Low Leakage memory

Kai Liknes 20.12.2015

Abstract

The purpose of this report is to consider several memory architectures for use in a standard CMOS technology with a technology node larger than or equal to 90nm, and to model and simulate their subthreshold leakage currents. Subthreshold leakage currents are explained, and methods to reduce them are presented. A novel way of analyzing leakage currents by hand was developed but not verified in simulation. Various standard CMOS memory cells were modelled and simulated in Cadence ADE L with the intent to measure their leakage currents, but the lack of precise models supplied for the simulation tool prevented any conclusive results. The inaccuracy of the simulation models also prevented the verification of the analyses presented, discrediting the work and report as a whole. Further work is needed.

NTNU

Norwegian University of Science and Technology

Table of Contents

List of Figures and Tables

1. Preface

I was tasked with writing a report on low leakage memory by Disruptive Technologies, a company that specializes in designing microchips for use in the Internet of Things.

A big part of my project was done in Cadence ADE L, implementing various memory cells and setting up testbenches and simulations. More than midway through my work, I discovered that the default technology library/transistor models provided by Cadence were not set up to model leakage currents. This caused a lot of frustration, as it made all my previous work pointless. I did not continue the work on setting up simulations, and the only cells that are presented in this report are three cells that I finished the simulation setups on. I wanted to do something useful with my project report, so I thought I would include a partial literature study in my report. This is the reason the report does not appear to have a purposeful structure. I apologize.

2. Acknowledgement

I give thanks to Bjørnar Hernes, Snorre Aunet and Trond Ytterdal for assisting me in my work. I also give thanks to Åsmund Oma and Erlend Hestnes for helping me discover new memory architectures to consider in my report.

3. Introduction

The Internet of Things (IoT) is a concept which is quickly gaining popularity and this causes the IC industry to gear towards designing microchips that are compatible with this concept. According to advocates of the Internet of Things concept [1], almost every physical object in use by people will eventually be connected to the internet. Microchips are designed to fit into even the most trivial applications such as clothes hangers. Sensor networks are created by spreading out a large amount of inexpensive sensors and having them communicate over the internet. In these cases, a change of batteries is impractical and therefore one must design to maximize the battery lifetime of the chip. In most applications in the Internet of Things, the chip is only active and computing/transmitting data a fraction of the time. This means the static power consumption (power leakage) will be the deciding factor in battery lifetime.

All IoT-chips will require some form of data storage. This report assumes a distributed shared memory (DSM), and that the memory is implemented as a single centralized memory cell array.

Memory accesses only happen when a chip is either computing or transmitting data, and because these actions are infrequent, memory accesses are also infrequent. Combined with the fact that the memory portion of a chip often makes up a large portion of the total chip area, this means that minimizing the power leakage of the memory is essential to reducing the static power consumption of the entire chip.

In the previous decade, reducing power consumption meant reducing the active power consumption. Active power is the power required to switch transistors on and off. As stated, the leakage power more of a concern in IoT-chips. Instead of designing for speed, area, or active power consumption, this report focuses on the static power consumption of memory circuits.

A broad range of different kinds of memory is discussed, and their viability in a simple CMOS technology is considered. In order to further compare viability, the remaining memory types are analyzed and an attempt at simulating their leakage power is made.

4. Theoretical background

4.1 Previous work

Previous work in minimizing power in memory circuits focus mostly on Active power consumption more than static power consumption. [2] tries to minimize read and write power consumption in SRAM while neglecting static power consumption.

[3] simulates several kinds of single-edge triggered D-flip-flops and does include information on their static power consumption. The D-flip-flop is the most widely used memory element in the System on Chip (SoC) IC design doctrine, relying on off-chip memory arrays to supplement memory requirements. Unlike [3], this report also includes other memory cell architectures.

[4] uses a technique called power gating to try to minimize the static power consumption of a ripple carry adder, a circuit which does not require any kind of memory.

4.2 Leakage / Static energy consumption

In CMOS technologies using a technology node of 90nm and larger, the most dominant source of static power is the subthreshold leakage power, P_{sub_leak}. For a single-V_{dd}-level circuit this is given as:

$$P_{sub\_leak} = V_{DD} * I_{sub\_leak} = \frac{{V_{DD}}^2}{R_{Vdd-gnd}} \quad [1]$$

Where I_{sub_leak} is the current going from V_{DD} to ground, through the drain-source subthreshold channel of the transistors. $R_{Vdd-gnd}$ is the resistance seen from V_{DD} to ground.

According to [5], the subthreshold leakage current through a single transistor can be approximated by the following function:

$$I_{DS,off}[\text{nA}] = 100 * \frac{W}{L} * 10^{-\frac{V_t}{S}} \quad [2]$$

Where W is the gate width, L is the gate length, V_t is the threshold voltage. S is the so-called **subthreshold swing**, given by:

$$S = \eta * 60\text{mV} * \frac{T}{100} \quad [3]$$

Where T is the temperature [K], and η is equal to:

$$\eta = 1 + \frac{C_{dep}}{C_{oxe}} \quad [4]$$

Where C_{dep} is the channel-depletion capacitance and C_{oxe} is the channel-oxide capacitance.

4.3 Power gating

Power gating means turning off the power for a part of the circuit, in order to gain almost zero active and static power consumption. Power gating relies on the circuit not requiring a continuous supply of power to function properly.

