figure 31: DRC, Well error check for sense amplifier layout

Write Driver Schematic:

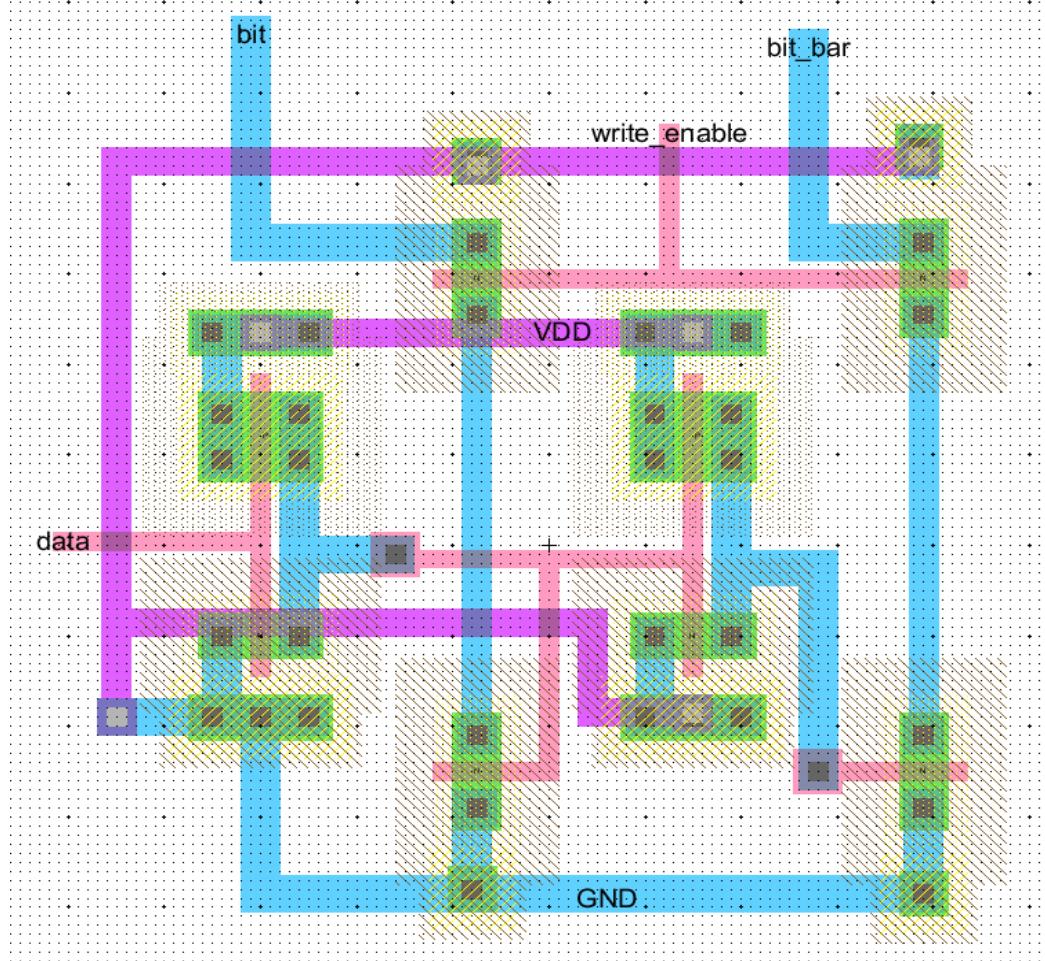


Figure 32: Write driver layout

```

=====
Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.004 secs)
Found 19 networks
Checking cell 'write_Driver[lay]'
    No errors/warnings found
0 errors and 0 warnings found (took 0.164 secs)
=====90=====
Checking Wells and Substrates in 'final_SRAM_DESIGN:write_Driver[lay]' ...
    Geometry collection found 32 well pieces, took 0.004 secs
    Geometry analysis used 4 threads and took 0.002 secs
NetValues propagation took 0.001 secs
Checking short circuits in 8 well contacts
    Additional analysis took 0.002 secs
No Well errors found (took 0.011 secs)
|
```

Figure 33: DRC, Well error check for write driver layout

Mux/Demux Layout:

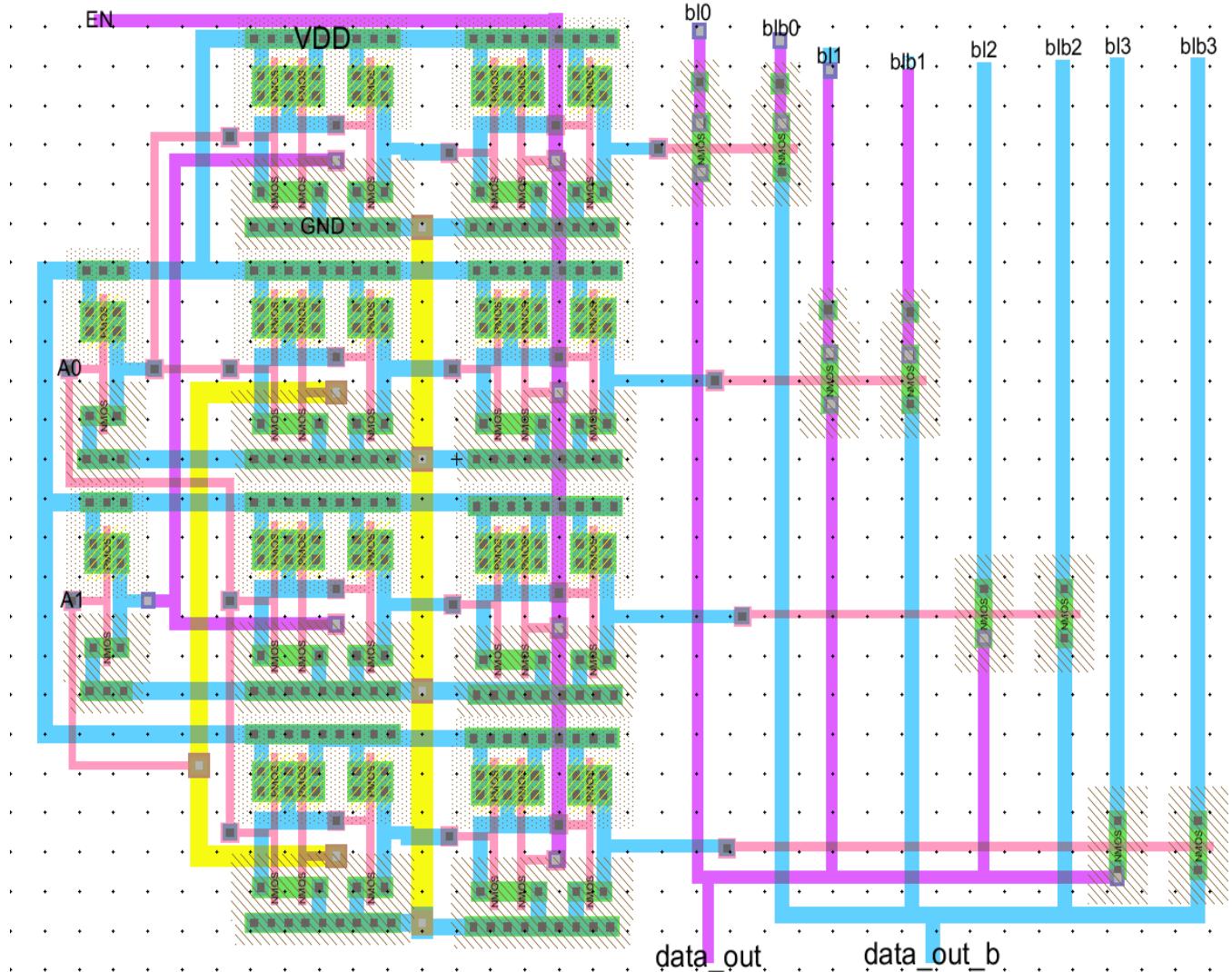


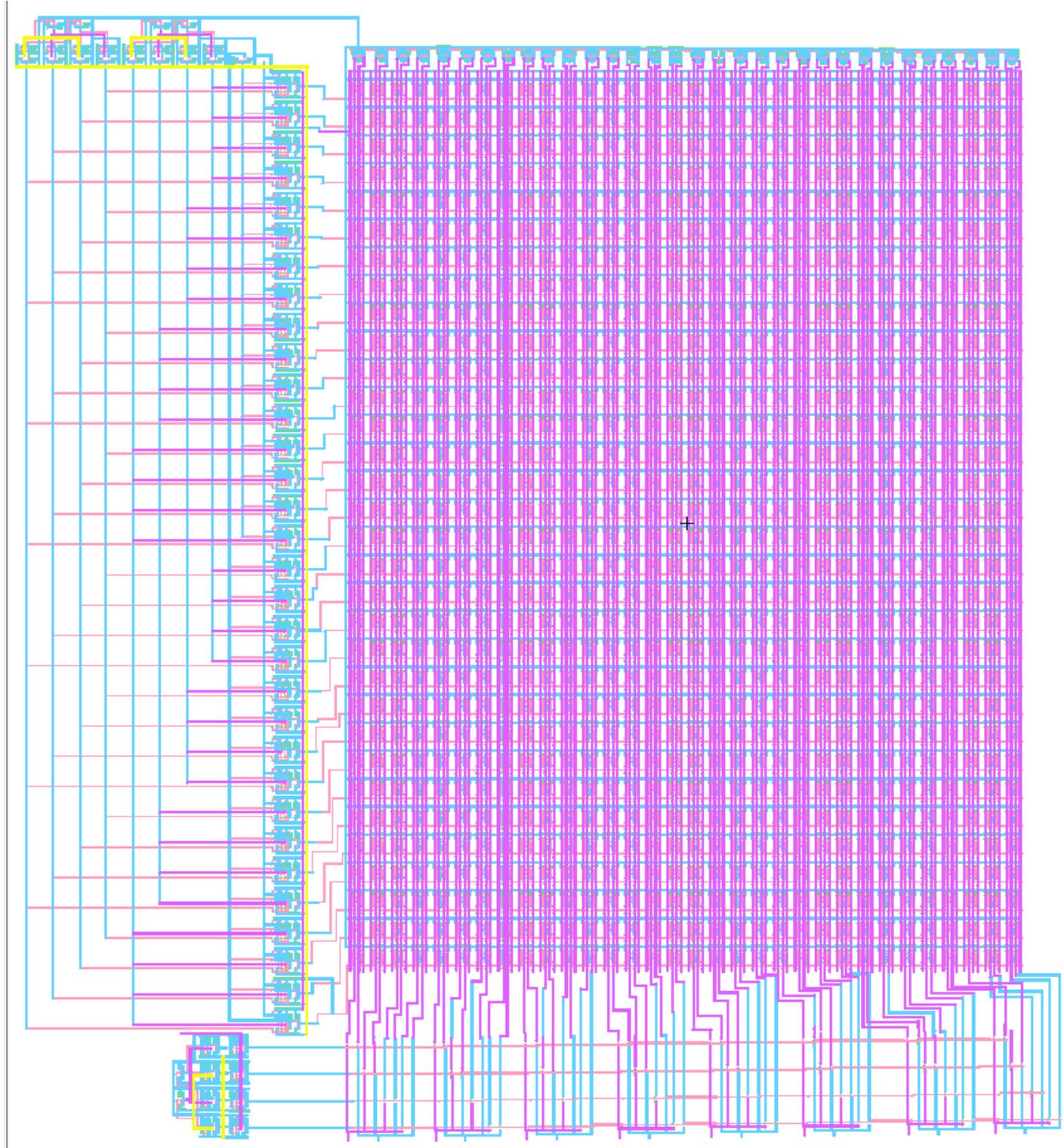
Figure 34: Mux/Demux Layout

```

Running DRC with area bit on, extension bit on, Mosis bit
Checking again hierarchy .... (0.009 secs)
Found 106 networks
Checking cell 'mux{lay}'
      No errors/warnings found
0 errors and 0 warnings found (took 0.28 secs)

```

Figure 35: DRC error check for Mux/Demux layout

1K SRAM Layout:**Figure 36: 1K SRAM Layout**

Zoom in view of 1K SRAM Layout:

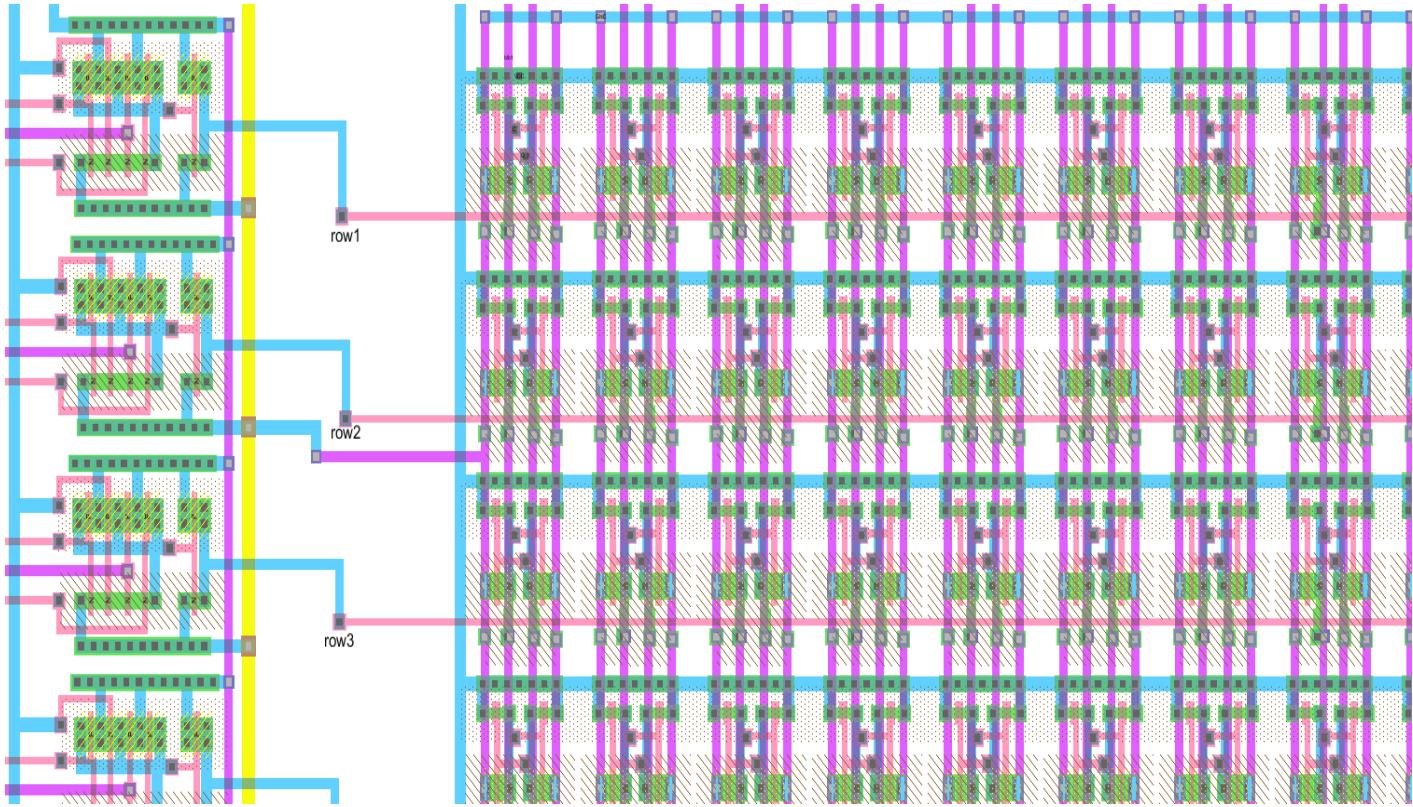


Figure 37: Zoom in view of 1K SRAM, 5 to 32 decoder connected with the word lines.

