University of Virginia
Charles L. Brown Department of Electrical and Computer Engineering

ECE 4332: Introduction to VLSI Design
Design Review 1

Spring 2024 2/29/2024

August Bresnaider, Robert Smith, Thomas Johnson

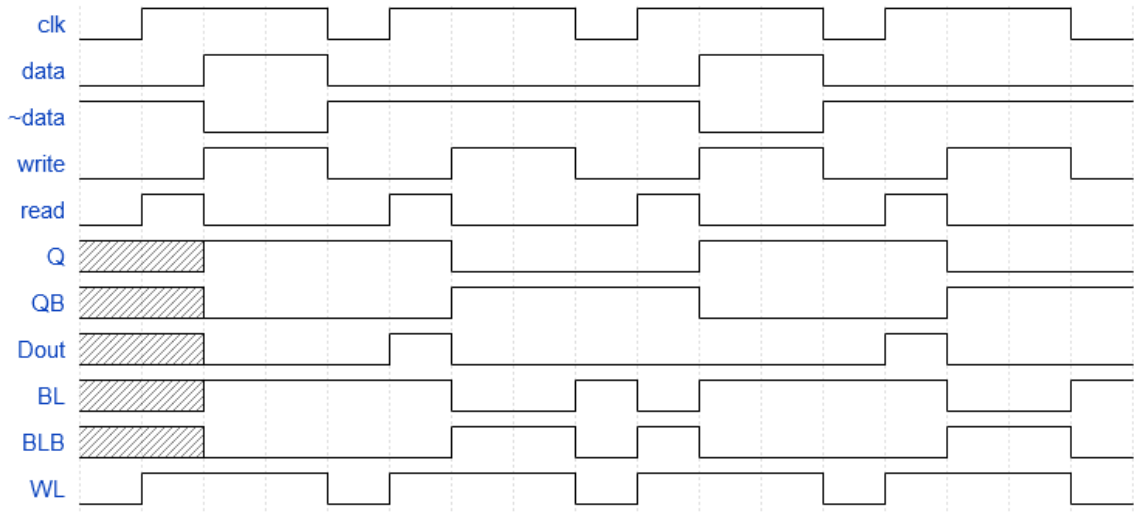
Specifications

Description	Value
Word size	32b
Number of words/row	2
Number of columns/block	64
Number of rows/block	16
Number of blocks in the array	1