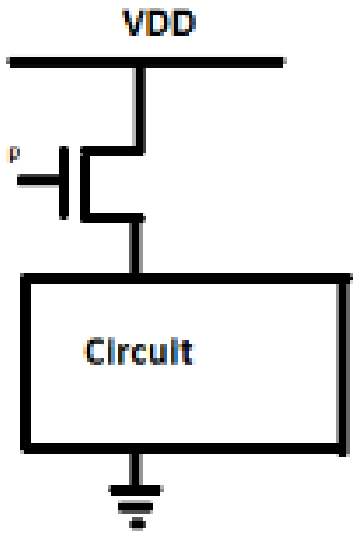

Figure 1: A simple form of power gating, the signal P turns the power on for the circuit.

4.4 Multi-level-VDD

As can be seen in equation 1, the leakage power is proportional to the square of V_{DD}. This means that by reducing VDD, one can significantly reduce the leakage power. A multi-level-VDD circuit reduces the supply voltage in parts of the circuit where one can afford to sacrifice speed and timing requirements for lower power. A concept called voltage islands is commonly used when implementing multi-level-VDD circuits. A voltage island refers to a single coherent part of the circuit employing its own supply voltage level.

4.5 Types of Memory

4.5.1 Non-volatile memory

Non-volatile memory is a kind of memory that retains data even when the supply voltage is switched off.

4.5.1.1 Flash memory

Flash memory retains its data by storing a charge on a secondary electrically disconnected gate on each memory transistor. This gate is called a floating gate. No significant current flows through to the floating gate,

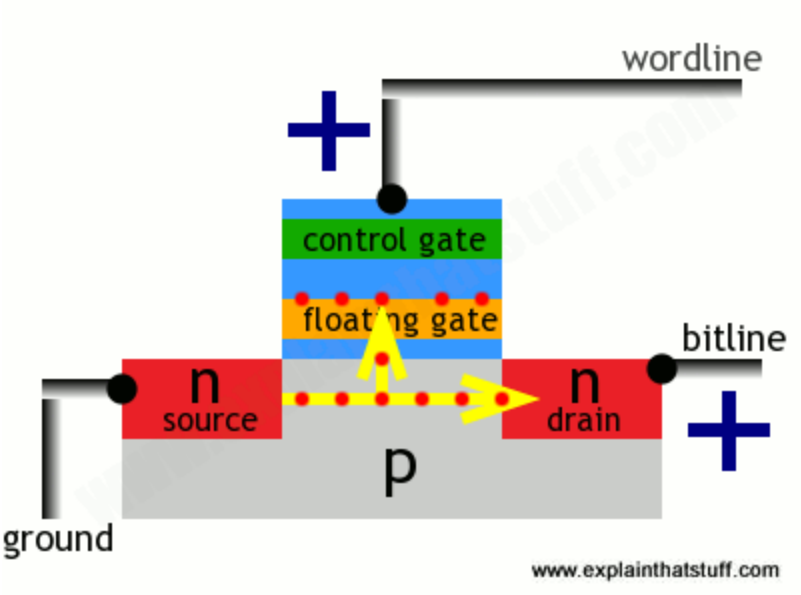

Figure 2: Illustration of a flash memory cell. [6]

The programming of a transistor to a binary '0' value happens by applying a strong voltage to both the wordline and bitline of the selected transistor. This strong positive charge attracts electrons from the source-drain current. These electrons are permanently stuck in the floating gate, creating a permanent negative charge on the gate, thus creating a permanent logic '0' value.

Resetting the transistor requires a strong negative charge to be applied to the word line, repelling the electrons stuck in the floating gate, creating a permanent logic '1' value.

The benefits of flash memory is that it is a relatively mature technology. Today (September 2015), flash memory is significantly cheaper than MRAM. [7]

4.5.1.2 MRAM (Magnetoresistive RAM)

MRAM is a type of memory which stores data by magnetizing one of two sandwiched layers, thereby changing the electrical characteristics of a tunnel region between the two layers due to an effect called the tunnel magnetoresistance effect. This change can be registered by sensing circuits and provides a permanent storage of data.

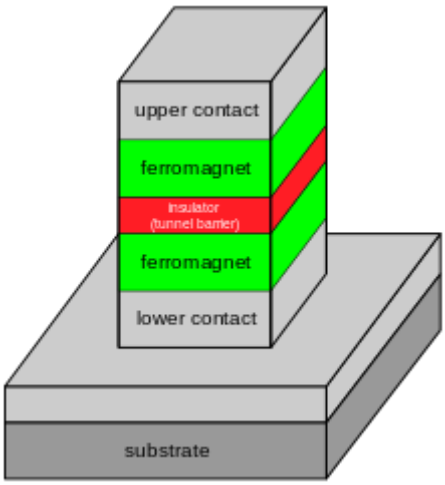

Figure 3: Illustration of the tunnel magnetoresistance effect. [8]

The benefits of MRAM is that it is faster and denser than SRAM and DRAM. Recent developments has also ensured that it is suitable for low power operation. [9] [10]

The drawbacks of MRAM is that in order to manufacture the different layers, more masks and more complicated processes are required, adding to the cost.

4.5.1.3 FRAM (Ferroelectric RAM)

FRAM stores data by applying an electrical field across a dielectric material, changing the polarization of the electrons within. This polarization is retained after the electrical field is removed. To read the stored value, a value is written over the dielectric. If the written value is the same as the previous stored value, nothing happens. If the written value is the opposite of the stored value, a brief pulse can be registered on the output lines. The read is destructive, requiring a write after reading.

