
Research and Design of Low Power SRAM

Everestus Ezike

Zhiqi Hao

Haolan Zhan

Date: 6/2/2022

Instructor: David Zaretsky

Course: COMP_ENG 392 VLSI SYSTEM DESIGN PROJECTS

Table of Contents

1. List of Figures (2)
2. List of Tables (4)
3. Executive Summary (5)
4. Body (6)
 - a. Introduction (6)
 - b. Broader Considerations (6)
 - c. Design Constraints and Requirements (7)
 - d. Design Description (8)
 - e. Design Optimizations (17)
 - f. Testing/Simulations (19)
 - g. Implementation/Synthesis (29)
 - h. Conclusions (33)
5. References (35)
6. Appendix (36)
 - a. 6T SRAM Introduction (36)
 - b. 8T SRAM Introduction (37)
 - c. Leakage Reducing Transistors (38)

1. List of Figures

- *Fig. 1 Top Level Block Diagram*
- *Fig. 2 Power Gating Using Multi-Threshold CMOS (MTCMOS) Design*
- *Fig. 3 6T Single-Ported MTCMOS SRAM Cell (6TSPMTCMOS)*
- *Fig. 4 Read Stability Criteria*
- *Fig. 5 Write Stability Criteria*
- *Fig. 6 Transistor Sizing of 6TSPMTCMOS*
- *Fig. 7 8x8 SRAM Cell Array Module*
- *Fig. 8 8x8 SRAM Cell Array*
- *Fig. 9 Write Circuitry*
- *Fig. 10 8x1 Write Circuitry Module*
- *Fig. 11 Precharge Circuit*
- *Fig. 12 8x1 Precharge Unit Module*
- *Fig. 13 Sense Amplifier Circuit*
- *Fig. 14 8x1 Sense Amplifier Module*
- *Fig. 15 Lyon Schediwy Decoder Circuit*
- *Fig. 16 Lyon Schediwy Decoder Module*
- *Fig. 17 Full System Level Architecture of SRAM Design*
- *Fig. 18 Single Column of SRAM design*
- *Fig. 19 Traditional 6T SRAM*
- *Fig. 20 6T SRAM with Leakage Reducing Transistors*
- *Fig. 21 6T SRAM with Sleep Transistors (Chosen Cell Design)*
- *Fig. 22 6T SRAM with Leakage and Sleep Transistors*
- *Fig. 23 Dual Ported SRAM with Sleep Transistors*
- *Fig. 24 Dual Ported SRAM with Leakage and Sleep Transistors*
- *Fig. 25 Simulation Schematic*
- *Fig. 26 Component Tested in Fig. 24*
- *Fig. 27 Traditional 6T SRAM design with 2 Added Sleep Transistors: Simulation Waveform*
- *Fig. 28 6T SRAM with Sleep Transistors: Simulated Waveform Demonstrating Cell Stability During Sleep Mode*
- *Fig. 29 Simulation Schematic for 64BitLPSRAM*
- *Fig. 30 and 31 64BitLPSRAM Simulation Waveforms for Test 1: 3 Consecutive Write-Read Sequences on Address 0*
- *Fig. 32 and 33 64BitLPSRAM Simulation Waveform for Test 2: Write-Read on Address 2, Write-Read on Address 5, Read Address 2 Again, Read Address 5 Again*
- *Fig. 34 and 35: Fig. 32 and 33 64BitLPSRAM Simulation Waveform for Test 3: Two Consecutive Writes on Address 7, then a Read*
- *Fig. 36 SRAM Cell Layout*

- *Fig. 37 Precharge Layout*
- *Fig. 38 Write Circuitry Layout*
- *Fig. 39 Sense Amplifier*
- *Fig. 40 Decoder*
- *Fig. 41 Full SRAM array*
- *Fig. 42 6T SRAM Schematic*
- *Fig. 43 6T SRAM Schematic*

2. List of Tables

- *Table 1 Chosen SRAM design with 2 Added Sleep Transistors: Energy and Average Power Measurements*
- *Table 2 Traditional 6T SRAM design: Energy and Average Power Measurements*

3. Executive Summary

SRAM cells are an important design component for many circuits, electronics, and processors found everywhere in industry. The task of minimizing power and area of SRAM cells is significant when considering the functionality and efficiency of SRAM cells. Through extensive research into different designs, schematics of a select few SRAM cell designs were created and simulated in Cadence Virtuoso for the purpose of measuring energy and power consumption. The SRAM cell utilized in our final design was chosen to meet specifications on minimized power consumption and area. The main feature of our SRAM cell is the addition of power gating, or sleep transistors that can isolate the cell from vdd and gnd when not in use. In comparison to a traditional 6T SRAM cell, our SRAM cell design on average demonstrates a 25.68% decrease in power usage for a write operation, a 34.99% decrease for a read operation, and a 93.56% decrease in leakage power when in standby. Additionally, a 8x8 (64 bit) memory array was built using this SRAM cell design, including a row decoder and necessary column circuitry to perform successful read and write operations in Cadence Virtuoso. This final array will be referred to as a 64BitLPSRAM chip (64 bit low power SRAM, named SAIRS in our Virtuoso project directory). Moreover, our SRAM design was limited to using 90 nm transistor technology, which is of comparable size to transistors currently used in industry for this application.

4. Body

4.a Introduction

The goal of this project is to research and explore different SRAM cell designs to implement a low-powered 8x8 SRAM array. Additionally, a row decoder and necessary column circuitry will be incorporated to successfully perform byte sized read/write operations. Schematics and layouts of each component in our SRAM design created in Cadence Virtuoso will be tested through simulation. The column circuitry consists of a pre-charge, a write circuitry, and a sense amplifier unit. Since the 8x8 SRAM array allows for 8 possible memory locations, each with 1 byte of memory, a 3-bit address provided by the row decoder as input to access particular memory cells is required. The designs of all components are described in further detail in Section 4.d.

Our approach to developing a low powered 8x8 SRAM array consisted of simulating the schematic of many proposed low-powered SRAM cell designs. Under simulation, power and energy measurements were taken. Due to the extensive nature of SRAM research and design, much prior work was considered when picking SRAM cell designs to build and test. Beginning with the simplest but most widely used standard 6T (single-ported) and 8T (dual-ported) SRAM cell designs as a benchmark for power and energy comparisons, other cells from prior publications were explored [1]. One publication proposes a stacking technique of two transistors beneath the existing pull-down transistors to reduce leakage power [2]. Another publication proposes a power gating technique using a multi-threshold CMOS design to reduce leakage power [3]. A combination of these features were also explored.

After thoughtful consideration of the power, energy, and area needed to implement a proposed SRAM cell into a full SRAM design, one SRAM cell was selected and integrated into an 8x8 SRAM array. The cell chosen is based on a traditional 6T SRAM cell design, with two additional transistors to implement power gating in a Multi-Threshold CMOS design application [3].

*see Appendix for Standard 6T and 8T SRAM cell design and description

4.b Broader Considerations

SRAM is one of the most important research topics under the umbrella of VLSI design. SRAM is ubiquitous across any ASIC or general purpose computer chip, with caches largely made from SRAM cells. Along with cache arrays, SRAM cells can be found in FPGAs and are

increasingly important for many areas under embedded systems, cyber physical systems, autonomous vehicles, IoT, and AI computing [4]. Due to increasing leakage power, power/energy usage no longer decreases with respect to decreasing transistor size under Dennards's Law. Considering this power bottleneck trend in modern ASIC circuits, low power SRAM is an urgent area of research [1]. Large SRAM arrays make up a large portion of the transistor count in modern day chips; thus, the power usage of a single SRAM cell significantly scales when magnified by the millions of cells that exist in an array. For context, a standard 256 KB cache consists of 2,048,000 individual SRAM cells - the majority of which are simply just holding their bit value while a small portion of the cells are being read from or written into. On average, each cell spends a significant amount of time with its worldline disconnected (i.e. in standby mode), and thus leakage power during such times is very significant for such large SRAM arrays. Consequently, reducing leakage power and dynamic power is critical for the design of powerful and efficient SRAM arrays and can have a significant impact in reducing power consumption for the entire chip as a whole.

Our 64-bit Low Power SRAM design has many applications in virtually any SRAM application when scaled up to a larger array. Given the significant reduction in power and energy consumption compared to the standard 6T SRAM design (demonstrated in section 4.f), our SRAM cell design is a great candidate for power-sensitive applications. One particularly interesting application is AI modeling. The topic of AI can be applied to virtually any industry, from medical image processing to autonomous vehicle systems. Circuits responsible for AI computing require an extensive amount of memory to function properly, and thus allow the opportunity for our low power SRAM design to enter the discussion for use as memory[4]. Furthermore, our SRAM design motivates the need for greater, more efficient use of power and lower supply voltage in not just memory chips, but VLSI circuits in general.

The future of low-powered SRAM does not end at the system level architecture, but material selection and area used in production can have significant effects on this topic. FINFET transistor technology, and Carbon Nanotube transistor technology are a few examples of alternatives to MOSFET transistors that not only have the potential of reducing power consumption but area used in these cell designs [4]. Such novel transistor technologies can be used in tandem with our SRAM cell design in order to further decrease supply voltage, power usage, and area.

4.c Design Constraints and Requirements

The primary design constraint for the SRAM cell was to obtain a reduction in power and energy consumption in comparison to measurements seen with a traditional 6T SRAM cell. Furthermore, a supply voltage of 1.1 V (VDD) and minimization of total layout area were

additional design constraints. In some cases, reducing power usage comes at the tradeoff of using more transistors and thus increasing area, at which point, power reduction becomes the priority.

In regard to layouts, our design is restricted to the use of metal layers 1 (M1), 2 (M2), and 3 (M3). M1 can have any orientation while M2 can only be placed horizontally and M3 vertically. Moreover, our layout design is further constricted by DRC rules in order to ensure manufacturability, and checked against schematic using LVS [1].

4.d Design Description

Our SRAM design will be described beginning with the top level design and going down to lower level designs.

The top level design of our 64BitLPSRAM takes as input a 3-bit address to access one of eight memory locations within the 8x8 SRAM array ($a[0:2]$), a clock (clk), a 8-bit data input value ($data_in[0:7]$), a read enable signal (read_en), a write enable signal (wr_en), vdd and gnd as inputoutputs, and outputs a 8-bit data output value ($data_out[0:7]$). Because our SRAM cell is single ported, a read and write can not be done synchronously on the same clock cycle. Instead a read or write operation must be done on the rising edge of the clk as demonstrated in Section 4.f. 64BitLPSRAM is depicted below.

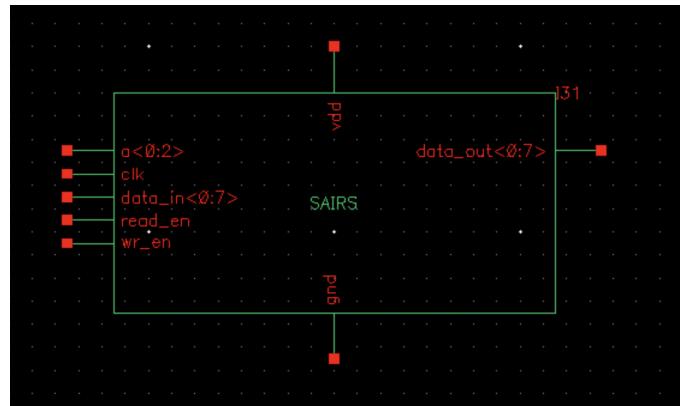


Fig. 1 Top Level Block Diagram

64BitLPSRAM is composed of lower level components: 8x8 SRAM array, 8x1 pre-charge unit, 8x1 write circuitry, 8x1 sense amplifier, and a 3:8 row decoder (Lyon Schediwy Decoder → LS38Decoder as named in Virtuoso).

