August Bresnaider, Thomas Johnson, Robert Smith
ECE 4332
Project Proposal

Prof. Calhoun and the Executives at PICo,

I hope this memo finds you well. I'm sure you will be pleased to know that after thorough R&D, our team (subdivision "Illogical Effort") will be working to develop an SRAM module made in accordance with your Low Power contract. As such, we will be developing 1Mb worth of memory, and working to aggressively minimize the following target metric:

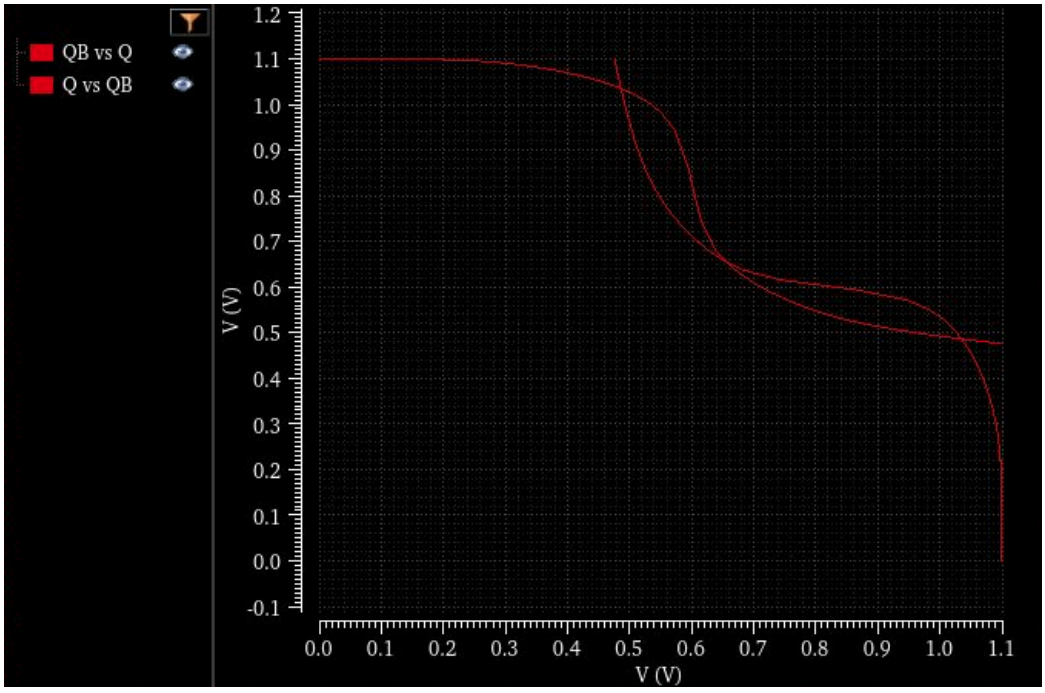
$$(Active\ Energy\ per\ Access)^2 * Delay * Area * (Idle\ Power)$$

This document intends to serve as a guide as to what design requirements have already been met, what challenges lay ahead still, and what critical features our design will entail.