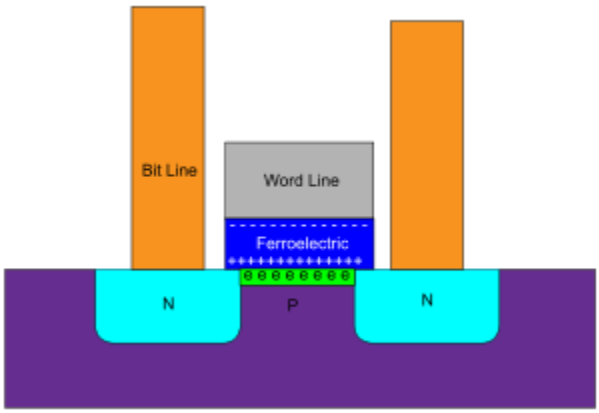

Figure 4: The setup of an FRAM cell. [11]

The benefits of FRAM is that it potentially only requires 2 additional masks during fabrication, significantly reducing the complexity and cost of fabrication. It is the main competitor to MRAM for the memory of the future. A drawback compared to MRAM is that FRAM's destructive read limits the speed of operation.

4.5.1.4 RRAM(Resistive random-access memory)

RRAM works by changing the resistance across a dielectric material. [12] shows that an RRAM circuit can be fabricated using a 0.18um TSMC technology. RRAM is still in development and the details on its layout and fabrication are unknown at this time.

4.5.2 Volatile memory

Volatile memory is a type of memory that requires a supply voltage to retain data. The two main types of volatile memory is DRAM and SRAM.

4.5.2.1 DRAM (Dynamic RAM)

DRAM is a compact way of creating memory, requiring only a single transistor. DRAM stores data as a charge across a capacitor connected to the bit line by a pass transistor. To write to a DRAM cell, set the word line to '0', opening the pass transistor, then the bit line is forced to a desired value, causing the capacitor to either charge or discharge. The bit lines of other cells connected to the word line are held stable by a sense amplifier circuit.

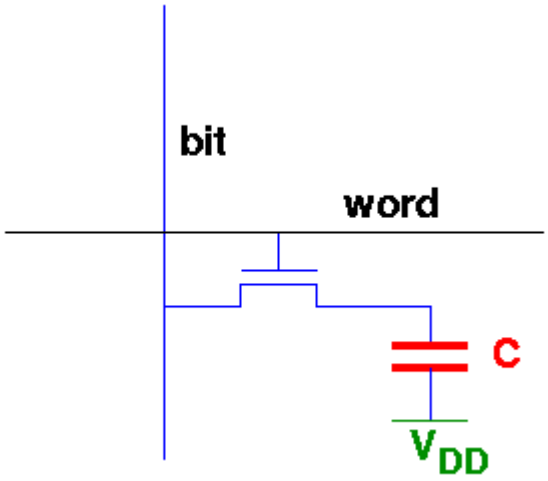

Figure 5: A single DRAM cell. [13]

Due to leakage through the pass transistor and capacitor, the capacitor will not hold its desired value for longer periods of time. This requires the value stored on the capacitor to be continually re-written. This process is called refreshing and is handled by the DRAM controller circuit.

The advantages of DRAM is its compact form. DRAM is much denser than SRAM. A DRAM memory cell requires only one transistor, as opposed to SRAM, where the standard solution used 6 transistors. The disadvantages of DRAM is that reading a value destroys the value, and the read value needs to be rewritten to the cell. This limits the speed of the DRAM cell. As a consequence, DRAM is often slower than SRAM. The typical access time of DRAM is 40ns, compared to 4ns for SRAM. [14]

4.5.2.2 SRAM (Static RAM)

Static RAM is called so because it retains its data as long as a supply voltage is present, unlike DRAM which needs refreshing.

6T SRAM cell

A 6T-SRAM cell retains its data by having two inverters connected in a feedback loop. The first inverter inverts the input given from the second inverter, and sends that output to the input of the second inverter, which in turn inverts and sends back to the first inverter. This means the voltage from either VDD or GND from the output of the first inverter reinforces the charge on the input of the second inverter, and visa versa. This mutual reinforcement of charges on the gates of the transistors in each inverter is what retains the data.

To write to the SRAM, the charges stored on either side of the inverter loop must be forced to the desired value. To do this, the two bit lines are forced to the desired voltage, one will be VDD and one will be GND. The word line transistors are then opened. This will draw the charges out of the inverter loop and force the loop to store the new value instead.

The simplest way to read from an SRAM cell is simply to open the word line pass transistors and read the voltages on the bit lines. The bit lines have a large parasitic capacitance and will take some time to charge. Using a sense amplifier to quickly sense the difference in voltage on the bit lines will help solve this problem.

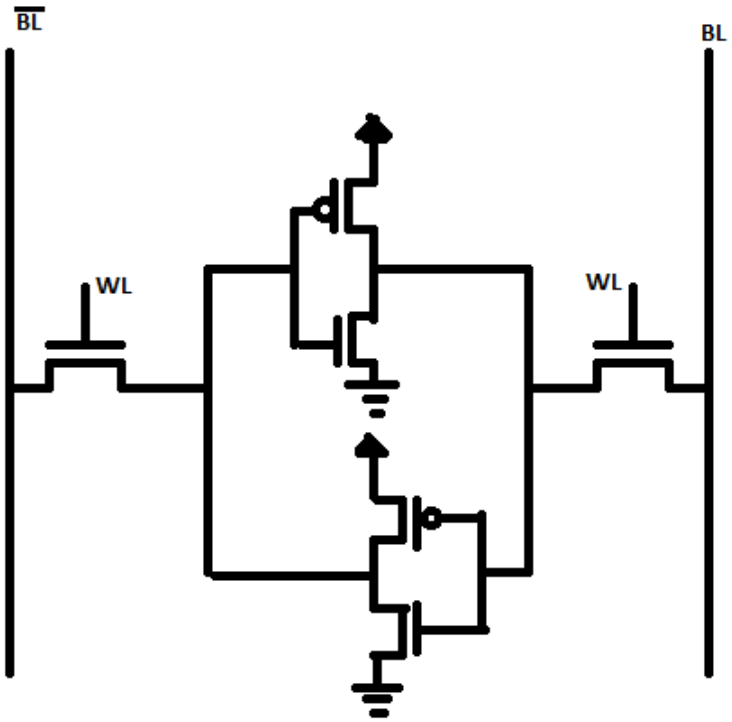

Figure 6: A standard 6T SRAM cell.