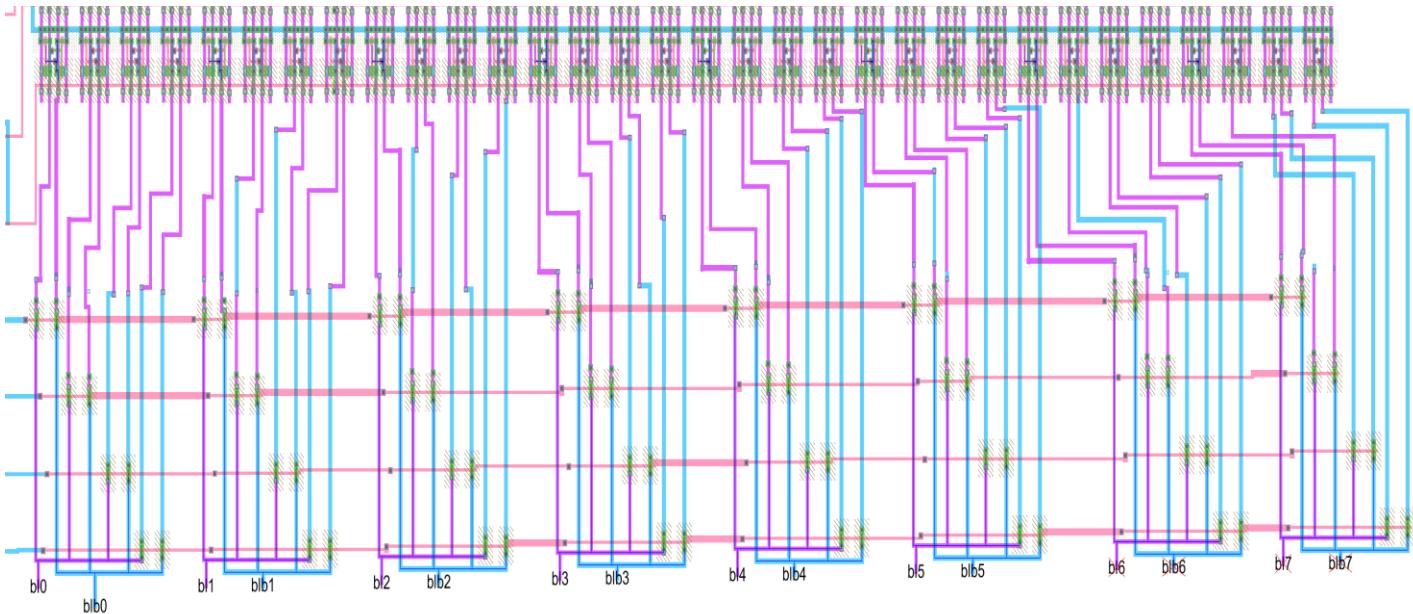


Figure 38: Mux/Demux Connected to the bit lines

```
-----  
Checking schematic cell 'lktest;l{lay}'  
  No errors found  
0 errors and 0 warnings found (took 0.464 secs)  
  
=====193=====  
Running DRC with area bit on, extension bit on, Mosis bit  
Checking again hierarchy .... (0.009 secs)  
Found 106 networks  
Checking cell 'lktest;l{lay}'  
  No errors/warnings found  
0 errors and 0 warnings found (took 0.28 secs)  
Toggle grid is on but grid is not drawn due to the resolution  
=====194=====
```

Figure 39: DRC, Well error check for 1K SRAM Layout

Section 5: IRSIM Logic Simulations and Measurements for Layout and Schematic with Comparison

6T SRAM Schematic:

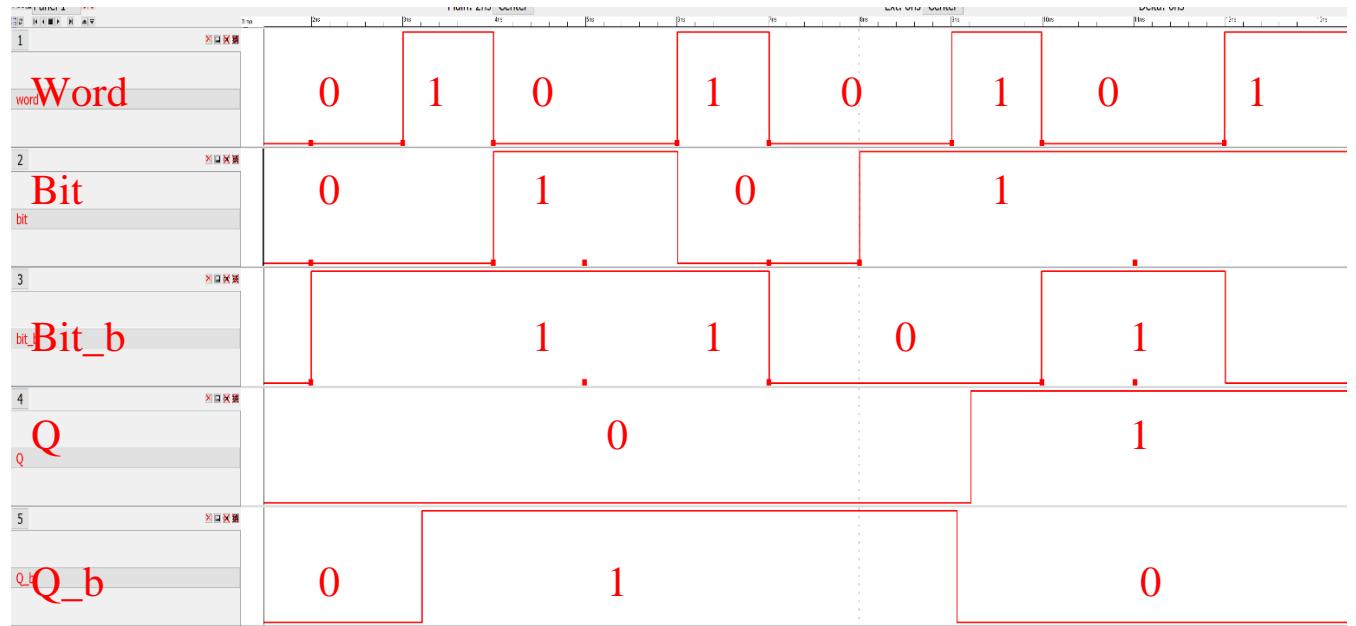


Figure 40: IRSIM simulation for 6T SRAM schematic

6T SRAM Layout:

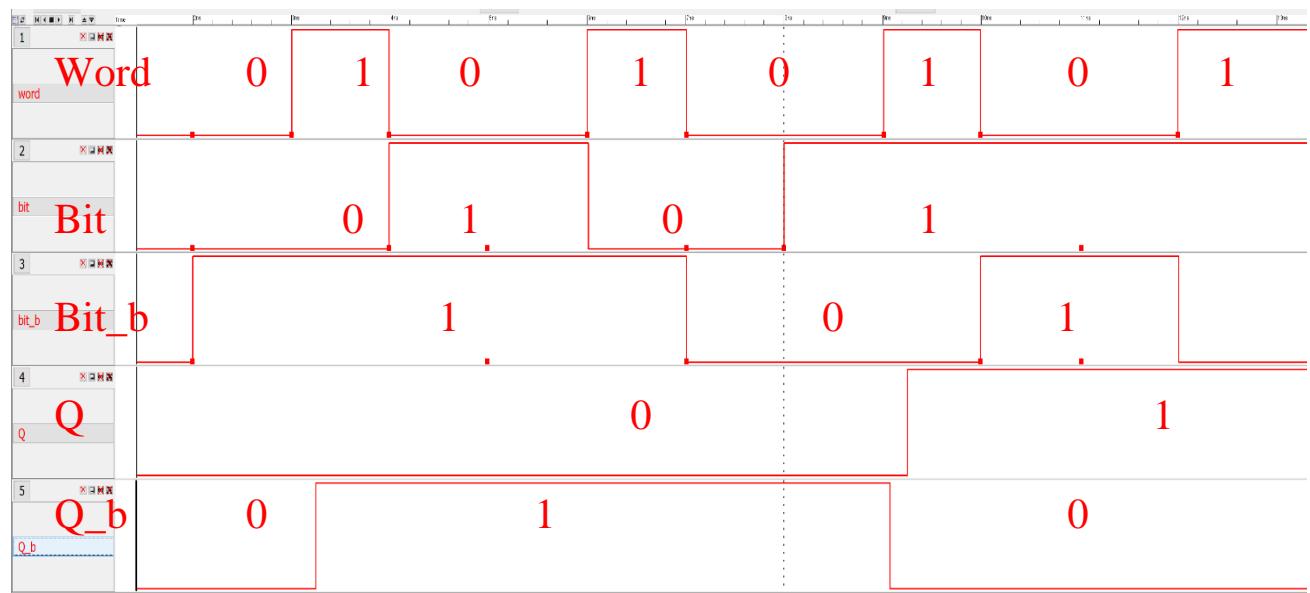


Figure 41: IRSIM simulation for 6T SRAM layout

2 to 4 Decoder Schematic:

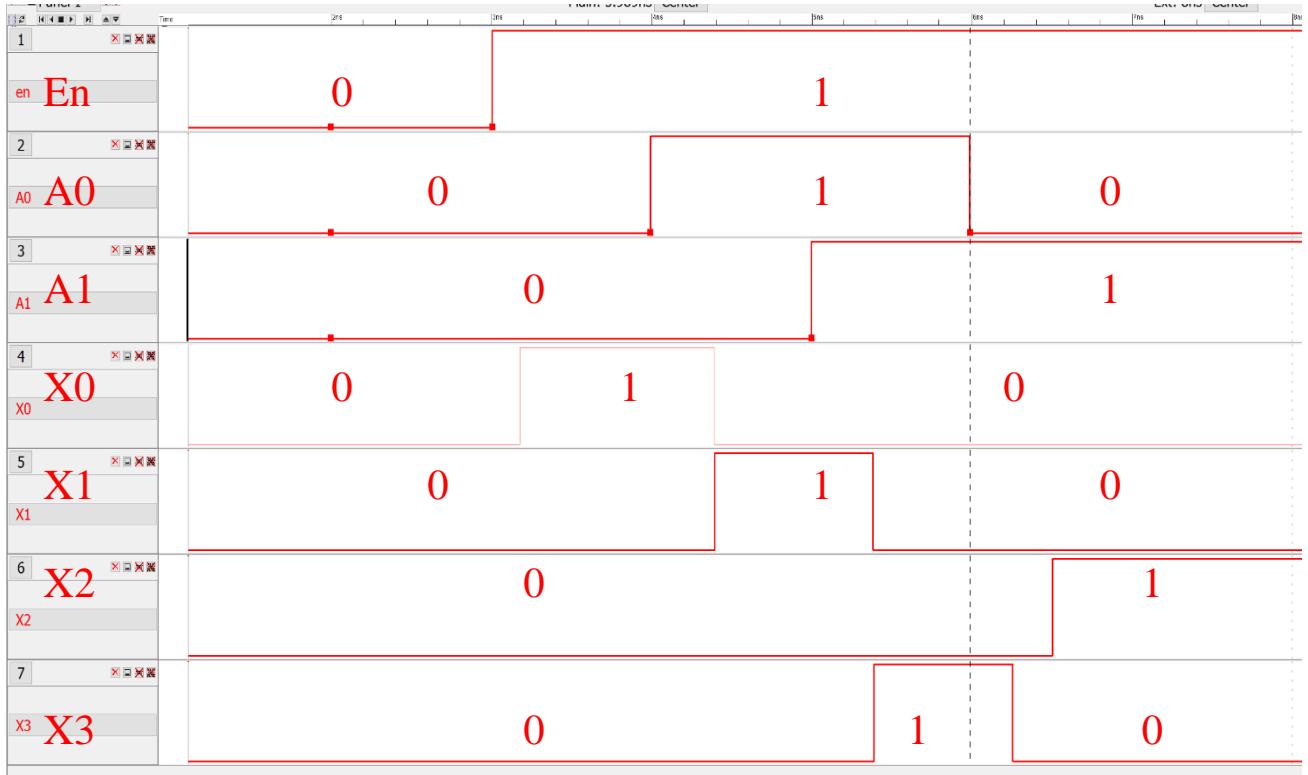


Figure 42: IRSIM simulation for 2 to 4 Decoder Schematic

2 to 4 Decoder Layout:

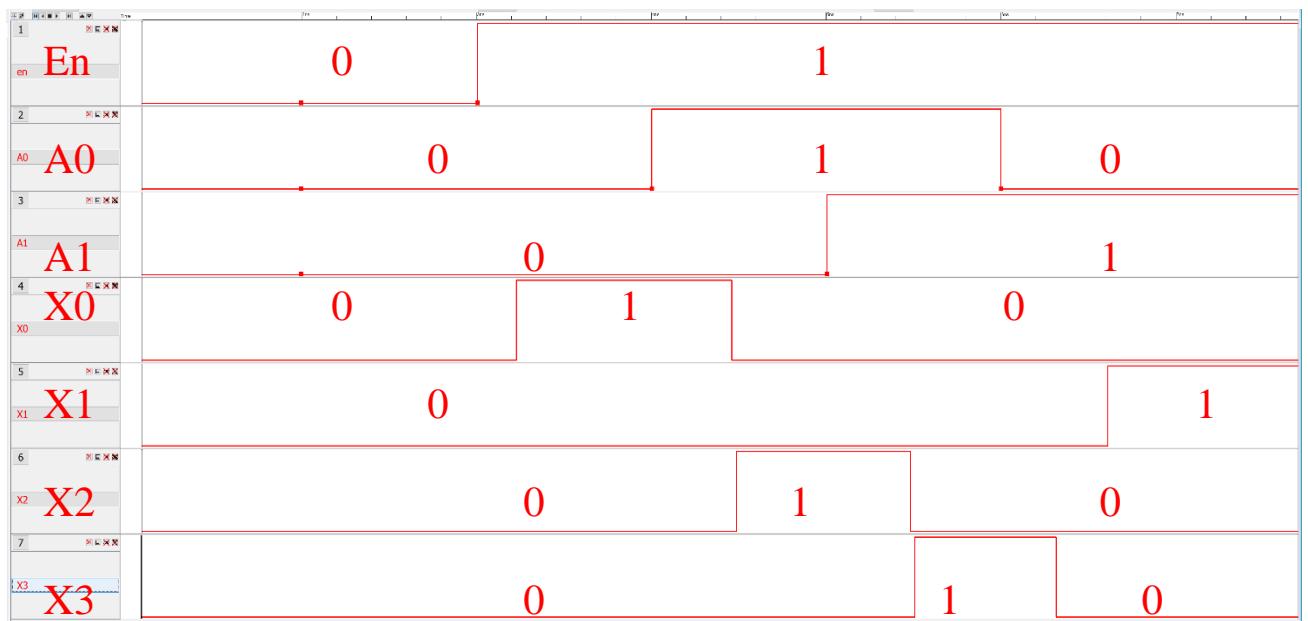


Figure 43: IRSIM simulation for 2 to 4 Decoder Layout

5 to 32 Decoder Schematic output 0-15:

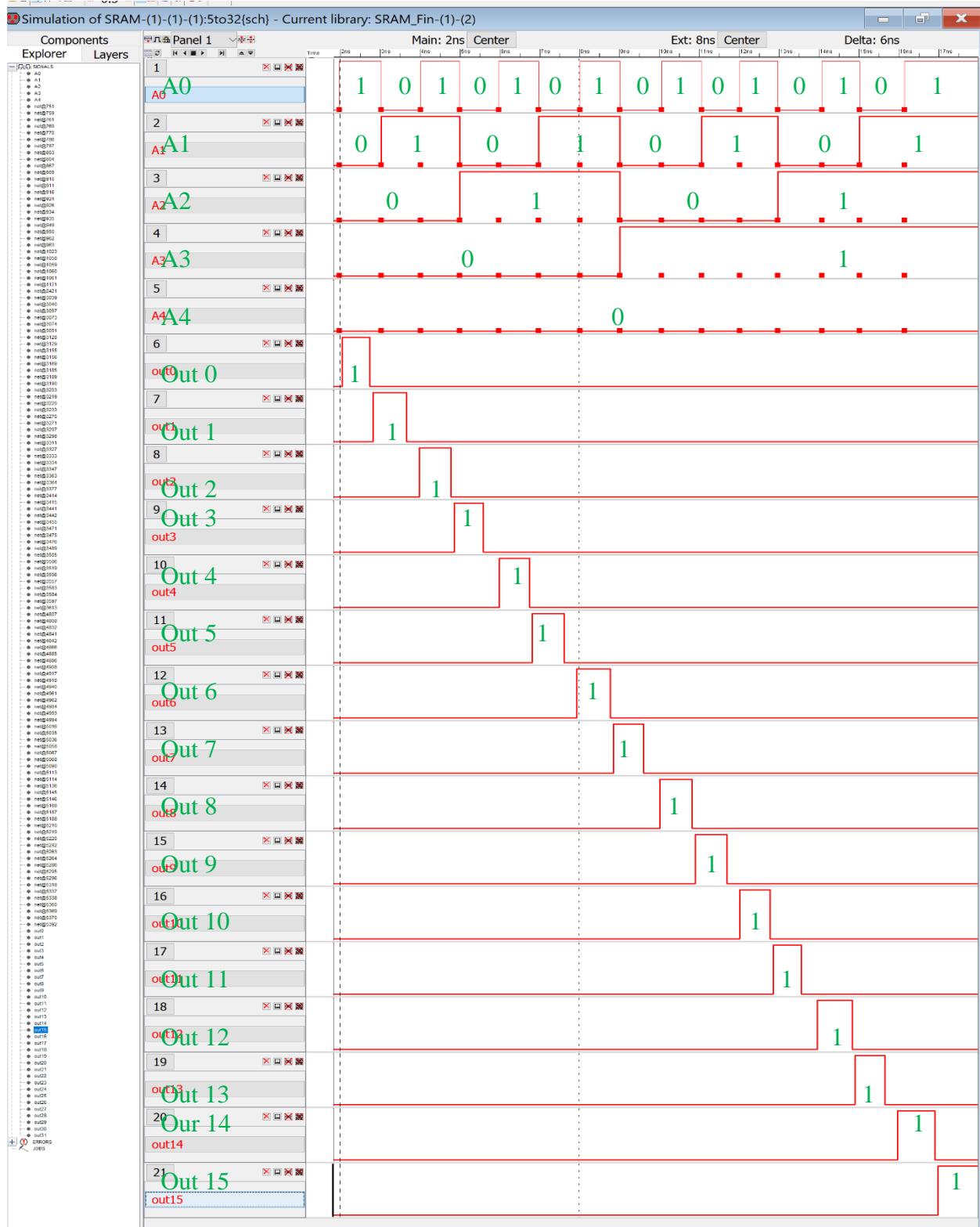


Figure 44: IRSIM simulation for 5 to 32 Decoder Schematic output 0-15

5 to 32 Decoder Schematic output 16-31:

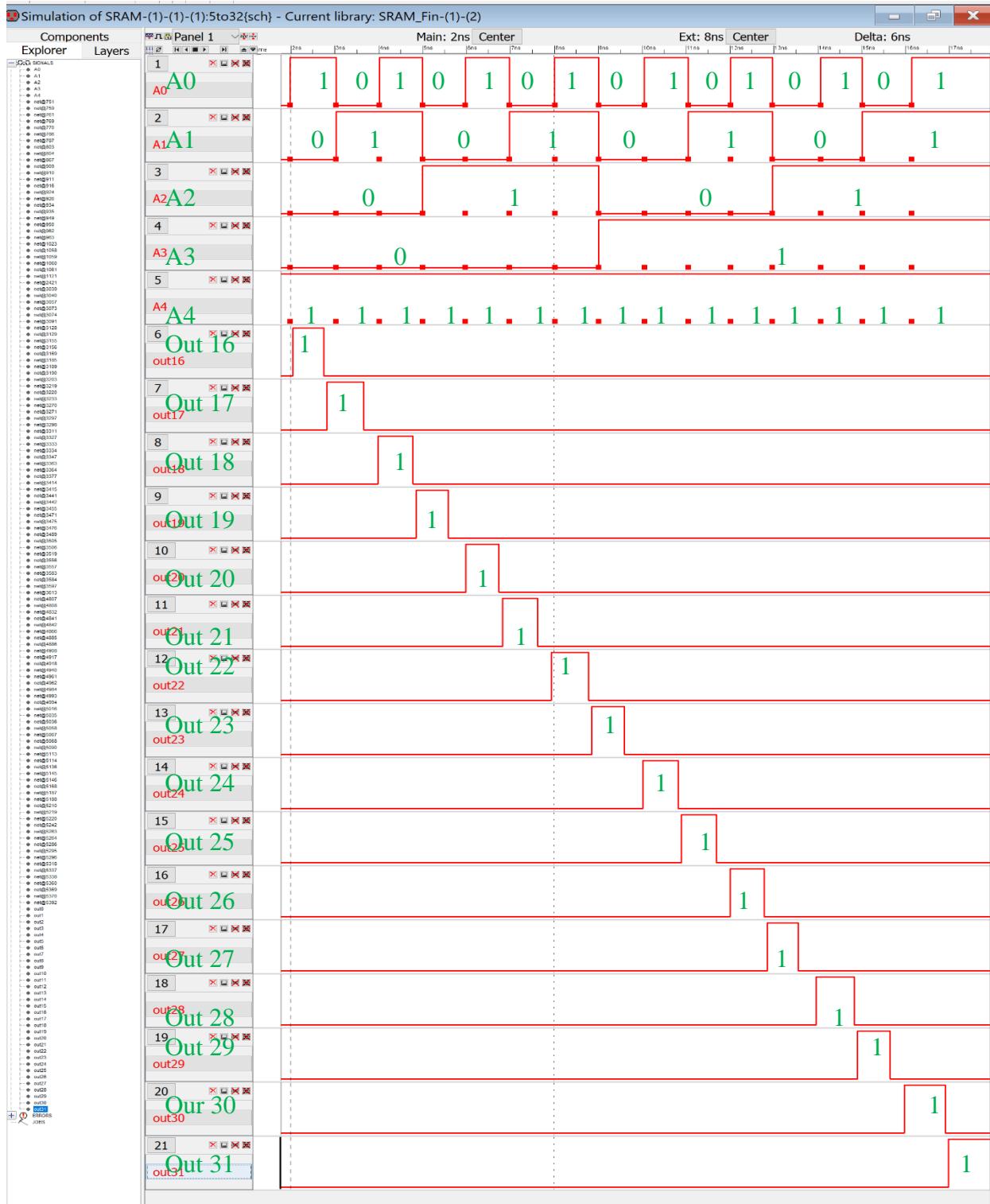


Figure 45: IRSIM simulation for 5 to 32 Decoder Schematic output 16-31

5 to 32 Decoder Layout output 0-15:

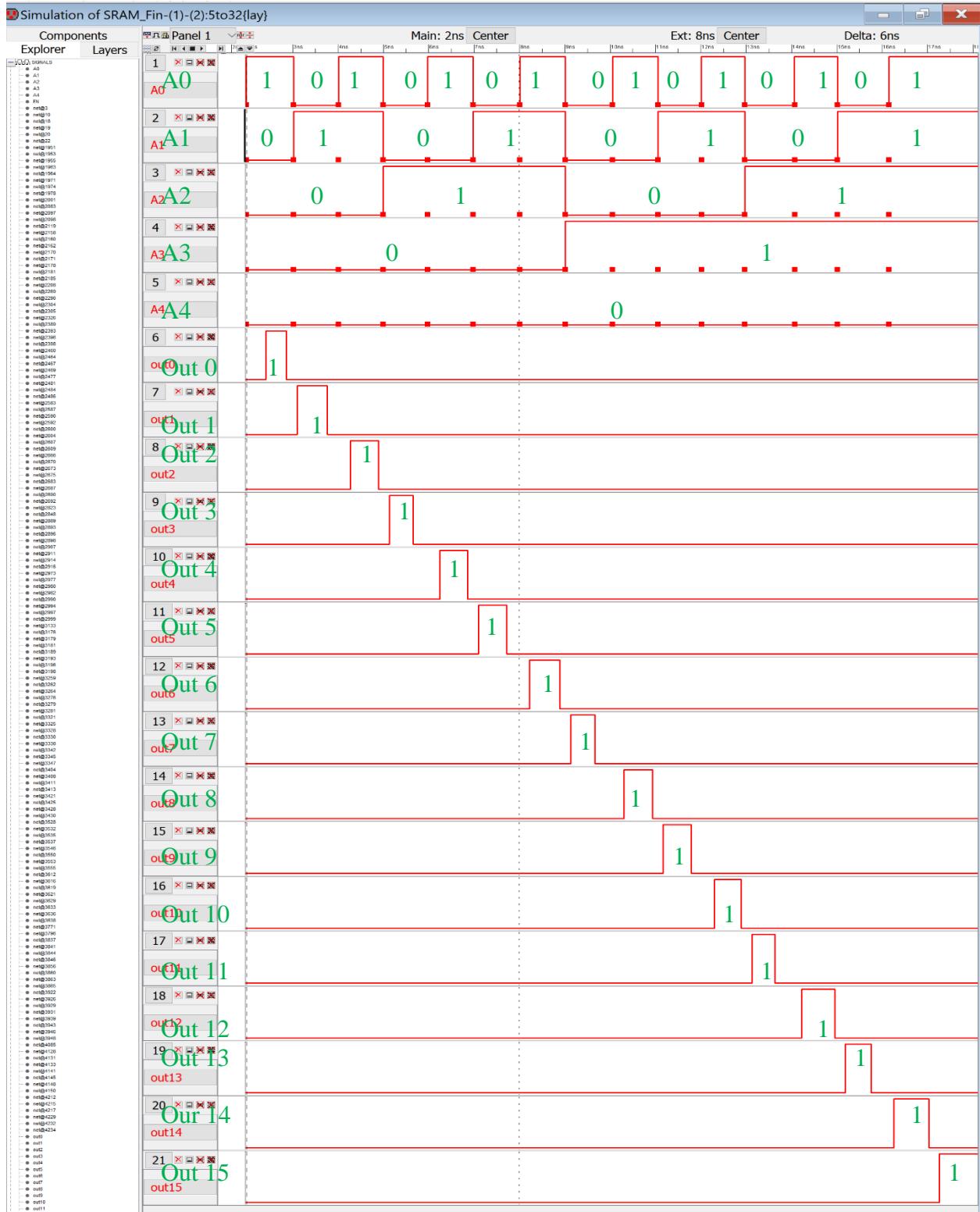


Figure 46: IRSIM simulation for 5 to 32 Decoder Layout output 0-15

5 to 32 Decoder Layout output 16-31:

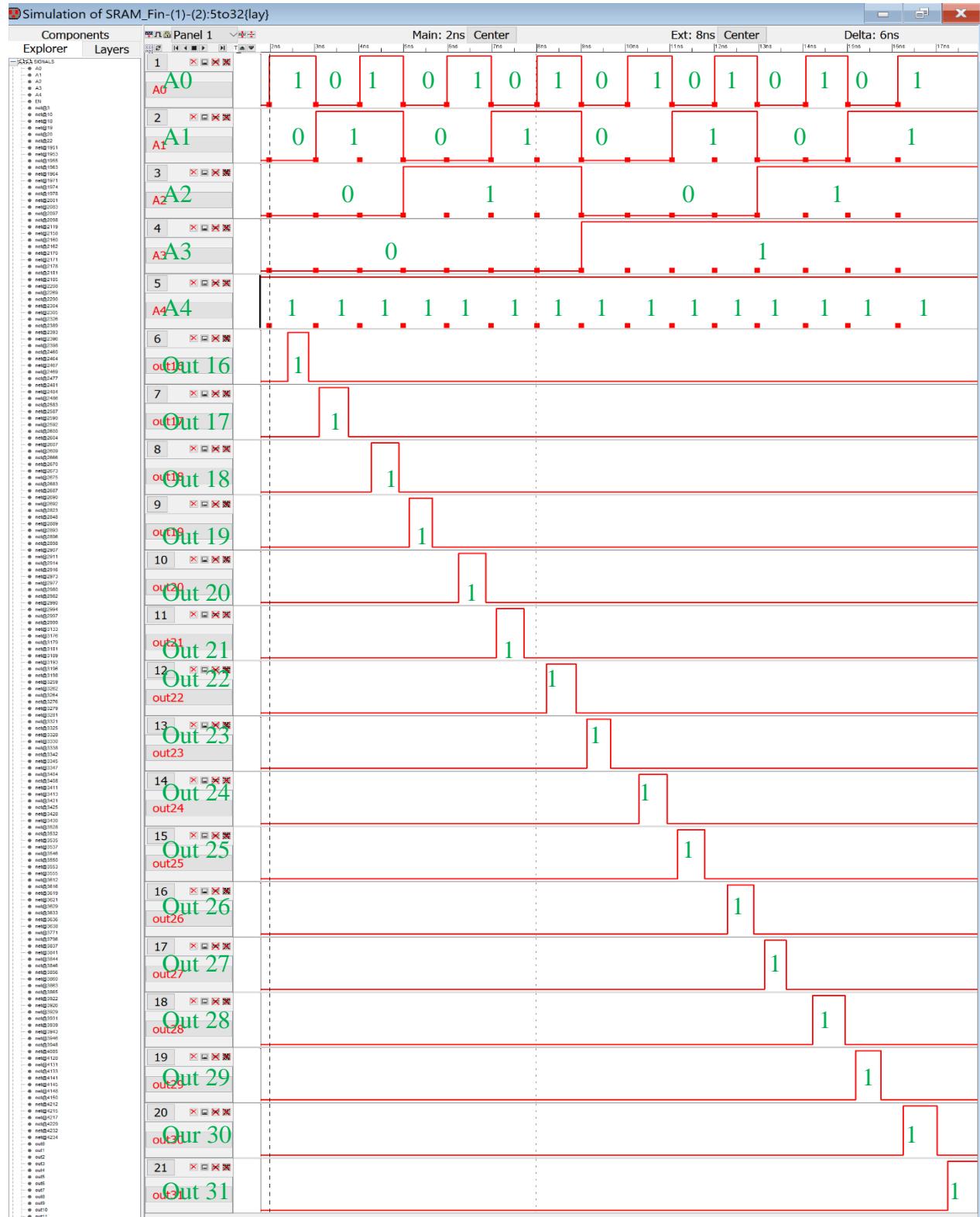


Figure 47: IRSIM simulation for 5 to 32 Decoder Layout output 16-31

Sense Amplifier Schematic:

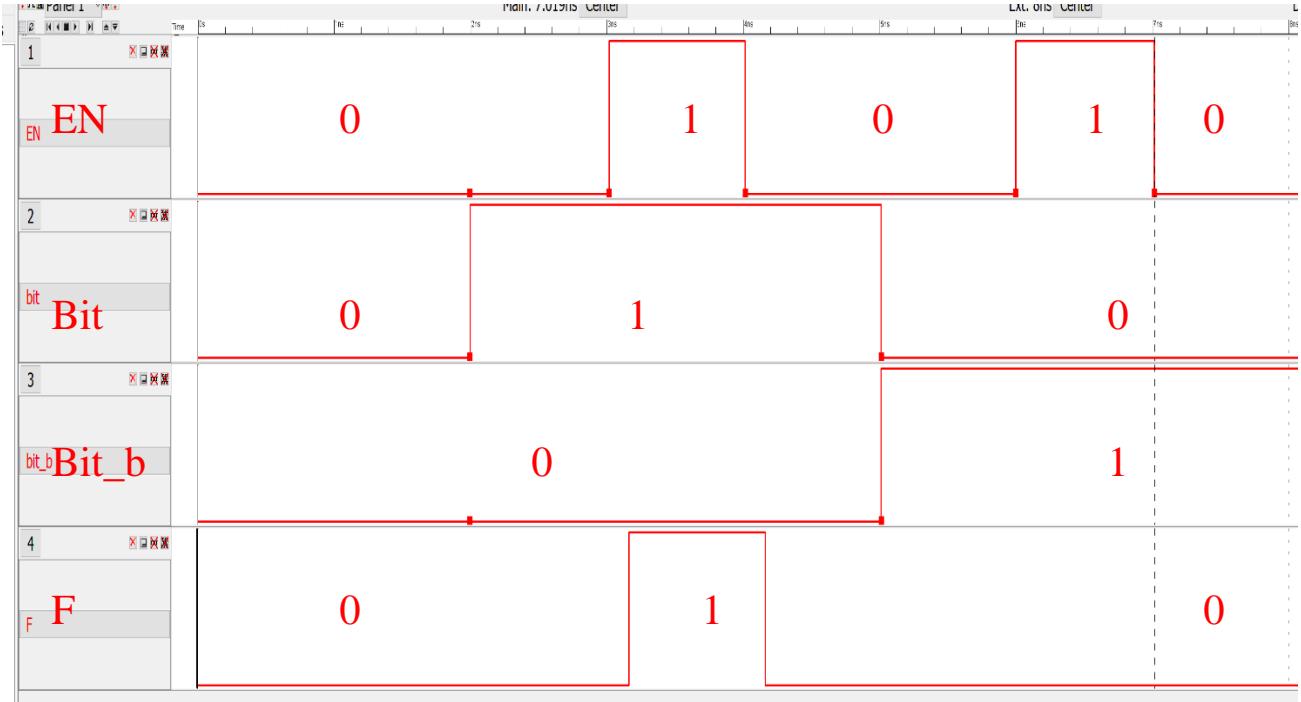


Figure 48: IRSIM Simulation for Sense Amplifier Schematic

Sense Amplifier Layout:

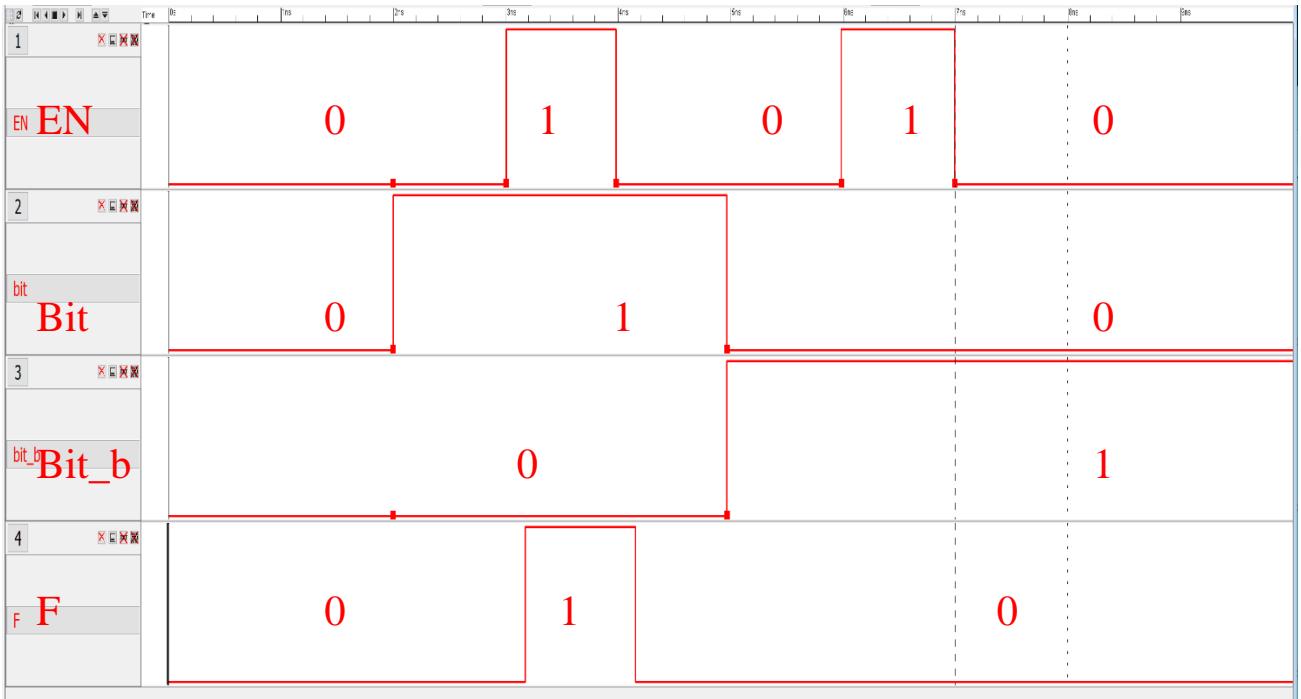


Figure 49: IRSIM Simulation for Sense Amplifier Layout

Write Driver Schematic:



Figure 50: IRSIM Simulation for write driver schematic with data high and low

Write Driver Layout:

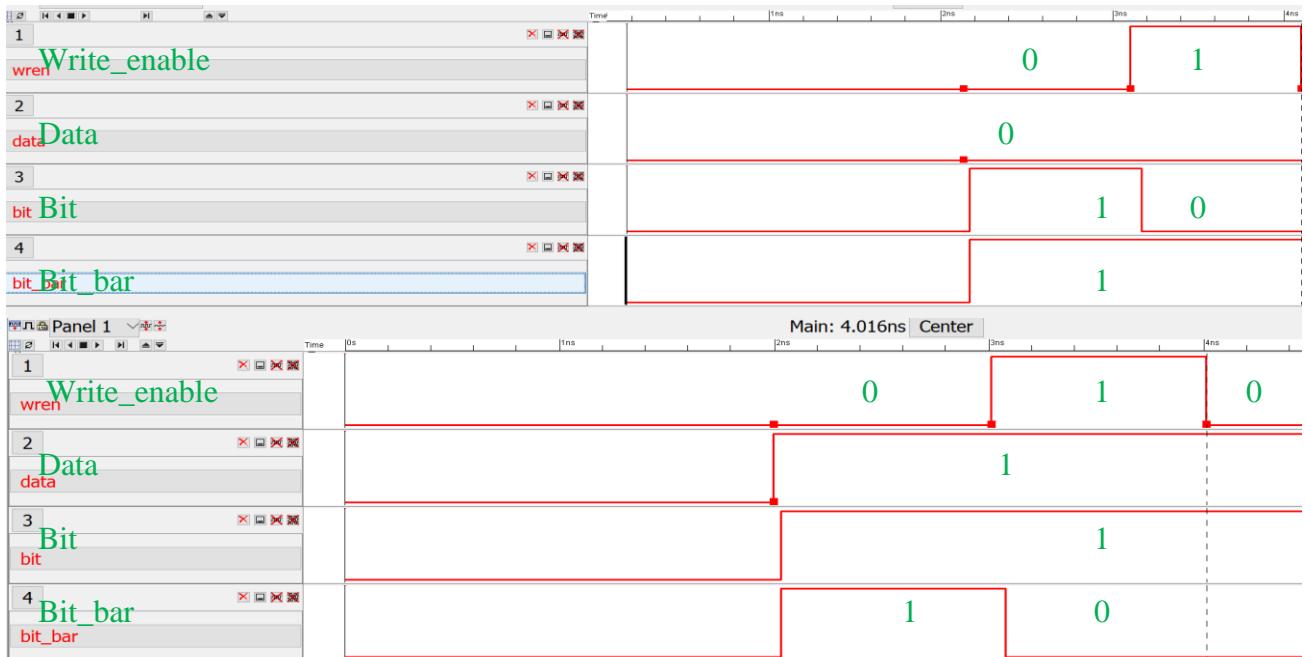


Figure 51: IRSIM Simulation for write driver layout with data high and low

Mux/Demux Schematic:

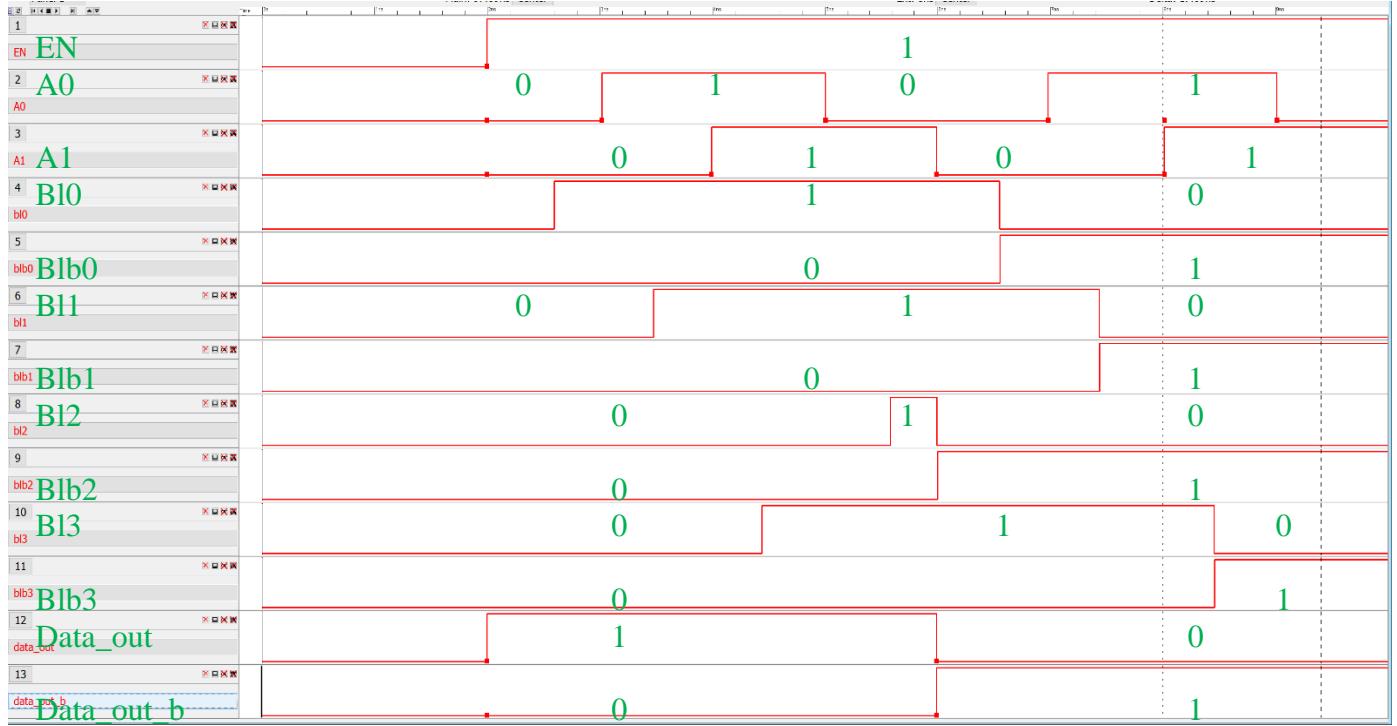


Figure 52: IRSIM for Mux/Demux Schematic

Mux/Demux Layout:



Figure 53: IRSIM for Mux/Demux Layout

8 Bit Word Read Mode Schematic:

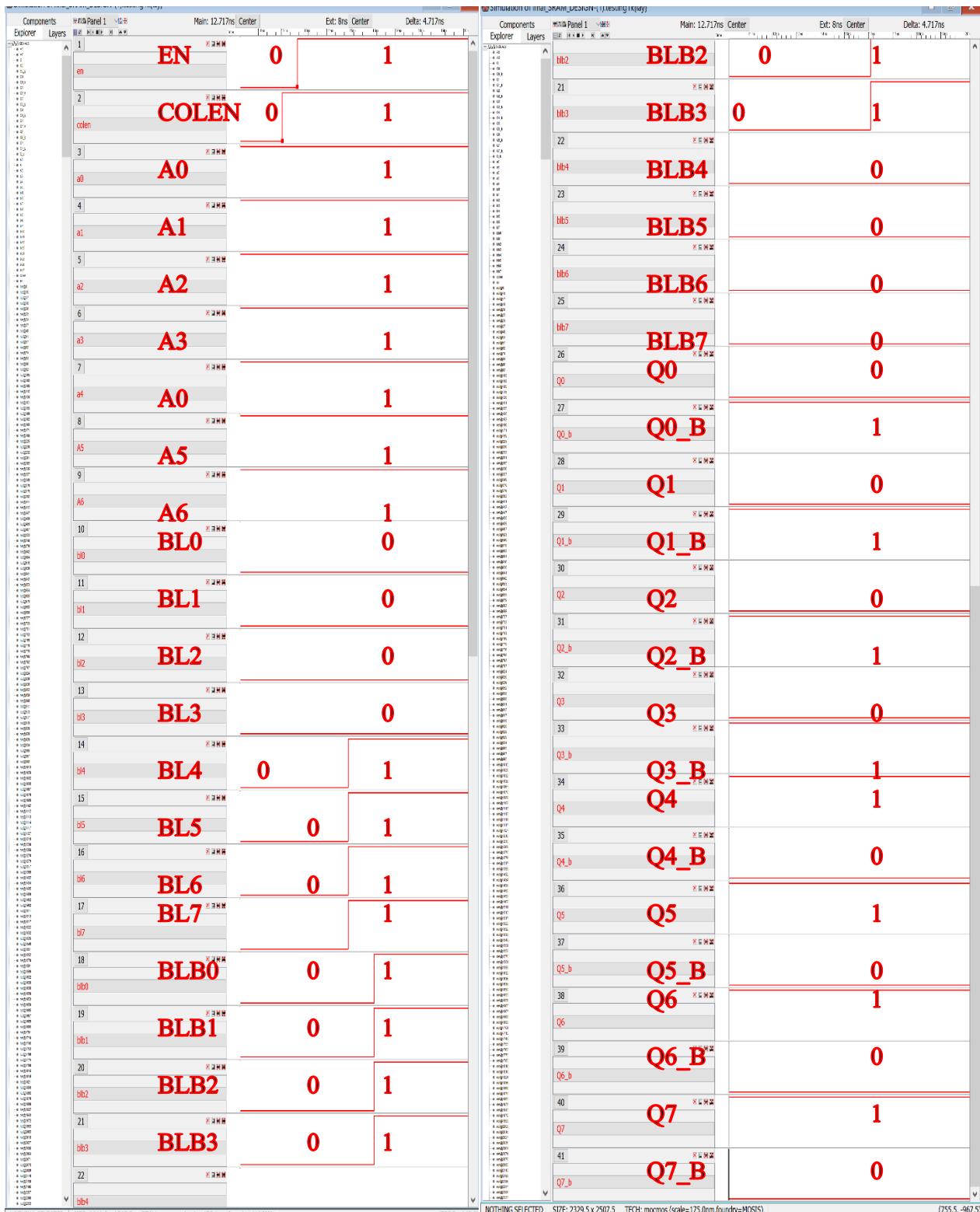


Figure 54: IRSIM for 8 Bit Word Read Mode Schematic

8 Bit Word Write Mode Schematic:

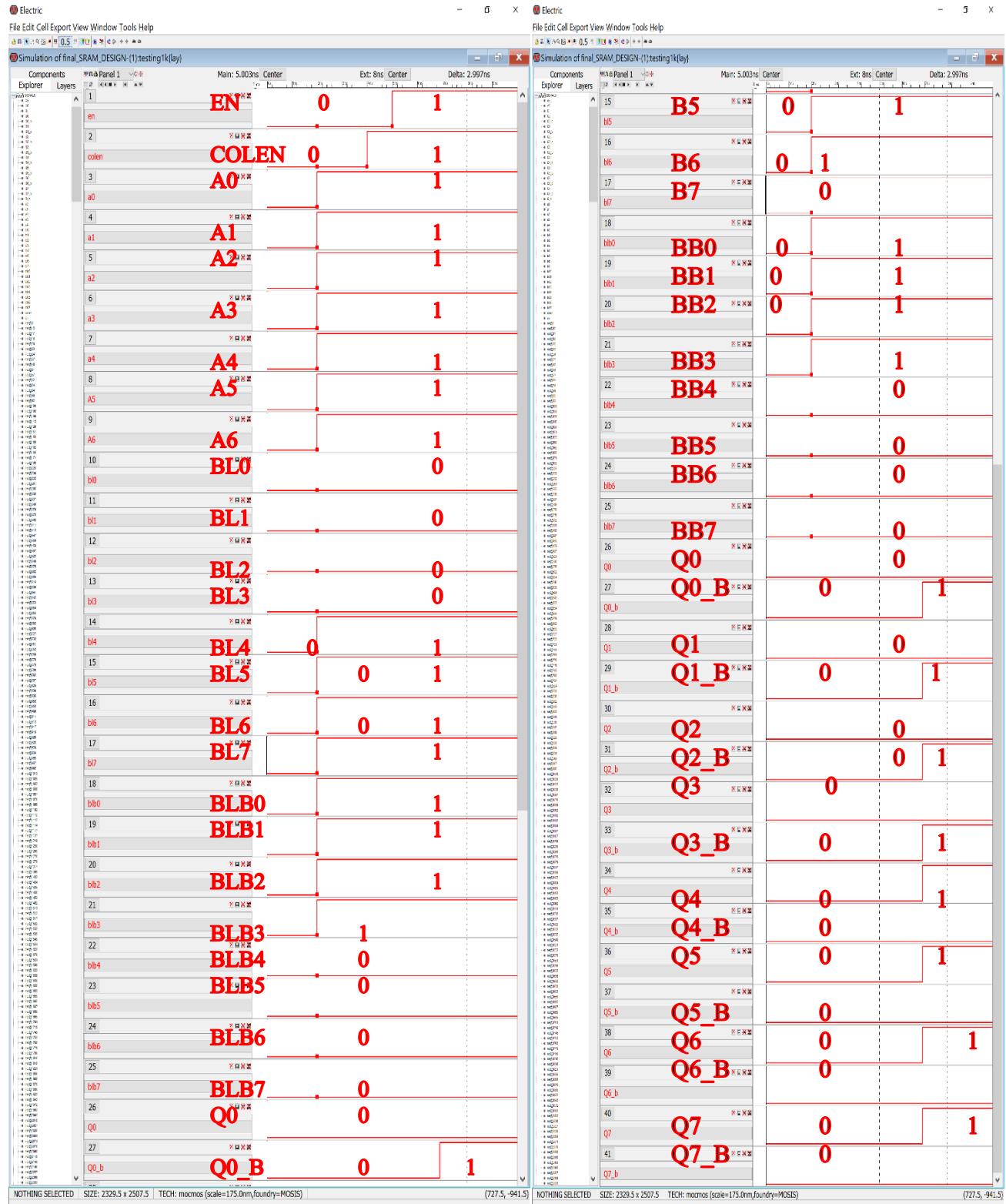


Figure 55: IRSIM for 8 Bit Word Write Mode Schematic

8 Bit Word Read Mode Layout:

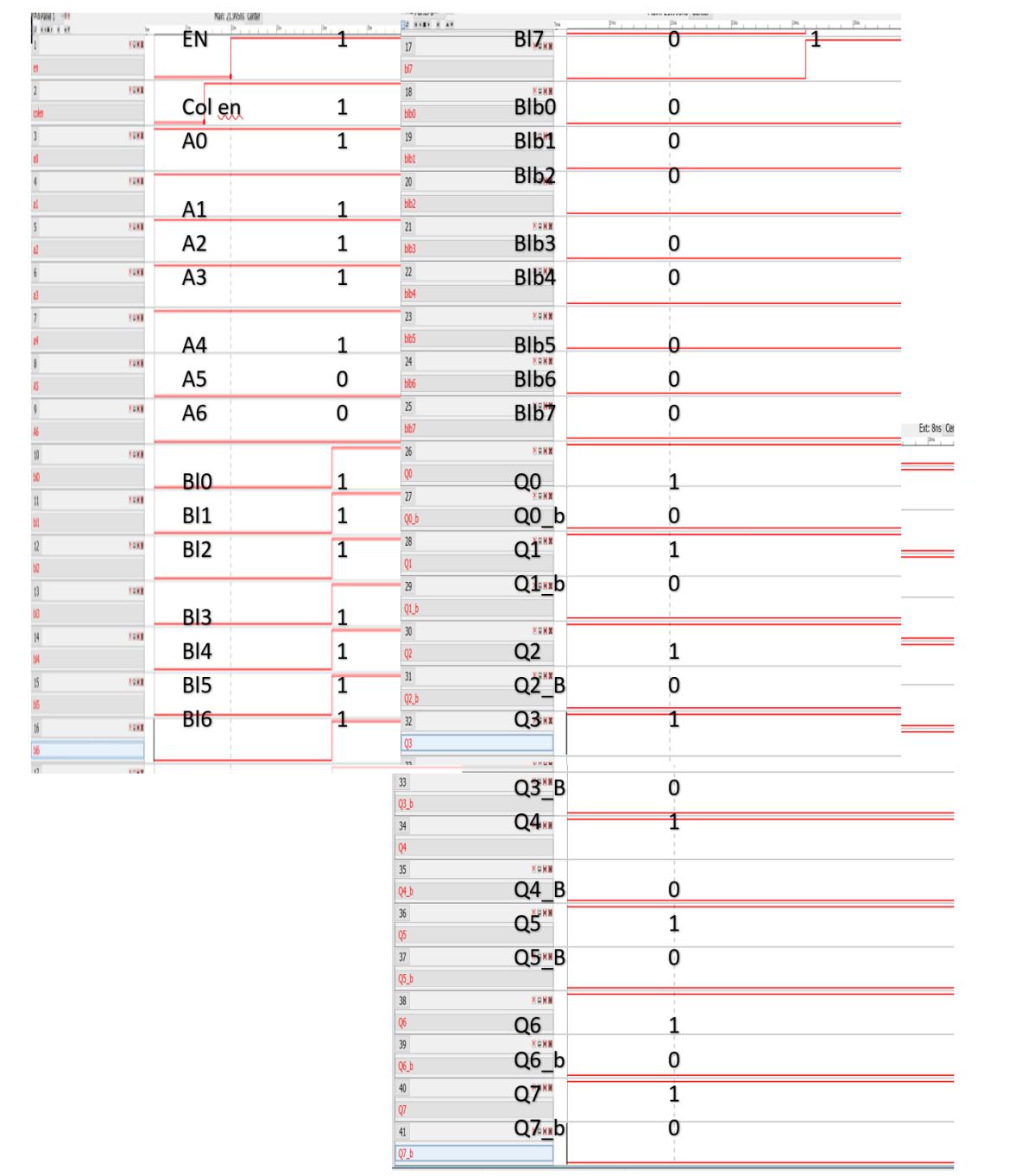


Figure 56: IRSIM for 8 Bit Word Read Mode Layout

8 Bit Word Write Mode Layout:

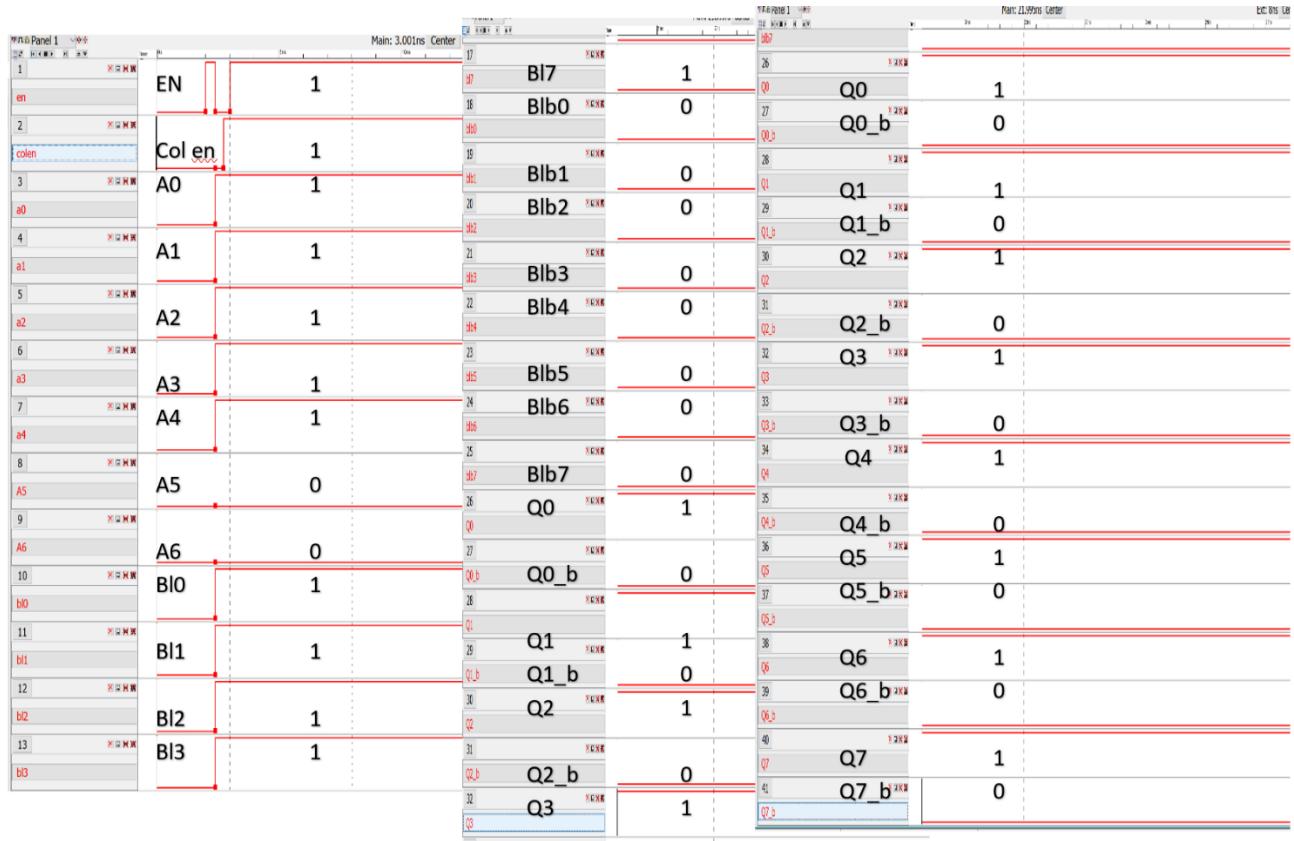


Figure 57: IRSIM for 8 Bit Word Write Mode Layout

Comparison Between all Schematic Waveforms and Layout Waveforms:

If we compare schematic and layout, we can see that they are little different than each other in terms of time delay. Since there are a lot more transistors used in the circuit, the time to turn on and off for all those transistors take extra time to show the output. Schematic takes less space since we don't need to worry about extra components that the layout need. So we can say Layout takes more time in general and it's the same case in this situation. We can learn more accurate delay in the next section where we will simulate and measure the delay in LTSPICE. Other than that, both schematic and layout give the output correctly which match with the truth table.

Section 6: LTSPICE code and parasitic extractions with calculation analysis

6T SRAM Spice Code:

```
* Spice Code nodes in cell cell '6TSRAM{sch}'
vdd vdd 0 DC 5
R1222222222222222 bit 0 1meg
R233333333333 bit_b 0 1meg
C5555555555555555 bit 0 100f
C5555555555553 bit_b 0 100f
Sw_bit_HI vdd bit_SbH 0 switch
Sw_bit_LO bit 0 SbL 0 switch
Sw_bitB_HI vdd bit_b SbBH 0 switch
Sw_bitB_LO bit_b 0 SbBL 0 switch
uSw_bit_HI sbH 0 PWL(0n 0 3.9n 0 4n 1 4.9n 1 5n 0 13.9n 0 14n 1 14.9n 1 15n 0 33.9n 0 34n 1 34.9n 1 35n 0)
uSw_bit_LO sbL 0 PWL(0n 0 23.9n 0 24n 1 30.9n 1 31n 0)
uSw_bitB_HI sbBH 0 PWL(0n 0 13.9n 0 14n 1 14.9n 1 15n 0 23.9n 0 24n 1 24.9n 1 25n 0 33.9n 0 34n 1 34.9n 1 35n 0)
uSw_bitB_LO sbBL 0 PWL(0n 0 3.9n 0 4n 1 10.9n 1 11n 0)
Uword word 0 PWL(0n 0 5n 0 6n 5 9n 5 10n 0 15n 0 16n 5 19n 5 20n 0 25n 0 26n 5 29n 5 30n 0 35n 0 36n 5 39n 5 40n 0)
.model switch Uswitch()
.tran 0 45n
.include C:\Electric\models.txt
.END
```

Figure 58: Spice Code for 6T SRAM

6T SRAM Parasitic Extractions:

```
** Extracted Parasitic Capacitors ***
C0 Q 0 1.428FF
C1 Q_b 0 1.357FF
C2 net@44 0 7.669FF
C3 bit 0 2.773FF
C4 bit_b 0 2.824FF
C5 word#1nmos@7_poly-left 0 0.104FF
C6 word#0nmos@8_poly-right 0 0.102FF
C7 Q#4pin@19 polysilicon-1 0 0.165FF
C8 Q_b#5pin@23 polysilicon-1 0 0.165FF
** Extracted Parasitic Resistors ***
R0 Q#3pmos@3_poly-left Q#3pmos@3_poly-left##0 5.425
R1 Q#3pmos@3_poly-left##0 Q#4pin@19 polysilicon-1 5.425
R2 Q#4pin@19 polysilicon-1 Q#4pin@19 polysilicon-1##0 8.913
R3 Q#4pin@19 polysilicon-1##0 Q#4pin@19 polysilicon-1##1 8.913
R4 Q#4pin@19 polysilicon-1##1 Q#4pin@19 polysilicon-1##2 8.913
R5 Q#4pin@19 polysilicon-1##2 Q 8.913
R6 Q_b#4nmos@5_poly-right Q_b#5pin@23 polysilicon-1 4.65
R7 Q_b#5pin@23 polysilicon-1 Q_b#5pin@23 polysilicon-1##0 9.3
R8 Q_b#5pin@23 polysilicon-1##0 Q_b#5pin@23 polysilicon-1##1 9.3
R9 Q_b#5pin@23 polysilicon-1##1 Q_b#5pin@23 polysilicon-1##2 9.3
R10 Q_b#5pin@23 polysilicon-1##2 Q_b#6pmos@2_poly-left 9.3
R11 Q_b#5pin@23 polysilicon-1 Q_b#5pin@23 polysilicon-1##0 8.913
R12 Q_b#5pin@23 polysilicon-1##0 Q_b#5pin@23 polysilicon-1##1 8.913
R13 Q_b#5pin@23 polysilicon-1##1 Q_b#5pin@23 polysilicon-1##2 8.913
R14 Q_b#5pin@23 polysilicon-1##2 Q_b 8.913
R15 word#0nmos@8_poly-right word#1nmos@7_poly-left 7.75
R16 word#0nmos@8_poly-right word#0nmos@8_poly-right##0 8.783
R17 word#0nmos@8_poly-right##0 word#0nmos@8_poly-right##1 8.783
R18 word#0nmos@8_poly-right##1 word 8.783
R19 word#1nmos@7_poly-left word#1nmos@7_poly-left##0 8.525
R20 word#1nmos@7_poly-left##0 word#1nmos@7_poly-left##1 8.525
R21 word#1nmos@7_poly-left##1 word#1nmos@7_poly-left##2 8.525
R22 word#1nmos@7_poly-left##2 word#4pin@49 polysilicon-1 8.525
R23 Q#11nmos@6_poly-right Q#11nmos@6_poly-right##0 7.75
R24 Q#11nmos@6_poly-right##0 Q#11nmos@6_poly-right##1 7.75
R25 Q#11nmos@6_poly-right##1 Q#11nmos@6_poly-right##2 7.75
R26 Q#11nmos@6_poly-right##2 Q#4pin@19 polysilicon-1 7.75
```

