

# Implementation and Simulation of a FreePDK 45nm SRAM Array

Pranav Mathews

Electrical and Computer Engineering  
Georgia Institute of Technology  
Atlanta, GA  
pmatthews30@gatech.edu

Garrett Botkin

Electrical and Computer Engineering  
Georgia Institute of Technology  
Atlanta, GA  
gbotkin3@gatech.edu

James Read

Electrical and Computer Engineering  
Georgia Institute of Technology  
Atlanta, GA  
jread6@gatech.edu

**Abstract**—In this report, we present a 512b SRAM memory system complete with peripheral circuitry necessary for operation. We designed the SRAM memory system using NCSU’s FreePDK45 library and simulated it in Cadence Virtuoso. The system operates at 1 GHz with a nominal supply voltage between 730mV and 800mV. The read and write operation complete in one clock cycle each, where one 8b word is accessed in parallel. We provide traces of a 3-cycle write followed by a 3-cycle read to verify the functionality of the system. Additionally, we analyzed the critical delay and power at the component and system levels as well as the trade-offs in performance with varied supply voltage.

**Index Terms**—SRAM, FreePDK 45nm, Cell, Array

## I. INTRODUCTION

In this section, we briefly describe the SRAM system components and operation, then in subsequent sections describe the design and performance of each individual component.

The memory array is comprised of 16x32 6T SRAM cells where each row stores 4 8b words. The bits of each word are grouped in 4’s by their significance e.g. the MSBs of each word are adjacent to each other. The bitlines (BLs) of each group are fed into 4:1 MUXes allowing for a full word to be read/written in parallel. 6-bit address stored in address registers are used to select a word in the array. The first 4 bits of an address are decoded to select 1 of the 16 rows and the remaining 2 bits are decoded separately to select 1 of the 4 BLs in each group of bits. The write operation is executed by loading an 8b word into data registers, and the selected BLs are driven with the data bits by a write driver circuit. To conduct the read operation, precharger circuits charge the selected BLs to the supply voltage enabling a sense amplifier to detect the memory state stored in each SRAM cell. Every circuit mentioned was designed by our team at the transistor level using NCSU’s FreePDK45 library.

## II. THE SRAM COMPONENTS

### A. Address and Data Registers

We designed the 6b Address and 8b Data registers using positive edge triggered mux-based latches and negative edge triggered mux-based latches linked together to build a positive edge master-slave flip-flops. Table I lists the DFF’s key power-performance metrics at a nominal supply voltage of 800 mV.

TABLE I  
PERIPHERAL POWER-PERFORMANCE PARAMETERS

Component	Delay (ps)	Power (fW)
WL Decoders	116	3.51 pW
DFF	Hold: 13 Setup: -13 Clk-Q: 92.5	28.3
Column Mux	400	15.18
Write Driver	275	5.729
Sense Amp	293	20.4
Precharger	100	22.28

Because the inverters were designed with mismatched pull-up and pull-down strength, the CLK-Q delay timings vary between a "0" to "1" transition and vice versa, thus the reported delay timings are the worst-case of the two transitions. Both the setup time and CLK-Q delay meet timing requirements of < 15% and < 10% of the 1ns clock period.

### B. Wordline and Column Select Decoders

We implemented the wordline (WL) and column select decoders using pseudo NMOS 3 input NOR gates and chained the 2x4 column selection decoders to form the 4x16 decoder necessary for the WL logic. The WL decoder delay meets timing requirements of < 50% of the clock high-time. We chose to implement the decoders in Pseudo NMOS for the low delay of the NOR gate in comparison to a Static CMOS NOR gate, at the expense of increased power consumption. Table I lists the WL decoder’s and column decoder’s key power-performance metrics at a nominal supply voltage of 800 mV.

### C. Column Multiplexers

The column multiplexers were designed by feeding the outputs from the column select decoders into four separate transmission gates, enabling the selection of individual columns for connection to the write and read circuits. The use of transmission gate based column multiplexers was essential to reducing the critical delay along the read and write path compared to a more traditional multiplexer. Table I lists the column multiplexers’ key power-performance metrics at a nominal supply voltage of 800 mV.

