ECE-GY 6473: COURSE PROJECT
64-BIT MEMORY SYSTEM DESIGN
New York University

I-Ting Chen	Anurag Marwah	Hard Patel	William Xia
itc233@nyu.edu	am8482@nyu.edu	hap338nyu.edu	wx312@nyu.edu

Group 5: Part 2 Submission - 12/10/18

This page is intentionally left blank.

Table of Contents

Table of Abbreviations:

CLK	Clock
RE	Read Enable
WE	Write Enable
WL	Word Line
BL	Bit Line
BL_BAR	Bit Line Bar
PRE	Precharge
SA	Sense Amplifier
SAE	Sense Amplifier Enable
D	Data (Input of flip-flop)
SRAM	Static Random Access Memory

1 SRAM Problem 1: Simulate Read and Write Operations

1.1 Schematic

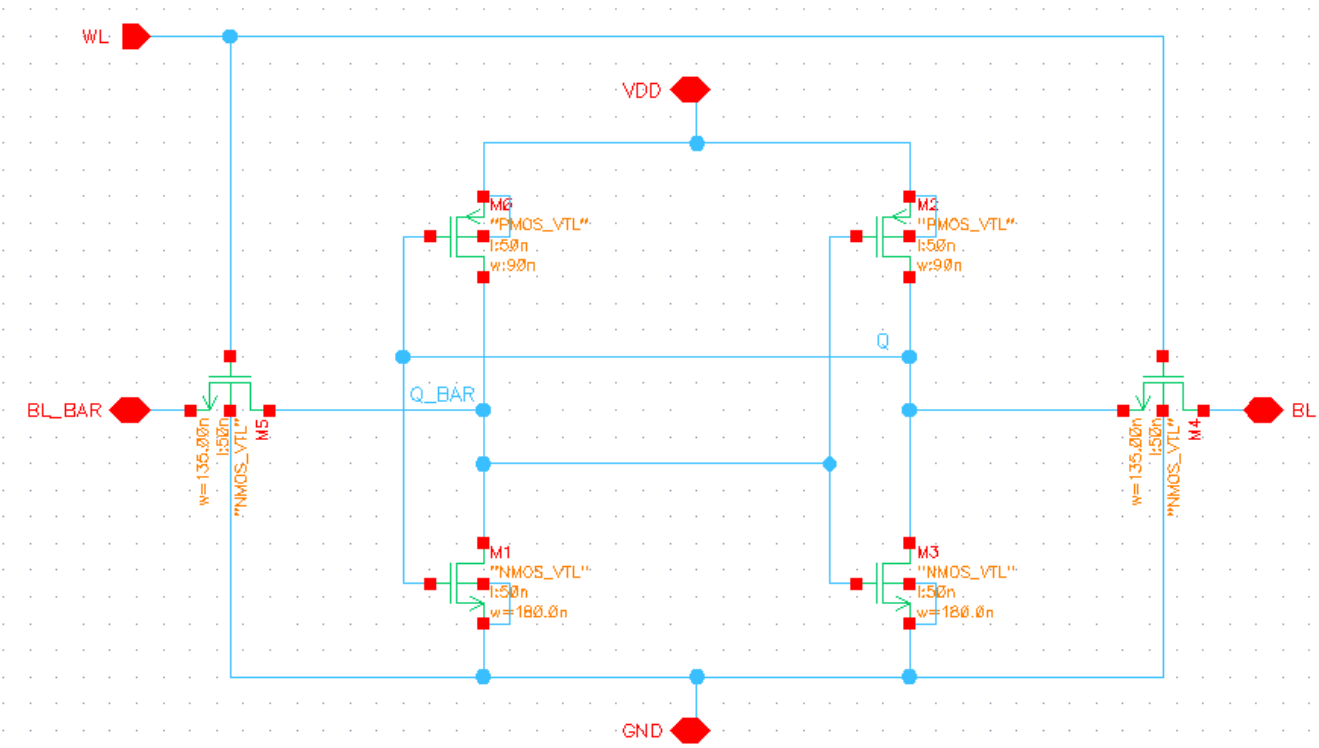


Figure 1: Basic SRAM Cell - Schematic

Figure 1 is the transistor level schematic of the 6-transistor SRAM cell used for memory system. Transistors M0, M1, M2, and M3 form cross-coupled inverters that hold the memory data in the nodes Q and Q_BAR. The access transistors M4 and M5 connect the cell to the bit lines BL and BL_BAR when the WL signal goes high.

1.2 Waveforms

The waveforms shown in Figure 2 display the write and read operations to, and from the SRAM cell. In this section, we assumed the sizing of the SRAM cell to be the ratio $W_{Access} : W_{PullUp} : W_{PullDown} = 1.5 : 1 : 2$. The read and write operations function correctly on the SRAM cell.

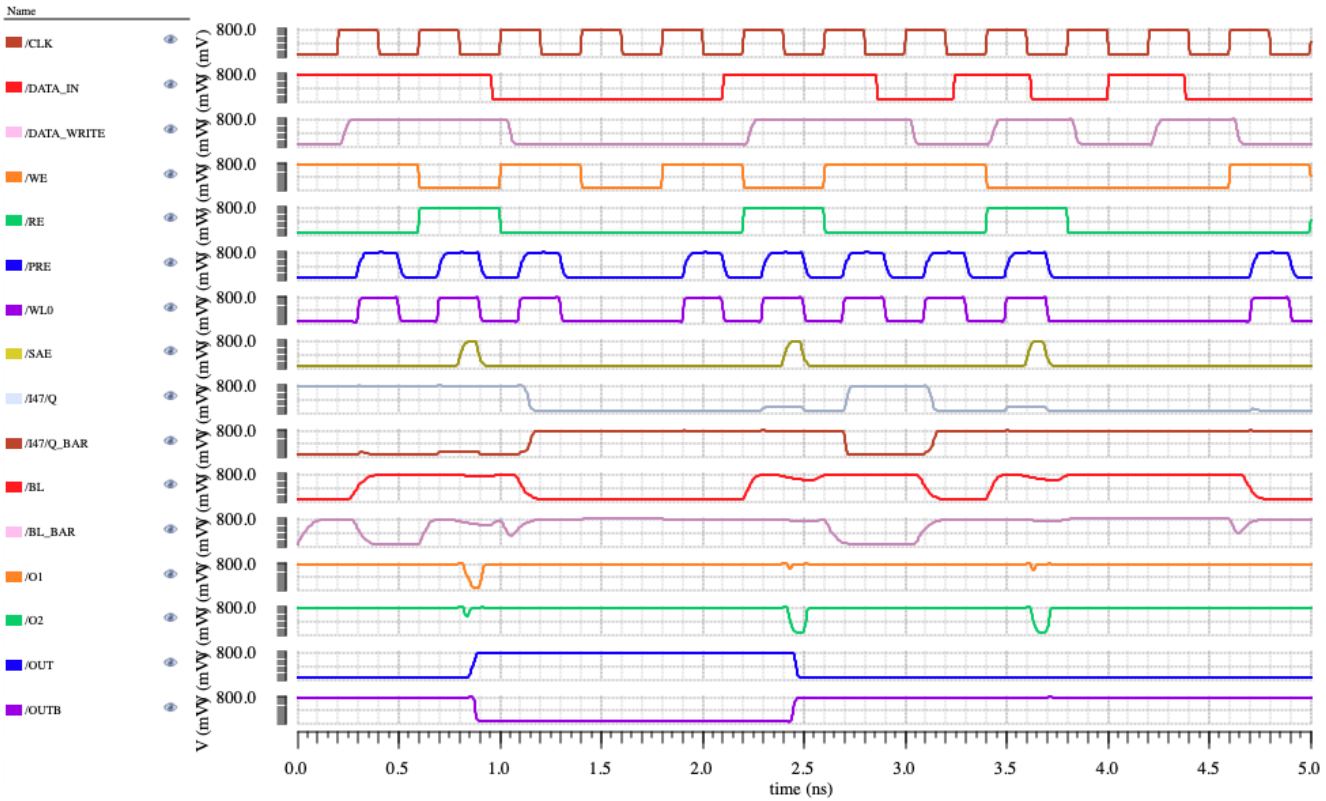


Figure 2: Read and Write Operations (Non-optimized Cell)

2 SRAM Problem 2: Optimized Cell Transistor Design

2.1 Transistor Sizing: Design

Designing the SRAM Cell required careful study of the width ratios between the access, pull-up, and pull-down transistors in the SRAM cell, as shown in Figure 1. The sections below will discuss in depth the process for developing the optimal sizing. The required specifications are as follows:

- Read Noise Margin: $> 25\%$ of $V_{DD} = 187.5mV$
- Write Noise Margin: $> 35\%$ of $V_{DD} = 262.5mV$
- SRAM Cell Area: $0.8\mu m^2$ (single cell)

2.2 Parameters

During the design process, it became clear that the area constraint would be the most difficult specification to meet. Thus, all subsequent sizing choices are made such that the area is be minimized, while still meeting the noise margins for the read and write operations. Please see Appendix A.3 for full tabulation of all experimental sizing used, and their respective noise margins.

2.2.1 Read Margin

The read noise margin depends on two important factors: the tripping voltage of the inverters in the cell, and the read voltage (or read disturb) that occurs when performing a read operation. Please see Appendix A.1 for the DC analysis used for testing and noise margin calculation. Figure 3 is a plot of the widths of all three transistors in the

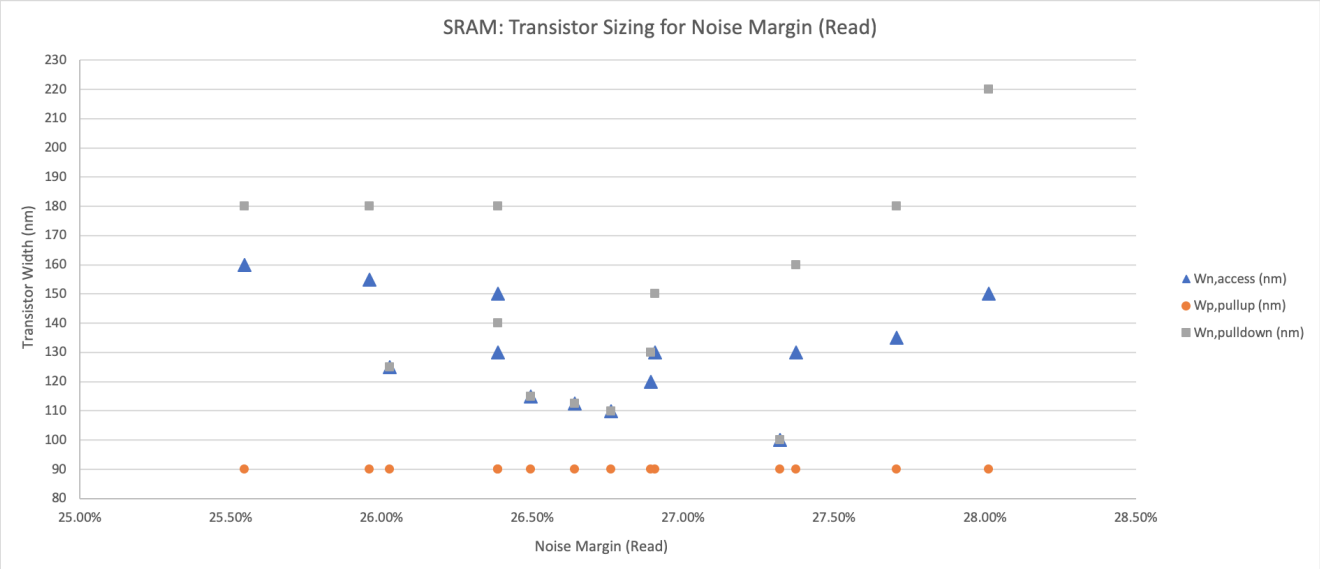


Figure 3: Transistor Sizing for Read Noise Margin

SRAM cell against the read margins found for each sizing. We can see that sizing the access and pull-down transistors the same width, and decreasing them can actually increase the read noise margin at smaller sizes. The behavior shown from sizing $W_{Access} : W_{PullUp} : W_{PullDown} = 110nm : 90nm : 110nm$ down to $100nm : 90nm : 100nm$ reflects this. Since the read margin continues to be satisfied, even as we bring the transistor widths down to minimum size (90nm), we conclude that this specification is not the limiting factor. Thus, only analyzing the sizing relationship with the other constraints will provide the needed information for applicable design choices.

2.2.2 Write Margin

The write noise margin depends on the voltage levels in the bit lines during the write operation. The margin describes the largest error in the bit line voltage for which the SRAM cell can still register the correct data. Please

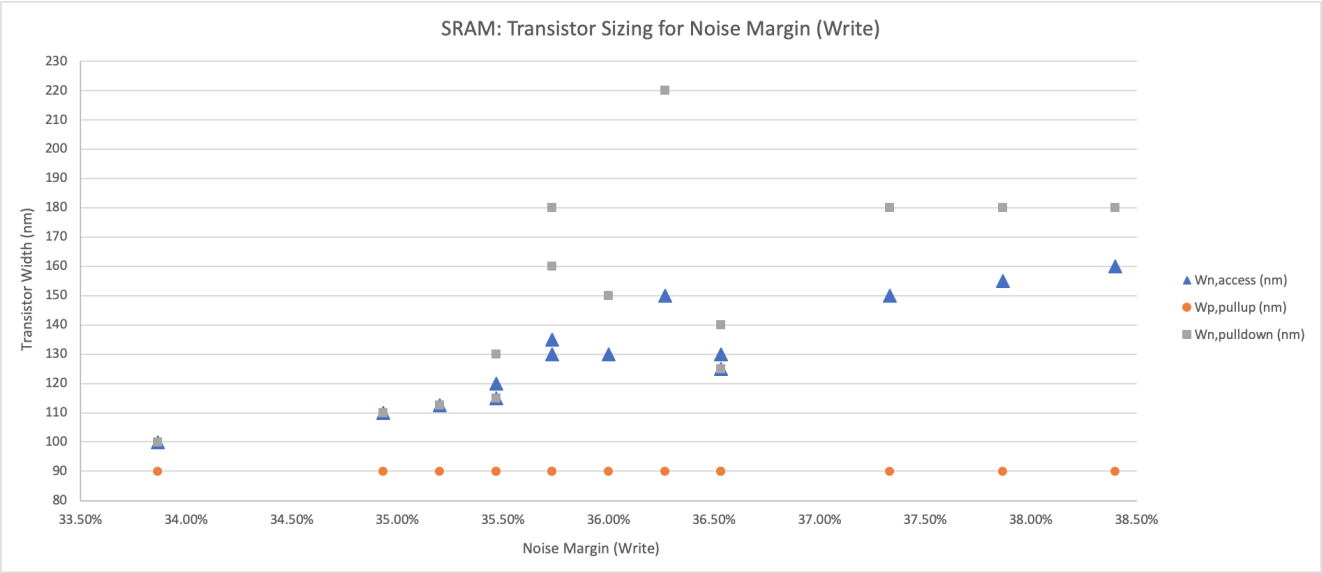


Figure 4: Transistor Sizing for Write Noise Margin

see Appendix A.2 for the circuit and parametric analysis used for testing and noise margin calculation. Similarly as for the read margin, Figure 4 is a plot of the widths of all three transistors in the SRAM cell against the write margins found for each sizing. It is clear that the write noise margin fails to meet specification once the access and pull-down transistor widths jointly reaches 110nm. As our prediction from studying the read margin suggests, the transistor sizing is indeed limited by the write margin. We take the smallest possible ratio from Figure 4 ($W_{Access} : W_{PullUp} : W_{PullDown} = 112.5nm : 90nm : 112.5nm$) and use this ratio for the SRAM design.