Figure 59: 6T SRAM Parasitic Extractions

2 to 4 Decoder Spice Code:

```
* Spice Code nodes in cell cell '2to4decoder{sch}'
UDD UDD 0 DC 3.3
UGND GND 0 DC 0
VIN A0 0 DC PULSE(0 3.3 0 1p 1p 100n 200n)
VIN1 A1 0 DC PULSE(0 3.3 0 1p 1p 200n 400n)
.tran 0 400n
.meas tr1 time TRIG V(OUT)=0.333 TD=0.2u RISE=1 TARG V(OUT)=2.97 TD=0.2u RISE=1
.meas tf1 time TRIG V(OUT)=2.997 TD=0.2u FALL=1 TARG V(OUT)=0.333 TD=0.2u FALL=1
.include C:\Electric\models.txt
.END
```

Figure 60: Spice Code for 2 to 4 Decoder

2 to 4 Decoder Parasitic Extractions:

```
Mpmos@34 X3 net@457#3pmos@34_poly-left vdd vdd P L=0.35l
** Extracted Parasitic Capacitors ***
C0 net@20#2contact@4_metal-1-polysilicon-1 0 0.149FF
C1 net@7 0 2.548FF
C2 net@1 0 3.054FF
C3 net@20#3contact@20_metal-1-n-act 0 2.003FF
C4 A0 0 0.148FF
C5 net@55#2contact@22_metal-1-polysilicon-1 0 11.006FF
C6 A1 0 0.145FF
C7 net@20#11contact@69_metal-1-polysilicon-1 0 0.149FF
C8 net@182 0 2.578FF
C9 net@181 0 3.054FF
C10 A1#5contact@78_metal-1-polysilicon-1 0 9.56FF
C11 A0#6contact@81_metal-1-polysilicon-1 0 0.237FF
C12 net@215 0 2.504FF
C13 net@214 0 3.054FF
C14 A0#10contact@93_metal-1-polysilicon-1 0 0.155FF
C15 net@248 0 2.548FF
C16 net@247 0 3.054FF
C17 X1 0 1.791FF
C18 net@351 0 3.054FF
C19 en 0 15.311FF
C20 X2 0 1.791FF
C21 net@389 0 3.054FF
C22 X0 0 1.791FF
C23 net@423 0 3.054FF
C24 X3 0 1.791FF
C25 net@457 0 3.115FF
C26 net@20#1pin@3_polysilicon-1 0 0.198FF
C27 net@1#9pin@4_polysilicon-1 0 0.182FF
C28 A0#1pin@11_polysilicon-1 0 0.168FF
C29 net@55#1pin@14_polysilicon-1 0 0.168FF
C30 A1#1pin@17_polysilicon-1 0 0.168FF
C31 net@20#10pin@45_polysilicon-1 0 0.198FF
C32 net@181#6pin@46_polysilicon-1 0 0.182FF
C33 A1#4pin@49_polysilicon-1 0 0.168FF
C34 A0#5pin@53_polysilicon-1 0 0.198FF
C35 net@214#6pin@54_polysilicon-1 0 0.182FF
```

Figure 61: 2 to 4 Decoder Parasitic Extractions

1K SRAM Spice Code:

```
* Spice Code nodes in cell cell '6TSRAM{sch}'
vdd vdd 0 DC 5
R1222222222222222 bit 0 1meg
R233333333333 bit_b 0 1meg
C55555555555555 bit 0 100f
C5555555555553 bit_b 0 100f
Sw_bit_HI vdd bit SbH 0 switch
Sw_bit_LO bit 0 SbL 0 switch
Sw_bitB_HI vdd bit_b SbBH 0 switch
Sw_bitB_LO bit_b 0 SbBL 0 switch
vSw_bit_HI sbH 0 PWL(0n 0 3.9n 0 4n 1 4.9n 1 5n 0 13.9n 0 14n 1 14.9n 1 15n 0 33.9n 0 34n 1 34.9n 1 35n 0)
vSw_bit_LO sbL 0 PWL(0n 0 23.9n 0 24n 1 30.9n 1 31n 0)
vSw_bitB_HI sbBH 0 PWL(0n 0 13.9n 0 14n 1 14.9n 1 15n 0 23.9n 0 24n 1 24.9n 1 25n 0 33.9n 0 34n 1 34.9n 1 35n 0)
vSw_bitB_LO sbBL 0 PWL(0n 0 3.9n 0 4n 1 10.9n 1 11n 0)
Uword word 0 PWL(0n 0 5n 0 6n 5 9n 5 10n 0 15n 0 16n 5 19n 5 20n 0 25n 0 26n 5 29n 5 30n 0 35n 0 36n 5 39n 5 40n 0)
.model switch Uswitch()
.tran 0 45n
.include C:\Electric\models.txt
.END
```

Figure 62: Spice Code for 1K SRAM

1K SRAM Parasitic Extractions:

```
Mpmos@34 X3 net@457#3pmos@34_poly-left vdd vdd P L=0.35I
** Extracted Parasitic Capacitors ***
C0 net@20#2contact@4_metal-1-polysilicon-1 0 0.149FF
C1 net@7 0 2.548FF
C2 net@1 0 3.054FF
C3 net@20#3contact@20_metal-1-n-act 0 2.003FF
C4 A0 0 0.148FF
C5 net@55#2contact@22_metal-1-polysilicon-1 0 11.006FF
C6 A1 0 0.145FF
C7 net@20#11contact@69_metal-1-polysilicon-1 0 0.149FF
C8 net@182 0 2.578FF
C9 net@181 0 3.054FF
C10 A1#5contact@78_metal-1-polysilicon-1 0 9.56FF
C11 A0#6contact@81_metal-1-polysilicon-1 0 0.237FF
C12 net@215 0 2.504FF
C13 net@214 0 3.054FF
C14 A0#10contact@93_metal-1-polysilicon-1 0 0.155FF
C15 net@248 0 2.548FF
C16 net@247 0 3.054FF
C17 X1 0 1.791FF
C18 net@351 0 3.054FF
C19 en 0 15.311FF
C20 X2 0 1.791FF
C21 net@389 0 3.054FF
C22 X0 0 1.791FF
C23 net@423 0 3.054FF
C24 X3 0 1.791FF
C25 net@457 0 3.115FF
C26 net@20#1pin@3_polysilicon-1 0 0.198FF
C27 net@1#9pin@4_polysilicon-1 0 0.182FF
C28 A0#1pin@11_polysilicon-1 0 0.168FF
C29 net@55#1pin@14_polysilicon-1 0 0.168FF
C30 A1#1pin@17_polysilicon-1 0 0.168FF
C31 net@20#10pin@45_polysilicon-1 0 0.198FF
C32 net@181#6pin@46_polysilicon-1 0 0.182FF
C33 A1#4pin@49_polysilicon-1 0 0.168FF
C34 A0#5pin@53_polysilicon-1 0 0.198FF
C35 net@214#6pin@54_polysilicon-1 0 0.182FF
```

Figure 63: 1K SRAM Parasitic Extraction

Section 7: Measurements in LTSPICE for delays for Layout and Schematic

6T SRAM Schematic:



Figure 64: Rising delay for 6T SRAM Schematic

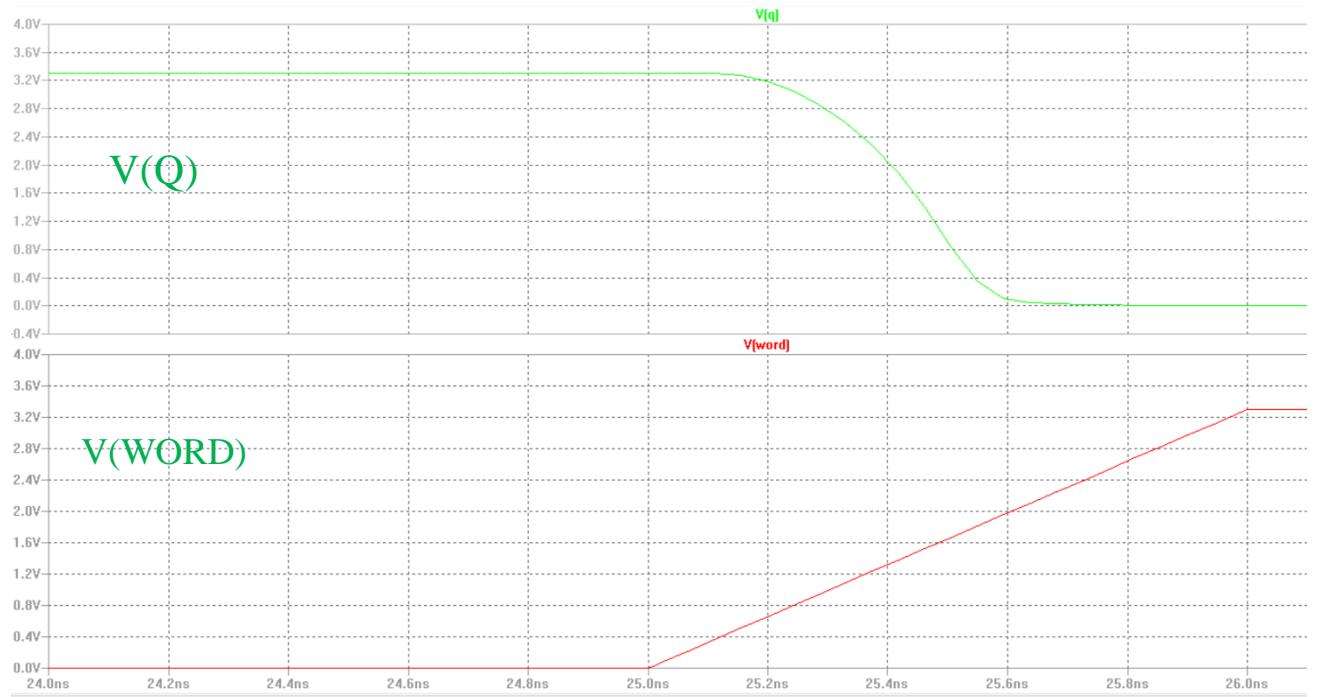


Figure 65: Falling delay for 6T SRAM Schematic

6t schematic

25.417ns, 25.92ns Rise: :0.503ns

25.23ns 25.52ns Fall:0.29ns

$T_{plh} = (25.68 - 25.53) = 0.15\text{ns}$

$T_{phl} = (25.38 - 25.57) = 0.19\text{ns}$

$(tplh+tphl)/2=0.17 = \text{Propagation Delay}$

6T SRAM Layout:

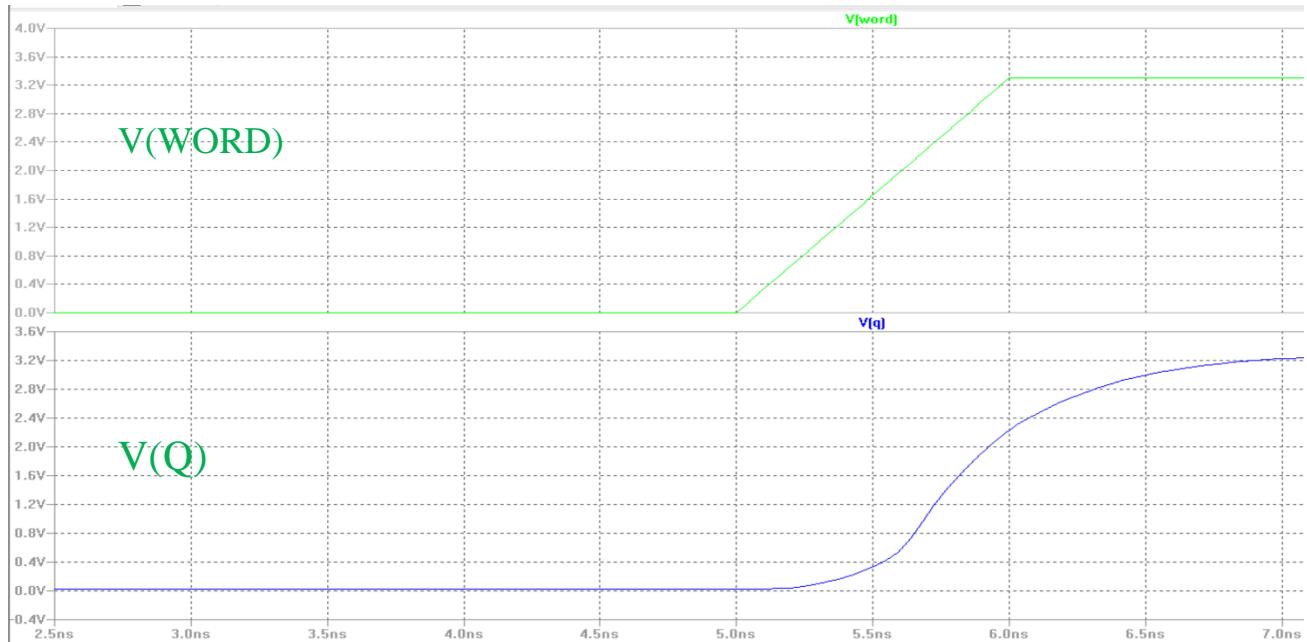


Figure 66: Rising delay for 6T SRAM Layout

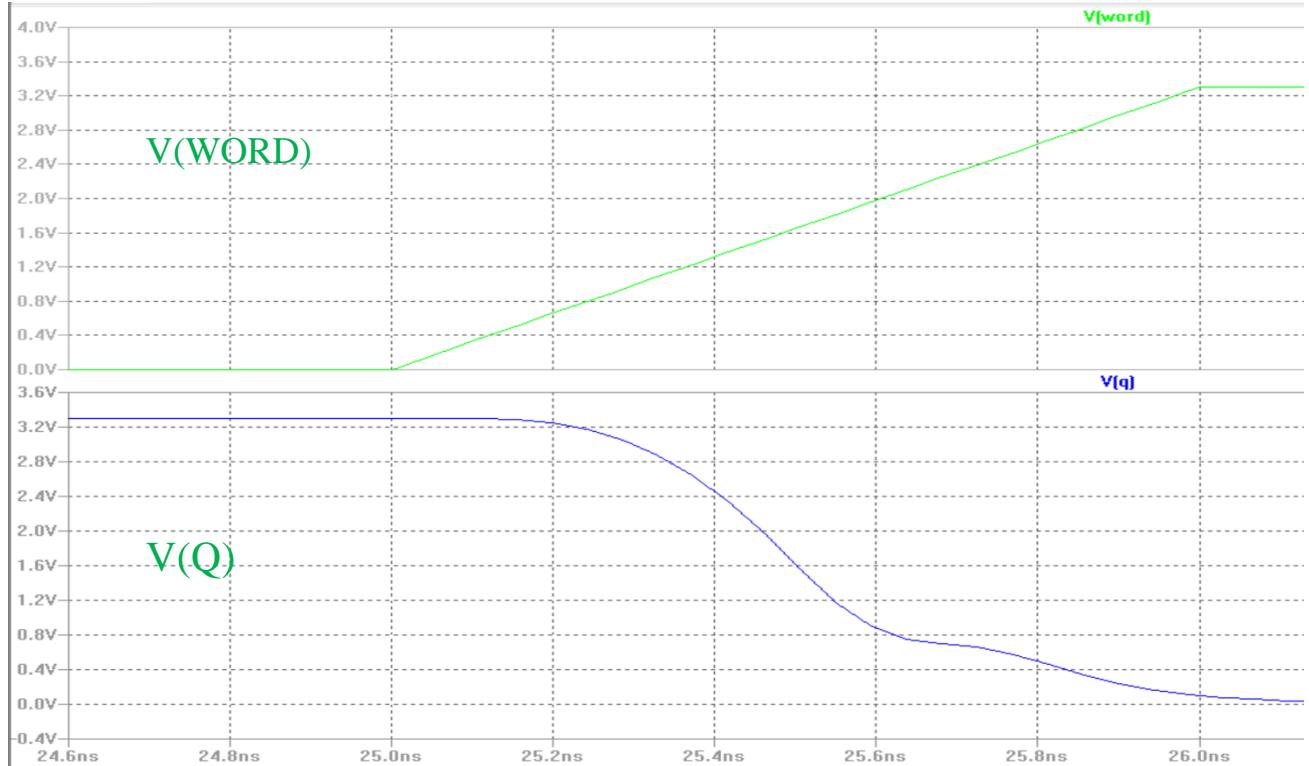


Figure 67: Falling delay for 6T SRAM Layout

6t Layout

Rise Time=(6.43ns-5.56ns)=0.87ns

Fall Time=(25.75-25.26)=0.49ns

Tplh=(5.78ns-5.49)=0.29

Tphl=(25.52ns-25.42ns)=0.1ns

Propagation Delay=0.2ns

From the results above, we can see that the schematic is much faster and has less delay than the layout.