Inputs	Outputs
Clock	Dout<31:0>
ADDR<4:0>	
Din<31:0>	
Read	
Write	

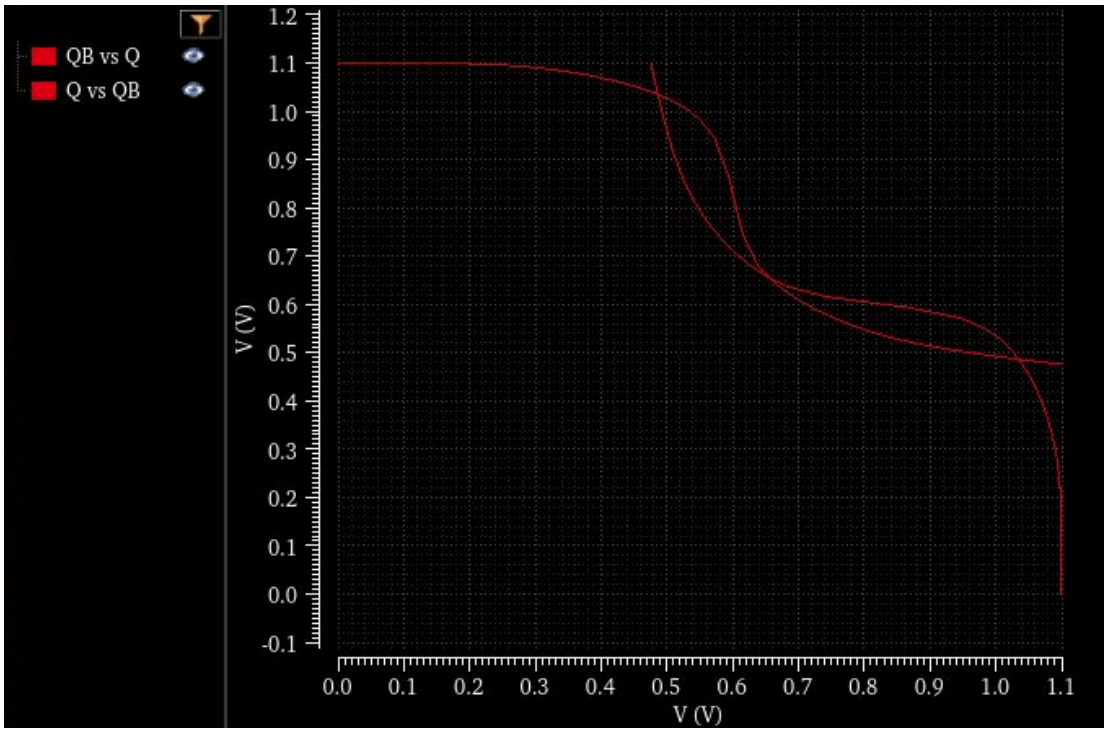
Progress

In this design review, our group laid the foundation of the project. Our group decided to implement our SRAM using 32-bit words, as we would need to use 32 bits in our next design reviews, and it would make our column logic 2x as simple, having half as many required words, at the expense of more bits to grapple with. We began by building a basis of static CMOS logic gates, test benching, and then using these gates to make more complicated structures. Our design uses a 4->16 demultiplexer, multiple 2->1 multiplexers, and a 1->2 demultiplexer, which were all individually built from gates and tested. We then created an expected timing diagram, so we have a testbench for our single bitcell:



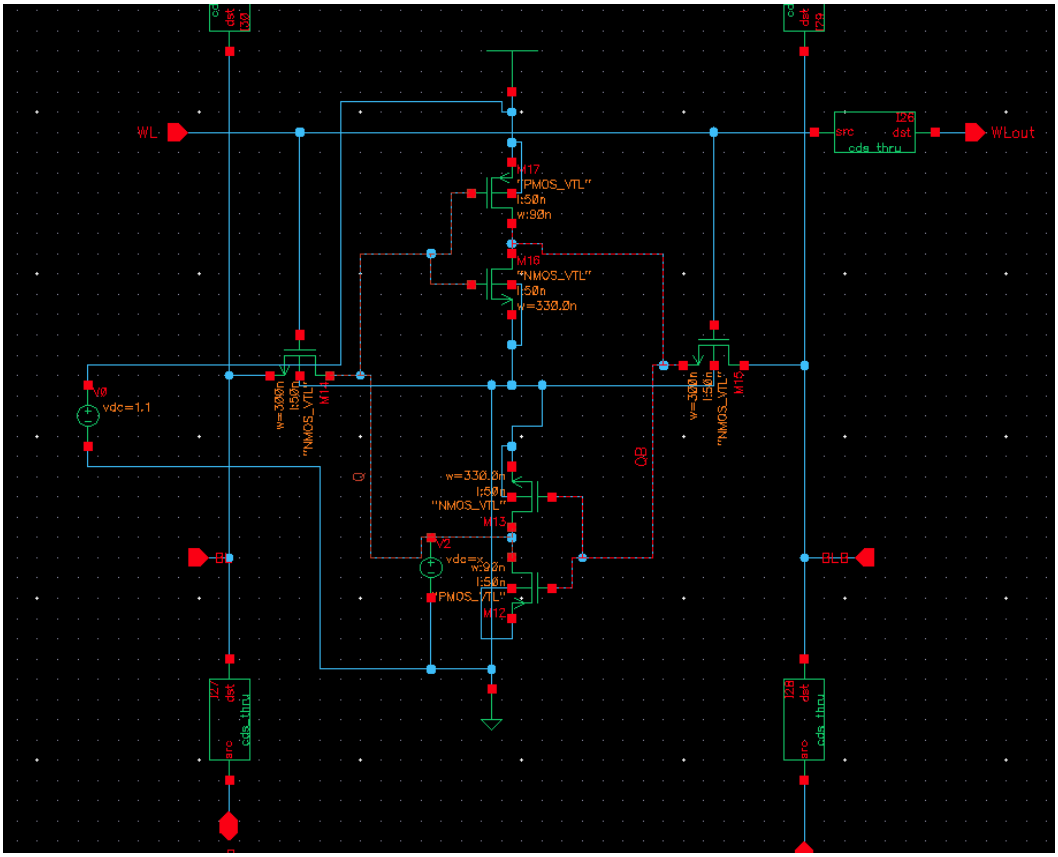
Expected Bitcell Output Waveform

We then performed research, calculations, and collaborated with other teams to find viable sizings for our pull-up, pull-down and access transistors, implementing these into our bitcell design, then simulating. We generated the following SNM plot to demonstrate the viability of our research:

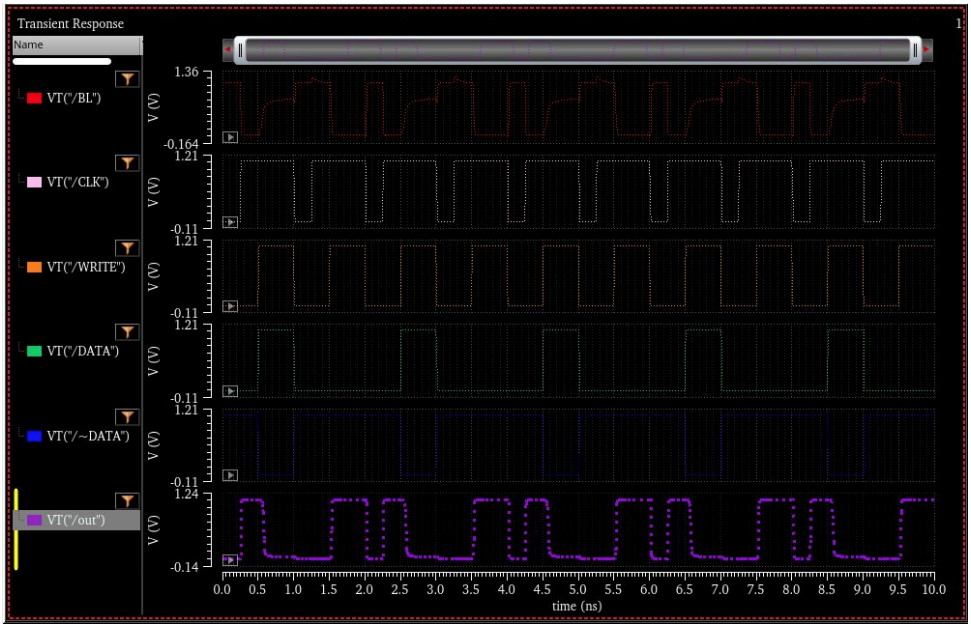


SNM Plot of Single Bitcell in “Hold” Operation

Below shows a diagram of our bitcell, as well as the operation of a single bitcell in isolation. It should be noted that the waveform ‘out’ is representative of the inverted BL signal.