The 8x8 SRAM array consists of a single SRAM cell placed in a manner where there are 8 rows and columns. The SRAM cell utilized is a single ported multi-threshold SRAM, which

uses a power gating technique to reduce static current (i.e. leakage power) by cutting off power to the cell. Power gating was implemented through use of multi-threshold transistors. More specifically, High-V_t transistors (i.e. sleep/power-gating transistors) were used to cut power to the SRAM cell which uses Low-V_t transistors. The diagram below illustrates this concept [3].

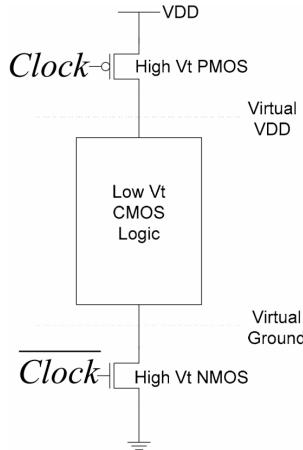


Fig. 2 Power Gating Using Multi-Threshold CMOS (MTCMOS) Design

As illustrated in Fig. 2 when Clock = 1 (the corresponding signal is named “sleep” in our design), the Low V_t CMOS Logic cell will be disconnected from its supply source, thereby creating a virtual power rail lower than the electrostatic potential of VDD and a virtual ground rail slightly higher than GND. In the opposite condition, the Low V_t CMOS Logic cell will have a connection to its supply sources VDD and GND [3].

Low-V_t transistors (i.e. small transistors) have a smaller distance between its source and drain, thereby decreasing the gate voltage required to form a conduction channel. Consequently, Low-V_t transistors switch faster but have higher static power leakage. On the other hand, High-V_t transistors reduce static power leakage but switch slower. Design of a SRAM cell using MTCMOS allows for optimization of delay and power [3].

Low-V_t transistors were used solely to design the SRAM cell and High-V_t transistors were used to create virtual power rails to the Low-V_t SRAM cell. As shown in Fig. 3, when sleep = 1, sleep mode is on; virtual power rails are created. Though the virtual power rail will always have a lower potential than VDD and the virtual ground rail higher than GND, virtual VDD and GND will be just enough for data reinforcement in the Low-V_t SRAM cell - meaning that the SRAM cell should keep its internal value even when disconnected from power sources. However, sleep is required to fall down to 0 to restore contact to VDD and GND for the purpose of performing successful read and write operations [3].

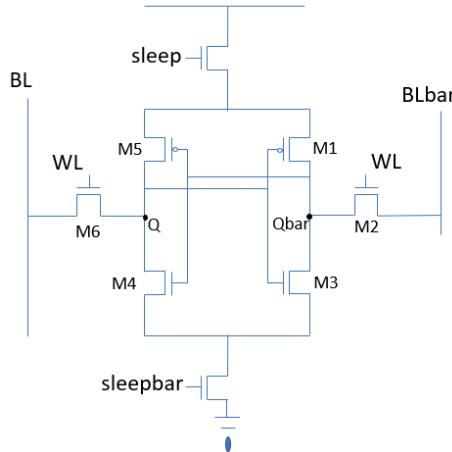


Fig. 3 6T Single-Ported MTCMOS SRAM Cell (6TSPMTCMOS)

Our SRAM design depicted in Fig. 3 was treated as a resistor circuit with the following parameters: 0.4V switch voltage, 1.1V vdd, $W_n = 380$ nm, and $W_p = 3/2W_n$. Also with the following resistance variable, $M_3 = M_1$, $M_1 = M_5$, $M_2 = M_6$ with M_4 and M_3 equal to R . More general information about how SRAM cells circuit functions during a read or write can be found in the Appendix. Fig. 4 - 6 depict sizing calculations done:

Read if $Q_{\text{bar}} = 0$

$R_{BL} = 1V$

$Q_{\text{bar}} < 0.4V$

$$\frac{1.1}{R_{M_2} + R_{M_3}} R_{M_3} < 0.40V$$

$$1.1R_{M_3} < 0.40V(R_{M_2} + R_{M_3})$$

$$R_{M_2} > \frac{7}{4}R_{M_3} = \frac{7}{4}R$$

Fig. 4 Read Stability Criteria

WRITE if $Q_{\text{bar}} = 1 \rightarrow 0$

$W_{BL\text{bar}} = 0V$

$Q_{\text{bar}} < 0.4V$

$$\frac{1.1}{R_{M_1} + R_{M_2}} R_{M_1} < 0.40V$$

$$1.1R_{M_1} < 0.40V(R_{M_1} + R_{M_2})$$

$$R_{M_1} > \frac{7}{4}R_{M_2} = \frac{49}{16}R$$

Fig. 5 Write Stability Criteria

$$M1: \frac{49R}{16} = \frac{2R}{k} \rightarrow k = \frac{32}{49} \rightarrow W_{M1} = \frac{32}{49} W_n = 248.1 \text{ nm}$$

$$M3: \frac{R}{1} = \frac{R}{k} \rightarrow k = 1 \rightarrow W_{M3} = W_n = 380 \text{ nm}$$

$$M2: \frac{7R}{4} = \frac{R}{k} \rightarrow k = 4/7 \rightarrow W_{M2} = \frac{4}{7} W_n = 217.1 \text{ nm}$$

$$sleepbar: 380 + 50 \text{ nm} = 430 \text{ nm}$$

$$sleep: W = \frac{3}{2} * 430 \text{ nm} = 645 \text{ nm}$$

Fig. 6 Transistor Sizing of 6TSPMTCMOS

Each individual SRAM cell takes wordline (WL), sleep, and sleepbar as inputs, vdd, gnd, bitline (BL) and bitline bar (BLbar) as input/outputs, and Q and Qbar (Q') as outputs. To have proper functionality, sleepbar = WL and sleep = not(WL). Q and Qbar represent the data value currently held within an SRAM cell. Below, Fig. 7 and 8 depict the 8x8 SRAM cell array module and the 8x8 SRAM cell array.



Fig. 7 8x8 SRAM Cell Array Module

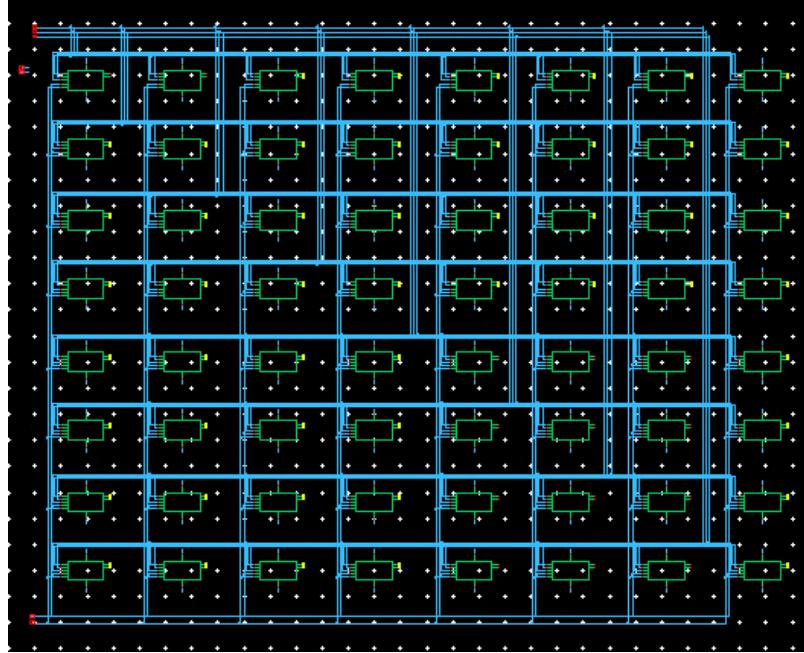


Fig. 8 8x8 SRAM Cell Array

During a write operation, BL is set to data_in and BLbar is set to not(data_in). After WL is set to 1, Q and Qbar will reflect the data signals at BL and BLbar respectively. The write operation is accomplished using the write circuitry depicted below in Fig 9.

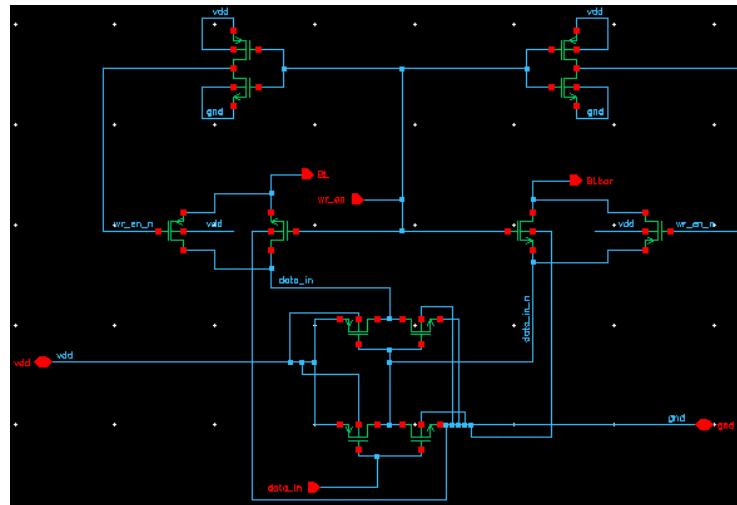


Fig. 9 Write Circuitry

The write circuitry displayed in Fig. 9 utilizes 2 back to back inverters to drive BL to the value held at data_in and BLbar to not(data_in). Notice the signal values driven by the back to back inverters meet a transmission gate before reaching the bitlines. Because a PMOS transistor struggles to pass a good 0 and a NMOS transistor struggles to pass a good 1, a transmission gate

is used as a means for optimal transmission of a good high (1) or low (0) signal from the input to the bitlines. Furthermore, the write circuitry was minimally sized to achieve optimal write conditions.



Fig. 10 8x1 Write Circuitry Module

During a read operation, BL and BLbar are first precharged to VDD when $\text{clk} = 0$. This condition is achieved through use of the precharge circuit which is shown in Fig. 11.

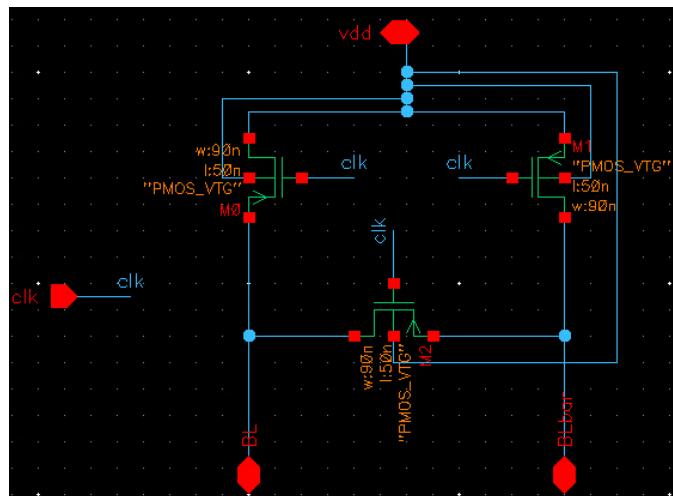


Fig. 11 Precharge Circuit

Observe in Fig. 11 an additional PMOS transistor connected to the drains of two PMOS transistors above. This additional transistor serves to equalize the bitlines to minimize the voltage difference seen on the bitlines when using a sense amplifier.



Fig. 12 8x1 Precharge Unit Module

After precharging BL and BLbar, WL is set to 1. Any difference in voltage seen at Q or Q' will cause the corresponding bitline to reduce in voltage as a direct result of a strong connection to ground from within the cell. The voltage swing (i.e. small voltage difference) seen between BL and BLbar will be interpreted by a sense amplifier depicted in Fig. 13, and outputted.