2 To 4 Decoder Schematic:

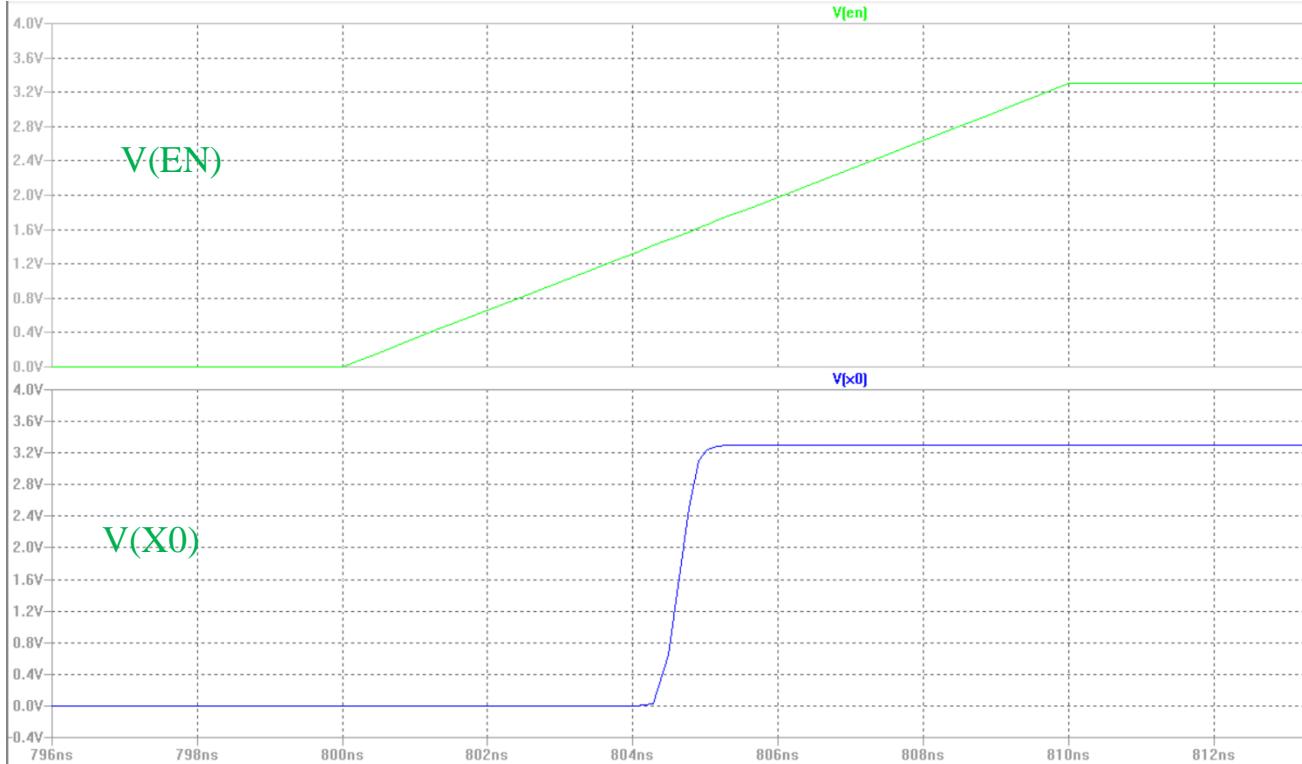


Figure 68: Rising delay for 2 to 4 Decoder Schematic

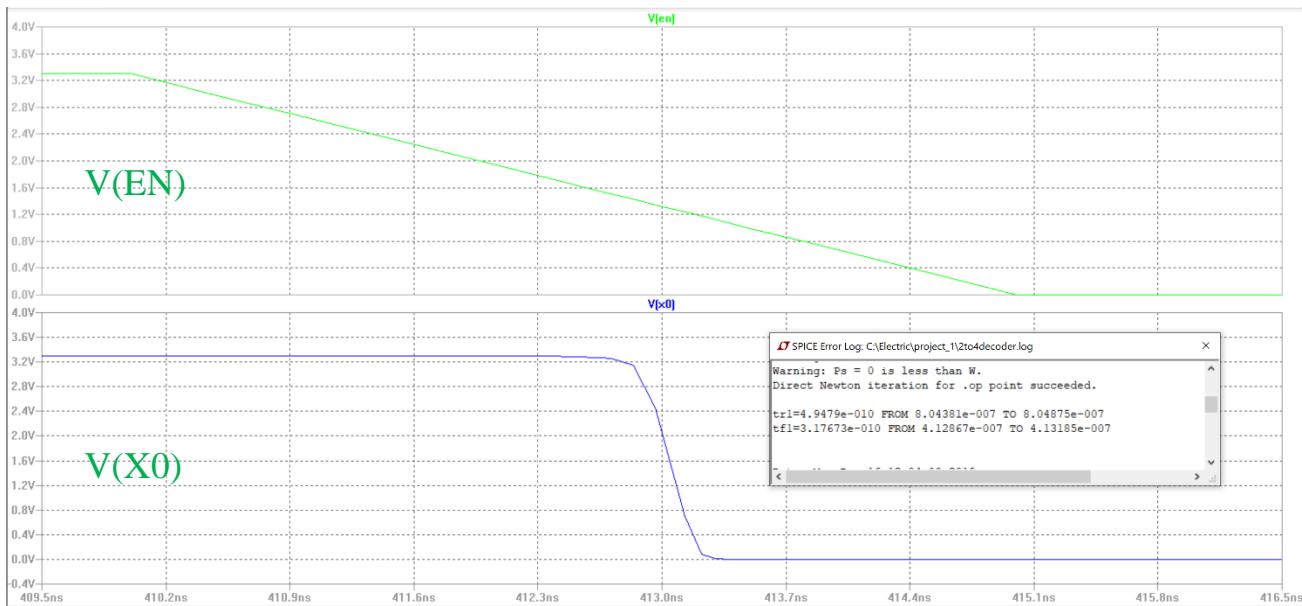


Figure 69: Falling delay for 2 to 4 Decoder Schematic

2 to 4 decoder schematic

Rise Time: 0.494ns

Fall Time: 0.317ns

$$T_{plh} = 804.88\text{ns} - 804.978 = 0.098\text{ns}$$

$$T_{phl} = 805.02\text{ns} - 805.01\text{ns} = 0.01\text{ns}$$

$$\text{Propagation Delay} = (0.098 + 0.1) / 2 = 0.099\text{ns}$$

2 To 4 Decoder Layout:

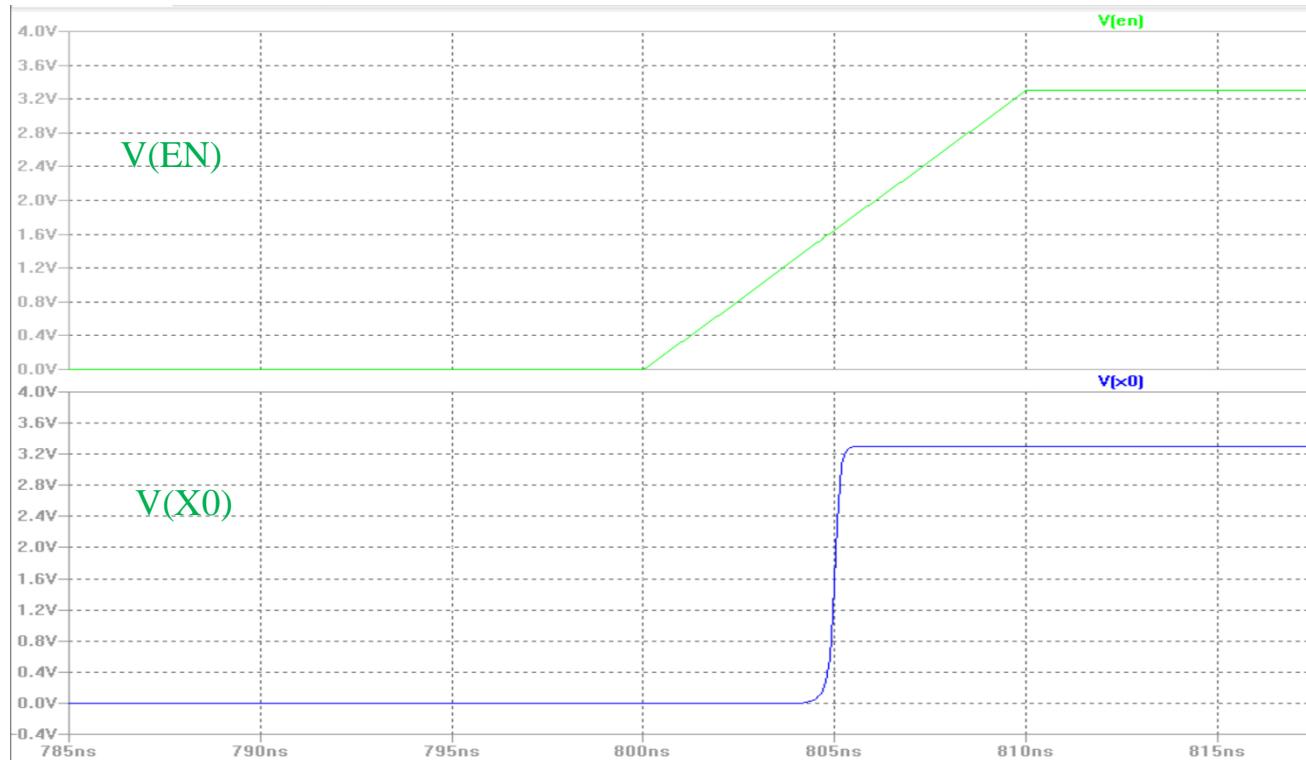


Figure 70: Rising delay for 2 to 4 Decoder Layout

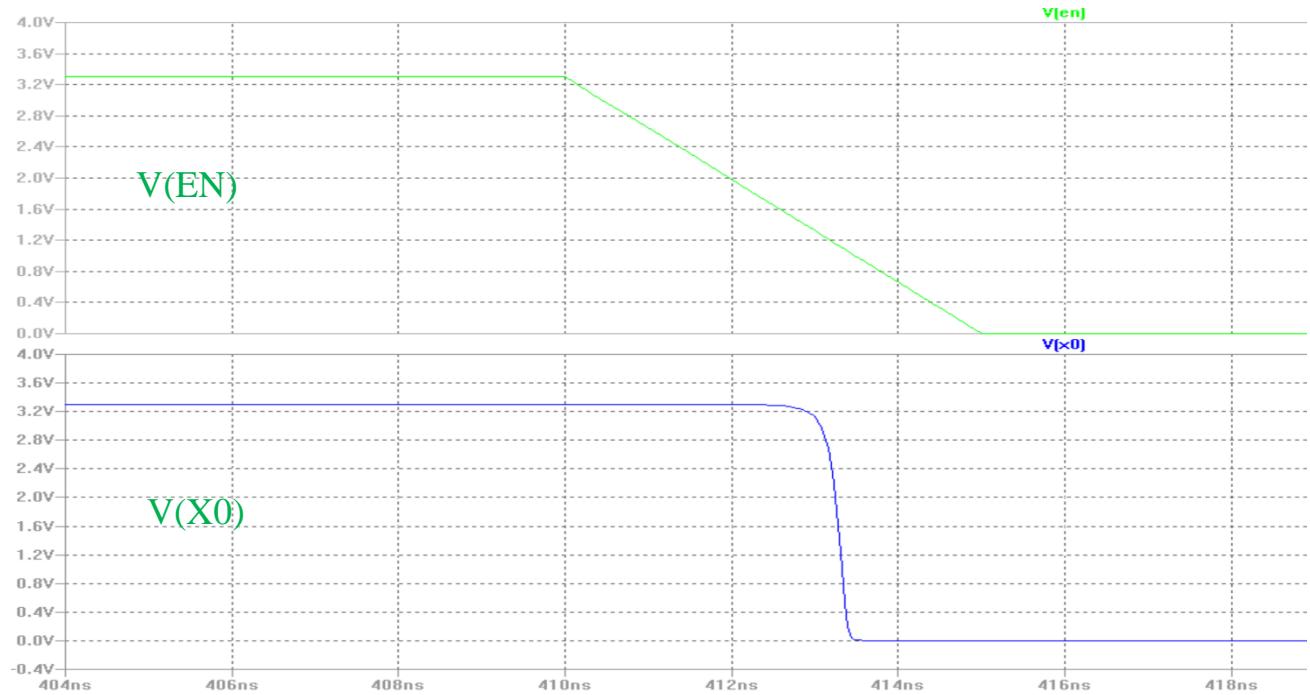


Figure 71: Falling delay for 2 to 4 Decoder Layout

2 to4 decoder layout

Rise Time: 0.478ns

Fall Time: 0.872ns

$$T_{phl}:(413.12\text{ns}-412.29\text{ns})=0.83\text{ns}$$

$$T_{plh}:(805.20\text{ns}-805\text{ns})=0.20\text{ns}$$

$$\text{Propagation Delay: } (0.83\text{ns}+0.20\text{ns})/2=0.515\text{ns}$$

From the result, we can also see that the propagation delay is much larger in the layout.