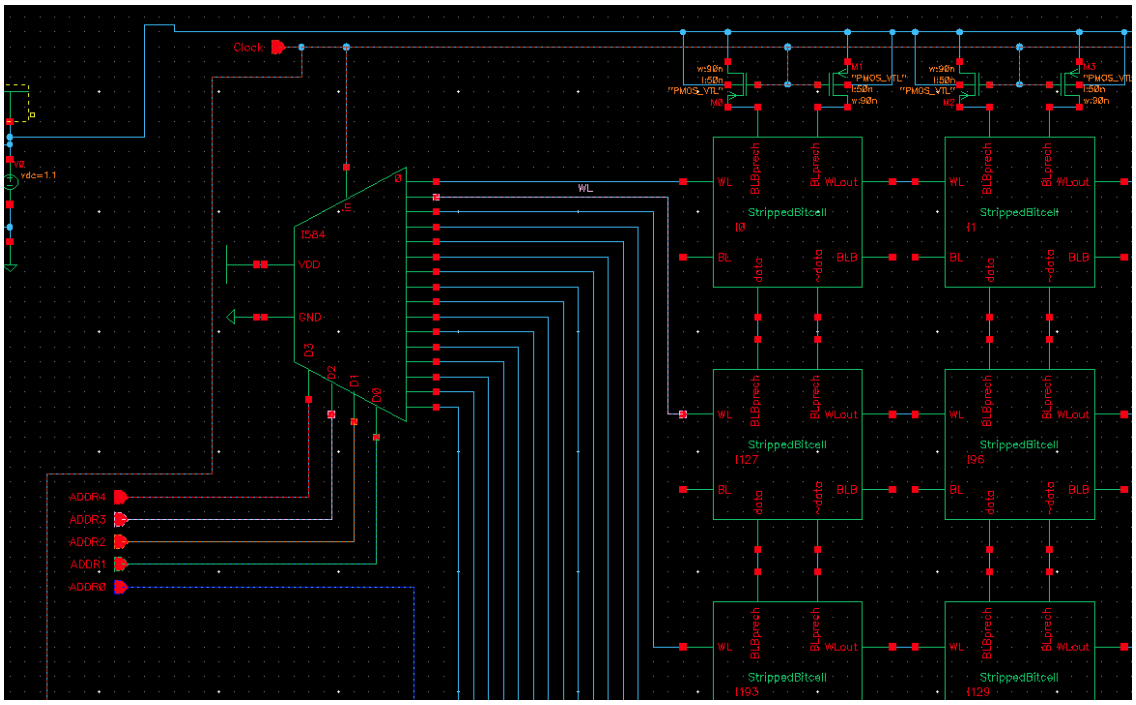
I. Challenges Met

As of the writing of this document, we have been able to develop a working block's worth of SRAM. This block consists of 1024 bits, with 64 bits (2 words) per row, and 16 rows, all randomly accessible. When scaling the size of our working memory up to 1Mb, it becomes prohibitively more and more difficult to continually add more rows and columns to the block size, meaning that we will most likely keep a similar block size in the final design. Being that this is the case, it is useful to highlight some important elements of said block:

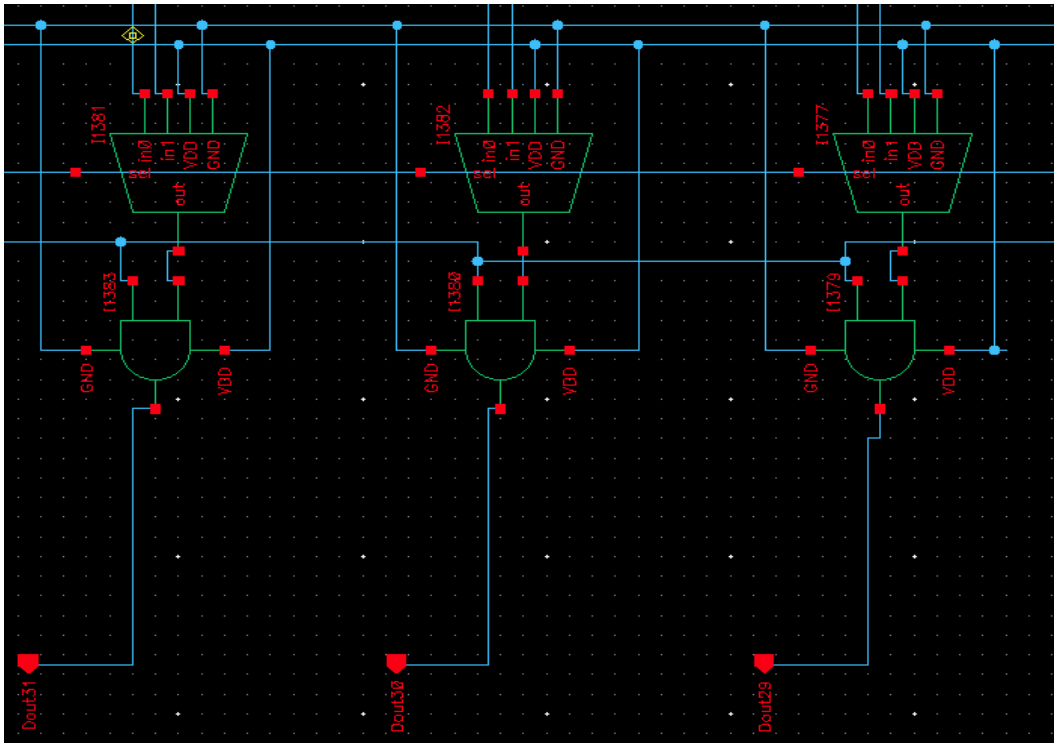
We created a preliminary cell ratio and pull-up ratio which was able to produce the following butterfly plot while in "hold" operation:



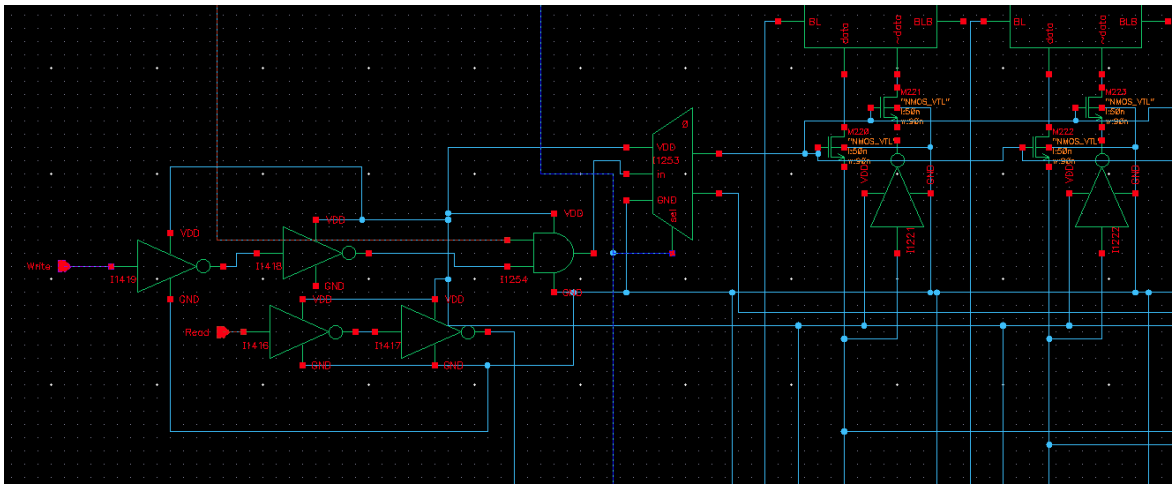
We were able to build and verify a 4-bit demultiplexer to send input to the correct wordlines:



We implemented output multiplexing to determine which word in a row is being read:



We implemented column demultiplexing to determine which bitlines to send write data to:



II. Further Challenges

In order to complete our design in a timely manner, there are still a decent number of hurdles we have yet to overcome. To begin, we will need to determine which bitcell model we will use. While the standard 6T transistor is considered the industry standard, we have come across papers showing that a 7T, 8T or 10T bitcell may be advantageous for low-power design [4]. Once a bitcell has been decided, we will run a series of parameter sweeps to determine the best transistor widths, taking into consideration the “read”, “write”, and “hold” SNMs, as well as the average power consumption of each configuration. When we are confident in the operation of our bitcell, we will redesign our 1024 bit block, verifying that its operation is sound. When we are confident in this, we will look at ways in which we can reduce leakage in our addressing logic. We have been able to find promising sources on reducing power consumption through our decoders, by implementing a pre-decode (using hierarchy and block decoders) [3] [5] which leverages principles of logical effort to increase decoder efficiency, as well as reducing leakage through unused decoder outputs.