2.2.3 Access Time

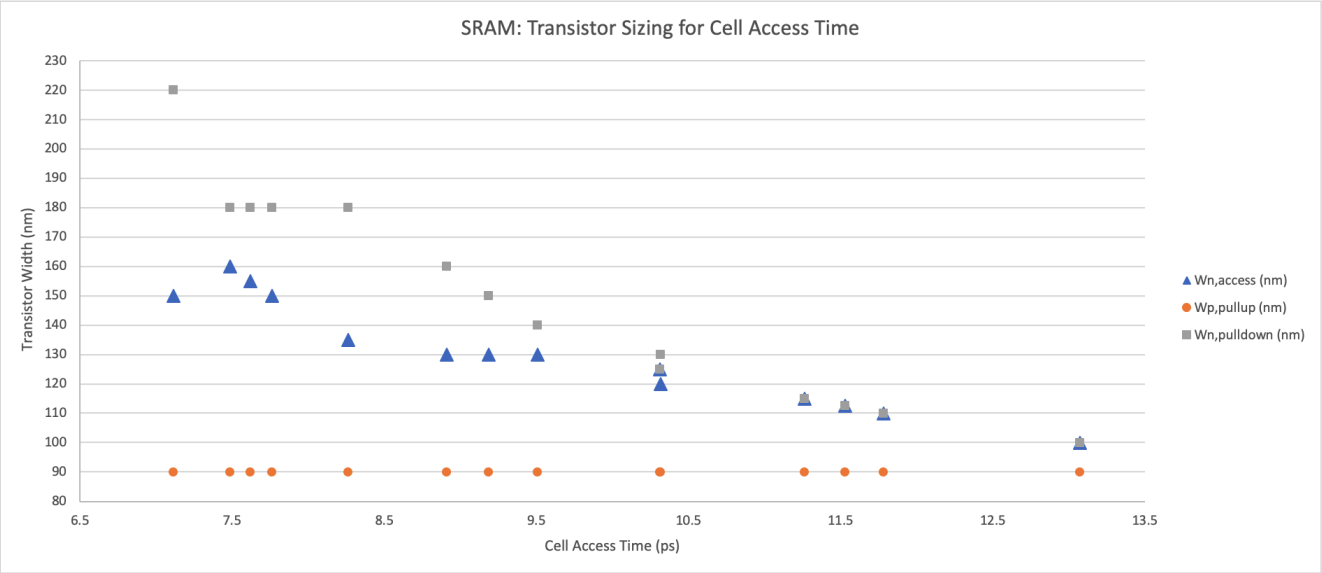


Figure 5: Transistor Sizing for Cell Access Time

The cell access time has been estimated for all the transistor ratios tested using the formula:

$$T_{Access} = \frac{C_{bit} \times \Delta V_{bit}}{I_{read}}$$

Where $C_{bit} = 40fF$ is the assumed bit line capacitance, $V_{bit} = 13mV$ is the sensitivity of our sense amplifer, and I_{read} is the DC current that flows through the access and pull-down transistors during a read operation.

Please see Appendix A.1 the for analysis used to determine I_{read}. The results for all width ratios tested are shown in Figure 5. These trends are expected since smaller transistor sizing increases the amount of time that is needed to access the cell. Note, that this is again not the limiting factor of the circuit.

2.2.4 Cell Area

To calculate the area of the SRAM cell, a formula is devised as below. This formula is only valid for the layout that is designed for the SRAM cell.

$Area = Length \times Width$
$Length = 2 \times [Max\{2 \times Max\{W_{Acc}, W_{pd}\}, W|^{min}_{P-well}\} + 2 \times Spacing|^{min}_{(N-well,P-well)}] + (Max\{2 \times W_{pu}, W|^{min}_{N-well}\})$
$Width = 2 \times Body\,Length\,of\,Mosfet - Overlap\,Length = (2 \times 370) - (30) = 710nm$

Variable	Description
W_{Acc}	Width of Access Transistor
W_{pd}	Width of Pull-down Transistor
W_{pu}	Width of Pull-up Transistor
$W\|^{min}_{P-well}$	Minimum Width of P-well (=200nm)
$W\|^{min}_{N-well}$	Minimum Width of N-well (=200nm)
$Spacing\|^{min}_{(N-well,P-well)}$	Minimum Spacing between N-well and P-well (=225 nm)

Table 1: Variables for Cell Area Formula

2.3 SRAM Cell Design Summary

The final sizing chosen for the SRAM cell was W_{Access} = 112.5nm, $W_{PullDown}$ = 90nm, and W_{PullUp} = 112.5nm. The resulting specifications for our design choice are: Please refer to section 3 for full area calculation details.

Specification	Value
NM_{read}	26.64%
NM_{write}	35.20%
Cell Access Time	11.53ps
Area	$0.781\mu m^2$

Table 2: SRAM Cell Specifications

3 SRAM Problem 3: Optimized Cell Layout

3.1 Layout

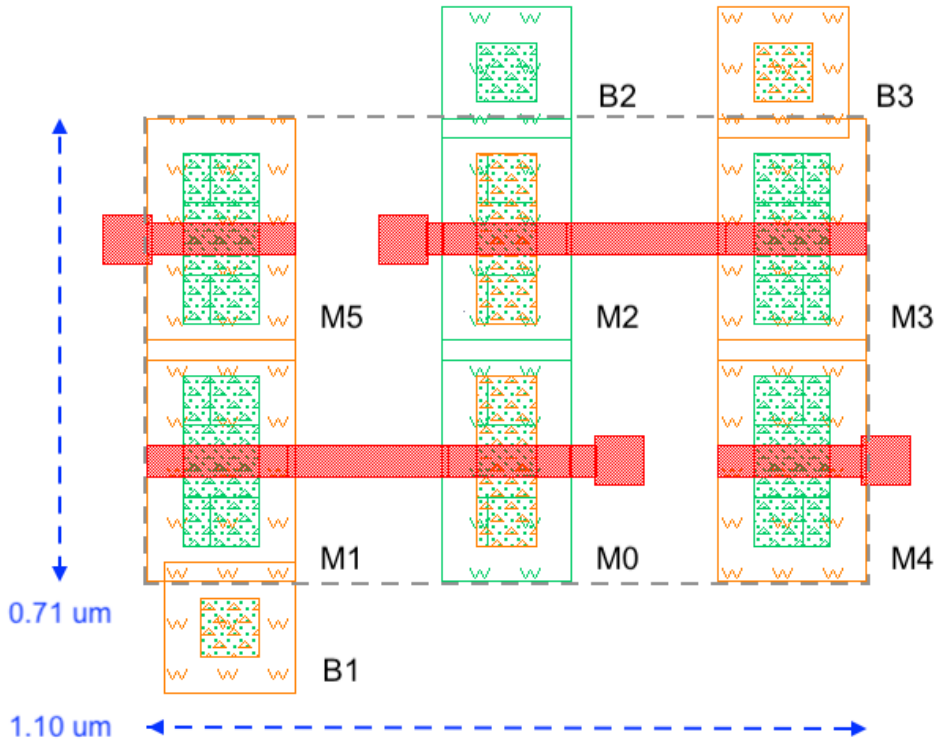


Figure 6: SRAM Cell - Placement

Figure 6 shows the placement of the SRAM cell. M0 and M2 are the Pull-up Transistors, M1 and M3 are the Pull-down transistors, while M4 and M5 are the Access Transistors. B1, B2, and B3 are the body contacts to the P-well, N-well, and P-well respectively. The gates of the transistors are shown in red. M0 and M1 form one cross-coupled inverter, while M2 and M3 form the other cross-coupled inverter. Figure 7 shows the complete layout of the SRAM cell, including higher metal layers. Part (a) shows the layout with Metal1 layer. Part (b) shows the layout with Metal2 layer. The WL signal is routed on the Metal2 layer. Part (c) shows the layout with Metal3 layer. The BL and BL_BAR signals along with the VDD and GND power lines are routed on the Metal3 layer. Part (d) shows the complete layout with Metal1, Metal2, and Metal3 layers.

3.2 Area

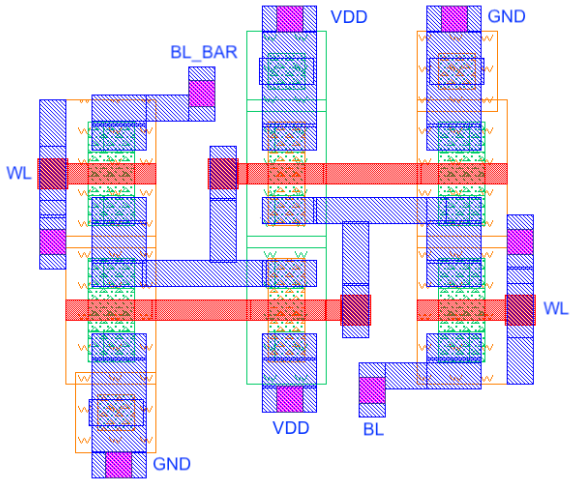
The area of the SRAM cell from edge-to-edge of the N-wells as calculated from Cadence is given below:

$$A = L \times W = 1.1\mu m \times 0.71\mu m = 0.781\mu m^2$$

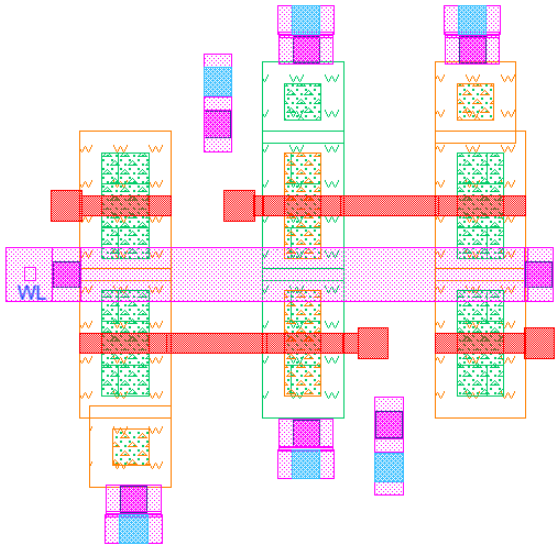
The area of the SRAM cell including the body contacts and the vias to higher metal layers is given below:

$$A = L' \times W' = 1.325\mu m \times 1.23\mu m = 1.629\mu m^2$$

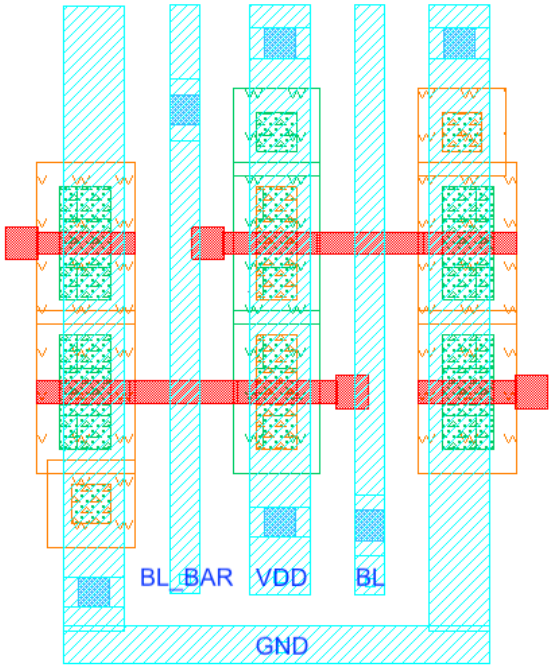
It is to note that the cell is repeatable on both the X and Y axes. So the effective area consumed per bit would be much lesser than $1.629\mu m^2$. This is because of the lithography friendly design practice, like orienting all the transistors in the same direction, and not having a T-junction on the gate lines. These practices make the design physically more realizable.