5 To 32 Decoder Schematic:

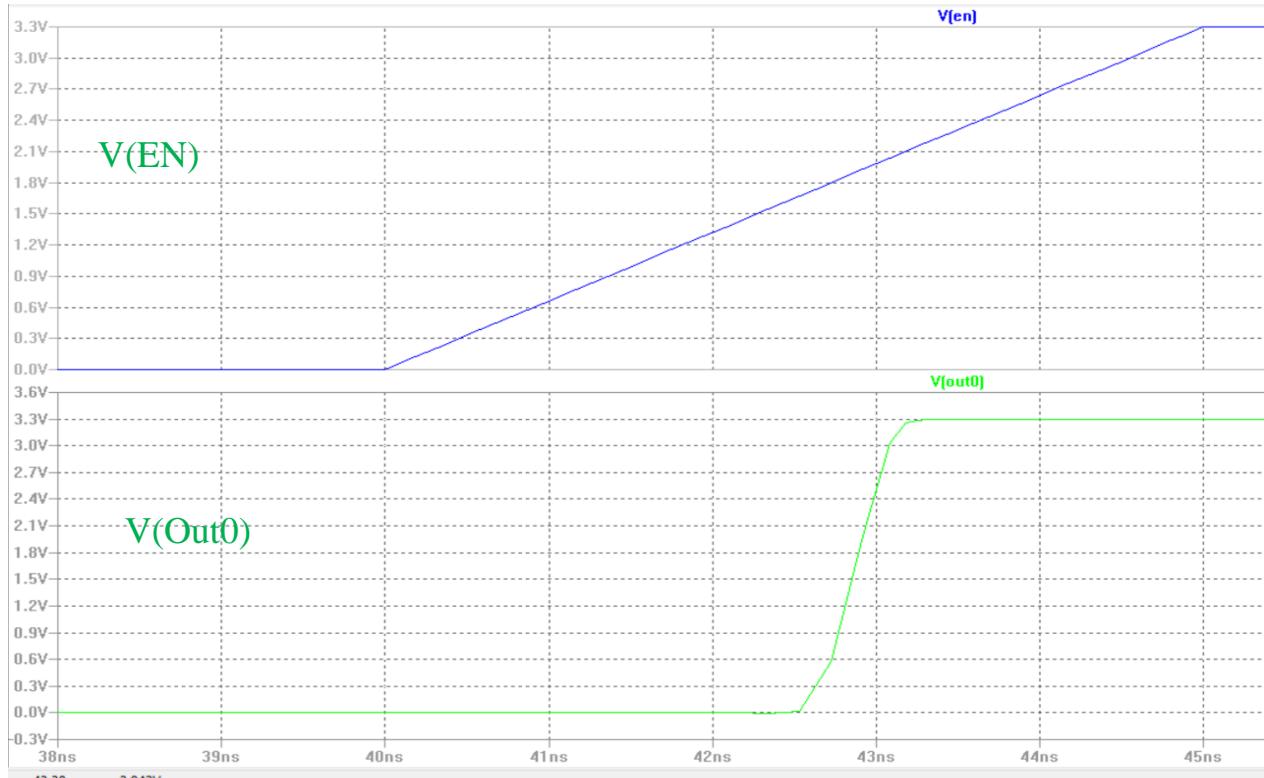


Figure 72: Rising delay for 5 to 32 Decoder Schematic



Figure 73: Falling delay for 5 to 32 Decoder Schematic

5 to 32 schematic

Rise Time: (42.6ns-43.4ns)=0.8ns

Fall Time: (70.2ns-71.157ns)=0.957ns

$T_{plh}=42.6\text{ns}-42.8\text{ns}=0.6\text{ns}$

$T_{phl}=68.4\text{ns}-70.98\text{ns}=2.58\text{ns}$

Propagation Delay=(2.58+0.6)/2=1.59ns

5 to 32 Layout

5 to 32 Decoder Layout:

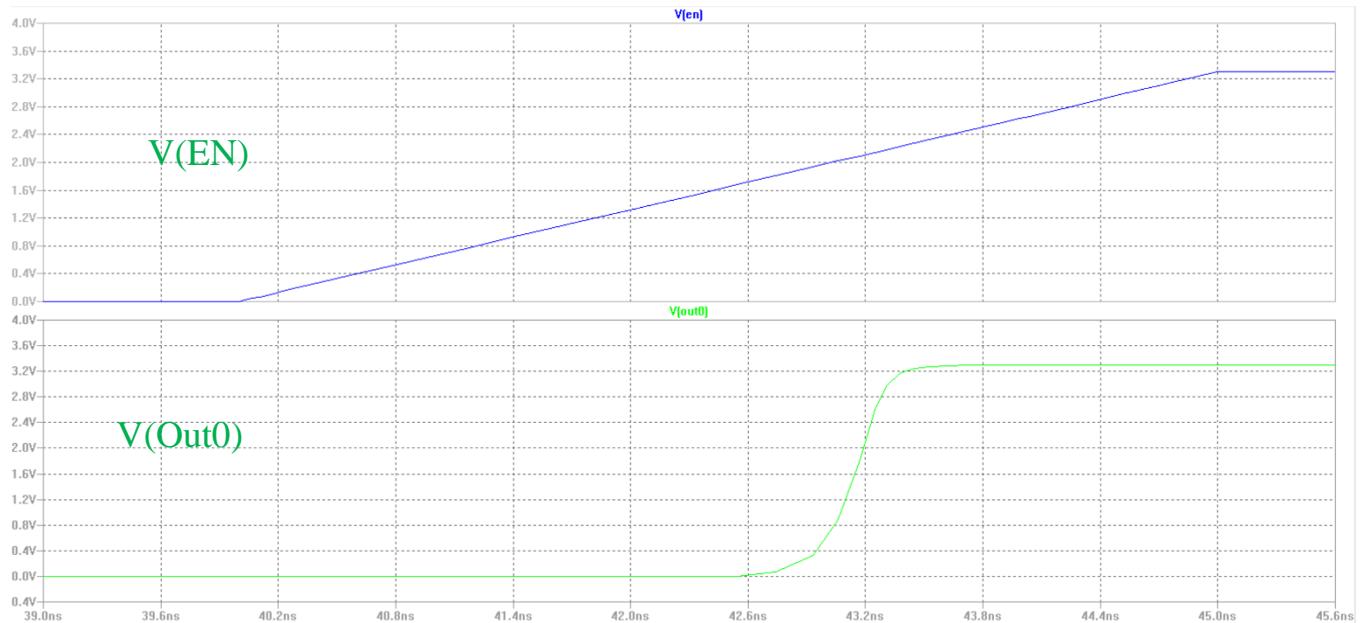


Figure 74: Rising delay for 5 to 32 Decoder Layout

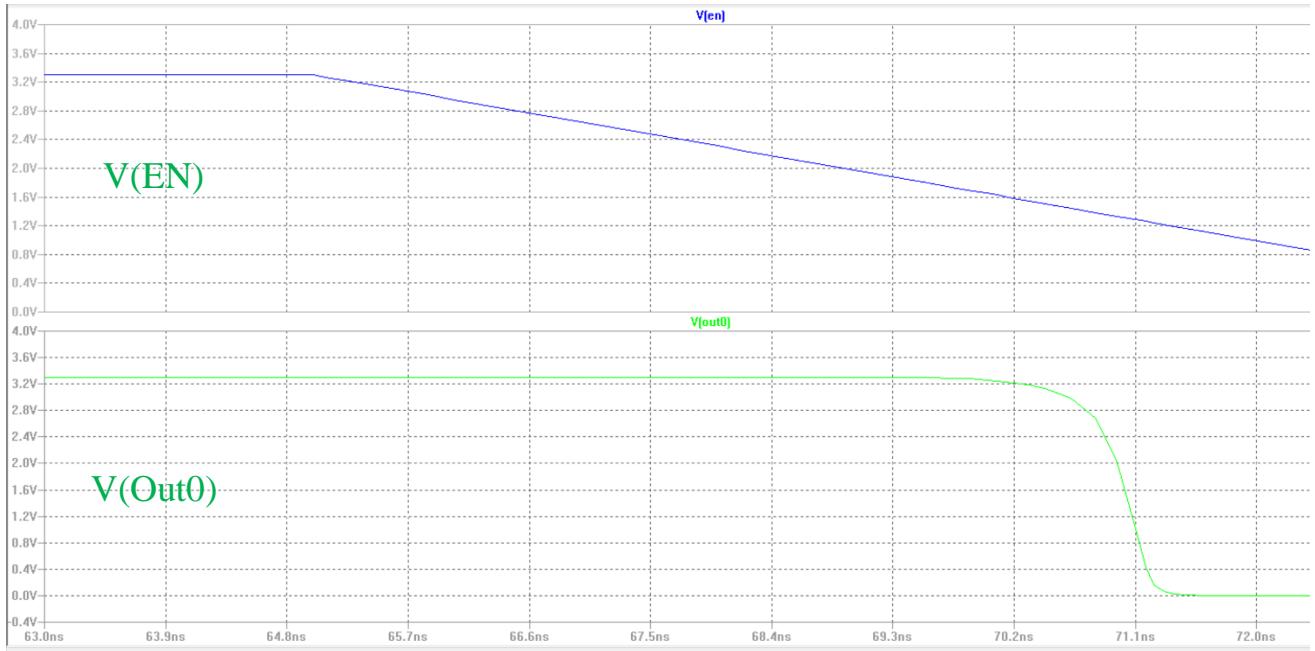


Figure 75: Falling delay for 5 to 32 Decoder Layout

5 to 32 Layout

Rise Time: 0.66ns

Fall Time: 0.84ns

$$T_{plh} = 42.97\text{ns} - 42.84\text{ns} = 0.13\text{ns}$$

$$T_{phl} = 70.89\text{ns} - 68.88\text{ns} = 2.01\text{ns}$$

$$\text{Propagation Delay} = (2.01 + .13)/2 = 1.07\text{ns}$$

Section 8: Measurements of power, delay, chip area, timing, number of transistors for the layout

Power:

6T SRAM Layout:



Figure 76: $I(VDD)$ of 6T SRAM Layout

6T SRAM power = 270 mW

2 to 4 Decoder Layout:

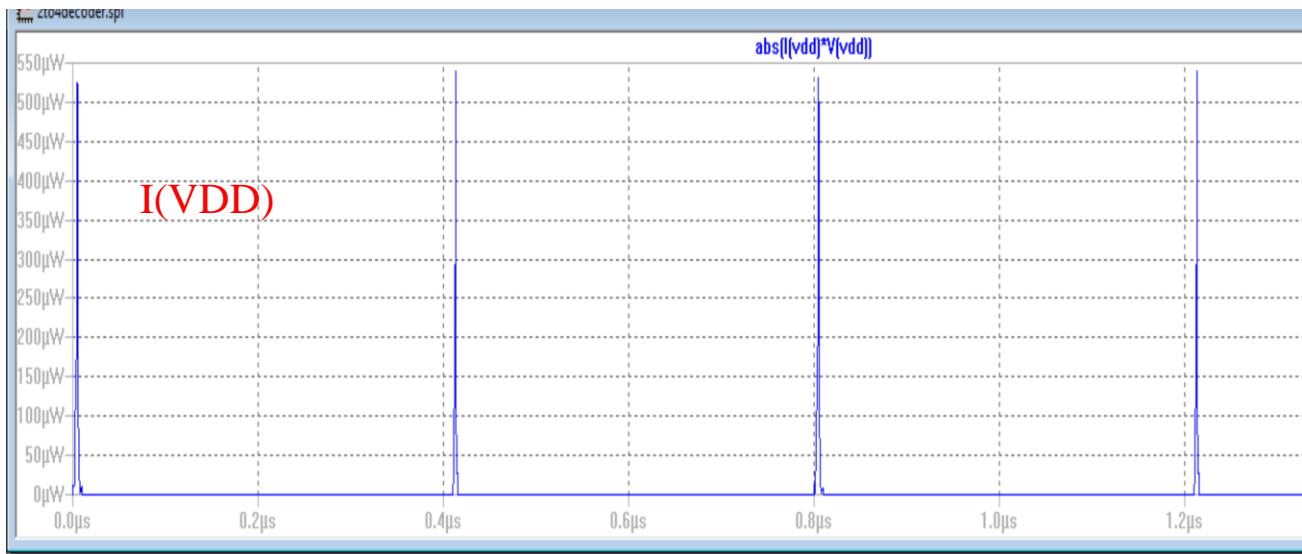


Figure 77: $I(VDD)$ of 2 to 4 Decoder Layout

2 to 4 Decoder power = 545 uW

5 to 32 Decoder Layout:

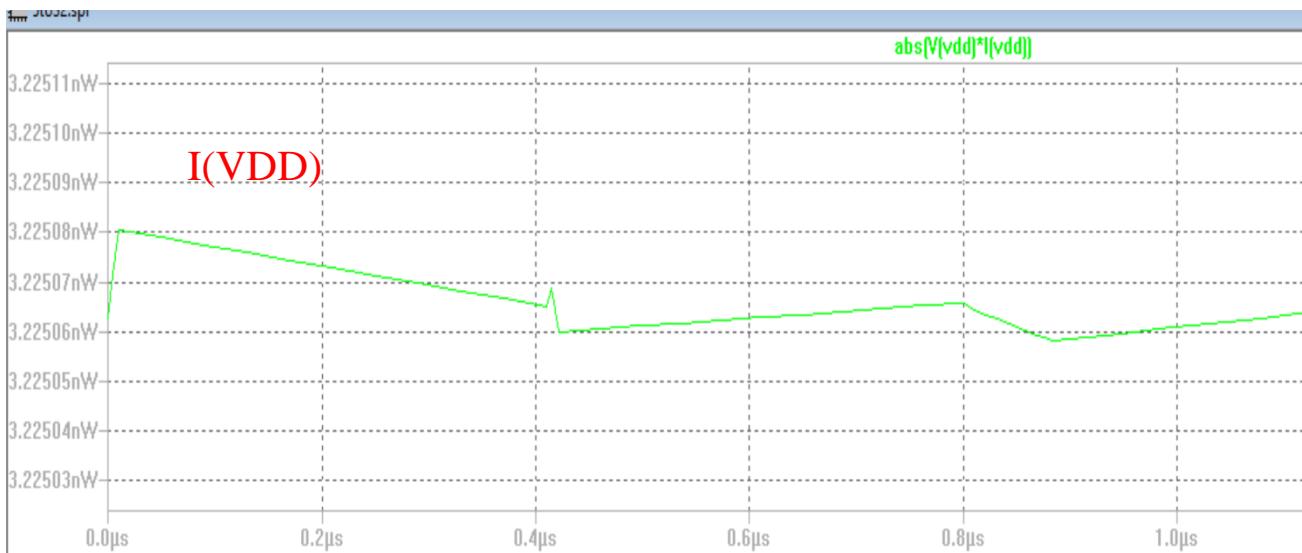


Figure 78: $I(VDD)$ of 5 to 32 Decoder Layout

5 to 32 Decoder power = 3.225 nW

Chip Area:

Chip area of CMOS Layout is = $(2329 \times 2507) \times (300 \times 10^{-3})^2 = 525,492.27 \text{ um}^2$

Number of transistors:

Table 6: Number of Transistors for 1K SRAM layout

	Number of Transistors
6T Cells	6144
5 to 32 Decoder	378
Amplifier	14
Write Driver	8
Mux/Demux	60
Total	= 6604 transistors

Section 9: Conclusion and References:

Conclusion: In this project we have learned how to design a 1K SRAM (Static Random-Access Memory) using conventional CMOS gates. In 1K SRAM we have used 6 transistor memory cells. 1K SRAM contains 1024 cells of 6T SRAM, a total of 6,144 transistors. To store and read data in SRAM we have used decoder, mux/demux and sense amplifier. Decoder was made using Inverter and AND gates. First, we have created a 2 to 4 decoder then used that decoder to design a 5 to 32 decoder with enable that connects to the word lines of the 1K SRAM cell. For the bit line and bit_b line we have used mux/ demux. It was designed by using a 2 to 4 decoder connected to an 8 NMOS all together connected to bit lines and bit_b lines. Output of mux/demux is connected to the write drive and sense amplifier. Lastly, we have used pre charge to charge the bit lines and bit_b lines that is needed before the read operation. To simulate the design, we have used the software called Electric and IRSIM. IRSIM gave us a complete overview of how the waveforms defer for schematic

design and layout design. On the other hand, LTSPICE gave us the propagation delay of the schematic and layout of the circuit.

References:

Chinonso, Ezeogu. "Design Principles of SRAM Memory in Nano-CMOS Technologies." *International Journal of Computer Applications*, vol. 178, no. 11, 2019, pp. 5–11., doi:10.5120/ijca2019918395.

Design and Evaluation of 6T SRAM Layout Designs at Modern ...

https://www.researchgate.net/publication/277075492_Design_and_evaluation_of_6T_SRAM_layout_designs_at_modern_nanoscale_CMOS_processes.

Pavlov, Andrej, and Manoj Sachdev. *CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies: Process-Aware SRAM Design and Test*. Springer, 2008.

Singh, Jawar, et al. *Robust SRAM Designs and Analysis*. Springer, 2013.

Zhang, K., et al. "SRAM Design on 65-Nm CMOS Technology with Dynamic Sleep Transistor for Leakage Reduction." *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, 2005, pp. 895–901., doi:10.1109/jssc.2004.842846.