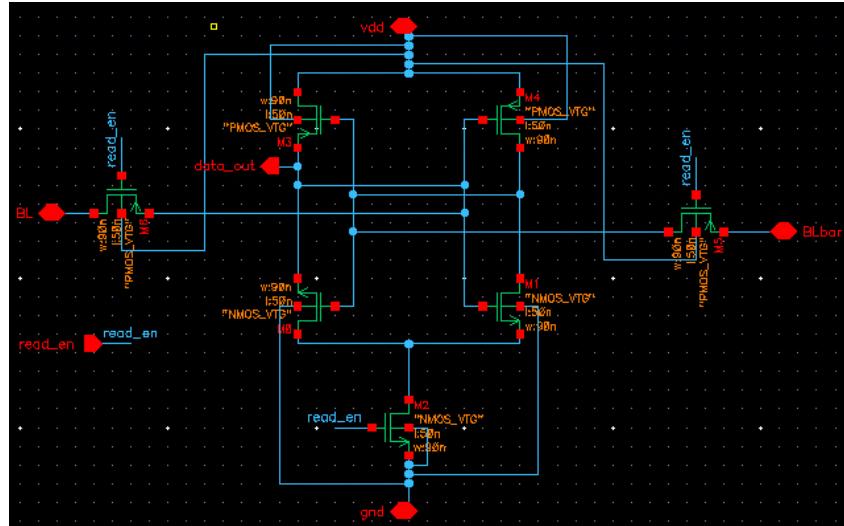


Fig. 13 Sense Amplifier Circuit

The sense amplifier shown in Fig. 13 uses a MTCMOS design to reduce power consumption through creation of a virtual ground rail when the cell is in standby (i.e. `read_en = 0`). Power is conserved in this stage, as even though the sense amplifier is connected to the bitlines, the feedback mechanism is inactive. When `read_en = 0 → 1`, the sense amplifier establishes a connection back to GND and amplifies the voltage difference seen between the bitlines to interpret the data value held in Q. During this operation, the connection to the bitlines is cut off by the `read_en` signal, so that the internal feedback mechanism can operate without outside noise.



Fig. 14 8x1 Sense Amplifier Module

The LS38Decoder interprets a 3-bit address input to access one of the 8 memory locations within the 8x8 SRAM cell array by setting one of 8 WLs to 1.

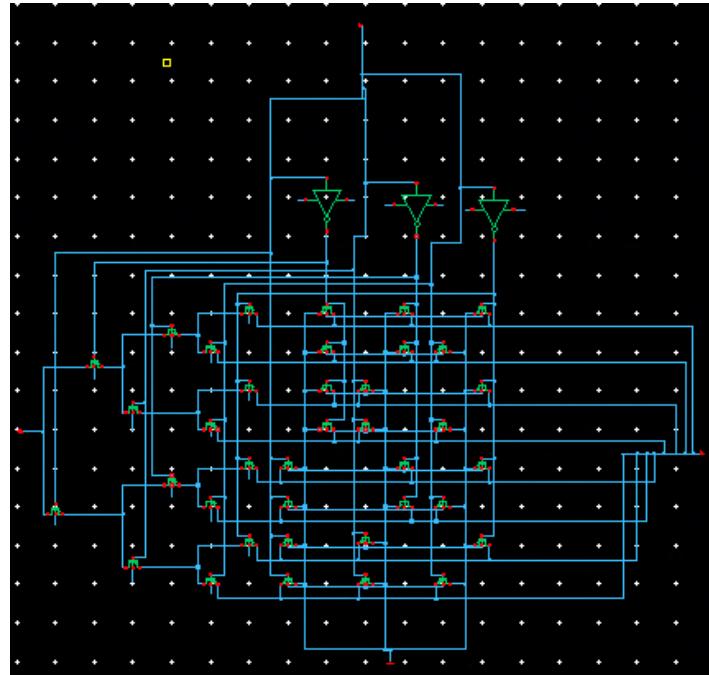


Fig. 15 Lyon Schediwy Decoder Circuit

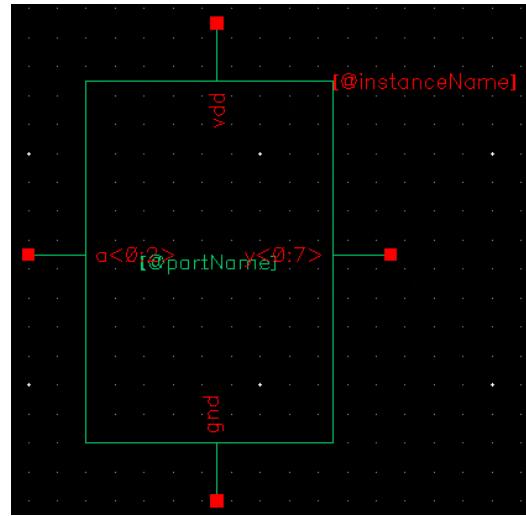


Fig. 16 Lyon Schediwy Decoder Module

The full system level architecture of our SRAM design is depicted in Fig. 17, as well as a single column circuitry in Fig. 18.

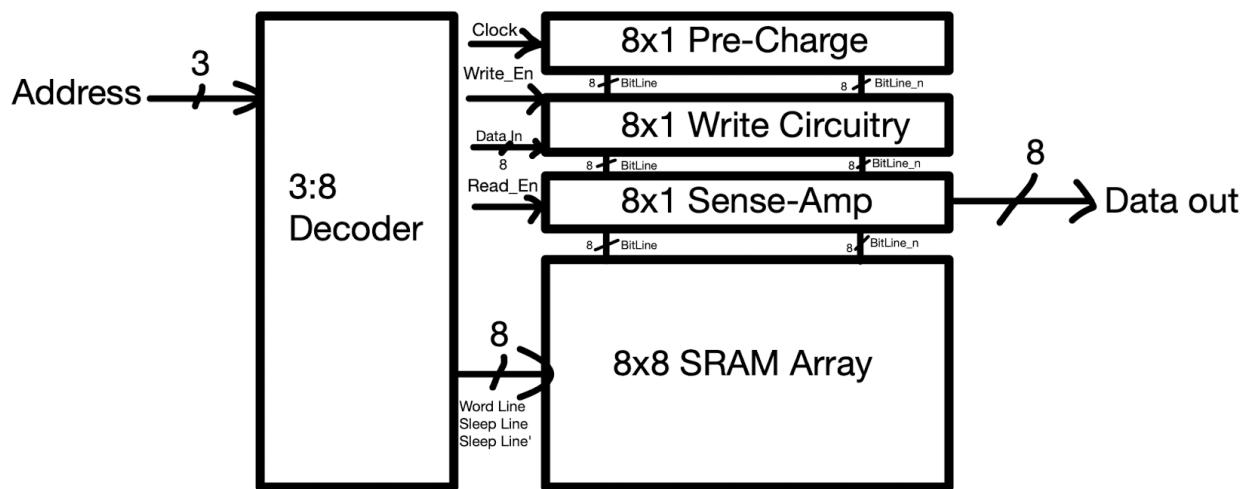


Fig. 17 Full System Level Architecture of SRAM Design

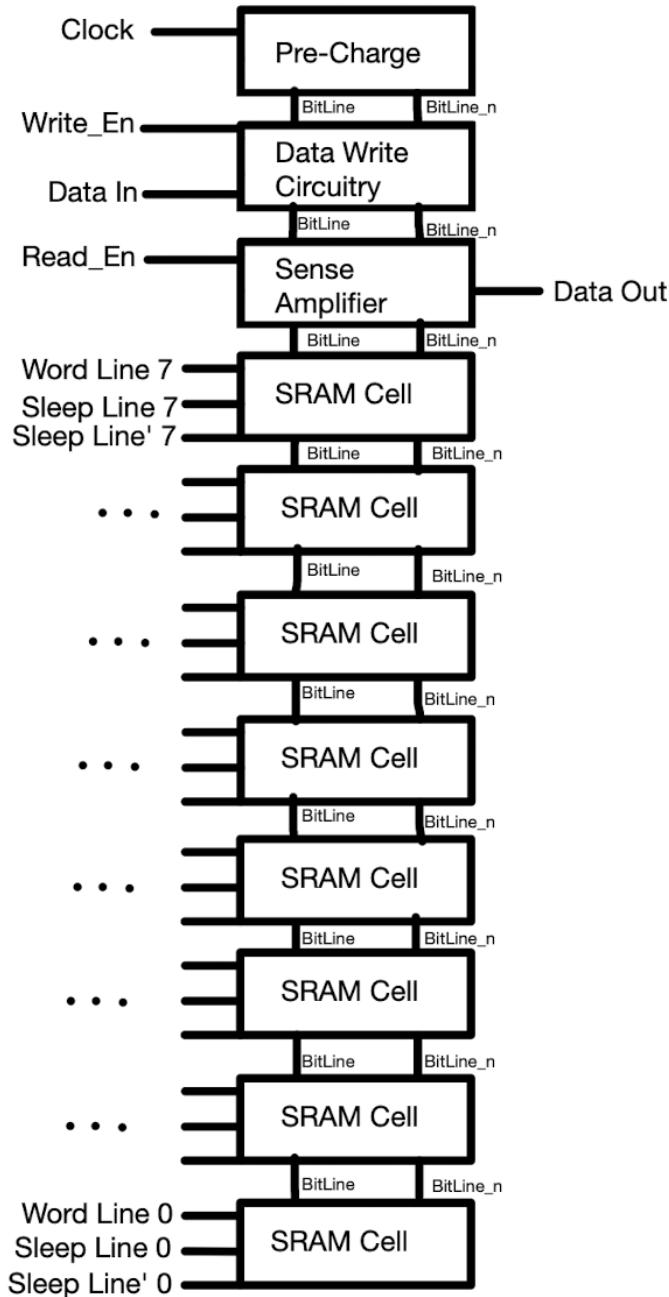


Fig. 18 Single Column of SRAM design

4.e Design Optimizations

Optimizations of the SRAM design primarily focused on reducing power and energy consumption. As discussed prior, the SRAM cell design with minimal energy and power usage in comparison to a traditional 6T SRAM cell was chosen for construction of our memory chip. Overall, there are three main optimization features for power and energy reduction.

The first feature is within the SRAM cell itself. Our SRAM cell design is based on the traditional 6T SRAM cell - see the Appendix and section 4.d. for further information. The optimization involves adding two additional power-gating transistors: one between ground and the pull down transistors within the feedback inverters, and one between VDD and the pull up transistors within the feedback inverters. These additional transistors disconnect the cell from VDD and ground when the Sleep signal is 1, and the Sleep_n signal is 0 [3]. The benefit here is that when the cell is not being read from or written into, the disconnected cell will no longer draw power from the supply source; thus, reducing leakage power and energy consumption. Reducing leakage power is critical since a SRAM cell is idle most of the time - in a SRAM bank with millions of cells, only a few are being read from or written to at a time. Power gating serves to solve this issue with incredible effectiveness - measurements will be shown in section 4.f.

The second optimization is with the Sense Amplifier design. The traditional Sense Amplifier includes a pair of feedback inverters that are always active; thus, using power even when the SRAM bank is not being read from. By adding a power-gating transistor controlled by a read_en signal between ground and the pull-down network, the feedback inverters are only active when read_en is 1, and thus only when a read is occurring [1].

Finally, the decoder design was greatly optimized using a Lyon-Schediwy decoder design. A traditional 3:8 decoder involves 3 inverters and 8 3-input AND gates for a total of 70 transistors - 2 transistors per inverter and 8 transistors per AND gate. With the Lyon-Schediwy design, the usage of traditional logic gates are forgone for a more efficient transistor level design that takes advantage of shared paths to VDD of which cause the need for less transistors. The Lyon-Schediwy 3:8 decoder only requires 44 transistors, which is an immense optimization on all counts of area, power, and speed [5]. Area is decreased by using 26 less transistors, power consumption is reduced from needing to power less transistors, and speed is increased by reducing the total path length from input to output.