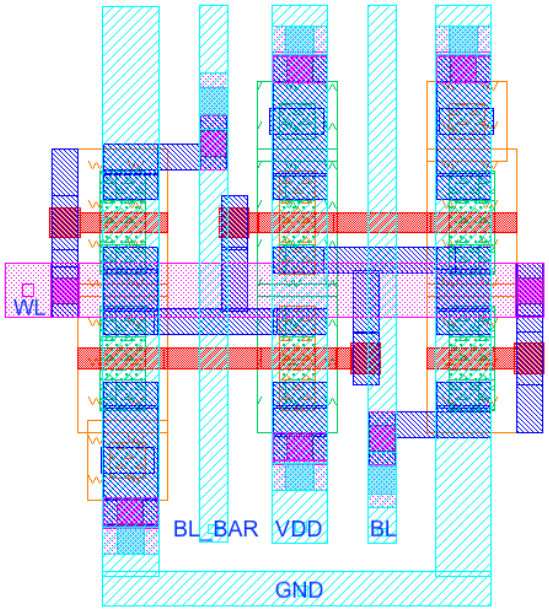
(a) Layout with Metal1 Layer



(b) Layout with Metal2 Layer



(c) Layout with Metal3 Layer

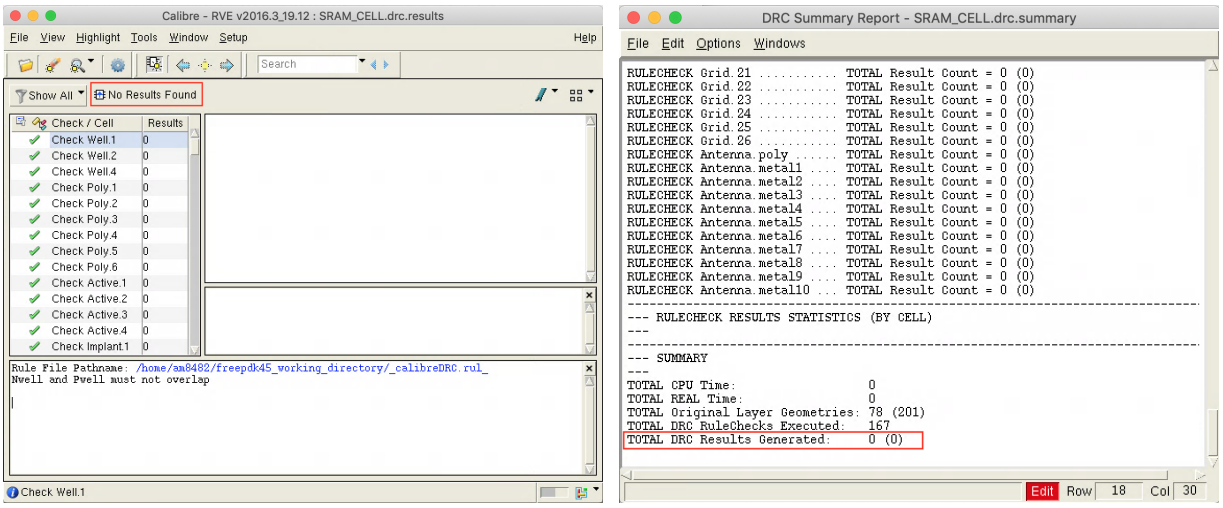


(d) Full Layout

Figure 7: SRAM Cell - Layout

3.3 DRC/LVS Report

There are no DRC errors in the design and the LVS report is also correct. Figure 8 shows a summary of the DRC report. As indicated by the red boxes, there are no DRC errors in the design. Figure 9 shows a summary of the LVS report. As per the report, the design is LVS correct. Please see Appendix B for full DRC and LVS reports.



(a) DRC Results (b) DRC Summary

Figure 8: SRAM Cell - DRC Report

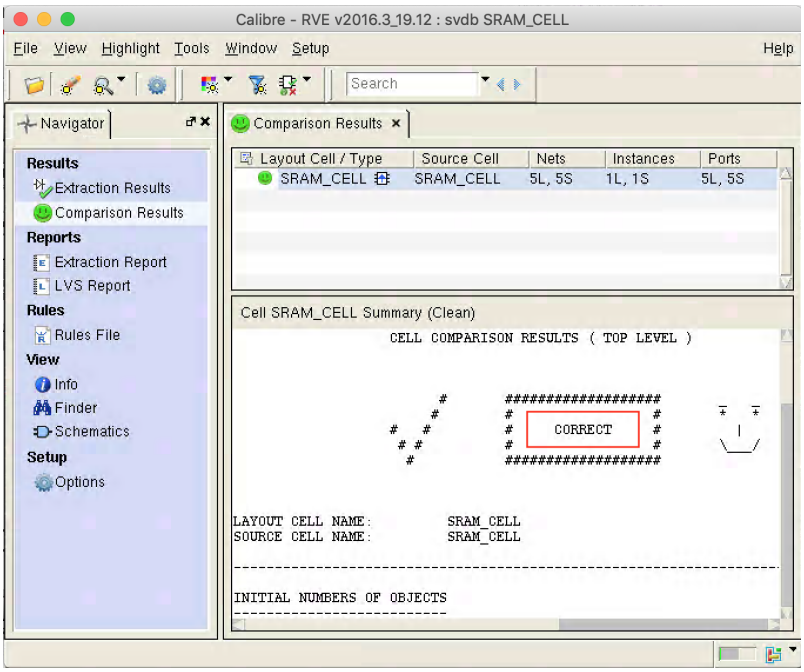


Figure 9: SRAM Cell - LVS Report

4 SRAM Problem 4: Read and Write with Optimized Cell

4.1 Schematic

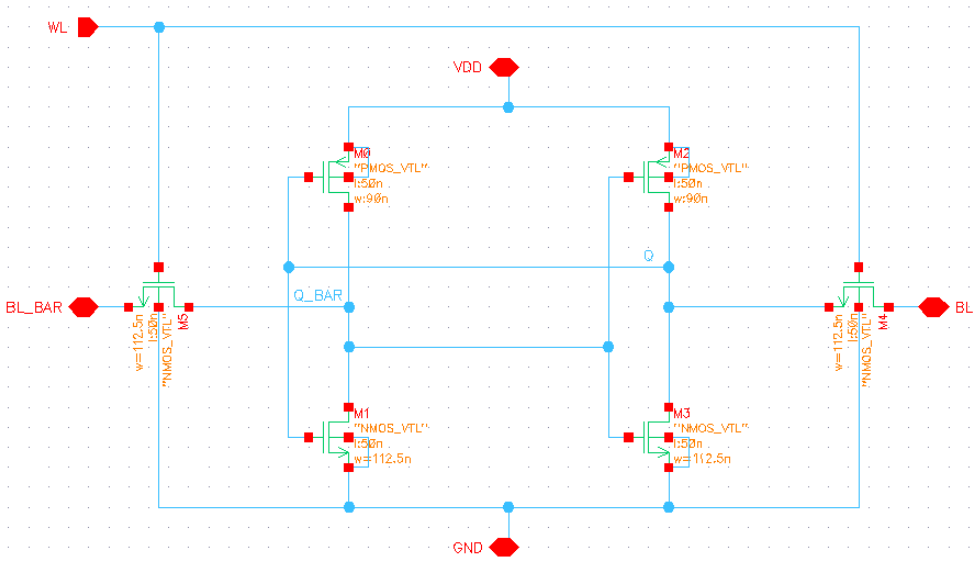


Figure 10: Optimized SRAM Cell - Schematic

4.2 Waveform

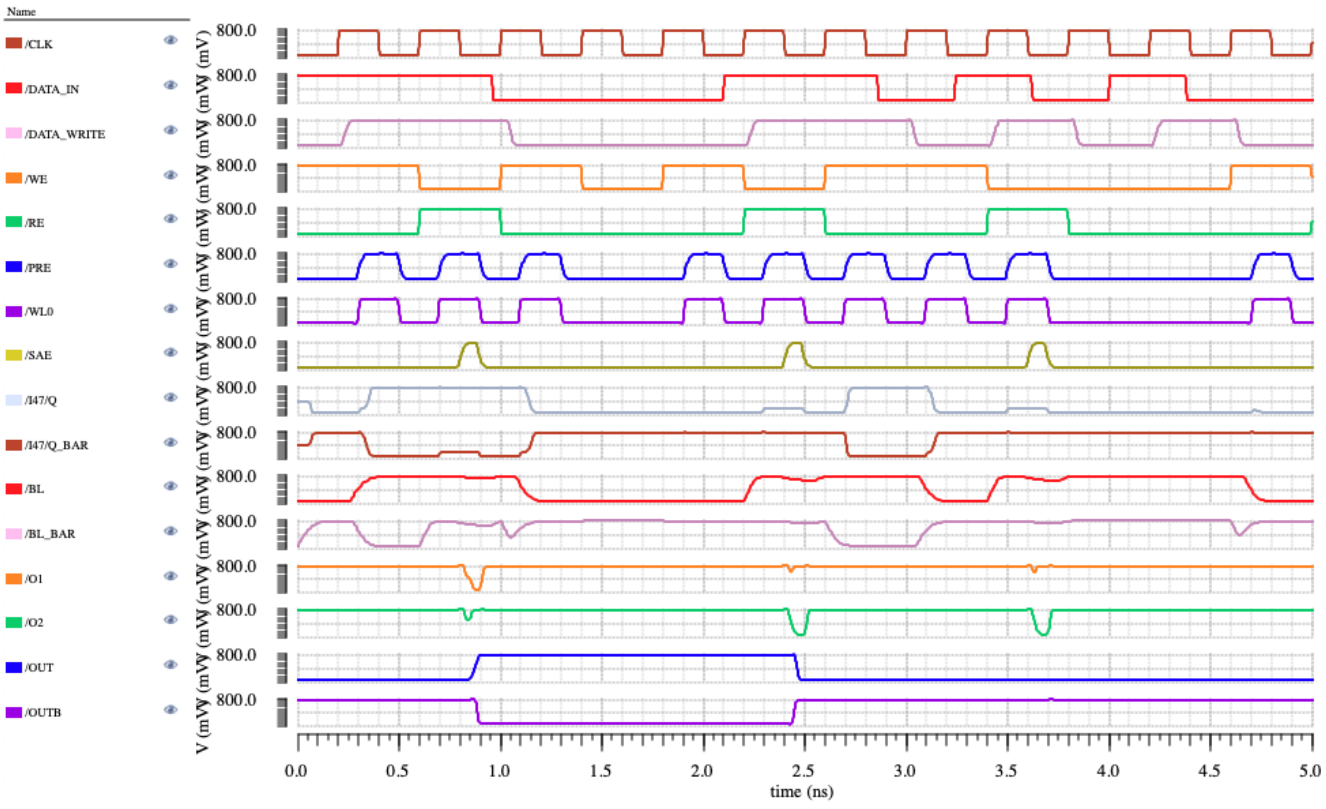


Figure 11: Read and Write Operations (Optimized Cell)

Figure 11 shows the read and write operation with the same parameters as those used for the simulation shown in Figure 2. Comparing the two, we see that the read and write operations are performed correctly.

4.3 Timing

The final timing specifications for the circuit are as follows:

- CLK to WL: 96.83ps
- Cell Access Time: 11.53ps
- Sense Amplifier: 38.18ps
- SAE to Output of latch: 79.10ps

Note that the delay from the rising edge of the clock to the rising edge of WL is approximately 100ps, as per our design. This will be further discussed in section 4.3.1.

4.3.1 Signal Generation

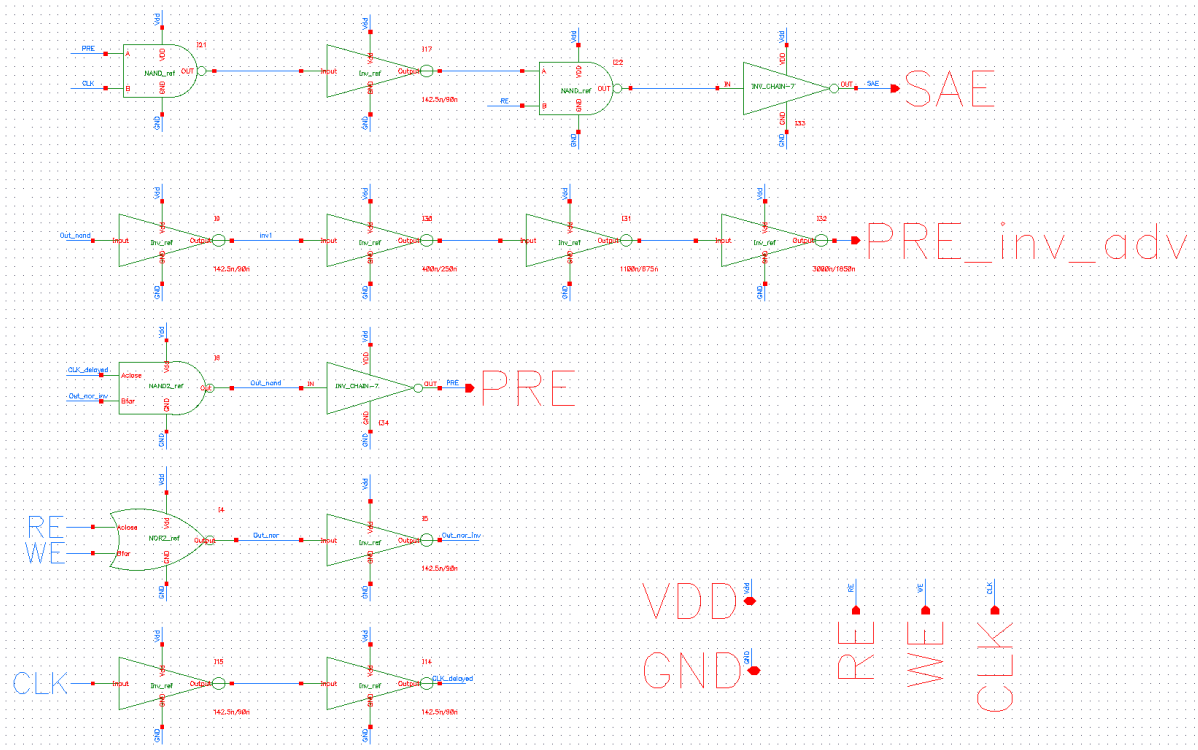


Figure 12: Signal Generation - Schematic

Figure 12 shows the circuit used for generating the control signals SAE, PRE, and an additional signal PRE_inv_adv. The PRE and SAE signals are same waveforms that are shown in Figure 11. The PRE_inv_adv signal is the time advanced version of inverted PRE. It is used for synchronizing WL with the PRE. The time advancing is done to synchronize PRE & WL to a greater extent.
PRE is a delayed version of clock which should be high only when memory is either reading or writing. That is, when $RE \cup WE = 1$. So basically, the signal generator implements below logic for PRE:

$$PRE = CLK_{delayed} \cap (RE \cup WE)$$

SAE signals should only be high when CLK is high, PRE is high, and RE is high. Thus, we use two NAND2 gates with inverters to generate the effective logic:

$$SAE = CLK \cap RE \cap PRE_{delayed}$$

The delayed version of CLK and PRE are generated using inverter chains. PRE uses minimum sized inverter chain, whereas PRE_inv_adv uses progressively sized inverter chain due to the fact that it observes large gate capacitances when synchronizing with WL.