4T Leaking SRAM cell

This four-transistor leaking SRAM cell takes advantage of leakage currents through the bit line pass transistors.

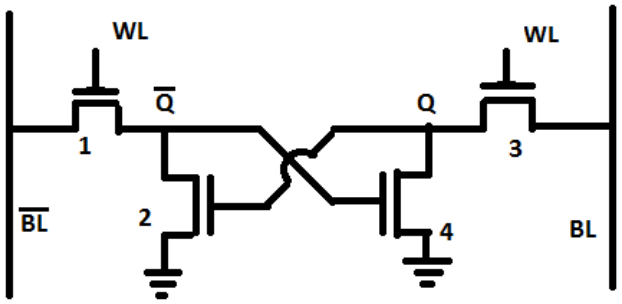

Figure 7: A 4T leaking SRAM cell using NMOS transistors.

In order to explain the operation of the 4T leaking SRAM cell, let's have a look at figure 7. When no reads or writes are being executed, both bit lines are held at a voltage of VDD. Let's look at transistors 3 and 4. If transistor 4's Rds_{off} is higher than transistor 3's Rds_{off}, then most of the voltage drop from VDD to GND will be on transistor 4. This causes the gate voltage on transistor 2 to be higher than V_{th}, turning on transistor 2. Because transistor 2 now has a much lower R_{ds} than transistor 1, there will be a very low voltage drop V_{ds} across transistor 2, which in turn means that the gate voltage V_{gs} on transistor 4 is below V_{th}, turning it off, causing a reinforcing feedback loop.

To read from the cell, the write line pass transistors, 1 and 3, are turned on and the transistor which was previously in an 'on' position (either 2 or 4) will begin lowering the voltage on the bit line. Care must be taken to assure that the bit line voltage does not drop too much in order to ensure the cell's value is not changed, along with all the cells connected to the same bit lines. The slight voltage difference between the two bit lines are read by a sense amplifier and produces a read logic value.

Writing to the cell is even harder to accomplish because one of the bit line voltages have to be forced to GND, potentially corrupting the values on all cells connected to the bit line. Care must be taken to assure that the duration of the 'low' GND pulse on the bit line is long enough to change the value on the cell whose bit line transistors are open, but short enough so that the charges stored inside other cells (on the source/drain capacitances inside the cell) are not discharged, corrupting their logic value.

4.5.2.3 Latches

SR-NOR-Latch

S:	R:	Action:	Qnext:
0	0	Hold state	Qprevious
0	1	Reset	0
1	0	Set	1
1	1	Undefined	Race condition

Table 2: SR-NOR-Latch truth table

An SR-latch is a simple latch that uses two NOR in a feedback loop to store a value. The principle of operation are as follows. If the input S is asserted high, the output Q is set, meaning Q will be set to '1'. If the input R is asserted high, the output Q will be set to '0'. If S and R both are low, the output will remain the same. S and R should not be high at the same time.

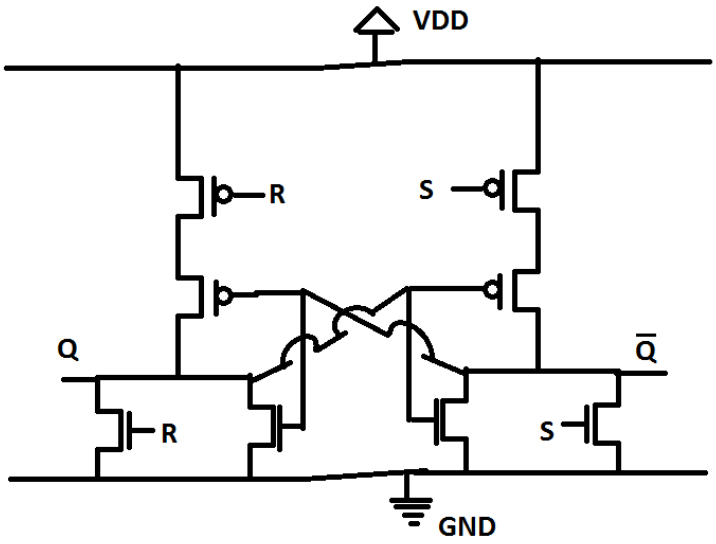

Figure 8: An SR-NOR-Latch transistor-level design.

Assuming that no other transistor-level implementations other than the straightforward implementation is possible, the total transistor count for an SR-NOR-Latch is 8 transistors, 4 for each logic gate.

D-Latch

A D-latch samples an input D, and if the enable signal Clk is high, it stores that value. Clk does not necessarily correspond to the Clock signal in the circuit.