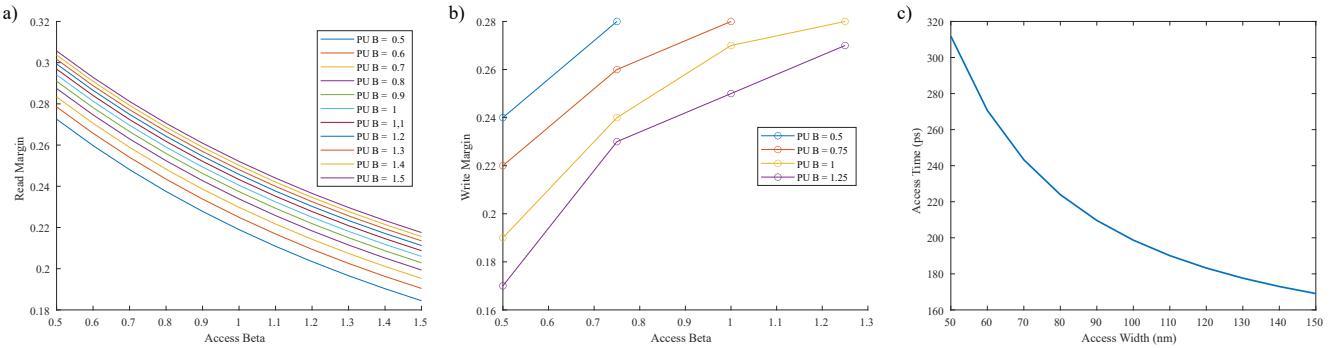


Fig. 1. SRAM timing values vs intermediate values tried. All  $\beta$  referenced to a PD W = 100nm. Access time assume  $\Delta$ bit of 200 mV

#### D. Write Circuit

To write to the SRAM cells, a write driver circuit connects the BL and BL bar to one transmission gate each. The data bit is applied to the BL transmission gate and an inverter inverts the data and applies it to BL bar. Table I lists the write driver's key power-performance metrics at a nominal supply voltage of 800 mV.

#### E. Read Circuit

We designed the read circuit using a current latch based sense amplifier and a bit line precharger for each bit line of the SRAM Array. 8 sense amplifiers in total are implemented to sense an 8b word in parallel.

With a worst case mismatch scenario (10%), we found the  $\Delta V_{bit}$  to be 300 mV by finding the minimum  $\Delta V_{bit}$  that still met the required sense amplifier delay (< 40% of the clock period). This delay increased as VDD decreased. This delay can be further decreased by adjusting the capacitance seen at the output of the amplifier. With the given 10fF we meet 300mV, but if the output is fed into a latch that then drives the 10 fF capacitance it can be reduced much further. Table I lists the sense amplifier's key power-performance metrics at a nominal supply voltage of 800 mV.

We designed the precharger using three PMOS gates that equalize and charge BL and BL bar when clock is low. Table I lists the precharger's key power-performance metrics at a nominal supply voltage of 800 mV.

#### F. SRAM Cell

Table II lists the various sizes of 6-T SRAM cell while Table III lists the resulting margins and performance based on the chosen transistor sizes. Both our read and write margin met requirements of > 30% of Vdd. Cell access speed meets the requirement of  $\approx 500$ ps.

Intermediate values were swept with a fixed  $W_{PD}$  as shown in Fig. 3. A higher width of the NMOS pulldown was initially set because both read speed and read noise margin are strongly affected by a strong pull down, while the write margin is affected less. For a more careful redesign, we could vary the pull down strength to get an even more optimal sizing.

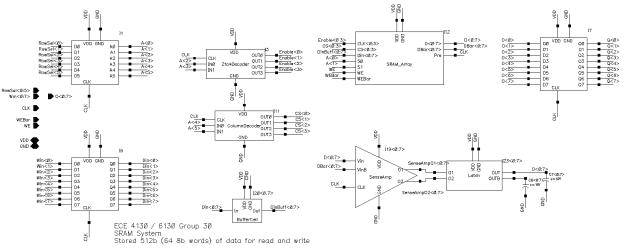


Fig. 2. Top-Level Schematic of the SRAM Array

TABLE II  
SRAM METRICS

Metric	Value
Pull-Up PMOS	50 nm
Pull-Down NMOS	100 nm
Access NMOS	75 nm
Read Margin	259.8 mV
Write Margin	260 mV
Access Speed	406 ps

#### G. Buffer Cell

We designed and built a simple four stage inverter chain with an up-sizing factor of three to assist in reducing the capacitive load on component outputs and the delay of these components as a result of the reduction in capacitive load.

### III. THE SRAM SYSTEM

#### A. Integration

Integration of the SRAM array components were performed hierarchically, resulting in a final schematic shown in Fig. 2. Integration was intentionally done this way to minimize human errors during place-and-route. Such an integration allows us to place transistors only in the lowest level modules, greatly reducing the time required to make adjustments to the system.