4.3.2 Maximum Frequency

The maximum frequency the circuit can be operated (with the current design) is with a clock period of 350ps. This corresponds to a maximum clock frequency of 2.86GHz. Figure 13 shows the full peripheral performing correct read and write operations to a single SRAM cell. Note that as the clock frequency is increased, the pulse width of the SAE signal is effectively decreased. This is due to our design choice to delay the SAE rising edge by 100ps after the rise of the WL. This choice serves to conserve power in the sense amplifier during each read cycle.

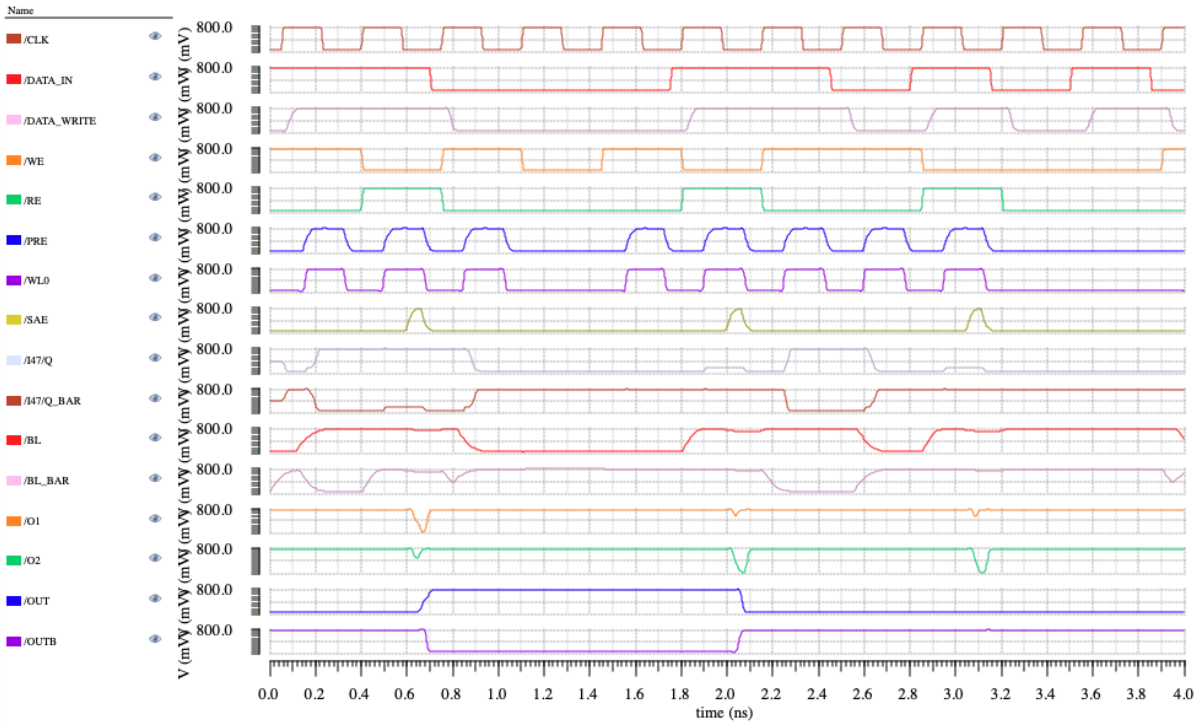


Figure 13: Read and Write Operation ($F_{CLK} = 2.86GHz$)

5 SRAM Problem 5: 16x4 Cell Array

5.1 Full Schematic

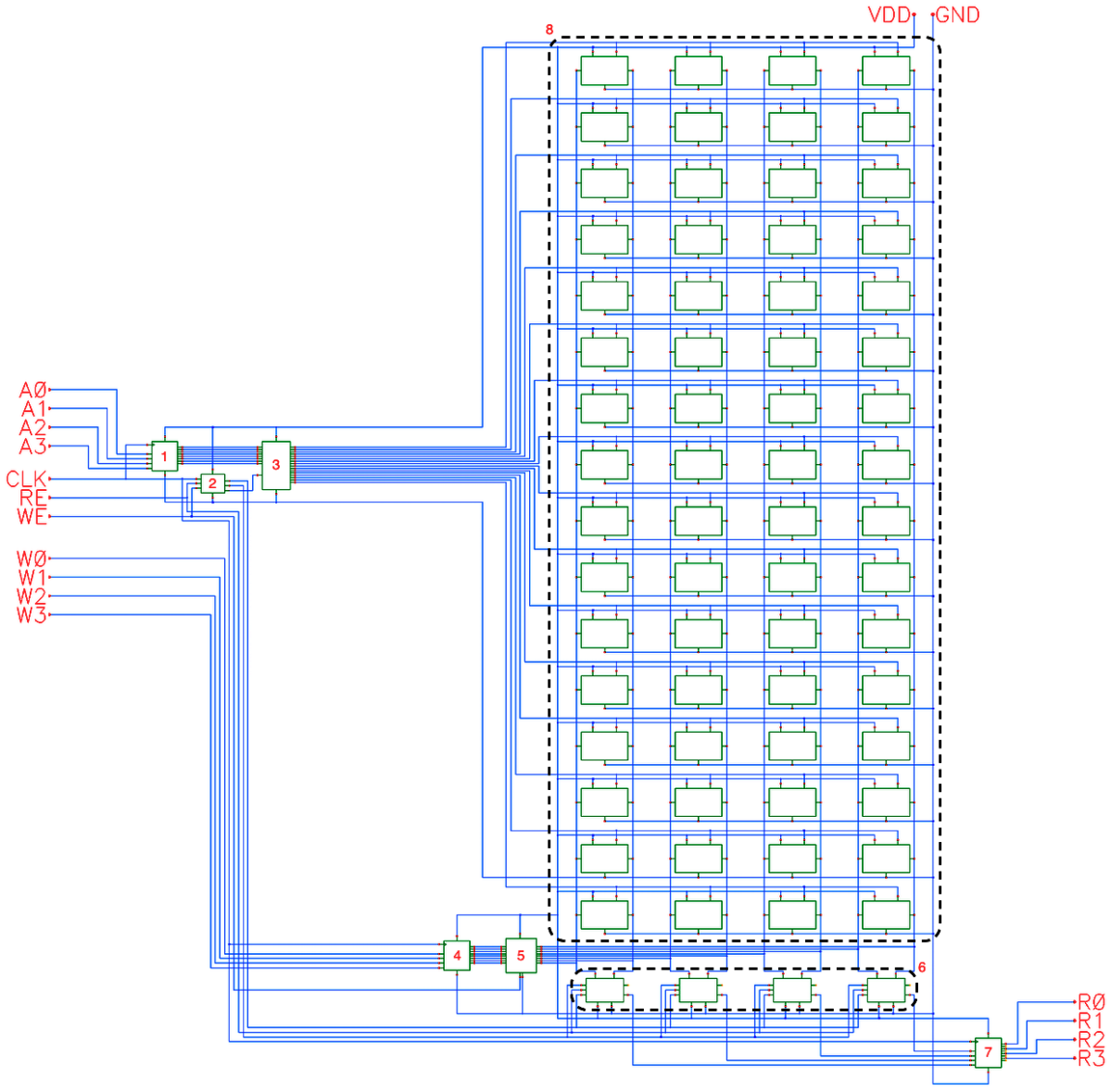


Figure 14: 16x4 SRAM Cell Array with Full Peripheral

Item No.	Description
1	Address Register
2	Control Signal Generator
3	Word Line Row Decoder
4	Data Register (Write Operation)
5	Write Circuit
6	Read Circuit
7	Data Register (Read Operation)
8	16×4 SRAM Cell Array

Table 3: Full Schematic Devices

[A0 A1 A2 A3] is a 4-bit address input used to choose which word line to write to. [W0 W1 W2 W3] is a 4-bit data input the user may write to the SRAM cell during a write cycle. [R0 R1 R2 R3] is a 4-bit output that results from reading the SRAM cell. The CLK, RE, and WE signals are user-provided signals for clock, read enable, and write enable, respectively.

5.2 Top Level Block Diagram

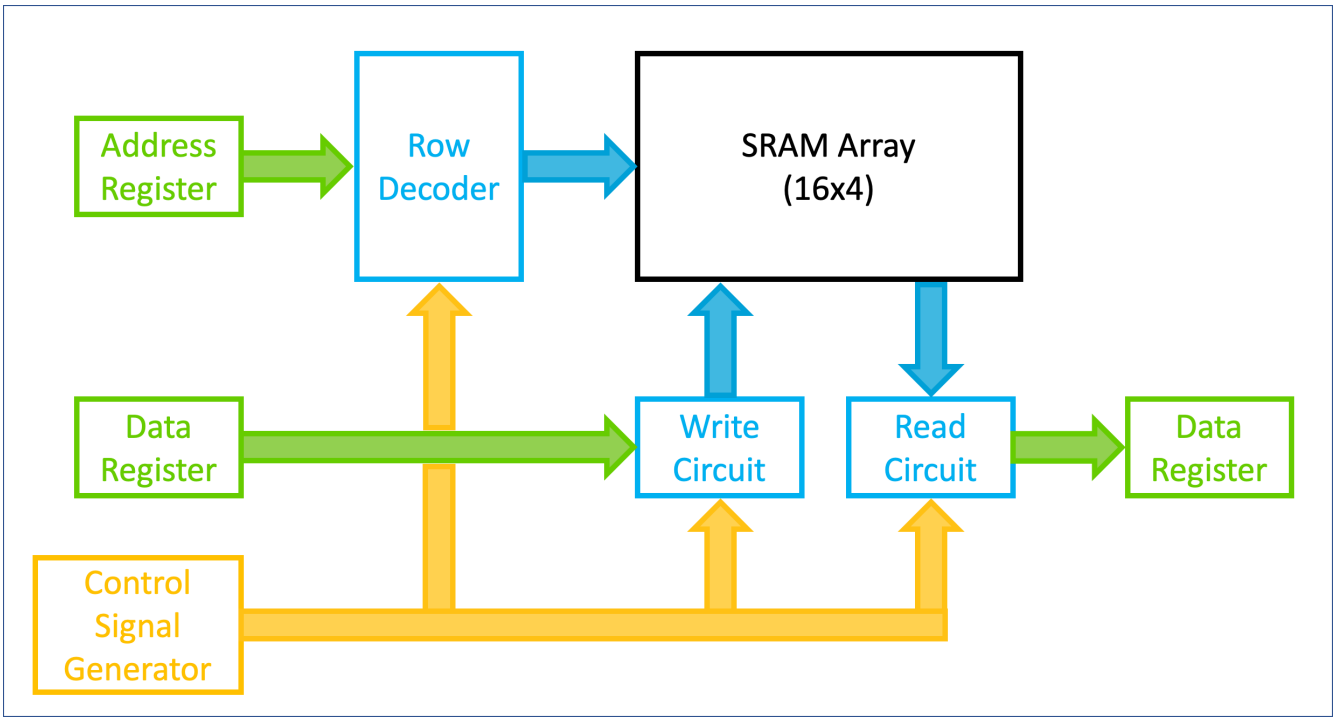


Figure 15: Top Level Block Diagram

6 Peripheral Optimizations

6.1 Row Decoder

6.1.1 Previous Design

This section briefly discusses the previous design of the decoder. A NAND-NOR-INV Predecoder design was used for realizing the decoder. A critical path of WL0 is shown in Figure 16 for reference.

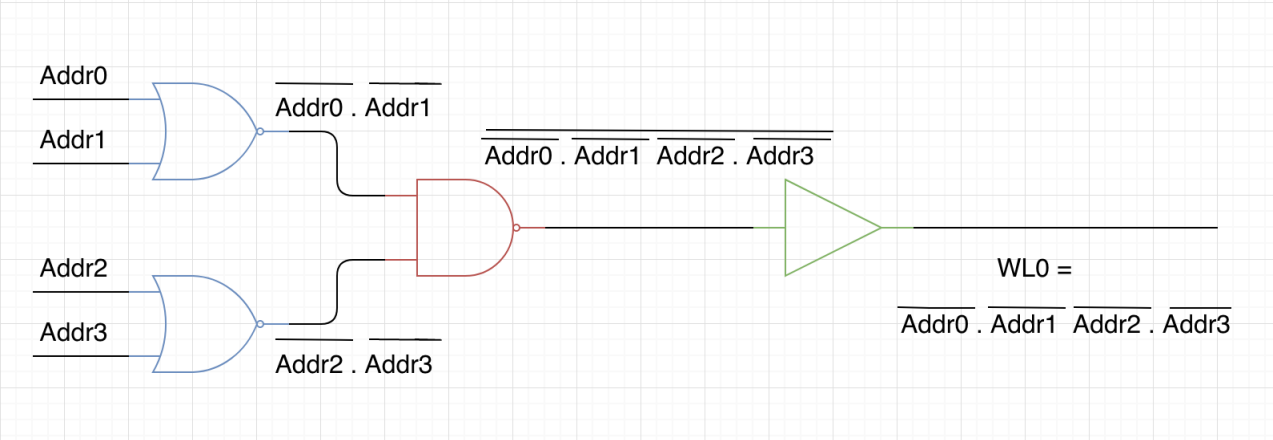


Figure 16: Critical Path of WL0

For the synchronization with PRE signal, a different kind of dynamic inverter (refered to as "Reverse Dynamic Inverter" from now on) was used instead of the regular Static CMOS inverter. Refer to Figure 17 for the same. This inverter takes PRE_inv (inverted PRE signal) and the WL signal as the input, and outputs WL which is synced with PRE signal. In case where PRE_inv is 1, i.e. when PRE = 0, the output defaults to 0.

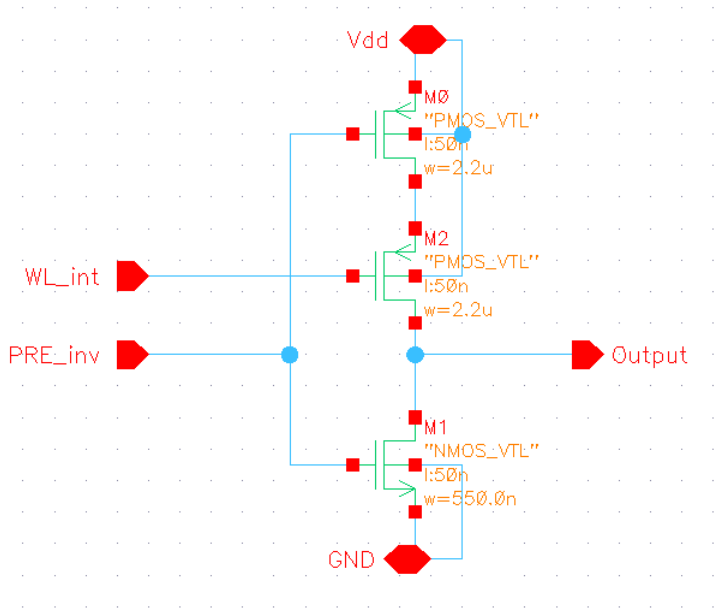


Figure 17: Reverse Dynamic Inverter (non-optimized)

The benefit of using this circuit was synchronization of WL and PRE at low area cost and default WL = 0 condition. However when pulling up, it requires large PMOS transistors (Refer to sizing in Figure 17) . It has certain undesirable effects which are discussed and solved in the following section.