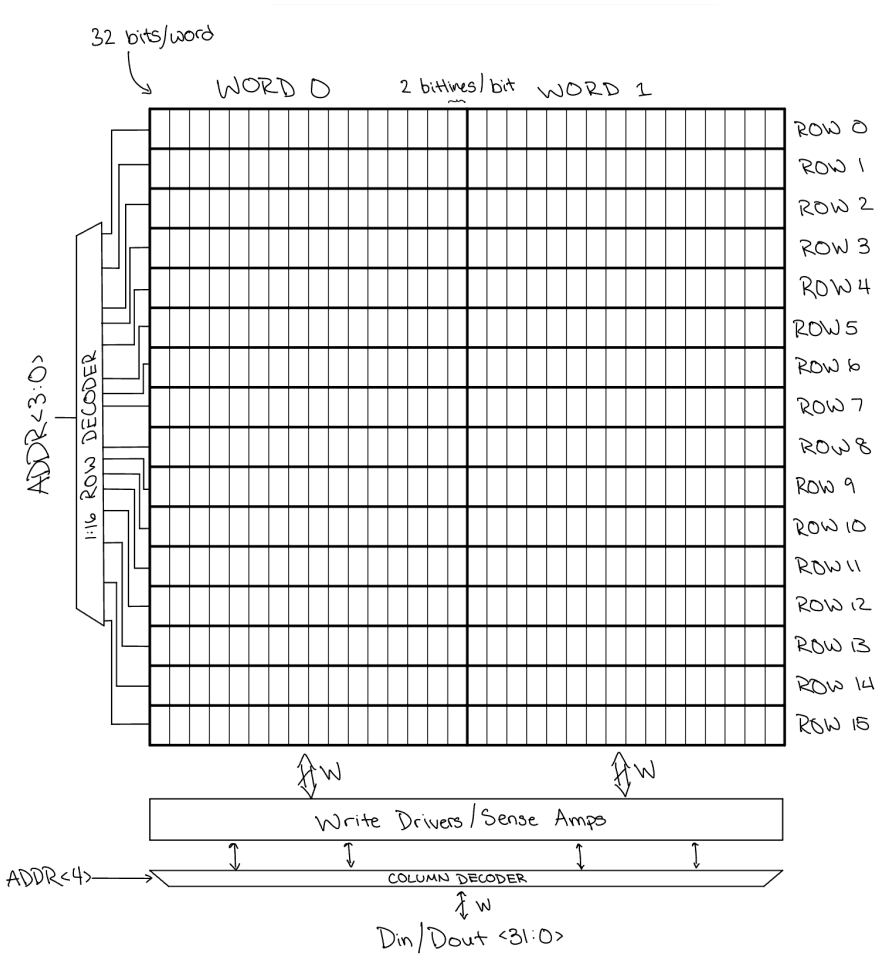


Single Bitcell Schematic



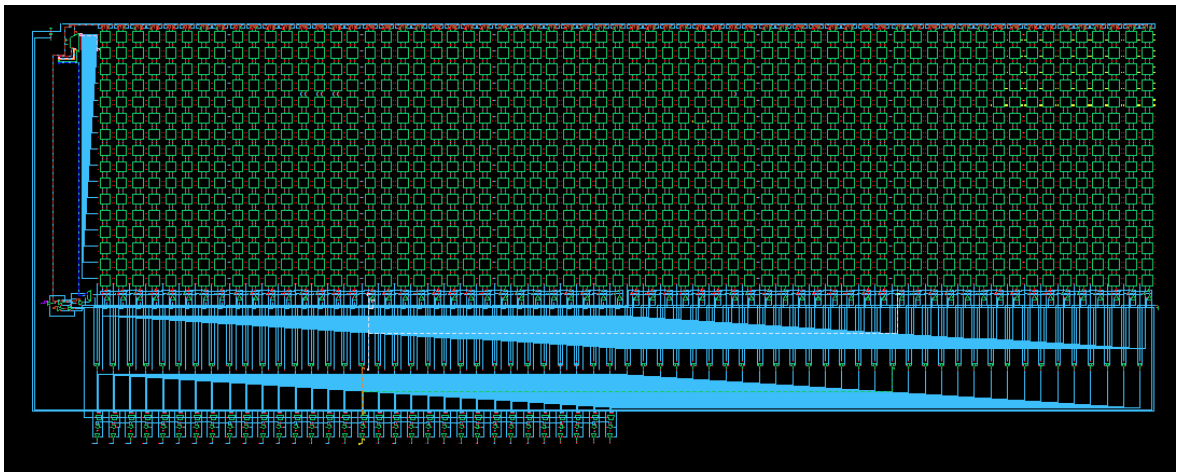
Single Bitcell Timing Diagram

Then, simulations and research was done to determine the ideal pull-up and cell ratios. Because the PMOS transistors are the smallest, it was set as the benchmark with its width as 90nm. From here, we created all of our switching logic to directly address and access individual blocks. The following block diagram was drawn to show our vision for implementation:



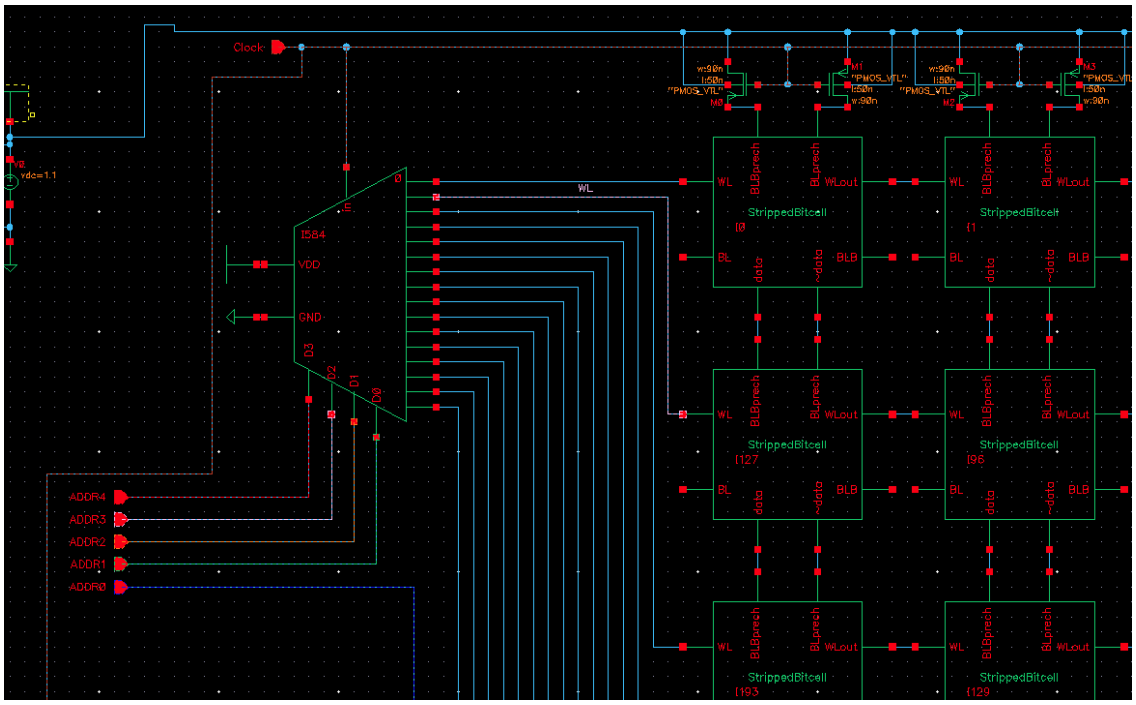
SRAM Block Diagram

From this conceptual design, we started working on simulation in Cadence, and made the following module:



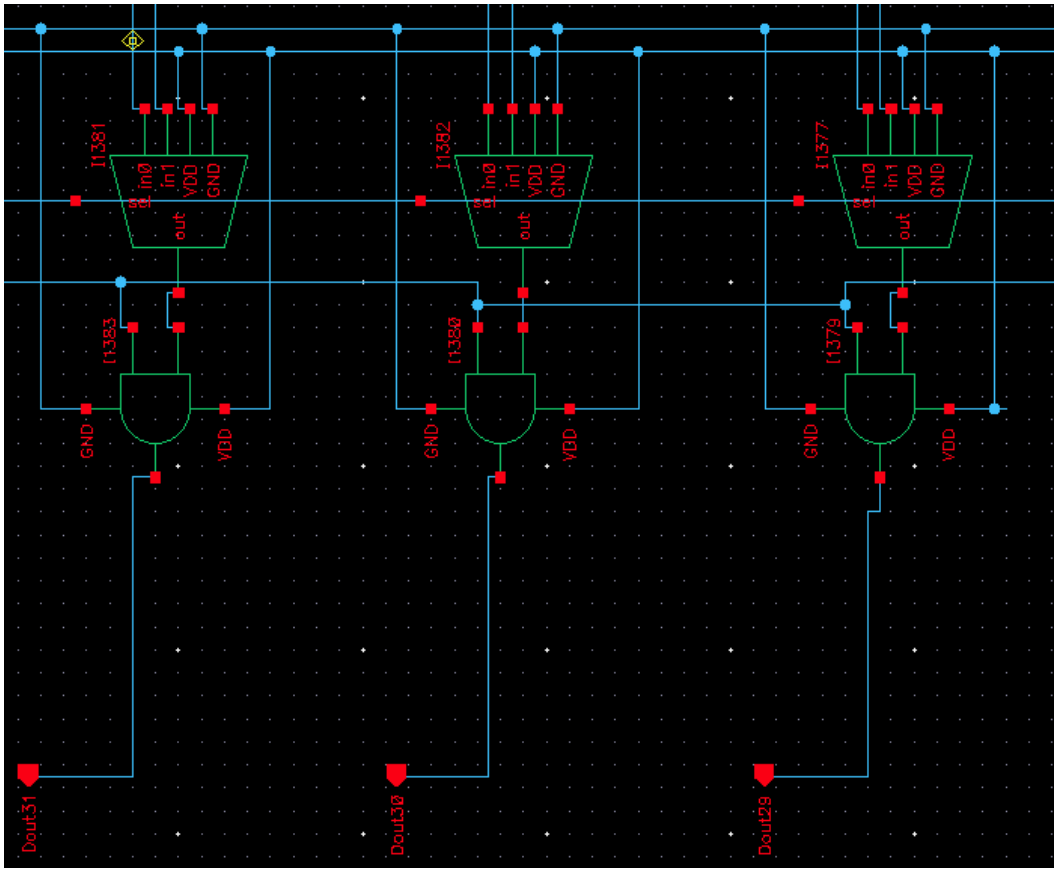
1024 Bit SRAM Block

Highlighting some key components of this design, we will first look at row addressing:



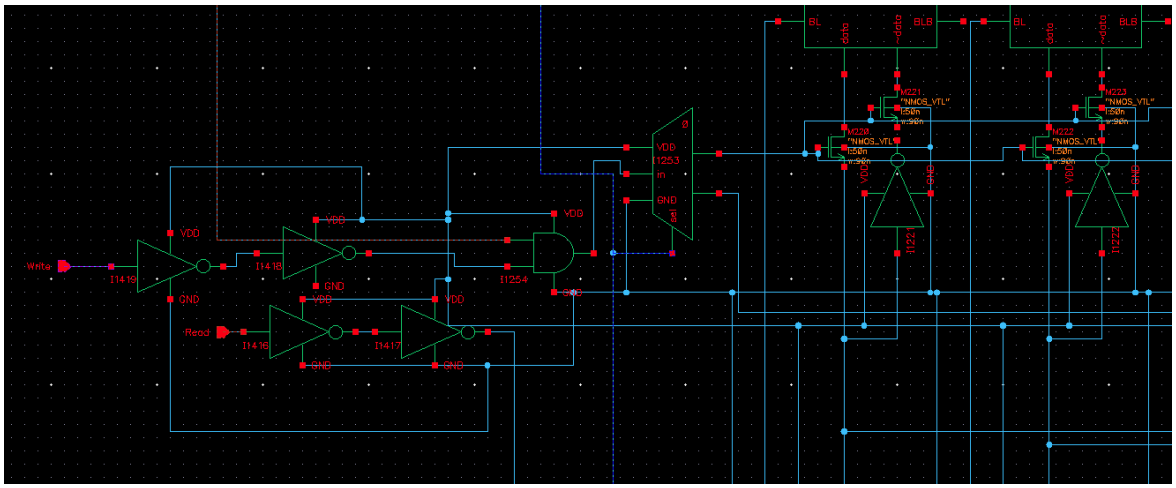
SRAM Row Demultiplexer

As can be seen, the highest order 4 bits are being used in a 4->16 bit demux to send clock signals to. This demux connects directly to the wordlines of each bitcell, and opens the access transistors for the requested row any time the clock signal is high (i.e., the circuit is not pre-charging). While the highest order bits are used to address the rows, the lowest order bit, ADDR0, is being used to select the word in the row:



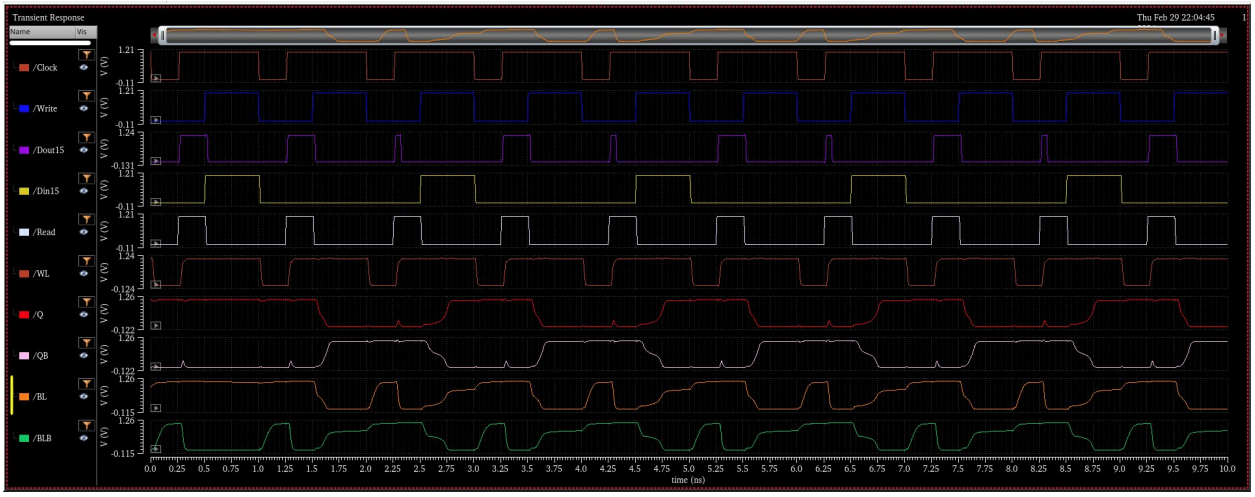
SRAM Output Multiplexing

At the bottom of the schematic, there is a row of 32 2->1 multiplexers, which are being used to decide whether to allow the first word or the second word to reach the output, based on the selector bit, ADDR0. After passing the mux, the output is AND'ed with the Read signal to see if an output is being requested at all.



SRAM Bitline Logic

Lastly, we have our logic that dictates when data may traverse the bitlines. This logic is set up such that ADDR selects which word to send data to, and data is allowed onto the bitlines if and only if both Write and Clock are high. Below is a timing diagram of a cell read / write:



SRAM Random Access Timing Diagram

While our design spikes high momentarily when the output is low, this is a rather short period of time, and is most likely due to not buffering our inputs. This is a conscious design decision, as we will not be able to buffer inputs in later design revisions, so we wanted to work without buffers for this first revision as well.

By designing the gates, then bitcell, and finally the complete block, our group has the circuitry to further develop and optimize into either Low Power SRAM or High Speed Cache. Before the proposal, our group needs to decide which path we want to take. This decision involves research into the differences between the two applications and the novelty our design could have.