Our final big challenge will be implementing hierarchy. As previously mentioned, splitting our memory model into blocks will drastically assist in scaling efforts, as it reduces the maximum capacitance that any one word-line or bit-line may contain. This will require the implementation of block decoders and more complicated output multiplexing, but design and verification of a hierarchy will be the final large hurdle to overcome in the design of the memory module. Crucially, this design requires determining an optimal block size, which balances the low maximum capacitances of smaller blocks with the simpler addressing / output multiplexing of larger blocks. The testing of this hierarchy requires the use of Monte Carlo simulations, in

which we will generate NMOS and PMOS parameters according to a Gaussian distribution to determine the operation of our memory at each of the process corners.

III. Special Features

This section notates some non-standard features we are considering adding to our design. Though our inclusion of certain features is not a guarantee of its ending up in our final design, we will be performing a wide range of simulations for power draw, static noise margin, delay, and area to determine the feasibility of each of these features. We discussed the idea of implementing sleep functionality on unused blocks. This would involve adding MTCMOS into each block, and keeping the block in sleep mode until it receives its first write, where it would no longer sleep for the rest of the SRAM's operation. We could use an SR latch to determine whether or not a block has been written to. This drastically reduces initial power consumption, while also providing a benefit for contiguous memory access, at the expense of increased area, having to use significantly more transistors. Another idea would be to implement a reverse body bias; when looking to reverse leakage current, we could drive the gate voltage of some transistors below 0V, which would increase V_{GS}, thus decreasing leakage. Lastly, we believe in implementing a strong-arm sense amplifier, as it provides our design with a fast, but power efficient way to read a value from a bitcell.

IV. Timeline

Items are to be due at midnight on the listed day, unless otherwise posted.

3/16: Research special features, sense amplifier (Robert)

3/18: Submit completed project proposal, due 11am (August, TJ, Robert)

3/19: Determine bitcell architecture (TJ, Robert)

3/22: Size cell ratio and pull-down ratio, implement working block (August)

3/28: Implement hierarchy, obtain minimum viable product (Robert, August)

3/31: Implement final special features (TJ, Robert)

4/2: Simulate, test, optimize (August, Robert, TJ)

4/4: Submit design review 2, due 11am (August, Robert, TJ)

References

(1) A. Bhaskar, "Design and analysis of low power SRAM cells," 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 2017, pp. 1-5, doi: 10.1109/IPACT.2017.8244888. keywords: {SRAM cells;Transistors;Delays;Logic gates;Switching circuits;Power demand;Low Power SRAM;Gated VDD SRAM Cell;MTCMOS;MTCMOS SRAM Cell;6T SRAM;Power Reduction in SRAM Cell}, https://ieeexplore.ieee.org/document/8244888

(2) Margala, Martin. (1999). Low-power SRAM circuit design. 115 - 122. 10.1109/MTDT.1999.782692.

(3) V. S. Melikyan, K. O. Petrosyan, A. K. Mkhitaryan and H. V. Margaryan, "The Method Of Low Power, High Performance And Area Efficient Address Decoder Design For SRAM," 2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO), Kyiv, Ukraine, 2020, pp. 276-279, doi: 10.1109/ELNANO50318.2020.9088809.

(4) N. Rathi, A. Kumar, N. Gupta and S. K. Singh, "A Review of Low-Power Static Random Access Memory (SRAM) Designs," *2023 IEEE Devices for Integrated Circuit (DevIC)*, Kalyani, India, 2023, pp. 455-459, doi: 10.1109/DevIC57758.2023.10134887.

(5) 1. ADAM TEMAN, Adam. VLSI - Lecture 9B: Row decoder design. *YouTube* [online]. 25 May 2021. [Accessed 17 March 2024]. Available from: https://www.youtube.com/watch?v=OJR-_onSvRg