6.1.2 Improvements in Previous Design

The drawbacks of using large pull up transistors in Reverse Dynamic Inverter are:

1. The decoder was causing higher power consumption.
2. Synchronization was generating voltage undershoot spikes
3. Large gate capacitance of pull up PMOS was causing unnecessary loading on PRE generator circuitry.

To mitigate these problems, transistor sizing were reduced (based on parametric analysis with trade off between delay and size). But in addition to that, one more design modification was also introduced. This new design exploits the fact that at any one time, maximum 1 WL line needs to be pulled up. Not more than that. In the previous design, large Pull Up Transistors were attached to all WL lines, but in the new design we reduce the size as well as the transistor count by introducing a common pull up network.

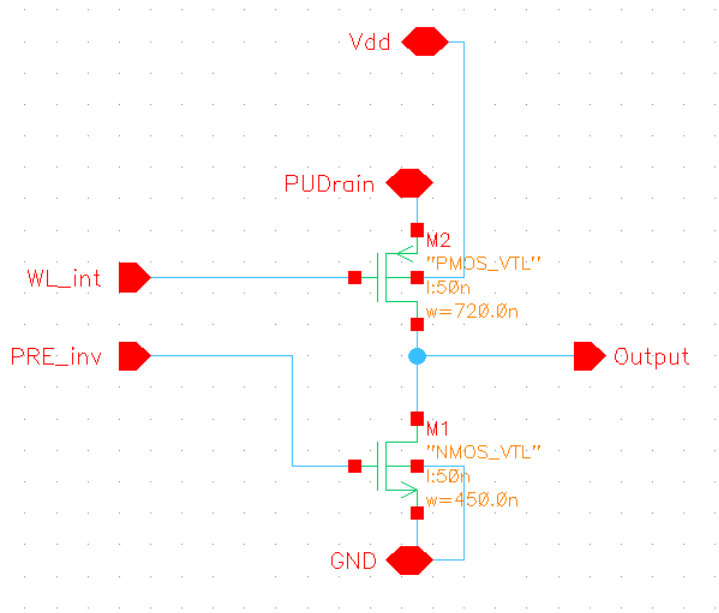


Figure 18: Reverse Dynamic Inverter (Shared Pull-Up)

As seen in the figure above, one large pull up transistor is removed. And instead of it, the source of PMOS M2 is connected to the Drain of the common pull up network. (Refer to Figure 19). This design update combines all the pull ups into one single pull up node.

Figure 19 shows common pull up implementation for 1 WL line. Here, PUDrain signal is common to all WL producing inverters. WL0_int is the non-synchronized WL output from the previous logic. Which gets synchronized with PRE signal at the output of this Inverter. This implementation is repeated for all WL lines, keeping the PUDrain node common.

The benefits of using common pull up is,

1. Reduction of 13 transistors (3 common pull up PMOS, instead of 16 individual WL PMOS)
2. Reduced transistor sizing (from 2.2μ PMOS & 550n NMOS to 720n PMOS & 450n NMOS)
3. Reduced power consumption (from 41 μW to 31.8 μW)

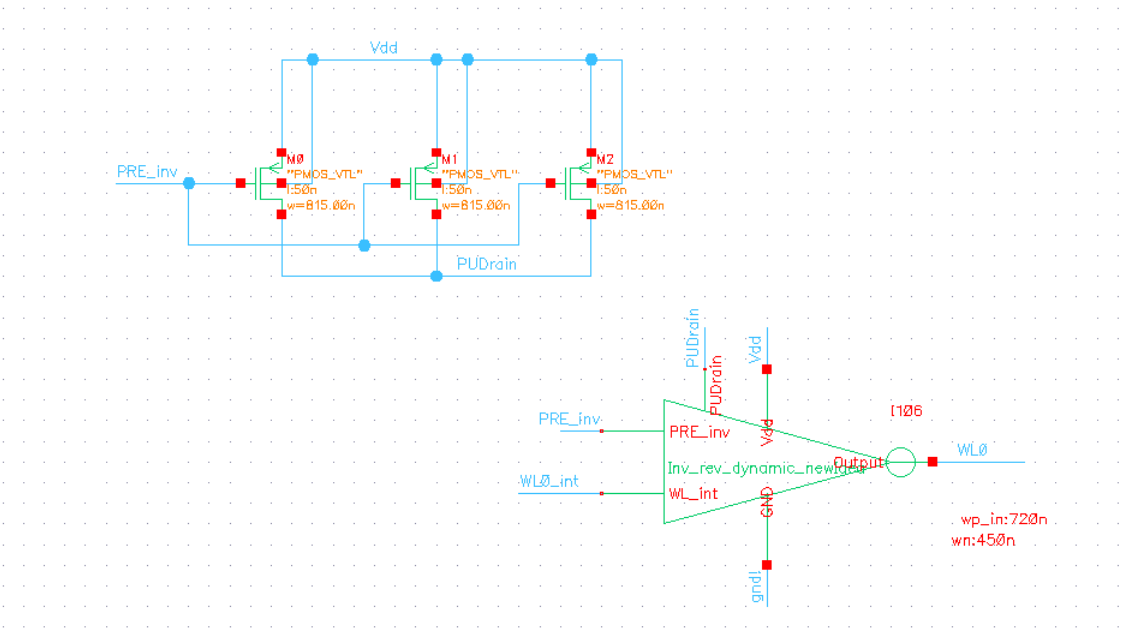


Figure 19: Common Pull Up Inverter Implementation

6.1.3 Power Consumption

Part of the reason for the implementing above mentioned optimization, was to reduce the power consumption as large transistors consume more power. The decoder outputs were simulated for the full truth table (i.e. WL0 to WL15) in ascending as well as descending order. In both the cases, simulation ran for 6.4 ns (400 ps clock period $\times$ 16 different inputs = 6.4 ns).
Switching Factor (number of inputs changing at a time) varies from minimum 1 (output switching from WL0 to WL1) to maximum 4 (output switching from WL7 to WL8).
The final result is as below,

- Power Consumption (Ascending Truth Table) : 31.8 μW
- Power Consumption (Descending Truth Table) : 31.8 μW

Figure 20 shows the Decoder Output simulation for which average power consumption was calculated. Note that here, WL is not synchronized with PRE signal. For the purpose of power consumption calculation, only correct functionality of WL is checked for all possible input combinations.

6.2 Register

6.2.1 Schematic and size

In this report, we improved our register in two perspectives. One is to make the C-Q delay of low-to-high and high-to-low balanced. Another one is to size it down to get a lower power consumption.
Optimization of the flip-flop involved trying to balance the C-Q delay for high-to-low and low-to-high transitions. The results are shown in Figure 23. In the previous flip-flop design, the low-to-high delay was shorter than the high-to-low delay. To fix this issue, we sized up the pull-down network. In addition, we made the farthest NMOS a little larger for a stronger pull-down, as seen in Figure 24.
For power consumption, we got 64.33μW in our previous report. In this report, since the smaller transistors, we had a better result.
The new specifications for the register are as follows:

- C-Q Delay: 28.5 ps
- Setup Time: 35 ps
- Hold Time: 0 ps

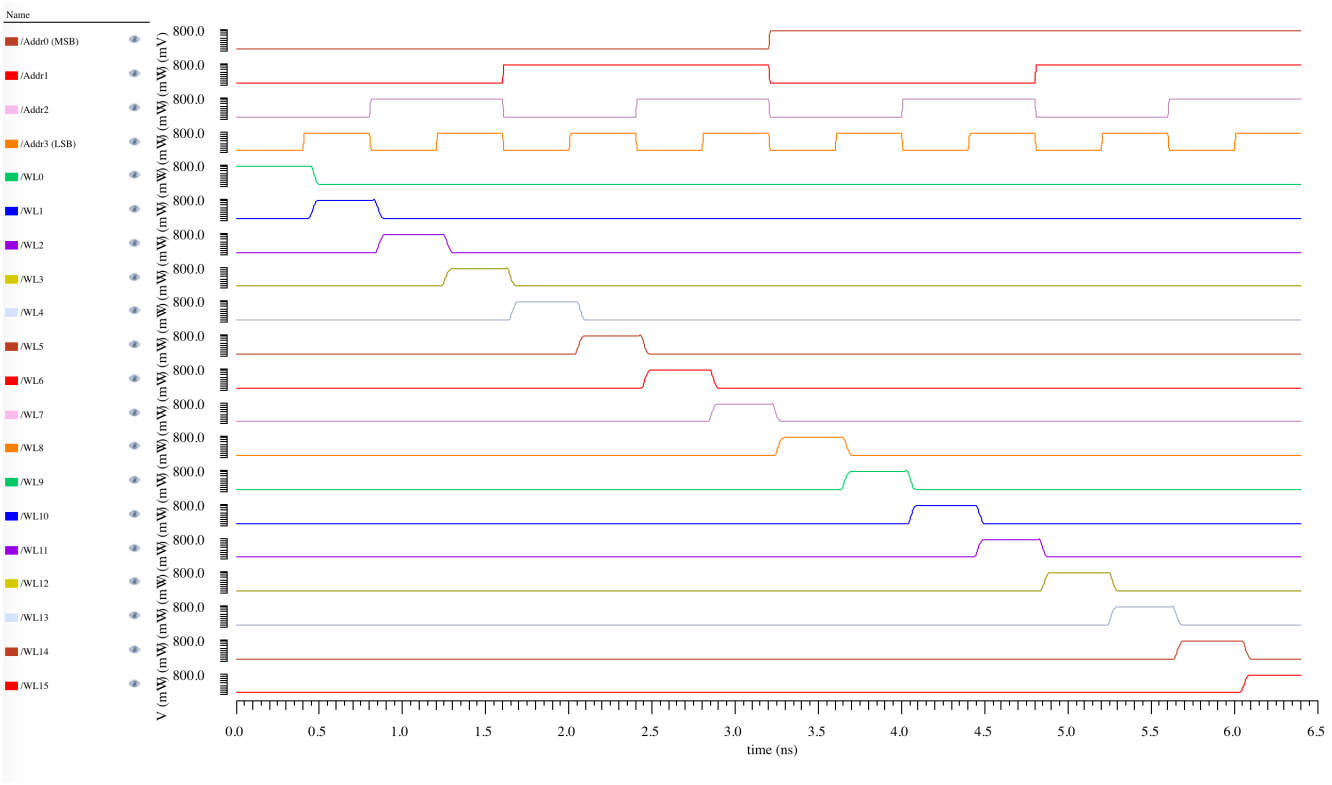


Figure 20: Ascending Truth Table of Decoder for Power Calculation

- Power Consumption - Four Bits Switching (Figure 21): $49.84\mu W$
- Power Consumption - Average Case (Figure 22): $44.15\mu W$

Figure 21 shows the simulation for how average power consumption of a register was calculated. This tested when four bits changed either from low to high or from high to low. On the other hand, since sometimes the output data might not change, so we also do this test which is shown in Figure 22. In Figure 22, A0 changed from high to low; A1 kept in high; A2 changed from low to high; A3 kept in low.

6.3 Write Circuit

For the write circuit, we worked to optimize the sizing for both the transmission gate and the inverter.
To find out the relation of inverter chain and delay, we set the size of transmission gate as 800nm, and tried different inverter chain ratio. Figure 25 shows our testing result.
Based on this result, we chose $e \times 1.7$ as the inverter chain ratio. The widths in the first stage are 245nm for NMOS ad 385nm for PMOS. The widths in the second stage are 1128nm for NMOS and 1773nm for PMOS. Compared to our previous values, we reduced the size to 63% of the previous sizes, but still kept the delay within 100p sec.
Figure 27 shows the testing results of one bit write circuit connected to flip-flop.