#### B. Signal Synchronization

The registers latch a signal and output it to the decoders. However, because of the tree-like structure of our decoders the

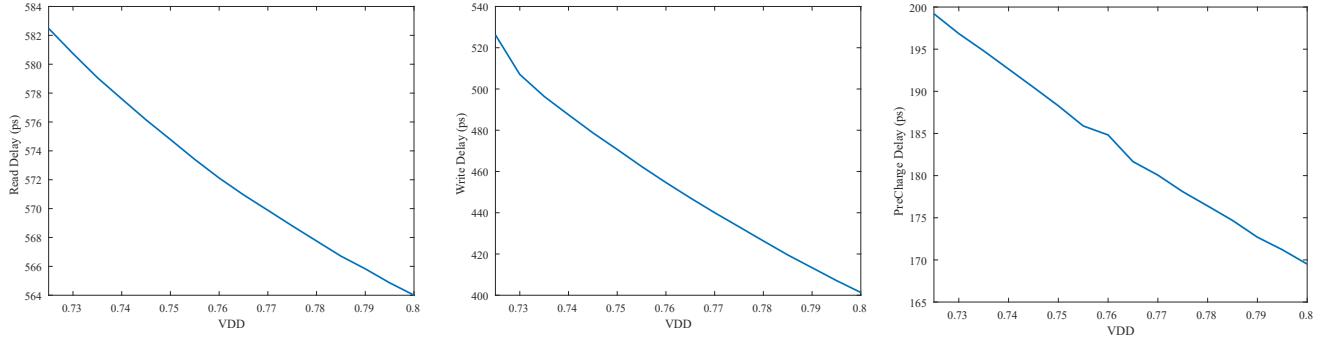


Fig. 3. System level delays vs VDD

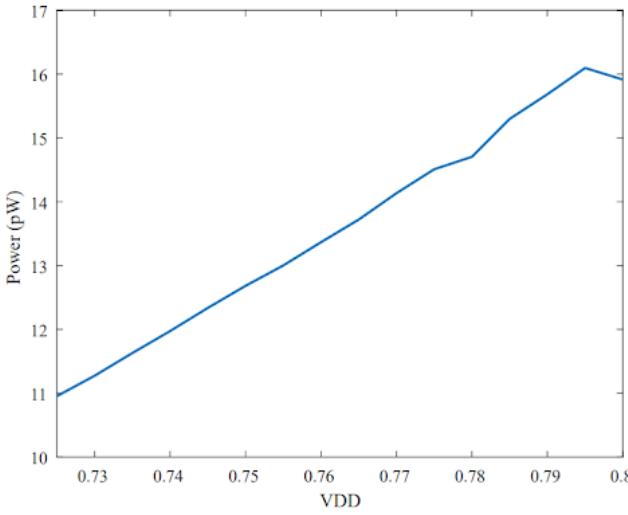


Fig. 4. Total Array Power Dissipation vs VDD

register in the worst case drives four 2to4 decoders and in the best case drives one. Delay requirements are met across both conditions, but the signal rises faster to the first part of the tree. The total delay between register to decoder 1 is less than register to the parts of the tree, which can result in glitching. To solve this, the registers were oversized so that the difference in delay is minimized.

Multiple system components require an inverter clk signal. A single inverter driving all of those gates would result in a large delay, so in order to synchronize each clk bar to the given clk signal, local inverters were used for minimum delay.

### C. Power and Performance

Table III lists the SRAM array's key power-performance metrics at a nominal supply voltage of 800 mV, over three clock cycles consisting of two writes and one read.

We also varied the nominal supply voltage from 800mV to 730 mV and plotted the affect on total read delay in Fig. 3a, total write delay in Fig. 3b, total pre-charge delay in Fig. 3c, and total array power dissipation in Fig. 4.

TABLE III  
SRAM ARRAY POWER-PERFORMANCE PARAMETERS

Metric	Value
Read Delay	564 ps
Write Delay	401 ps
Pre-Charge Delay	170 ps
Array Power	16 pW

## IV. TESTING METHODOLOGY

### A. Individual Validation

The SRAM array was tested using a two phase process that targeted both individual components of the array as well as integration of the components. Each component was given a personalized test bench schematic and simulation that were used to validate functionality before integration of the component into the SRAM array. The test benches ran simulations of all possible inputs into the components to ensure that there was no failure of the design or delays that exceeded required design specifications.