Clk	D	Q	Comment
0	X	Qprev	No change
1	0	0	Reset
1	1	1	Set

Table 2: D-latch truth table

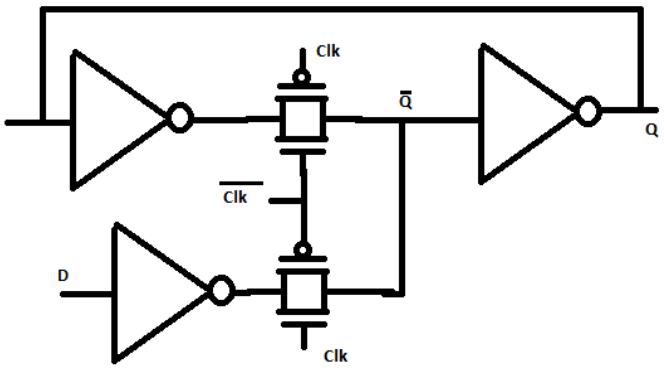

Figure 9: A design of a D-latch using clock gating

4.5.2.4 Flip Flops

D Flip flop

A positive-edge-triggered D flip flop stores the value D only when the signal Clk transistions from low to high, a so called rising edge.

Clk	D	Qnext
Rising edge	0	0
Rising edge	1	1
Non-rising	X	Qprev

Table 3: D-flip-flop truth table

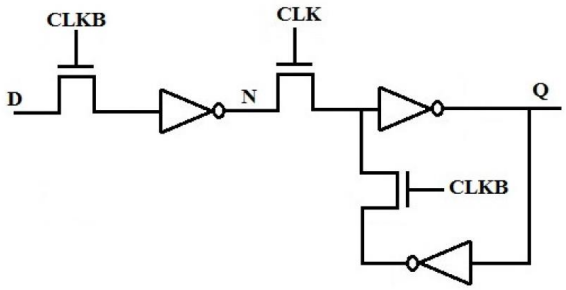

Figure 10: A low power, low area flip flop [3]

5. Implementation

5.1 Viability analysis

5.1.1 General requirements

Minimize leakage power

As explained before, minimizing leakage power is the goal of this report.

Large feature size / limited technology

Because of the requirement of low leakage power and because of economic considerations, choosing a simple and widely adopted feature size is recommendable. A large feature size reduces the leakage current in each transistor. The mask set for a simple large feature size process is significantly cheaper than the mask set for a newer technology node process. The simulations were run in a 90nm technology but the report is aimed at an 180nm standard process.

Area considerations

The area of the chip is not the highest priority consideration, but the smaller the chip, the cheaper it is to produce. A smaller area, i.e. fewer transistors, can also reduce the leakage power as each transistor contributes with its own leakage current.

5.1.2 Viability of non-volatile memory

The benefits of non-volatile memory in respect to leakage power is that it does not require a continuous supply of power to retain its data. Through power gating, the memory's power source can be switched off and only turned on when the memory needs to be accessed. If memory accesses are infrequent, which they are in most applications in the Internet of Things (IoT) industry, then the memory area can be in sleep mode (no supply voltage) most of the time. When the supply voltage is turned off, the leakage current is approximately zero.

Nonvolatile memory is generally also fairly dense. All nonvolatile memory mentioned in this report requires only 1 transistor per bit stored, with varying transistor sizes depending on fabrication

specifics. Nonvolatile memory also requires complex supporting circuitry to perform reads and writes.

The drawbacks of non-volatile memory is that it complicates the manufacturing of the chip. As opposed to DRAM and SRAM which require only standard CMOS technology to function, non-volatile memory requires unique transistor structures which require additional processes in fabrication.

FRAM is much less complicated to manufacture than Flash or MRAM, but still requires additional masks.

As this report targets a simple CMOS technology, Flash, FRAM and MRAM are not viable. RRAM has been shown to work on simple CMOS technology [12], but the details surrounding its fabrication are unknown.

5.1.3 Viability of DRAM

DRAM is the densest of the volatile memories, requiring a single transistor per bit stored. Though area is a factor, leakage is a much greater concern. The fact that DRAM requires continuous refreshing of the values stored on the capacitors, which in turn drains a lot of energy, means that it is completely unusable for this chip. [15] Estimates that for a 1um technology, 1Mbit memory, the data retention current for DRAM is 1mA while only 0.1mA for SRAM. If these estimations are true and also valid for a sub-0.18um technology, DRAM has a leakage power approximately 10 times greater than that of SRAM.

5.1.4 Viability of Latches, Flip Flops and SRAM

Both latches, Flip Flops and SRAM have the advantage of having static memory, meaning the value stored in each cell does not degrade as long as a supply voltage is present.

5.1.5 Peripheral circuits

When considering which type of memory to use it is not only important to look at the size and leakage characteristics of the memory cell itself, but also the circuit surrounding it which is required for the cell to function as intended. Examples of peripheral circuits are multiplexer circuits for addressing each individual cell, and state machines or timing circuits for controlling reads and writes.

D-latch

The latch requires two inputs to be demultiplexed to each individual cell, the Clk signal and the D signal. Each output also has to be multiplexed onto the output bus.

SR-Latch

The SR-latch is similar to the D-latch except that the S and R values are not taken available from outside the memory cell array. The SR-latch does not take input in the form of a ‘0’ or ‘1’ bit value, but stores a ‘1’ by setting the S signal high, and stores a ‘0’ value by setting the R signal high. A circuit with a truth table as shown in table 4 could be used to convert the input to the corresponding S and R signals.

D	W	S	R
0	0	0	0
1	0	0	0
0	1	0	1
1	1	1	0

Table 4: Truth table of a 1 to 2 decoder with active enable for use with the SR latch.

This control logic does not have to be a part of the S-R cell, the input value from the bus could be demultiplexed into an S and R signal and then the S and R signals could each be demultiplexed and sent to the cell being written to.