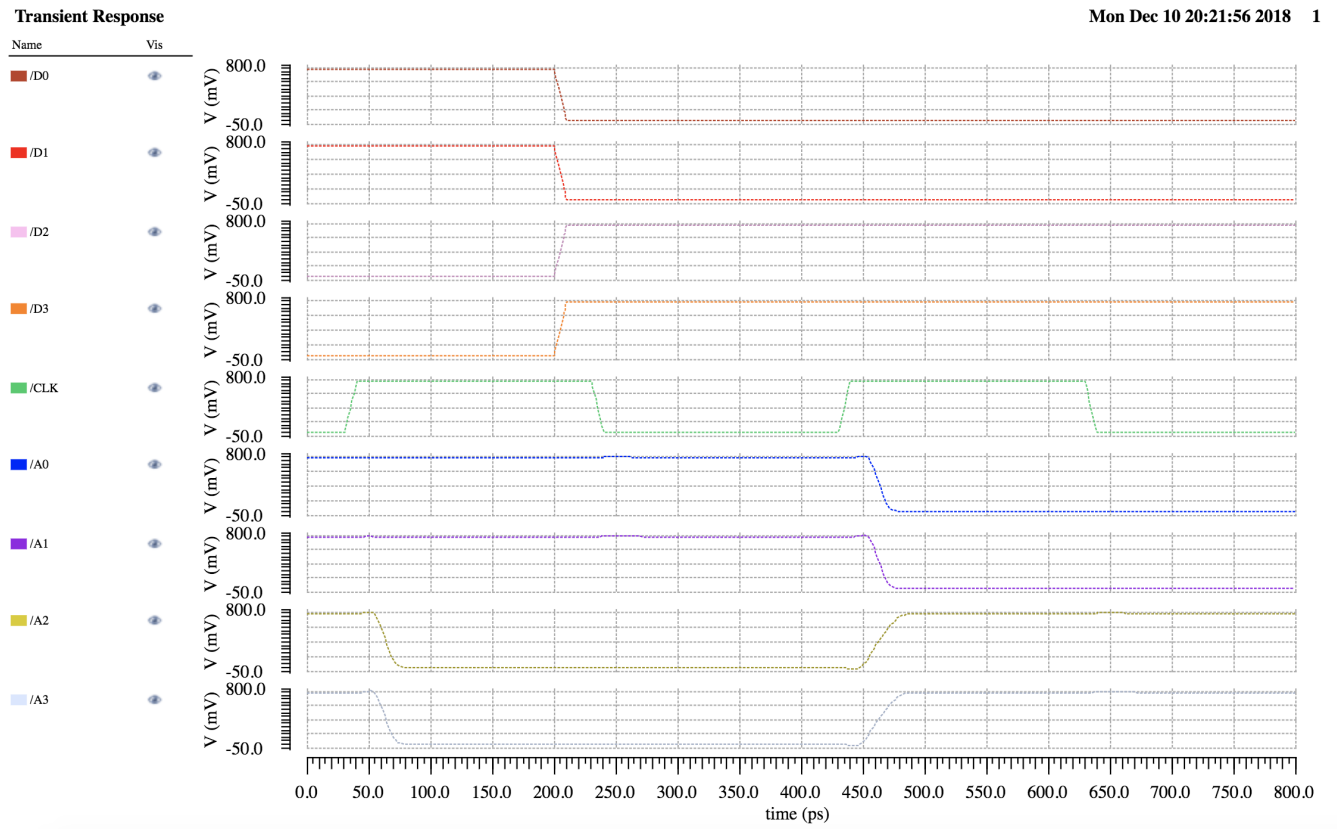


Figure 21: Register Power Calculation - Four Bits Switching

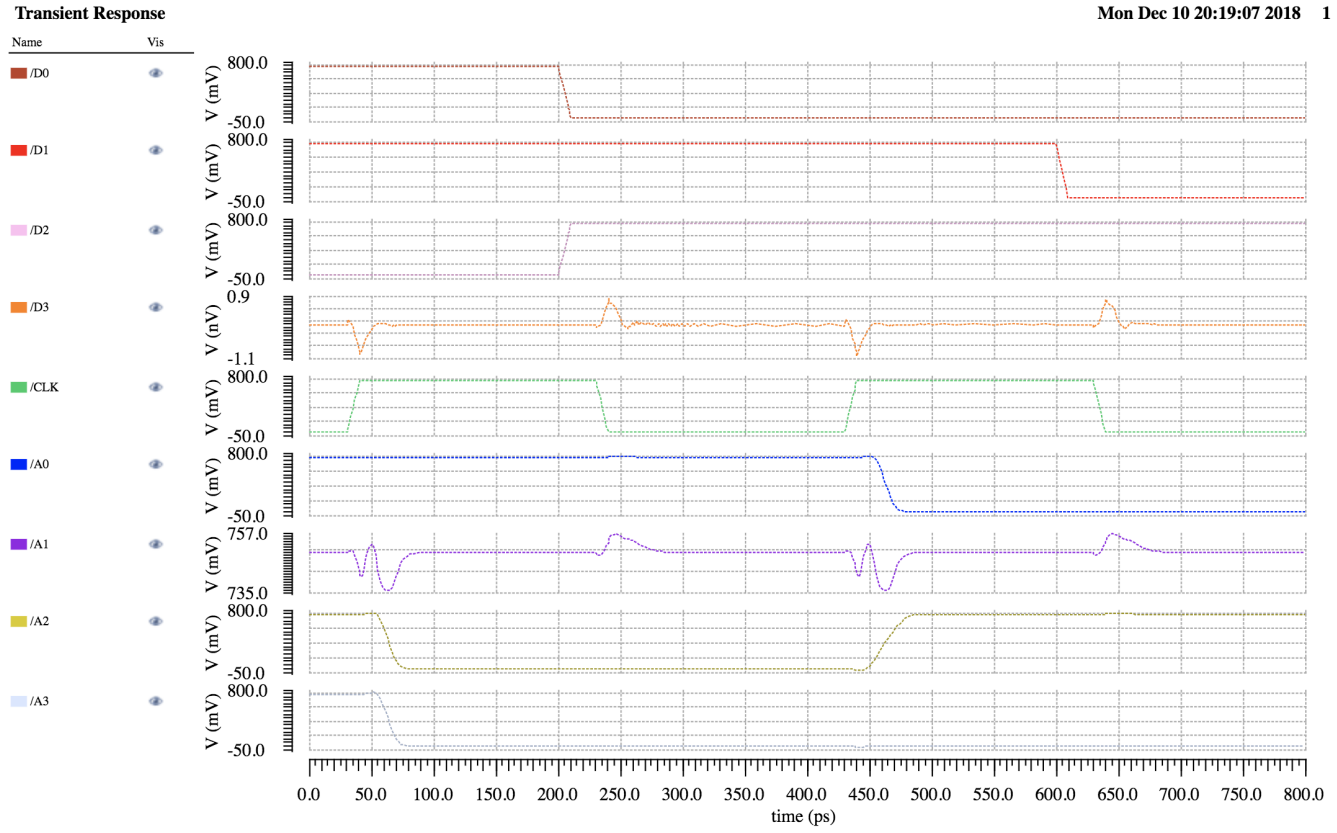


Figure 22: Register Power Calculation - Average case

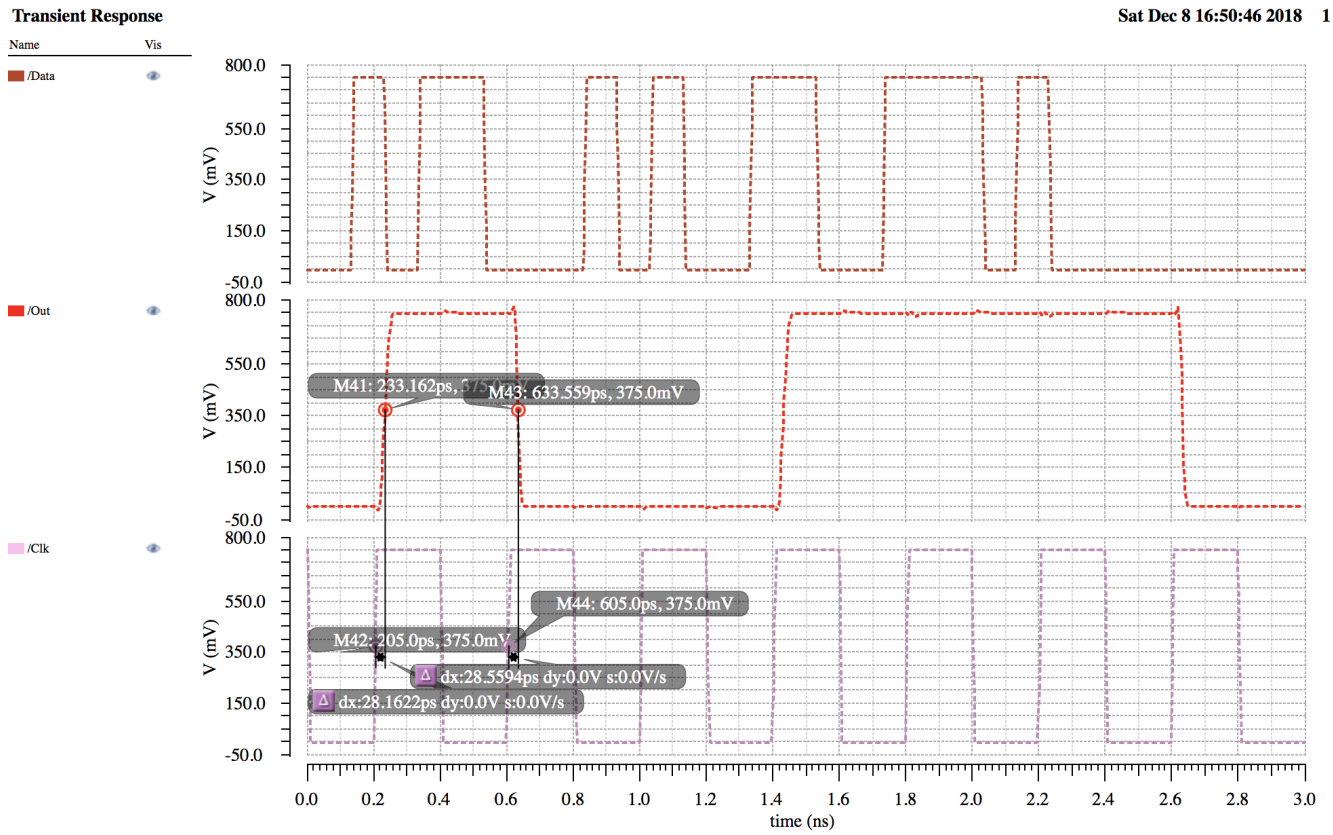


Figure 23: Clk-Q delay

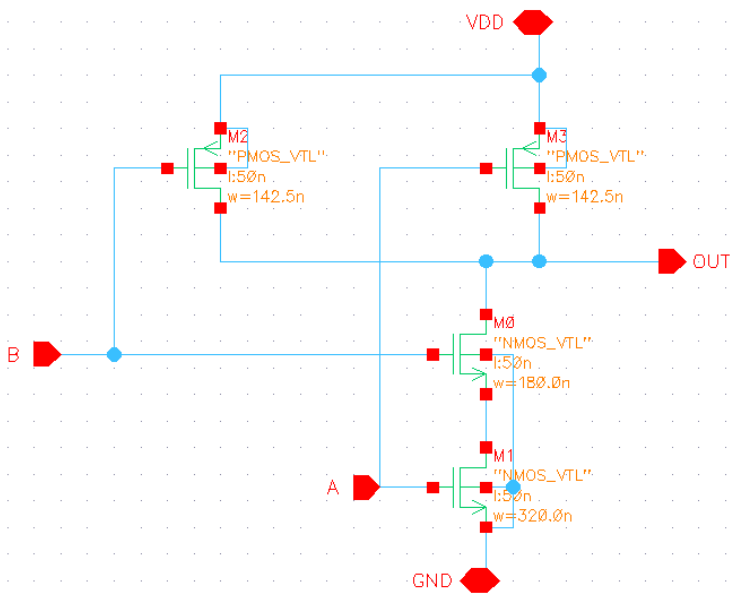


Figure 24: Register NAND2 Gate - Schematic

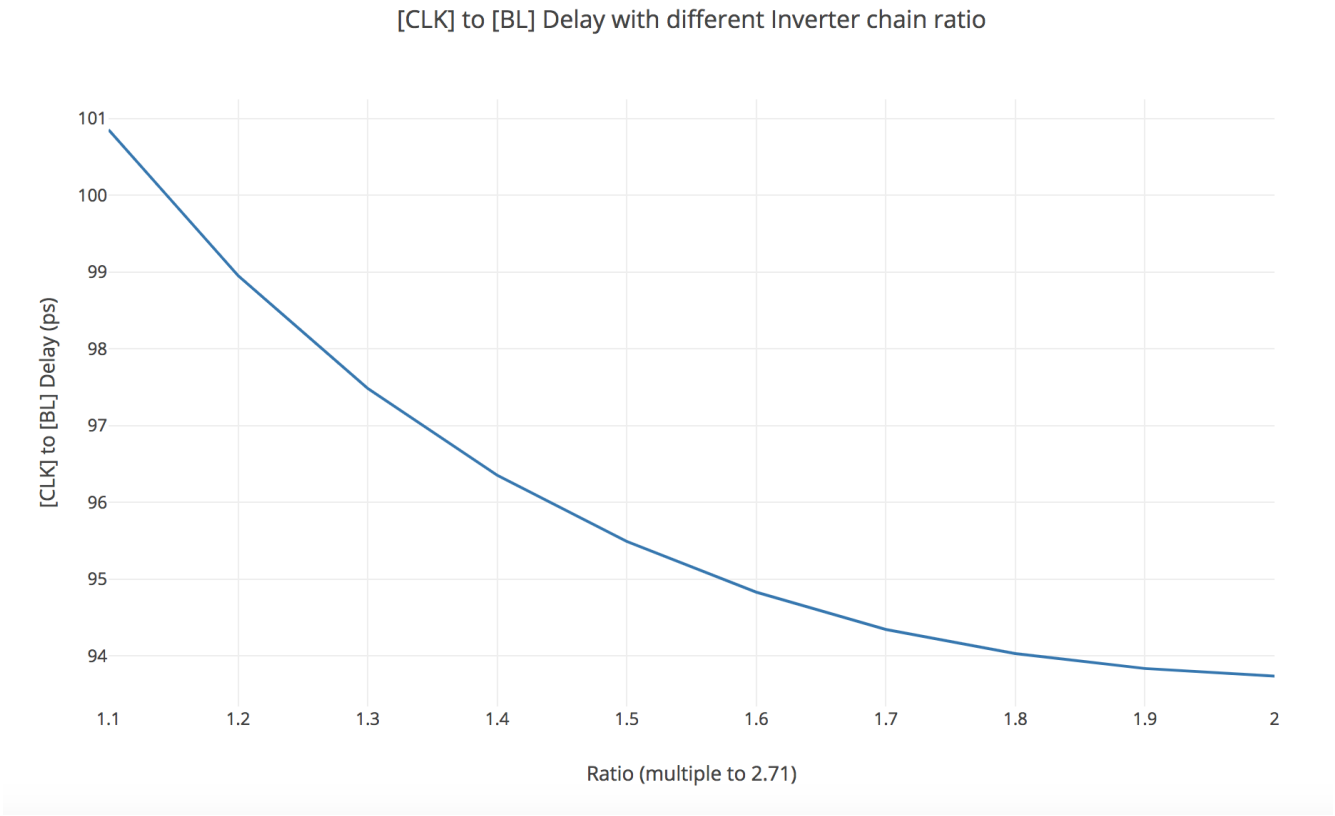


Figure 25: CLK to BL Delay vs. Inverter Chain Ratio

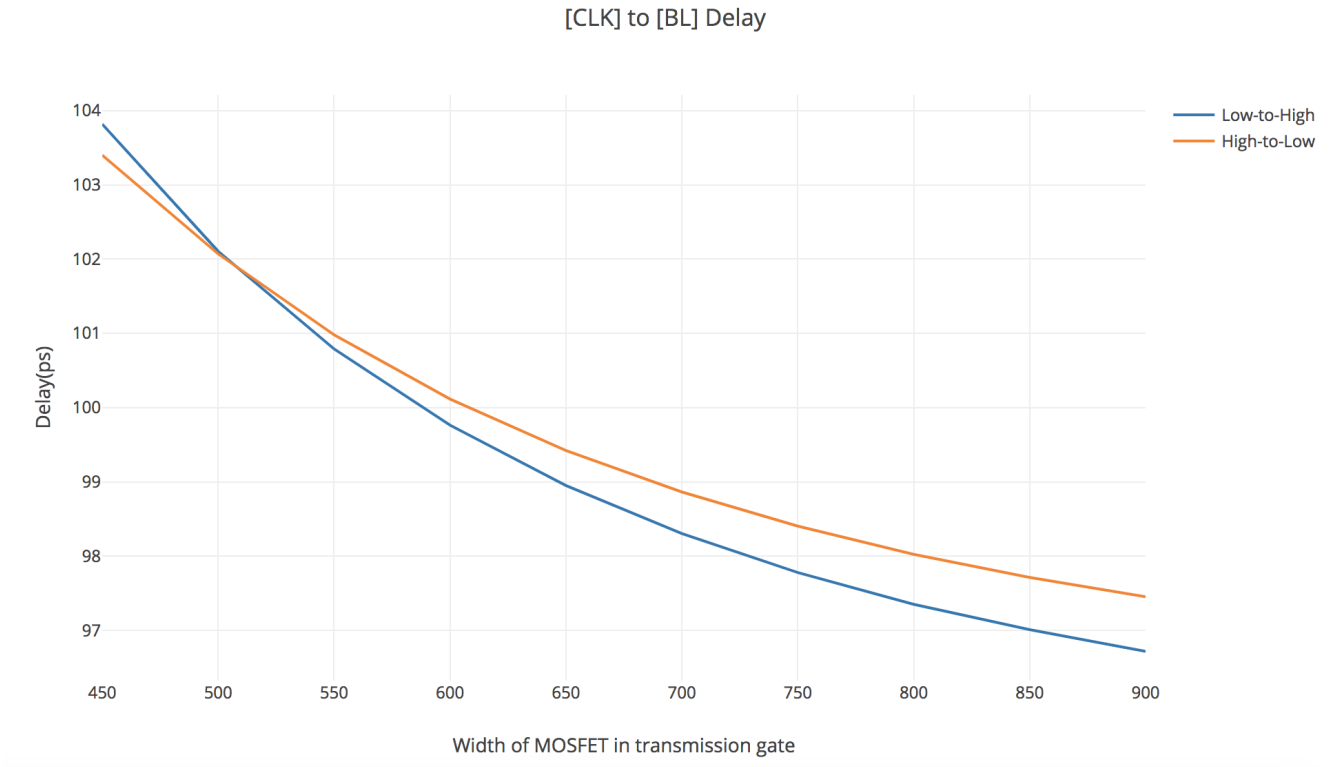


Figure 26: CLK to BL Delay vs. Transmission Gate Sizing

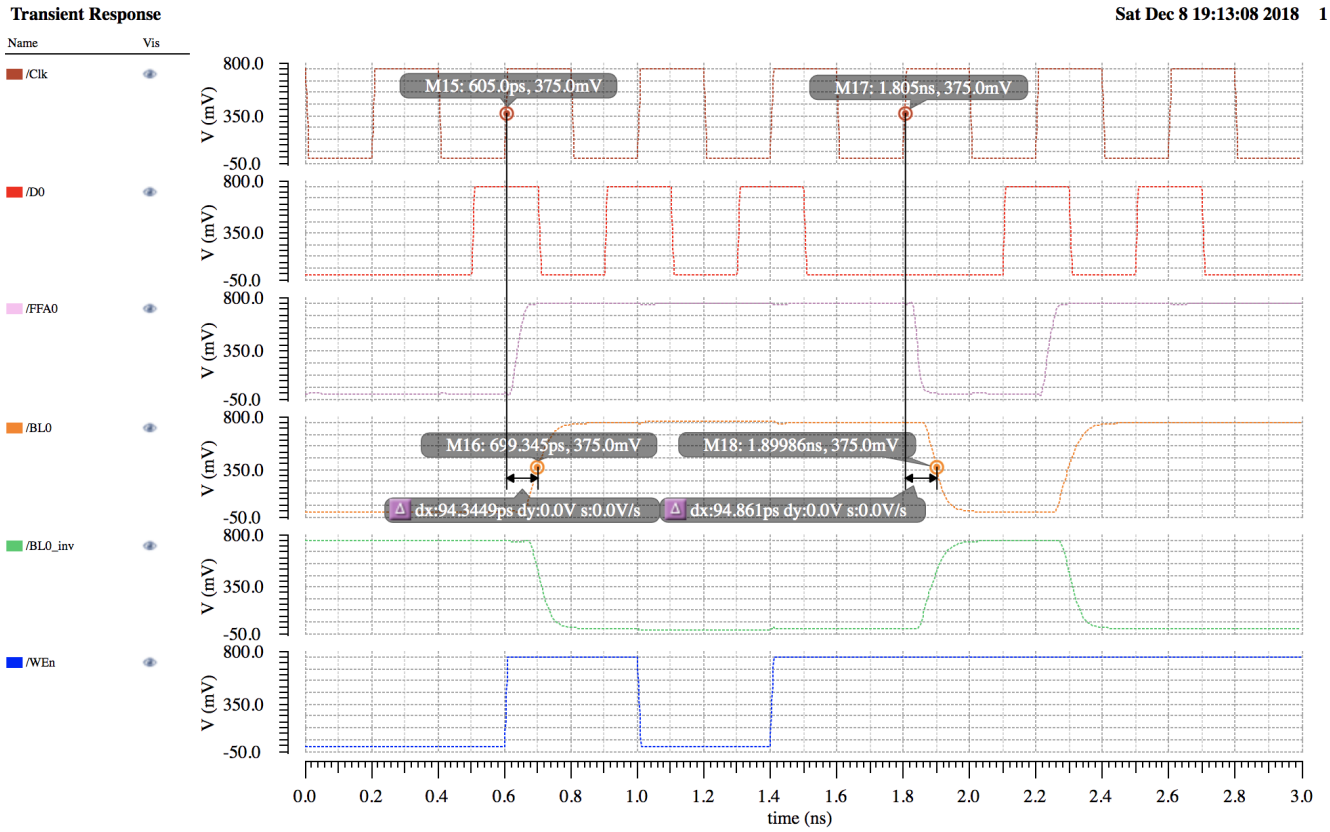


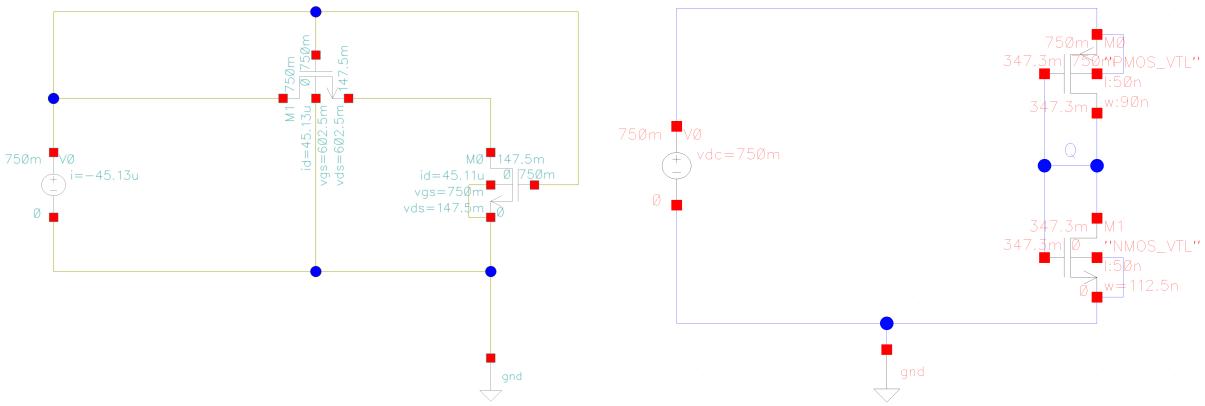
Figure 27: Single Bit Write Operation

APPENDIX

Appendix A

1. Read Noise Margin Testing

As discussed in the section 3.2.1, two tests are needed to verify the read noise margin for every width ratio of $W_{Access} : W_{PullUp} : W_{PullDown}$ that is tested. Figure 28 shows the DC analyses used to find the read voltage & read current, and trip voltage, respectively.



(a) Test Circuit for V_{read}

(b) Test Circuit for V_{trip}

Figure 28: Test Circuits for Read Noise Margin

The DC voltage at the node connecting the access transistor to the pull-down transistor in Figure 28(a) shows the voltage that Q or QBAR will rise to during a read operation. This voltage should be minimized such that the risk of data destruction is low. The DC current through the two transistors is a factor that affects the cell access time as discussed in section 2. The other contributing factor to this destructive read is the trip voltage of cross-coupled inverters inside the SRAM cell. If the read disturb (V_{read}) reaches the trip voltage, then the two inverters will switch to their opposite steady states, thus destroying the stored bit. Figure 28 shows the analysis performed for the optimal sizing described in section 2.3. The read noise margin is then calculated by the equation:

$$NM_{read} = \frac{V_{trip} - V_{read}}{V_{DD}} \times 100\%$$

2. Write Noise Margin Testing