In general, minimizing area of the final SRAM chip does not follow any formal algorithm, but involves time and effort spent placing components as close as possible without violating DRC. A layout technique called multi-f fingering was used wherever possible in each component of our SRAM design for the purpose of achieving optimizations in speed and area. Multi-f fingering involves using a single transistor with multiple fingers (i.e. polysilicon gates) to allow for source and drain diffusion areas to be shared, which reduces area when compared to simply using multiple single-fingered transistors to accomplish the same circuit [1].

4.f Testing/Simulation

There were two phases in testing - the first phase involved testing multiple single-celled SRAM designs in order to take energy and power measurements, and eventually choose the final SRAM cell design. The chosen SRAM cell design is explained in detail in section 4.d, while the technicalities of the other designs will be explained in the Appendix. In total, there were 6 SRAM cells that were tested. These 6 cells are shown below and involve some combination of 3 design features: dual ports vs single ports, stacking (i.e. leakage reduction transistors), and sleep/power gating transistors.

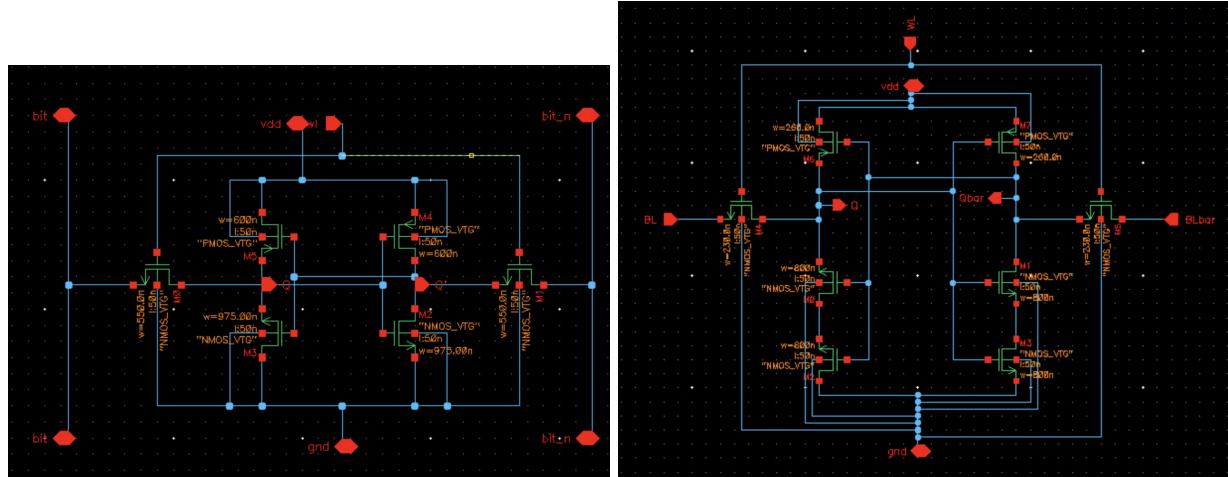


Fig. 19 Left: Traditional 6T SRAM, Fig. 20 Right: 6T SRAM with Leakage Reducing Transistors

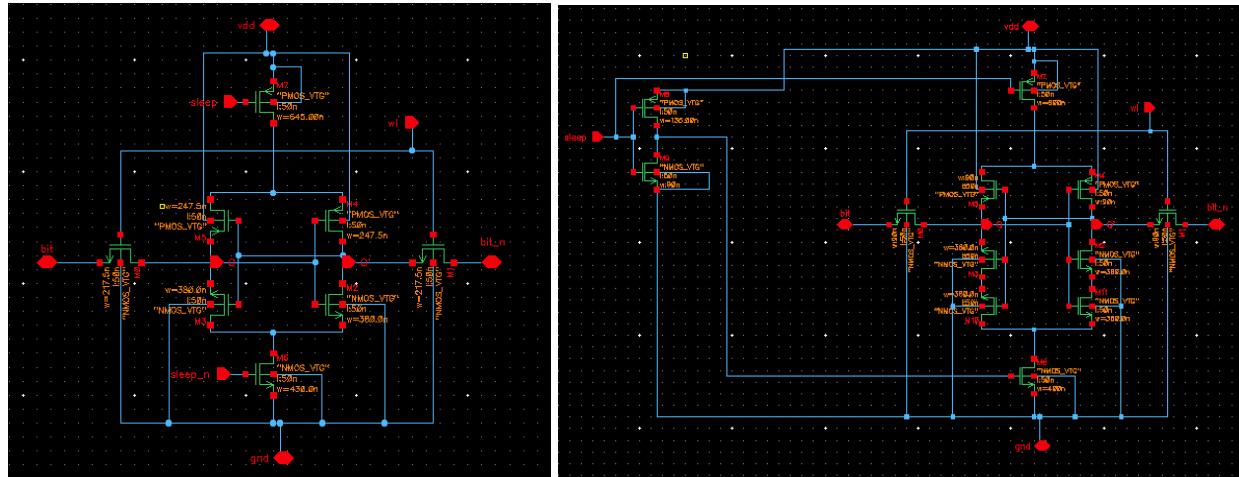


Fig. 21 Left: 6T SRAM with Sleep Transistors (Chosen Cell Design), Fig. 22 Right: 6T SRAM with Leakage and Sleep Transistors

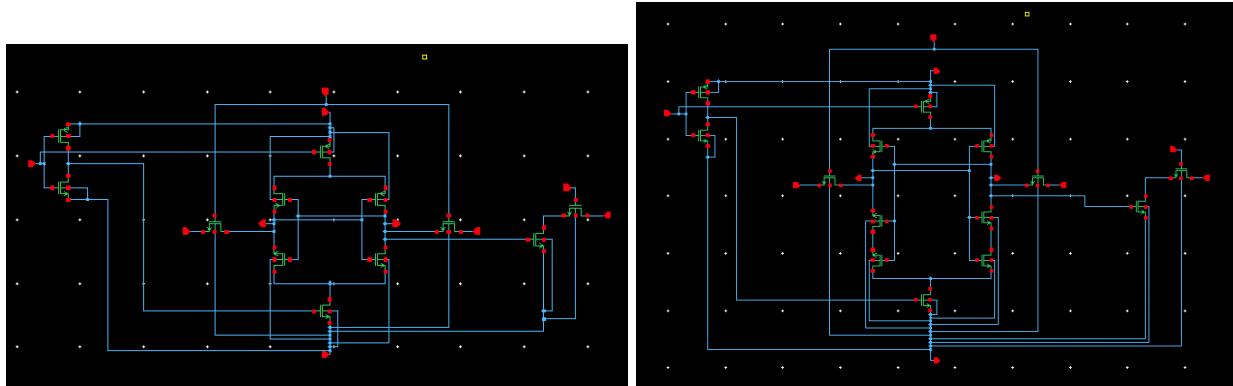


Fig. 23 Left: Dual Ported SRAM with Sleep Transistors, Fig. 24 Right: Dual Ported SRAM with Leakage and Sleep Transistors

A simulation schematic was designed to be able to both test the functionality of all 6 cells, and to allow energy and power measurements to be made. The Q and Q' outputs of each cell being tested are loaded with 4 inverters to mimic a real load when reading the SRAM, in order to take realistic energy and power measurements. The rest of the components are VBIT input sequences and muxes that select for different input sequences in order to validate the functionality of the SRAM cell. All simulations are running on SPICE parameters based on the tutorial for CE391 labs.

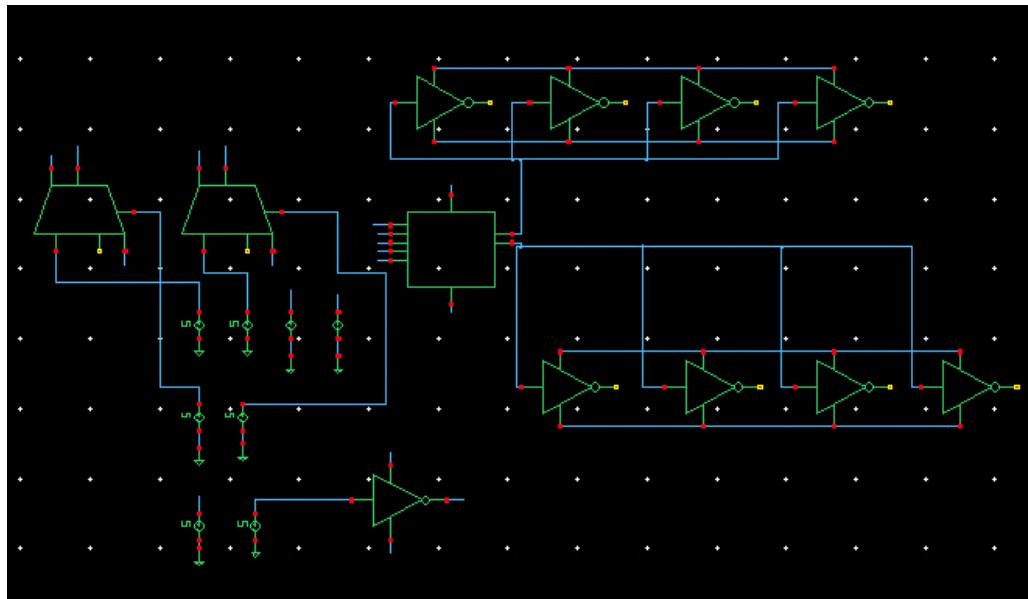


Fig. 25 Simulation Schematic

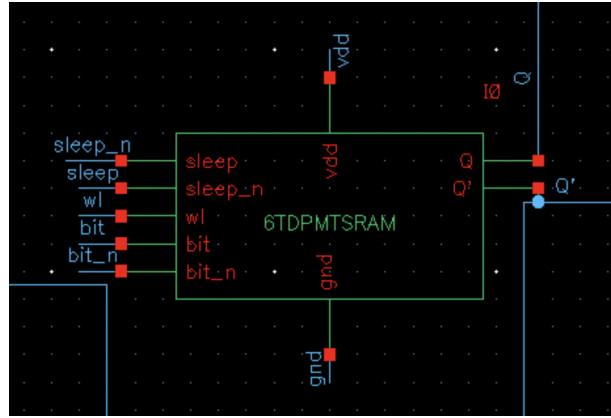


Fig. 26 Component Tested in Fig. 24

Formulas are entered into the Outputs window in the ADE L simulation environment in order to measure energy use and average power when writing a 0, writing a 1, reading a 0, and reading a 1. Example of a formula for energy is shown below. This formula takes the integral of the current measured through the VDD port of the SRAM multiplied by the voltage value of VDD [1]. This measured value is a simple way to calculate the total energy drawn from the power source, which is what we are ultimately trying to minimize. The integral is taken from 1.5ns to 2.5ns, which is the interval where a 1 is being written into the SRAM. The other energy formulas are the same, except taken at different time intervals.

$$(integ(IT("/I0/vdd") 1.5e-08 2.5e-08) * VAR("vdd"))$$

An example of a formula for average power is shown below. The formula simply divides the energy by the length of the time interval in order to obtain the value for average power in a specific time interval [1].

$$((integ(IT("/I0/vdd") 1.5e-08 2.5e-08) * VAR("vdd")) / 1e-08)$$

An example of the simulation waveform is shown below. This waveform corresponds to the traditional 6T SRAM design with two added transistors for power gating, which is the design that was chosen based on the small area, lower power/energy consumption, and its feasibility. This was decided after all the SRAM cells were sized minimally (using 90nm as the smallest PMOS size), and simulated for energy measurements. The Dual Ported SRAM design with the power gating transistors technically had lower power measurements - however, the column circuitry for a dual ported design is too complicated to be feasibly built effectively with our current resources, and the column circuitry may add additional power usage and layout area in the end as well.