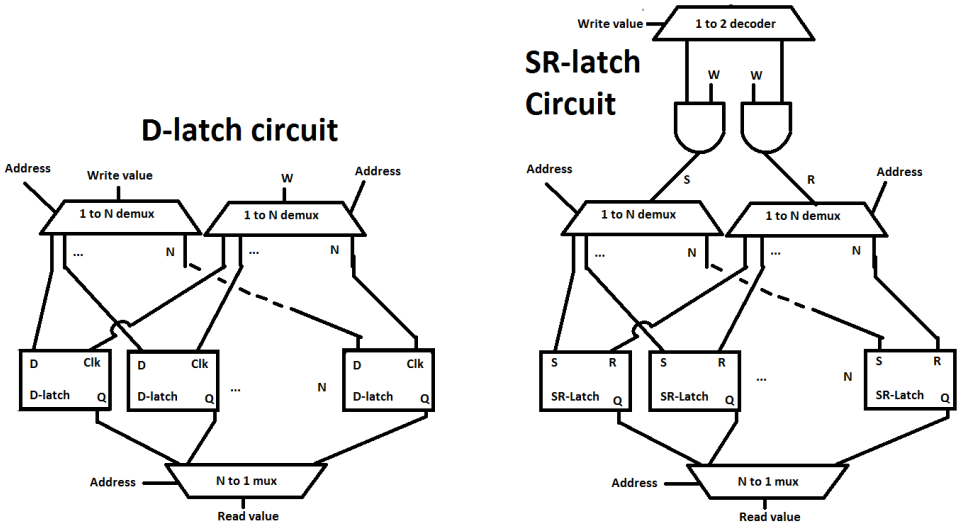

Figure 11: Proposed peripheral circuits surrounding each of the proposed latch arrays

Figure 11 shows proposed solutions to the peripheral circuitry surrounding the latch arrays. The downside to using latches is that certain timing requirements have to be met in the logic circuit in order to retain functionality.

D flip flop

The peripheral circuitry for the D flip flop could be identical to the circuitry for the D-latch proposed in figure 11. Timing requirements are a lot easier to meet for the D flip flop than for the latches.

SRAM

SRAM is entirely different in that the multiplexing of each cell happens within the cell arrays. A cell will not be written to unless the WL voltage is high, but the cell will also retain its value even when WL goes high as a result of another cell being written to. This is because the column circuitry makes sure that only the cell being written to will have its bit lines connected to any external voltage source or ground. The charge on the bit lines of the cell not being written to is not strong enough to change the value of the cell.

Because a lot of the multiplexing is done inside the cell array itself, the size of the multiplexer/decoders can be reduced. The control circuit would require a 1 to R row multiplexer (R is the number of rows) for the write line logic. The column / bit line logic would require more than just multiplexers, but at least one 1 to C column multiplexer or decoder (C is the number of columns)

would be present to distinguish between columns. Given C = R, the number of output pins on each multiplexer or decoder would be $\sqrt{N}$, where N is the number of cells in the array.

Each column has two bit lines attached to it, and requires a sense amplifier circuit and control logic to handle which bit line is to be charged (connected to VDD) or closed off (High impedance) depending on inputs from the SRAM controller. The size of this column circuitry can't be determined until the SRAM controller is designed.

The downside of SRAM as opposed to latches is that SRAM requires a state machine circuit to handle every read and write, and the complexity of said circuit is unknown before the design of the state machine itself. The design of this state machine and sense amplifier circuits is not covered in this report. The 4-T SRAM cell requires a more complex state machine than the 6-T SRAM cell due to more precise timing requirements.

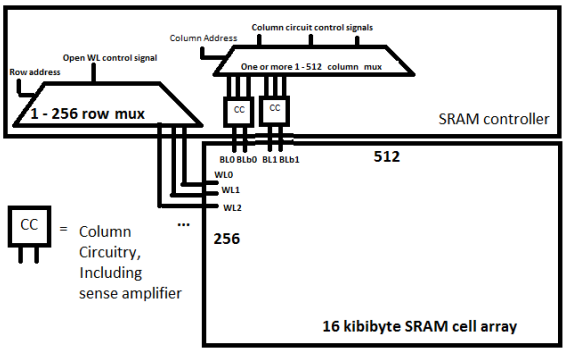

Figure 12: A proposed naive implementation of the SRAM peripheral circuitry

Voltage gating the peripheral circuits

As most of the peripheral circuitry is not needed to keep the value stored, its power supply could be turned off by voltage gating in order to save power during dormancy.

5.1.6 Table of viability for each kind of memory

Memory type	Advantages	Disadvantages	Viability in a simple 0.18um CMOS technology
Non-volatile			
Flash	Mature technology, dense	Complicated manufacturing process	Not viable
MRAM	Better performance than all other kinds of memory	Complicated manufacturing process, immature technology	Not viable
FRAM	Better than all kinds of memory except MRAM. Simpler to manufacture than MRAM	Complicated manufacturing process, immature technology	Not viable
RRAM	Simpler than flash, can be produced in a simple CMOS technology (?)	Immature technology. uncertainty about fabrication details	Not viable
Volatile			
DRAM	Denser than SRAM and latches	High leakage from refreshing	Not viable
SRAM, 4T and 6T	Denser than latches	Possibly large peripheral circuit	Viable
Latches	Very mature technology, dating back to the 60s	Often not very dense	Viable
Flip Flops	Same as latches, but with easier timing constraints	Same as latches	Viable

Table 5: A table of viability preceding analysis and simulation of all the mentioned types of memory.

5.1.7 Analysis of leakage power

After a viability check performed in the previous section, a simple by-hand leakage power analysis developed for this report is performed on the 5 remaining memory cell architectures, SR-latch, D-latch, D-Flip-Flop 6T-SRAM and 4T-SRAM.