The determination of the write noise margin is relatively simpler than the calculation of the read noise margin. Figure 29 shows the simulation of the scenario where a '0' is being written to the SRAM cell. This operation ideally means [BL] = 0V, and [BL_BAR] = 750mV. The write margin is a measure of how large the bit line at logic '0' ([BL] in this case), can become before the write operation fails the the data in the SRAM cell doesn't switch. The parametric analysis done in Figure 29 sweeps this bit line voltage from 260mV to 270mV and the voltage at which the write operation fails we call V_{write}. The write noise margin is then calculated by the equation:

$$NM_{write} = \frac{V_{write}}{V_{DD}} \times 100\%$$

3. Experimental Sizing Data

Using the methods described in Appendix A.1 and A.2, the read margin, write margin, and cell access time can be tabulated for all size ratios used in the design process. These values are shown in Table 4 below. Note that the data shown in Figures 3, 4, and 5 result from plotting the data collected in Table 4.

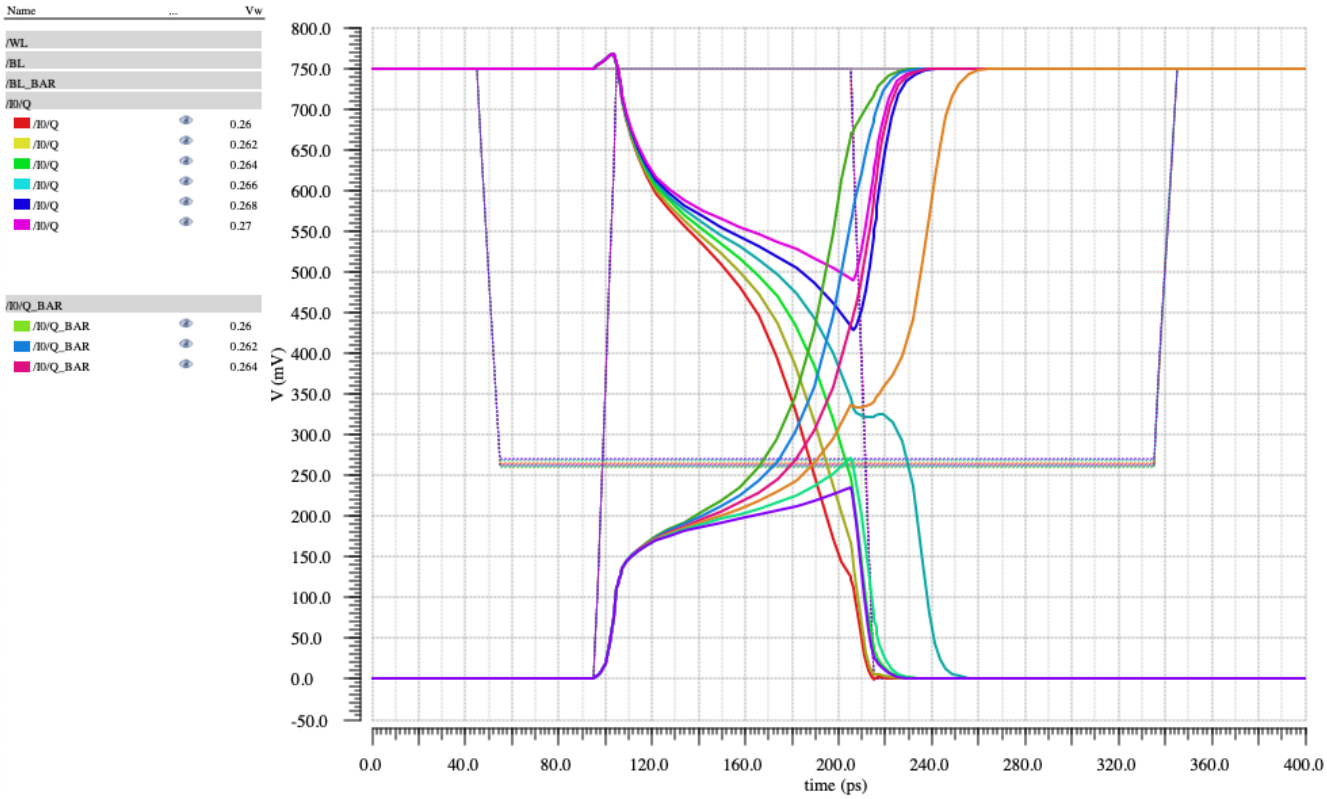


Figure 29: Parametric Analysis for V_{write}

Cell Sizing Parameters			Read Parameters				Write Parameters		Timing Parameters		
Wn,access (nm)	Wp,pullup (nm)	Wn,pulldown (nm)	Vtrip (mV)	Vread (mV)	Iread (uA)	Noise Margin (Read)	Vwrite (mV)	Noise Margin (Write)	Bitline Cap (fF)	Vbit (mV)	Cell Access Time (ps)
135	90	180	328.1	120.3	62.93	27.71%	268	35.73%	40	13	8.263149531
150	90	180	328.1	130.2	67	26.39%	280	37.33%	40	13	7.76119403
155	90	180	328.1	133.4	68.26	25.96%	284	37.87%	40	13	7.617931439
160	90	180	328.1	136.5	69.47	25.55%	288	38.40%	40	13	7.48524543
150	90	220	322.2	112.1	73.08	28.01%	272	36.27%	40	13	7.115489874
130	90	160	332.8	127.5	58.36	27.37%	268	35.73%	40	13	8.910212474
130	90	150	335.5	133.7	56.6	26.91%	270	36.00%	40	13	9.187279152
130	90	140	338.3	140.4	54.68	26.39%	274	36.53%	40	13	9.50987564
120	90	130	341.3	139.6	50.4	26.89%	266	35.47%	40	13	10.31746032
125	90	125	342.9	147.7	50.44	26.03%	274	36.53%	40	13	10.30927835
115	90	115	346.3	147.6	46.18	26.49%	266	35.47%	40	13	11.26028584
112.5	90	112.5	347.3	147.5	45.11	26.64%	264	35.20%	40	13	11.52737752
110	90	110	348.2	147.5	44.14	26.76%	262	34.93%	40	13	11.78069778
100	90	100	352.2	147.3	39.78	27.32%	254	33.87%	40	13	13.07189542