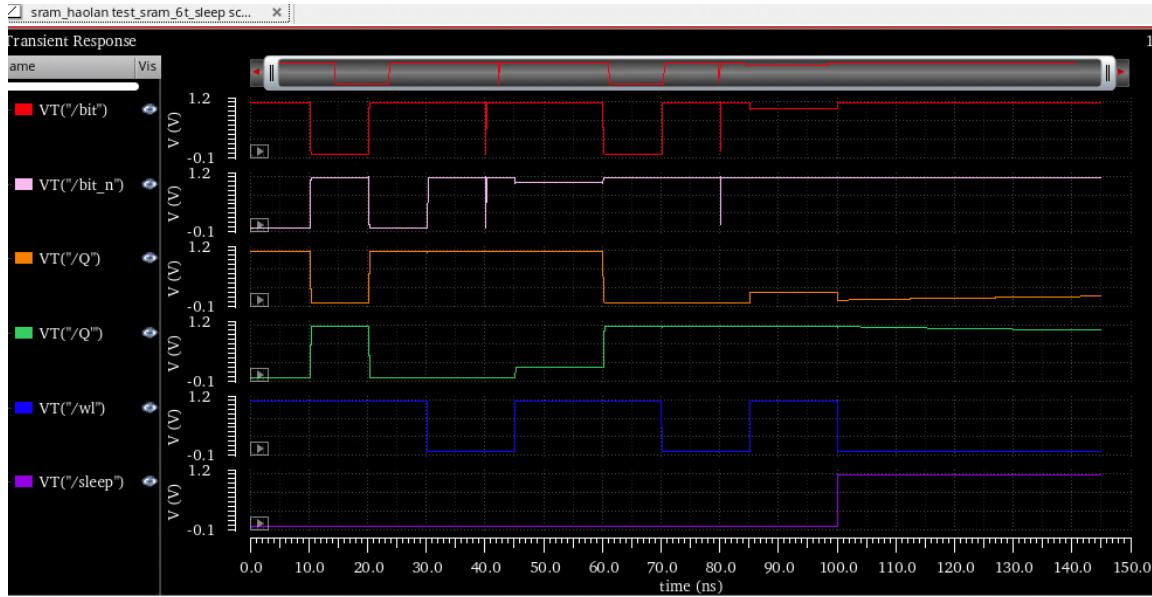


Fig. 27 Traditional 6T SRAM design with 2 Added Sleep Transistors: Simulation Waveform

Here are the verification details that demonstrate correct behavior of the SRAM cell:

0-10ns:

- Write function verified in order to write a 1
- Manually setting Bit = 1 and Bit_n = 0
- Observing that Q = 1 and Q_n = 0 in response

10-20ns:

- Write function verified in order to write a 0
- Manually setting Bit = 0, Bit_n = 1
- Observing Q = 0, Q_n = 1

20-30ns:

- Write 1 to SRAM (bit = Q = 1, bit_n = Q' = 0)

30-40ns:

- Turning off the world line before a read
- Manually setting bit = bit_n = 1 (precharging)

40-45ns:

- Leave bit, and bit_n floating high
- This is accomplished through switching the selector signal to the muxes that produce the input to bit and bit_n. The mux transitions from producing a 1 to producing a floating input for bit and bit_n. Floating the inputs is necessary for the read function to work.

45-60ns:

- Read function verified for SRAM holding 1
- Reconnecting the word line
- Observing bit_n dragged low by Q' (dips by over 100 mV, which is the necessary difference for the sense amplifier to produce a zero for Q', and a 1 for Q).

- Q' and Q do not flip

60-70ns:

- Write 0 to SRAM (bit = Q = 0, bit_n = Q' = 1)

70-80ns:

- Turning off the word line before a read
- Precharging bit = bit_n = 1

80-85ns:

- Leave bit, and bit_n floating high similar to before

85ns-105ns:

- Read function verified for SRAM holding 0
- Reconnecting the word line
- Bit dragged low by Q (again, dips by over 100mV)
- Q' and Q do not flip

105-145ns:

- Disconnecting the word line, and activating the sleep line
- This leaves the cell on Idle for leakage energy measurements

Energy and power measurements are shown below for the chosen SRAM cell, and the standard 6T SRAM cell as a baseline (both cells have matching transistor sizes for comparison). The unit for power is in joules/second, and the unit for energy is in joules. The following prefixes are used: u = 10^{-6} , n = 10^{-9} , f = 10^{-15} , and a = 10^{-18} . Energy and average power are measured for the relevant transition (writing/reading a 1/0), and for leakage, which is defined as a period where the SRAM is idle (the word line is off, sleep is on, and the cell is merely just retaining its value without being read from or written into). This measurement is arguably the most important, since a single SRAM cell may spend most of its time, on average, in idle mode. According to both tables, here are the improvements of using the design with sleep/power-gating transistors for average power:

- Average Power to write a 1: 26.08% decrease
- Average Power to write a 0: 25.28% decrease
- Average Power to read a 1: 35.63% decrease
- Average Power to read a 0: 34.34% decrease
- Average Leakage Power: 93.56% decrease

Outputs		
	Name/Signal/Expr	Value
1	pave_write_1	1.145u
2	pave_write_0	1.191u
3	pave_read_1	41.38n
4	pave_read_0	40.77n
5	pave_leakage	1.887n
6	energy_write_1	11.45f
7	energy_write_0	11.91f
8	energy_read_1	1.034f
9	energy_read_0	1.019f
10	energy_leakage	75.49a

Table 1 Chosen SRAM design with 2 Added Sleep Transistors: Energy and Average Power Measurements

Outputs		
	Name/Signal/Expr	Value
1	pave_write_1	1.549u
2	pave_write_0	1.594u
3	pave_read_1	64.28n
4	pave_read_0	62.09n
5	pave_leakage	29.31n
6	energy_write_1	15.49f
7	energy_write_0	15.94f
8	energy_read_1	1.607f
9	energy_read_0	1.552f
10	energy_leakage	1.172f

Table 2 Traditional 6T SRAM design: Energy and Average Power Measurements

Because the chosen design implements power gating, the simulation environment must test and verify that the SRAM will asymptotically retain its relative internal values (low and high) within the feedback inverter system when power is disconnected (sleep is active). It must also be verified that the cell will retain its original internal value once power is reconnected (sleep is deactivated). The following waveform verifies this. At $t = 300$ seconds, the sleep line is activated, and the value Q and Q' decreases from 1.1V and increases from 0V respectively. However, the internal value never flips during the 300 seconds that the cell is sleeping. At $t = 600$ seconds, the sleep line is deactivated - at which point, Q and Q' retain its original value of 1.1V and 0V respectively, demonstrating the viability of this SRAM design.

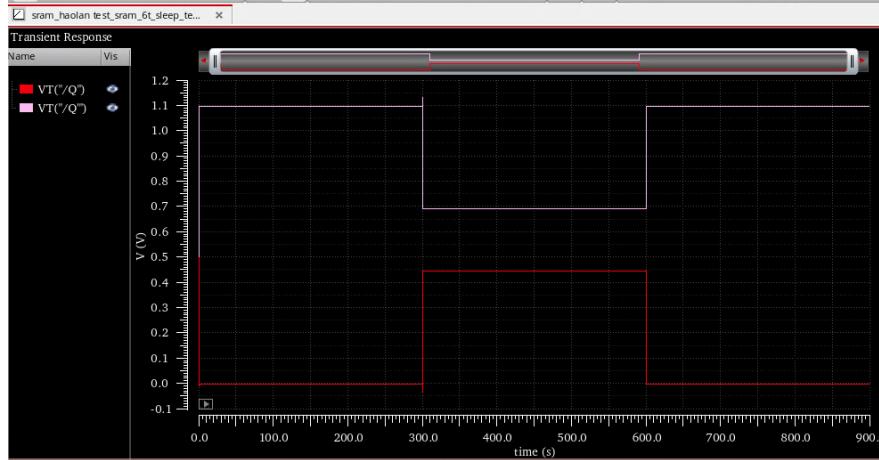


Fig. 28 6T SRAM with Sleep Transistors: Simulated Waveform Demonstrating Cell Stability During Sleep Mode

The second phase of testing verifies that the entire 64BitLPSRAM chip is functional. Again, a simulation environment is designed, with VBIT input sequences provided in order to measure the output signals from the memory array. For the following tests and waveforms, a write occurs when the write_en signal is high and the clock signal is high - at this point, the data in the data_in[0:8] input signals are written into the specified address. A read occurs when the read_en signal is high, and the clock signal is high - at this point, the y[0:7] output signals must be verified in the waveform in order to match whatever was previously written into at that specific address.

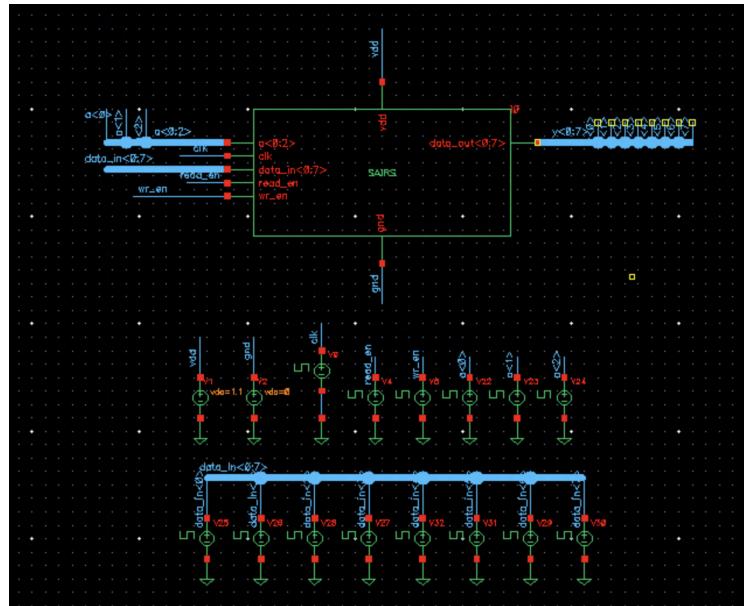


Fig. 29 Simulation Schematic for 64BitLPSRAM

The first test case involves using address 0 only (word line 0). The test involves writing the sequence 10101010 into address 0, and reading from the cell. Based on the $y[0:7]$ output signals in the below waveforms, the correct values are written and read from the SRAM array. The following two writes input the sequence 01010101 and 10101010 into address 0, and the read function demonstrates correct behavior after both writes.