For a simple analysis, the following assumptions are made for every cell except the 4T SRAM cell:

- All PMOS transistor dimensions W and L are assumed to be equal.
- All NMOS transistor dimensions W and L are assumed to be equal.
- A transistor is either in an ‘on’ or ‘off’ state.
- An ‘on’ transistor in series with an ‘off’ transistor is regarded as a short circuit because of the huge difference in R_{ds} between the two transistor states.
- All VDD and GND voltages are the same in all circuits.
- The R_{ds} for a transistor in the ‘off’ state remains the same even when placed in series with another transistor in the ‘off’ state.

- Q (the data voltage) = ‘1’, S = ‘0’ and R = ‘0’ for the SR-latch, D=’0’ for the D-latch and D-Flip-Flop, bit lines are held at VDD for SRAM.
- Gate leakage is insignificant.

If these assumptions apply, then in order to determine the leakage current, only the resistance from VDD to Ground in the cell needs to be calculated. Refer to formula 1. A higher resistance means less leakage.

Applying these assumptions, simple analyses can be performed:

SR-Latch

See figure 8 for the original circuit.

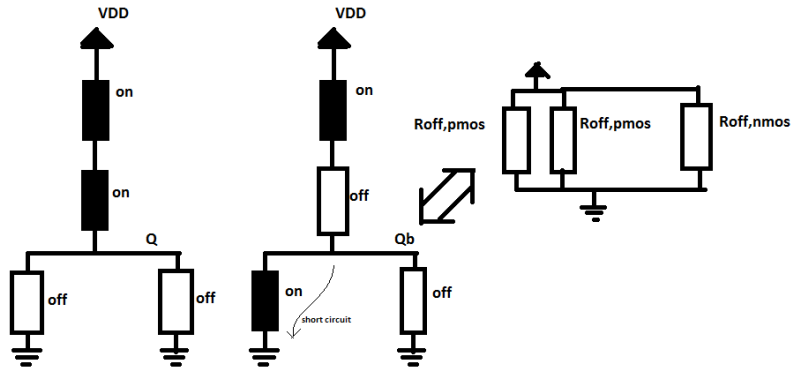

Figure 13: Simple resistance analysis for the SR-latch cell

The formula for the resistance in the SR-Latch becomes:

$$R_{VDD-GND} = \frac{R_{off_{pmos}} * R_{off_{nmos}}}{R_{off_{pmos}} + 2 * R_{off_{nmos}}} \quad [6]$$

D-Latch

See figure 9 for the original circuit.

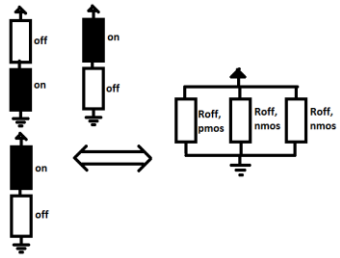

Figure 14: Simple resistance analysis for the D-latch cell

The formula for the resistance in the D-Latch becomes:

$$R_{VDD-GND} = \frac{R_{off_{pmos}} * R_{off_{nmos}}}{2 * R_{off_{pmos}} + R_{off_{nmos}}} \quad [7]$$

D-Flip-Flop

Through inspection of figure 9 and 10, one can see that the D-Flip-Flops inverters are in the same state as the D-Latch.

The formula for the resistance in the D-Flip-Flop becomes:

$$R_{VDD-GND} = \frac{R_{off_{pmos}} * R_{off_{nmos}}}{2 * R_{off_{pmos}} + R_{off_{nmos}}} \quad [8]$$

6T-SRAM

See figure 6 for the original circuit.

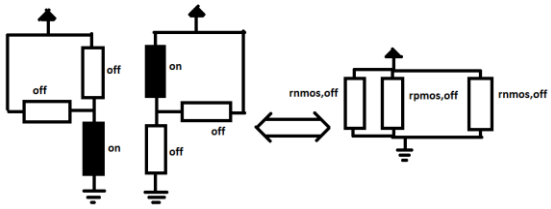

Figure 15: Simple resistance analysis for the 6T SRAM cell

The formula for the resistance in the 6T SRAM becomes:

$$R_{VDD-GND} = \frac{R_{off_{pmos}} * R_{off_{nmos}}}{2 * R_{off_{pmos}} + R_{off_{nmos}}} \quad [9]$$

4T-SRAM

The 4T-SRAM cell is different from the others in that it requires careful sizing of the transistors in order to function. Not all of the previous assumptions can be applied.

Looking at figure 7, $R_{DS_{pass}}$ will be significantly lower than the pass transistor would be in a 6T-SRAM cell. $R_{DS_{nmos1}}$ and $R_{DS_{nmos2}}$ will be higher than the resistance of an NMOS transistor in a 6T-Transistor.

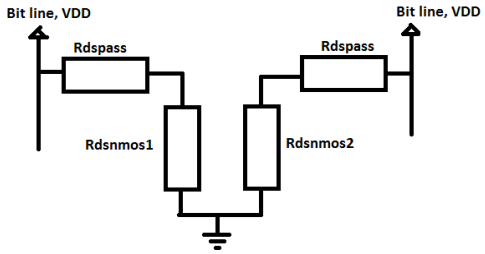

Figure 16: Simple resistance analysis for the 4T SRAM cell

The formula for the resistance in the 4T SRAM becomes:

$$R_{VDD-GND} = \frac{R_{DS_{pass}}{}^2 + R_{DS_{pass}} * R_{DS_1} + R_{DS_{pass}} * R_{DS_2} + R_{DS_1} * R_{DS_2}}{2\,R_{DS_{pass}} + R_{DS_1} + R_{DS_2}} \quad [10]$$

This is an unwieldy equation to work with. The R_{DS_2} and R_{DS_1} values cannot be known pre-simulation. Comparison between the 4T-cell and the other cells can't be done pre-simulation.