Table 4: SRAM Experimental Sizing Results

Appendix B

1. Full DRC Report

```
==========================================================================
   === CALIBRE::DRC-H SUMMARY REPORT
   ===
   Execution Date/Time: Mon Dec 10 00:14:38 2018
   Calibre Version: v2016.3_19.12 Thu Aug 4 11:48:27 PDT 2016
   Rule File Pathname: /home/am8482/freepdk45_working_directory/_calibreDRC.rul_
   Rule File Title:
   Layout System: GDS
   Layout Path(s): SRAM_CELL.calibre.dbs
   Layout Primary Cell: SRAM_CELL
   Current Directory: /home/am8482/freepdk45_working_directory
   User Name: am8482
   Maximum Results/RuleCheck: 1000
   Maximum Result Vertices: 4096
   DRC Results Database: SRAM_CELL.drc.results (ASCII)
   Layout Depth: ALL
   Text Depth: PRIMARY
   Summary Report File: SRAM_CELL.drc.summary (REPLACE)
   Geometry Flagging: ACUTE = NO SKEW = NO ANGLED = NO OFFGRID = NO
   NONSIMPLE POLYGON = NO NONSIMPLE PATH = NO
   Excluded Cells:
   CheckText Mapping: COMMENT TEXT + RULE FILE INFORMATION
   Layers: MEMORY-BASED
   Keep Empty Checks: YES
   ----------------------------------------------------------------------
   -- RUNTIME WARNINGS
   --
   ----------------------------------------------------------------------
   -- ORIGINAL LAYER STATISTICS
   --
   LAYER pwell ..... TOTAL Original Geometry Count = 2 (6)
   LAYER nwell ..... TOTAL Original Geometry Count = 2 (3)
   LAYER active ..... TOTAL Original Geometry Count = 13 (45)
   LAYER poly ..... TOTAL Original Geometry Count = 11 (26)
   LAYER pimplant ... TOTAL Original Geometry Count = 2 (4)
   LAYER nimplant ... TOTAL Original Geometry Count = 2 (5)
   LAYER vth ..... TOTAL Original Geometry Count = 0 (0)
   LAYER vtg ..... TOTAL Original Geometry Count = 0 (0)
   LAYER metal1 ..... TOTAL Original Geometry Count = 21 (43)
   LAYER contact .... TOTAL Original Geometry Count = 4 (19)
   LAYER metal2 ..... TOTAL Original Geometry Count = 8 (20)
   LAYER metal3 ..... TOTAL Original Geometry Count = 11 (16)
   LAYER metal4 ..... TOTAL Original Geometry Count = 0 (0)
   LAYER metal5 ..... TOTAL Original Geometry Count = 0 (0)
   LAYER metal6 ..... TOTAL Original Geometry Count = 0 (0)
   LAYER metal7 ..... TOTAL Original Geometry Count = 0 (0)
   LAYER metal8 ..... TOTAL Original Geometry Count = 0 (0)
   LAYER metal9 ..... TOTAL Original Geometry Count = 0 (0)
   LAYER metal10 .... TOTAL Original Geometry Count = 0 (0)
   LAYER via1 ..... TOTAL Original Geometry Count = 1 (8)
   LAYER via2 ..... TOTAL Original Geometry Count = 1 (6)
   LAYER via3 ..... TOTAL Original Geometry Count = 0 (0)
   LAYER via4 ..... TOTAL Original Geometry Count = 0 (0)
```

LAYER via5 ..... TOTAL Original Geometry Count = 0 (0)
LAYER via6 ..... TOTAL Original Geometry Count = 0 (0)
LAYER via7 ..... TOTAL Original Geometry Count = 0 (0)
LAYER via8 ..... TOTAL Original Geometry Count = 0 (0)
LAYER via9 ..... TOTAL Original Geometry Count = 0 (0)

— RULECHECK RESULTS STATISTICS
—
RULECHECK Well.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Well.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Well.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Poly.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Poly.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Poly.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Poly.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Poly.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Poly.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Active.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Active.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Active.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Active.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Implant.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Implant.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Implant.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Implant.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Implant.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Contact.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Contact.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Contact.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Contact.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Contact.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Contact.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Via1.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via1.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via1.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via1.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Via2.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via2.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via2.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via2.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal3.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal3.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal3.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal3.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Via3.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via3.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via3.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via3.4 ..... TOTAL Result Count = 0 (0)

RULECHECK Metal4.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal4.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal4.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via4.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via4.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via4.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via4.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal5.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal5.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal5.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via5.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via5.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via5.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via5.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal6.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal6.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal6.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via6.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via6.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via6.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via6.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal7.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal7.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal7.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via7.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via7.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via7.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via7.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal8.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal8.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal8.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via8.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via8.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via8.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via8.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal9.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal9.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal9.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via9.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Via9.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Via9.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Via9.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal10.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal10.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal10.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.8 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal1.9 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.8 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal2.9 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal3.5 ..... TOTAL Result Count = 0 (0)

RULECHECK Metal3.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal3.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal3.8 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal3.9 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal4.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal4.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal4.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal4.8 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal5.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal5.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal5.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal5.8 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal6.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal6.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal6.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal6.8 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal7.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal7.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal7.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal8.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal8.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal8.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal9.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal9.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal10.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Metal10.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.1 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.2 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.3 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.4 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.5 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.6 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.7 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.8 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.9 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.10 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.11 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.12 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.13 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.14 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.15 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.16 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.17 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.18 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.19 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.20 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.21 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.22 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.23 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.24 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.25 ..... TOTAL Result Count = 0 (0)
RULECHECK Grid.26 ..... TOTAL Result Count = 0 (0)
RULECHECK Antenna.poly ..... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal1 .... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal2 .... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal3 .... TOTAL Result Count = 0 (0)

RULECHECK Antenna.metal4 .... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal5 .... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal6 .... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal7 .... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal8 .... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal9 .... TOTAL Result Count = 0 (0)
RULECHECK Antenna.metal10 ... TOTAL Result Count = 0 (0)
——-
— RULECHECK RESULTS STATISTICS (BY CELL)
—
——-
— SUMMARY
—
TOTAL CPU Time: 0
TOTAL REAL Time: 0
TOTAL Original Layer Geometries: 78 (201)
TOTAL DRC RuleChecks Executed: 167
TOTAL DRC Results Generated: 0 (0)

2. Full LVS Report

C A L I B R E S Y S T E M
L V S R E P O R T
REPORT FILE NAME: SRAM_CELL.lvs.report
LAYOUT NAME: /home/am8482/freepdk45_working_directory/SRAM_CELL.sp ('SRAM_CELL')
SOURCE NAME: /home/am8482/freepdk45_working_directory/SRAM_CELL.src.net ('SRAM_CELL')
RULE FILE: /home/am8482/freepdk45_working_directory/_calibreLVS.rul_
RULE FILE TITLE: LVS Rule File for FreePDK45
CREATION TIME: Mon Dec 10 00:15:35 2018
CURRENT DIRECTORY: /home/am8482/freepdk45_working_directory
USER NAME: am8482
CALIBRE VERSION: v2016.3_19.12 Thu Aug 4 11:48:27 PDT 2016
OVERALL COMPARISON RESULTS

* *
CORRECT —

-/
****
CELL SUMMARY
****
Result Layout Source
——— ——— ———
CORRECT SRAM_CELL SRAM_CELL
****
LVS PARAMETERS
****
o LVS Setup:
LVS COMPONENT TYPE PROPERTY element
LVS COMPONENT SUBTYPE PROPERTY model
// LVS PIN NAME PROPERTY
LVS POWER NAME "VDD"
LVS GROUND NAME "VSS" "GROUND"
LVS CELL SUPPLY NO
LVS RECOGNIZE GATES ALL
LVS IGNORE PORTS NO
LVS CHECK PORT NAMES NO
LVS IGNORE TRIVIAL NAMED PORTS NO

```
LVS BUILTIN DEVICE PIN SWAP YES
LVS ALL CAPACITOR PINS SWAPPABLE NO
LVS DISCARD PINS BY DEVICE NO
LVS SOFT SUBSTRATE PINS NO
LVS INJECT LOGIC YES
LVS EXPAND UNBALANCED CELLS YES
LVS FLATTEN INSIDE CELL NO
LVS EXPAND SEED PROMOTIONS NO
LVS PRESERVE PARAMETERIZED CELLS NO
LVS GLOBALS ARE PORTS YES
LVS REVERSE WL NO
LVS SPICE PREFER PINS NO
LVS SPICE SLASH IS SPACE YES
LVS SPICE ALLOW FLOATING PINS YES
// LVS SPICE ALLOW INLINE PARAMETERS
LVS SPICE ALLOW UNQUOTED STRINGS NO
LVS SPICE CONDITIONAL LDD NO
LVS SPICE CULL PRIMITIVE SUBCIRCUITS NO
LVS SPICE IMPLIED MOS AREA NO
// LVS SPICE MULTIPLIER NAME
LVS SPICE OVERRIDE GLOBALS NO
LVS SPICE REDEFINE PARAM NO
LVS SPICE REPLICATE DEVICES NO
LVS SPICE SCALE X PARAMETERS NO
LVS SPICE STRICT WL NO
// LVS SPICE OPTION
LVS STRICT SUBTYPES NO
LVS EXACT SUBTYPES NO
LAYOUT CASE NO
SOURCE CASE NO
LVS COMPARE CASE NO
LVS DOWNCASE DEVICE NO
LVS REPORT MAXIMUM 50
LVS PROPERTY RESOLUTION MAXIMUM 32
// LVS SIGNATURE MAXIMUM
// LVS FILTER UNUSED OPTION
// LVS REPORT OPTION
LVS REPORT UNITS YES
// LVS NON USER NAME PORT
// LVS NON USER NAME NET
// LVS NON USER NAME INSTANCE
// LVS IGNORE DEVICE PIN
// Reduction
LVS REDUCE SERIES MOS YES
LVS REDUCE PARALLEL MOS YES
LVS REDUCE SEMI SERIES MOS YES
LVS REDUCE SPLIT GATES YES
LVS REDUCE PARALLEL BIPOLAR YES
LVS REDUCE SERIES CAPACITORS YES
LVS REDUCE PARALLEL CAPACITORS YES
LVS REDUCE SERIES RESISTORS YES
LVS REDUCE PARALLEL RESISTORS YES
LVS REDUCE PARALLEL DIODES YES
LVS REDUCTION PRIORITY PARALLEL
LVS SHORT EQUIVALENT NODES NO
// Trace Property
```

TRACE PROPERTY mn(nmos_vtl) l l 4e-09 ABSOLUTE
TRACE PROPERTY mn(nmos_vtl) w w 4e-09 ABSOLUTE
TRACE PROPERTY mp(pmos_vtl) l l 4e-09 ABSOLUTE
TRACE PROPERTY mp(pmos_vtl) w w 4e-09 ABSOLUTE
TRACE PROPERTY mn(nmos_vth) l l 4e-09 ABSOLUTE
TRACE PROPERTY mn(nmos_vth) w w 4e-09 ABSOLUTE
TRACE PROPERTY mp(pmos_vth) l l 4e-09 ABSOLUTE
TRACE PROPERTY mp(pmos_vth) w w 4e-09 ABSOLUTE
TRACE PROPERTY mn(nmos_vtg) l l 4e-09 ABSOLUTE
TRACE PROPERTY mn(nmos_vtg) w w 4e-09 ABSOLUTE
TRACE PROPERTY mp(pmos_vtg) l l 4e-09 ABSOLUTE
TRACE PROPERTY mp(pmos_vtg) w w 4e-09 ABSOLUTE
TRACE PROPERTY mn(nmos_thkox) l l 4e-09 ABSOLUTE
TRACE PROPERTY mn(nmos_thkox) w w 4e-09 ABSOLUTE
TRACE PROPERTY mp(pmos_thkox) l l 4e-09 ABSOLUTE
TRACE PROPERTY mp(pmos_thkox) w w 4e-09 ABSOLUTE
CELL COMPARISON RESULTS (TOP LEVEL)

* *
CORRECT —

-/
LAYOUT CELL NAME: SRAM_CELL
SOURCE CELL NAME: SRAM_CELL
———
INITIAL NUMBERS OF OBJECTS
———————————
Layout Source Component Type
———— ———— ——————————
Ports: 5 5
Nets: 7 7
Instances: 4 4 MN (4 pins)
2 2 MP (4 pins)
———— ————
Total Inst: 6 6
NUMBERS OF OBJECTS AFTER TRANSFORMATION
———————————————————
Layout Source Component Type
———— ———— ——————————
Ports: 5 5
Nets: 5 5
Instances: 1 1 _bitv (5 pins)
———— ————
Total Inst: 1 1
****
INFORMATION AND WARNINGS
****
Matched Matched Unmatched Unmatched Component
Layout Source Layout Source Type
————- ————- ——————— ——————— ———————
Ports: 5 5 0 0
Nets: 5 5 0 0
Instances: 1 1 0 0 _bitv
————- ————- ——————— ———————
Total Inst: 1 1 0 0
o Layout Names That Are Missing In The Source:
Ports: GND! VDD!

Nets: GND! VDD!
o Initial Correspondence Points:
Ports: WL BL_BAR BL
****
SUMMARY
****
Total CPU Time: 0 sec
Total Elapsed Time: 0 sec