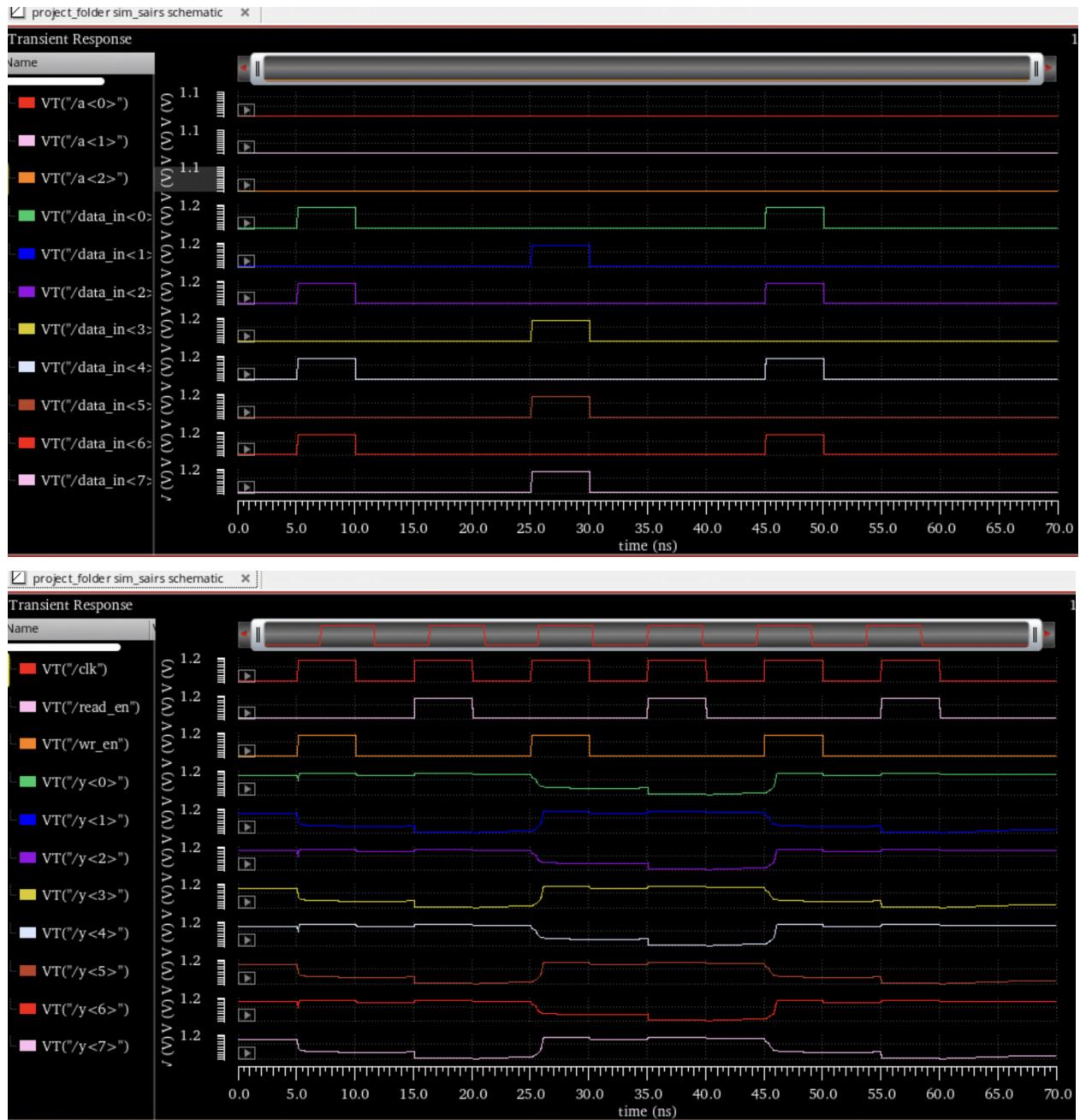


Fig. 30 and 31 64BitLPSRAM Simulation Waveforms for Test 1: 3 Consecutive Write-Read Sequences on Address 0

The second test sequence involves first writing 11011101 to address 2, and reading from address 2 to ensure correctness. Next, 00001111 is written into address 5, and a read is done on address 5. Finally, address 2 is read, and address 5 is read again to ensure the previous written values are still stored into the memory array despite switching addresses. Based on the $y[0:7]$ output waveform, all writes and reads were done successfully.

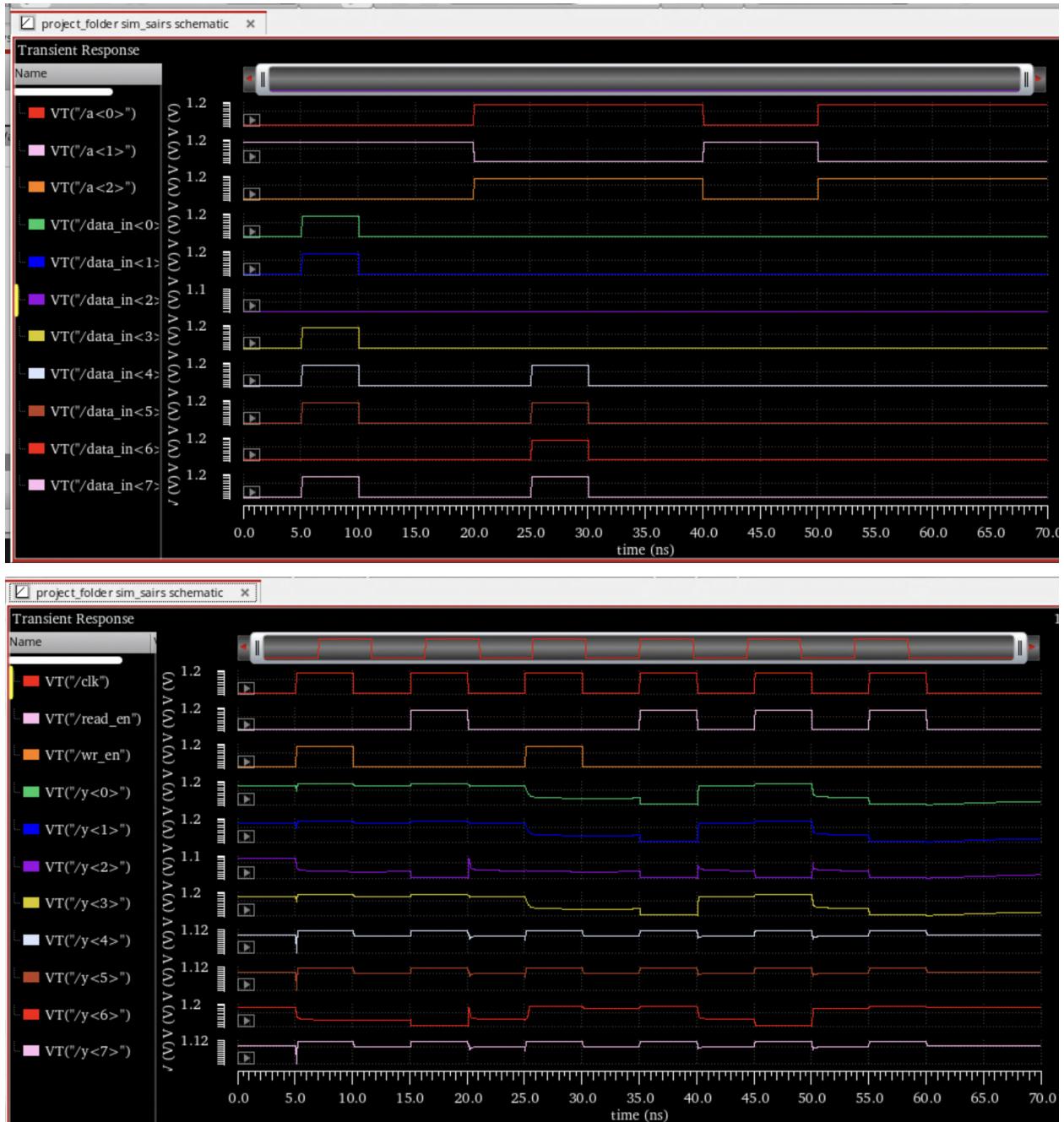
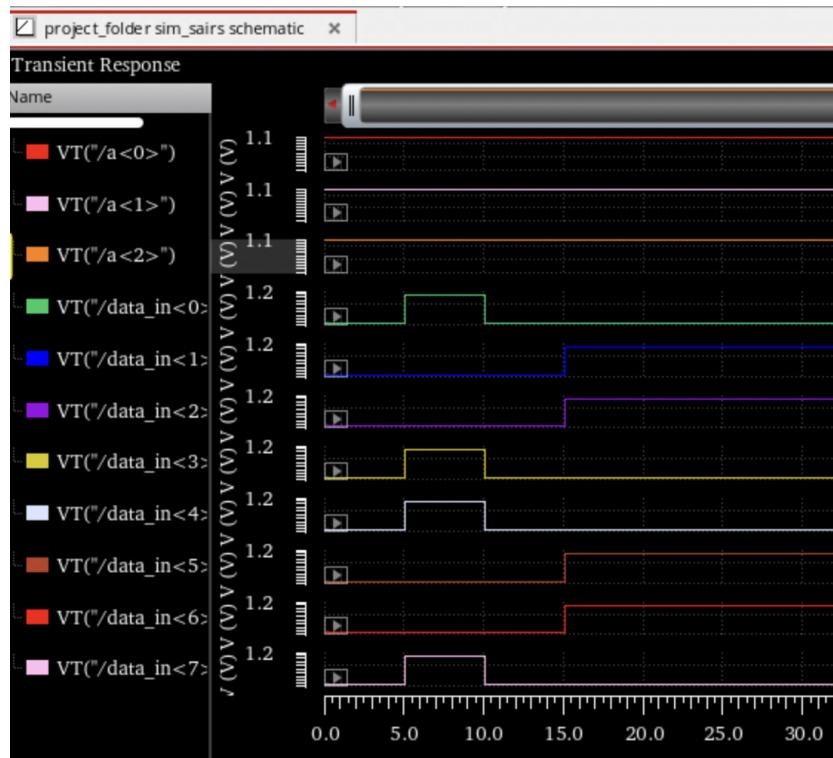


Fig. 32 and 33 64BitLPSRAM Simulation Waveform for Test 2: Write-Read on Address 2, Write-Read on Address 5, Read Address 2 Again, Read Address 5 Again

The third simulation scenario is using address 7 for the entire test. Two consecutive writes occur, writing 10011001 and 01100110 into row 7 of the cell. A read then occurs that should output 01100110 (the results of the second write), which is verified by $y[0:7]$ in the following waveforms. Of course there are much more rigorous tests to be done, including testing all word lines, and testing much more complex series of read and write operations over a larger range of time. The team ultimately ran out of time to complete such rigorous tests, yet through these 3 tests that were completed, there is great confidence in the functionality of the 64BitLPSRAM.



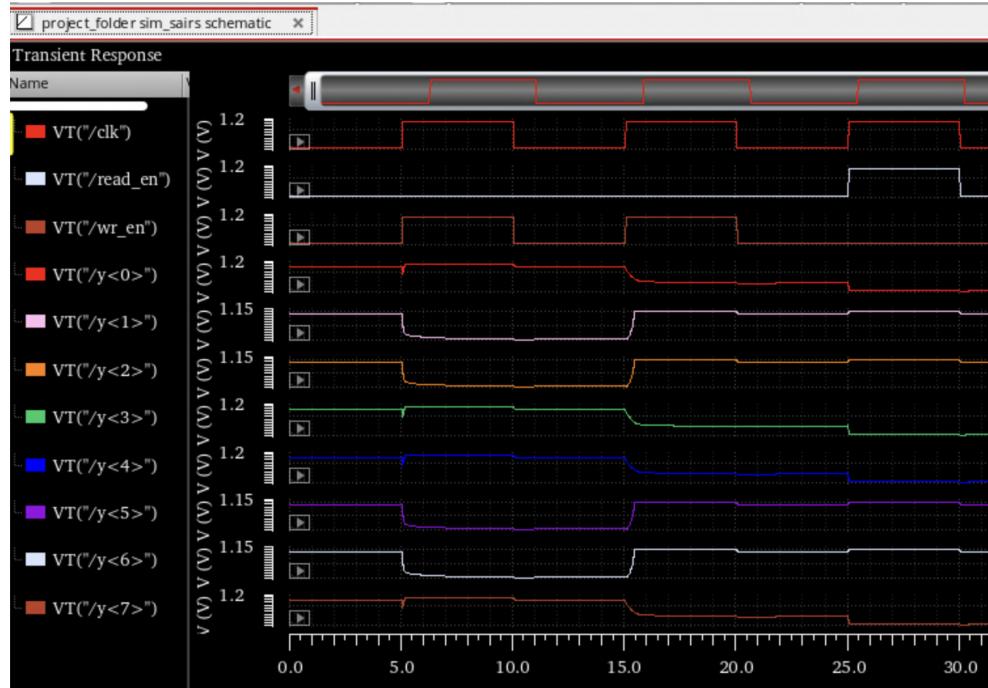


Fig. 34 and 35: Fig. 32 and 33 64BitLPSRAM Simulation Waveform for Test 3: Two Consecutive Writes on Address 7, then a Read

4.g Implementation/Synthesis

The 64BitLPSRAM design was implemented in schematic and layout on Cadence Virtuoso and was verified using DRC to ensure our design is manufacturable and using LVS to ensure our layout matches the schematic. All inputs and outputs to the layout are the same as the schematic design, and thus will not be elaborated here due to being covered in section 4.d of this report. The layout meets area minimization constraints and metal usage constraints. Figures of all significant layouts are depicted below.