### B. Integrated Validation

After individual validation, components were integrated together and test benches for combinations of different components were created. Once the operation of each group of components was verified, a system-level test bench was created. In all test benches, we made sure to simulate all possible input combinations to guarantee the components and ultimately the entire system operate correctly in all conditions. In the system-level test bench, the 8b write and read operations to multiple different addresses were simulated in a sequence. Fig. 5 shows writing to three different addresses, then reading those same addresses immediately after. These addresses were chosen to cover different parts of the array, and the stored word bits were chosen to test different possible transitions of the bitlines and decoders throughout the read/write process.

## V. A SECOND CHANCE REVIEW

### A. Limitations

Our system had multiple sources of signal propagation delay. When the power supply is lowered, some operations

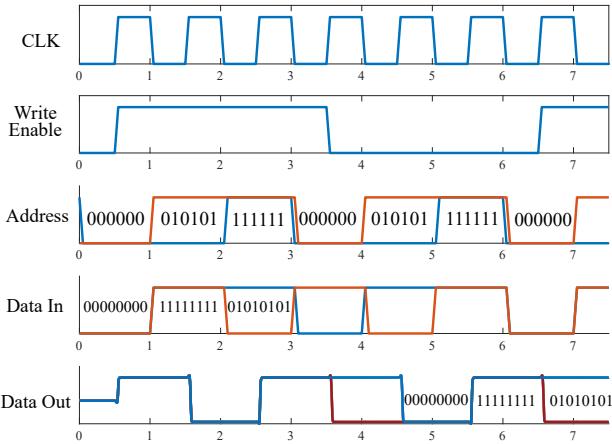


Fig. 5. SRAM system tested in a full 3 write 3 read cycle

do not work because of the extended delays while keeping to the same clock frequency. These delays also lead to large sizing on all peripherals.

Ideally the access time for the read operation is most of the clock high period. However in our design, from what we suspect is clock feedthrough and charge injection from the precharge circuit, the bitlines initially rise above Vdd and then start discharging after some delay. This massively reduces access time, meaning we need a large access transistor size.

Our system also consumes a large amount of power due to unoptimized sizings. We initially sized our transistors to make the system work, however some T-gates, inverters, etc ... are arbitrarily large to drive big capacitive loads. Instead logical effort and inverter chains could be used to optimize area with delay and power.

### B. Revisions

If we had a chance to redesign, we would have spent more time on system level integration testing to narrow down optimal sizing on components. For example, column select T-gate sizing had a significant effect on making the read circuit work, and could be further optimized. The wordline decoders are also extremely large because they directly drive the 40 FF wordlines, but if we had designed driving buffers this could be mitigated.

The SRAM cell could also be more carefully designed. As shown in Fig 3 all intermediate values were swept assuming that  $W_{PD} = 100$  nm. A more careful design would also consider the changing effect of the pulldown width rather than fixing it.

## VI. CONCLUSION

In this report, we presented a holistic design of a 512b SRAM system complete with peripheral circuitry. We describe our design methodology, integration and testing strategy and validated the pre-layout operation of the system. Additionally,

TABLE IV  
GROUP MEMBER CONTRIBUTIONS

Group Member	Milestone 1
Pranav Mathews	Creation of DFF
Garrett Botkin	Creation of the Address and Data Registers. Integration of the Address Registers and WL/CS Decoders. Creation of Milestone 1 Slides and Necessary Documentation for Slides.
James Read	Creation of Decoders
	Milestone 2
Pranav Mathews	Creation of the SRAM Cell and Cell Array Creation of Sense Amplifier System-level integration Creation of Milestone 2 Slides
Garrett Botkin	Creation of the Column Multiplexers. Creation of Buffer Cell. Editing of Milestone 2 Slides and Creation of Necessary Documentation.
James Read	Creation and Integration of Write Circuits Creation of and Integration Precharge Circuits Creation of Milestone 2 slides Gathering Power-Performance Data for Peripherals
	Report
Pranav Mathews	Generated all plots for DFF, Sense Amp, SRAM Cell, and System Level Array. General report cleanup, Section IV and V
Garrett Botkin	Worked on The SRAM Components Section, The SRAM System Section. The Testing Methodology Section and added most of the Tables and Figures.
James Read	Abstract, Introduction, Conclusion System Integration, Write Circuit Sections General Editing and Formatting

we present an analysis of the SRAM cell design parameters and system level power-performance trade-offs. Ultimately, our design can operate correctly at a minimum nominal supply voltage of 730mV at a frequency of 1GHz.