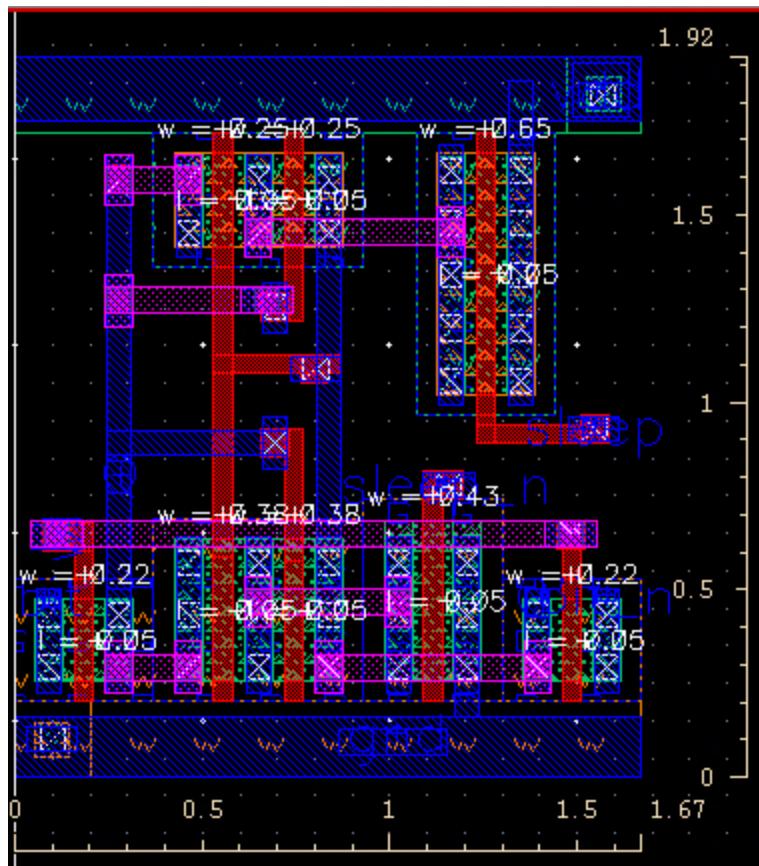


Fig. 36 SRAM Cell

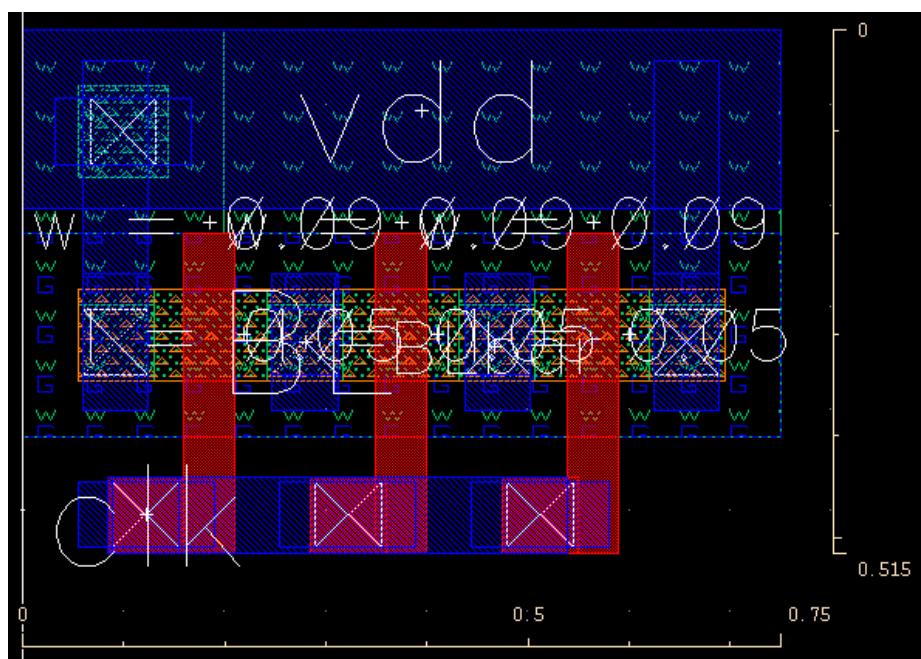


Fig. 37 Precharge

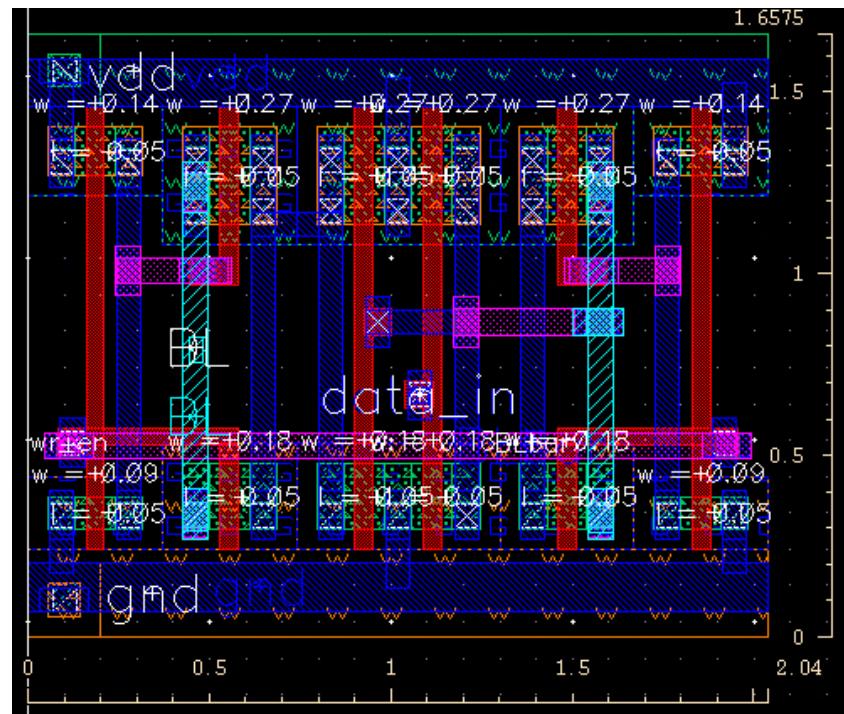


Fig. 38 Write Circuitry Layout

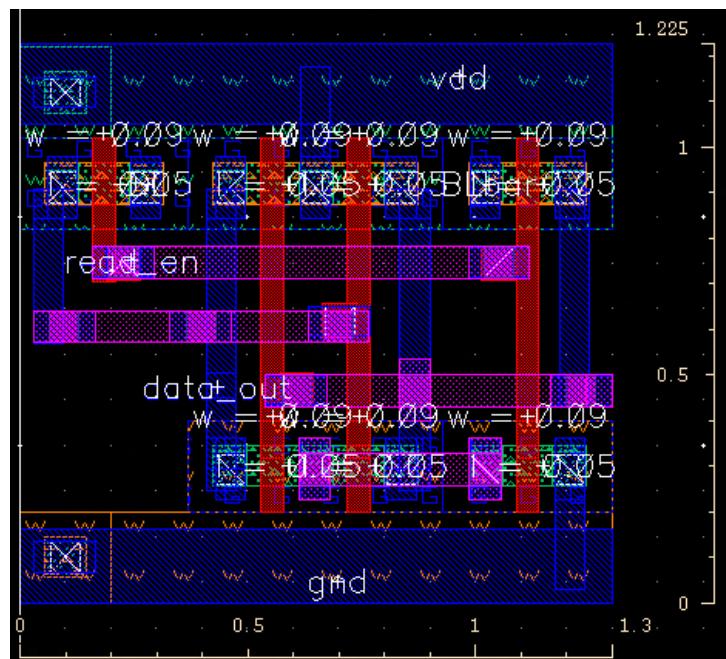


Fig. 39 Sense Amplifier Layout

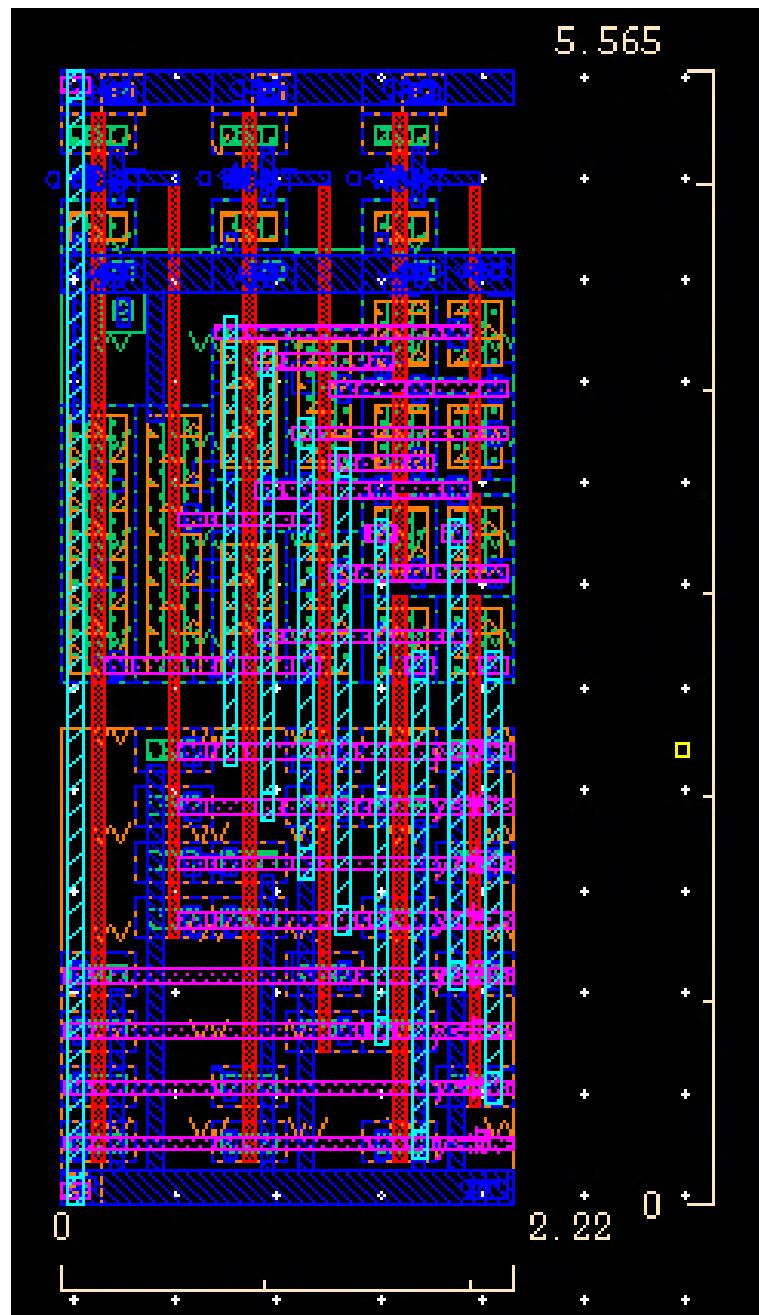


Fig. 40 Decoder Layout

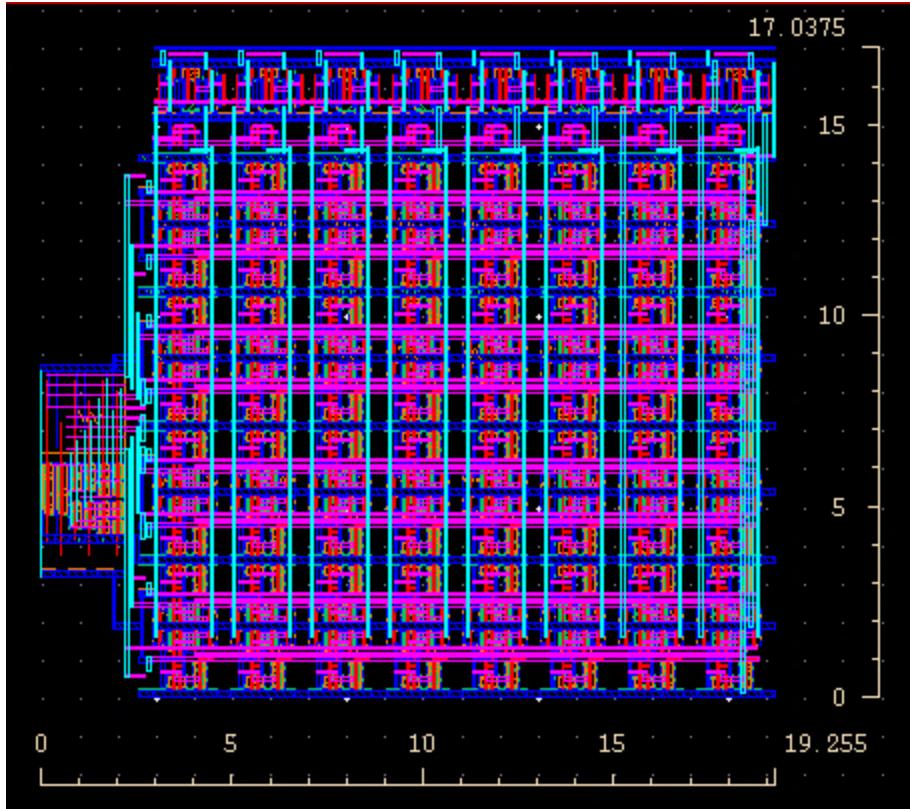


Fig. 41 Full 64BitLPSRAM Layout

4.h Conclusions

Overall, the design of the 64BitLPSRAM chip and the layout meets requirements of reducing power consumption very well, based on the measurements of a single cell of the chosen design. The final 64BitLPSRAM chip works exactly to the original design specifications, and can successfully complete both read and write operations in an 8 byte memory chip.

Possible improvements are with revising each layout component in order to see if area or metal usage can be further reduced anywhere. Specifically with the write circuitry, we had issues with the original design where it was difficult to pass a 1 onto the correct bitline when a write operation occurs (refer to the Appendix for more details on the write operation). This was due to NMOS transistors being weak at passing 1s, but strong at passing 0s. We opted for using transmission gates in order to pass both 0 and 1 successfully as a solution, but this may have over complicated the write circuitry and increased the area and power usage of the component. Further research into the write circuitry could provide simpler solutions. In fact, traditional write circuitry designs are only responsible for passing a 0 value onto the correct bitline, since the other bitline should be already holding a 1 value from being precharged. Yet this design caused issues for us that could not be resolved, therefore we opted for a more complicated but rigorous

write circuitry design. One of the inverters used within the write circuitry is redundant and minimization of area can be achieved by removing this component.

There may be other SRAM cell designs that could have reduced power even more, but may have not appeared in our research or we did not have time to build and test it. We could have tested multiple sizing of transistors for the same cell design in order to optimize power reduction, but ran out of time for this as well.

New features for this design would be mainly to scale the design up to a reasonable SRAM cache size, and to implement row decoding in order to keep the design dimensions similar to a square. Ultimately, the memory array that was built functions extremely well, and has significant power and energy improvements over the traditional 6T SRAM cell design based on the measurements given in section d.f.

5. References

- [1] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design: A circuits and systems perspective*, 4th ed. Addison-Wesley, Boston: Pearson Education, Inc., 2011.
- [2] R. K. I. Kumar, V. C. Kalal, H. P. Rajani, and S. Y. Kulkarni, "Design and Verification of Low Power 64bit SRAM System using 8T SRAM:Back-End Approach," *International Journal of Engineering and Innovative Technology*, vol. 1, no. 6, pp. 1–7, Jun. 2012.
- [3] A. Bhaskar, "Design and analysis of low power SRAM cells," 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), 2017, pp. 1-5, doi: 10.1109/IPACT.2017.8244888.
- [4] Y. Kim, S. Patel, H. Kim, N. Yadav, and K. K. Choi, "Ultra-Low Power and High-Throughput SRAM Design to Enhance AI Computing Ability in Autonomous Vehicles," *Electronics*, vol. 10, no. 3, p. 256, Jan. 2021, doi: 10.3390/electronics10030256.
- [5] D. Yagain, A. Parakh, A. Kedia and G. K. Gupta, "Design and Implementation of High Speed, Low Area Multiported Loadless 4T Memory Cell," *2011 Fourth International Conference on Emerging Trends in Engineering & Technology*, 2011, pp. 268-273, doi: 10.1109/ICETET.2011.23.

6. Appendix

6.a 6T SRAM Introduction

The 6T SRAM is the most widely used SRAM cell design across the industry, and it is the backbone architecture for our final SRAM cell design.

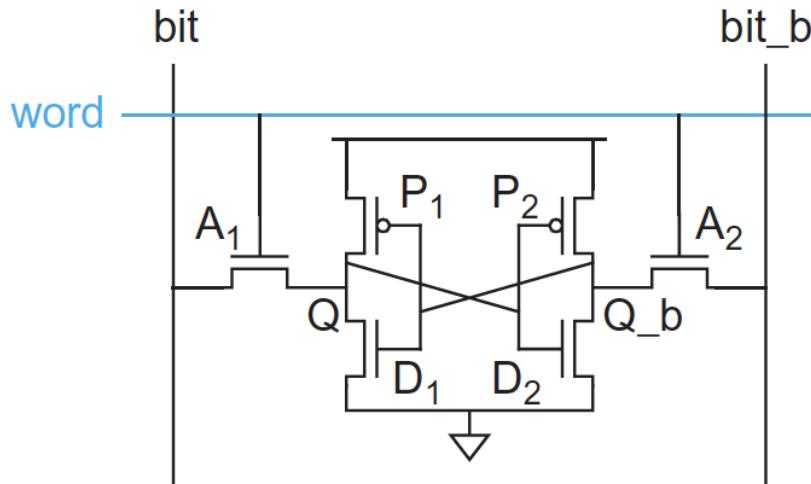


Fig. 42 6T SRAM Schematic

The following is a brief summary on how the 6T SRAM cell operates. The feedback inverters (P1 and D1 making up one inverter, and P2 and D2 making up the other) keep the internal value of the cell at Q, and the inverse value at Q_b. The access transistors A1 and A2 connect the cell to the bitlines once the worldline is set to 1; all SRAM cells in a column will share the same bitlines. Therefore, the cell being read from or written to will be the cell with the worldline set to 1 [1].

During a write operation, when the clock signal is 0, both bit and bit_b are set to a 1 by the pre-charge circuit. When the clock signal becomes 1 and the write_en signal is 1, the pre-charge circuit is disconnected, and the write driver drives one of the bitlines to 0. For example, if a 0 needs to be written into the cell, the write driver will drive bit low and keep bit_b floating high. This will cause Q to take on the 0 value. During a write, the A1 and A2 NMOS transistors must be sized larger (thus having less resistance) than the P1 and P2 PMOS transistors in order for the written value to overpower the existing value in the cell [1].

During a read operation, the bitlines are again set to 1 by the pre-charge circuit when the clock is 0. Once the clock is set to 1 and the read_en signal is also 1, the internal node that is 0

will drag down the corresponding bitline. For example, if Q_b is 0 (the cell is storing a 1), then the bit_b line will be dragged down. Because the cell is not powerful enough to drag the corresponding bitline all the way to 0 volts, the sense amplifier will detect which bitline was dragged lower, and amplify the difference until a value can be read. For example, if the bit_b was dragged down, the sense amplifier would output a 1 to represent the value held in node Q of the cell. During a read operation, it is important that the pre-charged bitline does not accidentally flip the value in the cell - for this purpose, the D1 and D2 NMOS transistors must be sized larger (thus having less resistance) than the A1 and A2 NMOS transistors to ensure read stability [1].

6.b 8T SRAM Introduction

The 8T SRAM is another design that is very popular in industry. Commonly known as the dual-ported design, this cell separates the read and write into separate modular components.

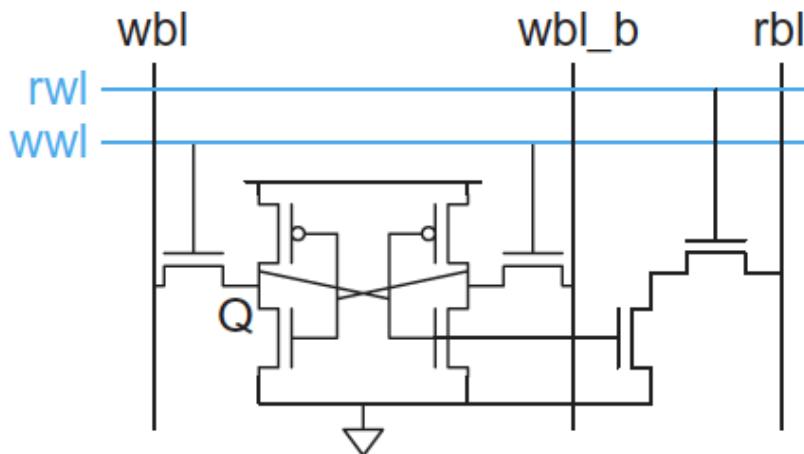


FIGURE 12.18 8T dual-port SRAM cell

Fig. 43 6T SRAM Schematic

The finer details of the 8T SRAM design will be omitted due to not playing a large role in our final cell design. Briefly, the benefits of the 8T SRAM design allows for greater read stability because the read module (the two transistors on the right) is separate, and less likely to accidentally flip the internal value of the cell during a read. This allows for this design to run at lower VDD values - which is not entirely relevant to our design due to our constraint value of 1.1V. With this design, the write operation functions exactly the same as the 6T SRAM design. However, during the read operation, there is a separate read word line that must be set to 1, and a third bitline is used in order to read the value of the cell. For example, if Q is holding a 1, then

rb1 will be dragged down. A sense amplifier would be able to sense the change in voltage, and thus output a 1 [1]. The reason that this design was dropped was because the team did not find a sense amplifier design that used a single bitline, instead of two bitlines.

6.c Leakage Reducing Transistors

Another design feature that was tested when comparing different single cell SRAM designs was the usage of leakage power reducing transistors (i.e. stacking). These transistors are shown in fig. 20 and involve just adding extra NMOS transistors in series with the existing pulldown NMOS transistors in the feedback inverter system [2]. In practice, this feature only offered slight reductions in power and energy consumption, with the added cost of two additional transistors. Therefore, the final SRAM cell design only uses power-gating transistors, and not a stacked transistor design.