5.2 Simulation

3 Cells were simulated, the gated D-latch shown in figure 9, the 6T-SRAM shown in figure 6 and SR-NOR-Latch shown in figure 8. The reason for these 3 cells being simulated is arbitrary, based on the order of completion before the inaccuracy of the simulation models was detected. A circuit supposed to determine $R_{off_{nmos}}$ and $R_{off_{pmos}}$ was also simulated. The program used was Cadence ADE-L IC6.1.6. The technology library used is gdpk090, a reference 90nm process distributed by Cadence. The simulations were run using the "conservative" default simulation setting.

The wire capacitances and resistances were calculated as part of an SRAM model developed for this report. A 1kByte memory made up of an array of 200 cells in height and 400 cells in width was assumed. See appendix B.

In all memory cells and testbenches, a minimum transistor width of 150nm and a minimum transistor length of 120nm is applied. The chosen transistor type is nmos_2v and pmos_2v with a supply voltage of 2.5 V.

The Memory cells and testbenches are shown in appendix A.

The circuit for determining $R_{off_{nmos}}$ and $R_{off_{pmos}}$ is shown in figure 17. While measuring the leakage current through the transistors, the PMOS and NMOS lengths and the PMOS width are swept from 200nm to 1um. The operating temperature is swept from 10°C to 70°C.

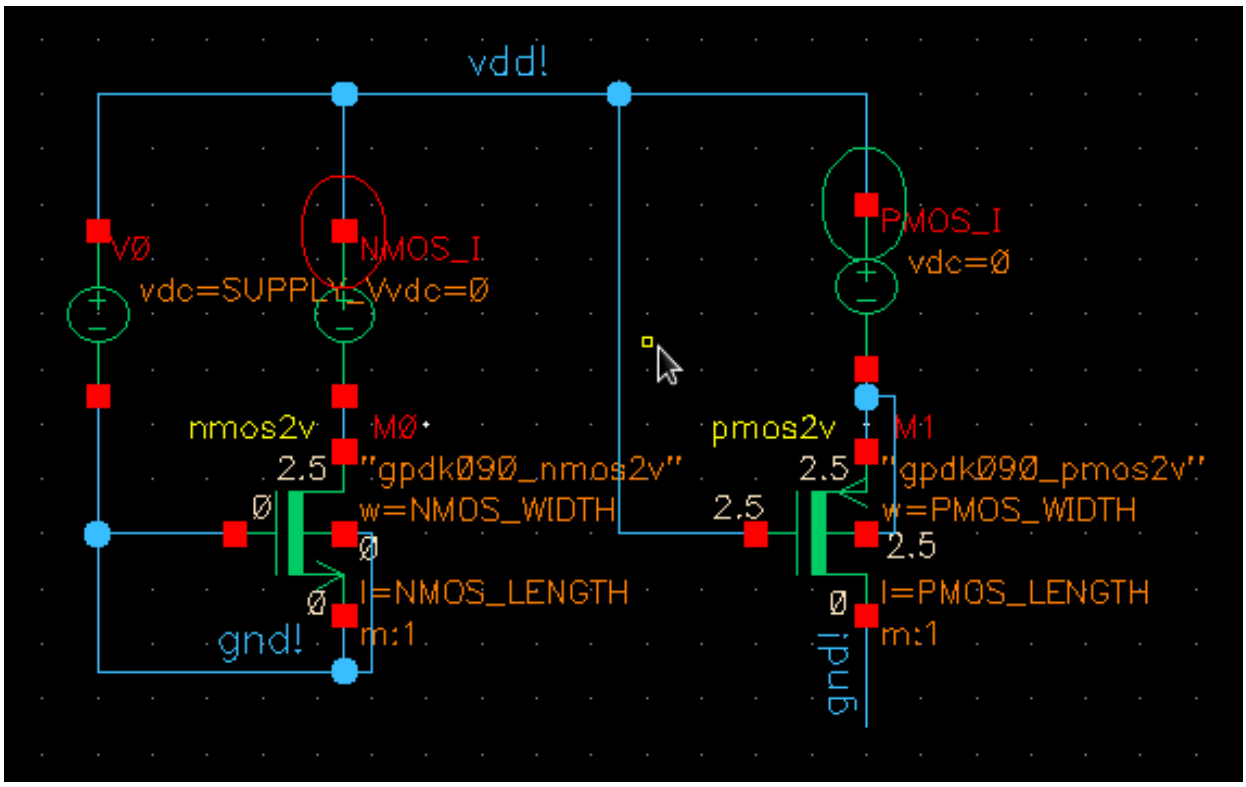

Figure 17: The testbench for determining Rdsoff for bboth the nmos2v and the pmos2v transistors

6. Results

As explained in sections 1 and 5.2, three cells were simulated, based on the order the simulation setups were completed before it was discovered that the results would be meaningless.

Nevertheless, the simulations were set up to be transitive, meaning the cells would be simulated in the time domain. The advantage of running a transitive analysis is that it allows one to run the cells through a read and write cycle before measuring the leakage currents, assuring that the cell's internal voltages assume feasible static values. The leakage was also supposed to be measured at both a stored logic '1' and a logic '0'. The results are not included in this report, as they are meaningless.

The following simulations are simulations of the testbench in figure 17, sweeping the gate length of the NMOS transistor, and sweeping the gate width and length of the PMOS transistor.

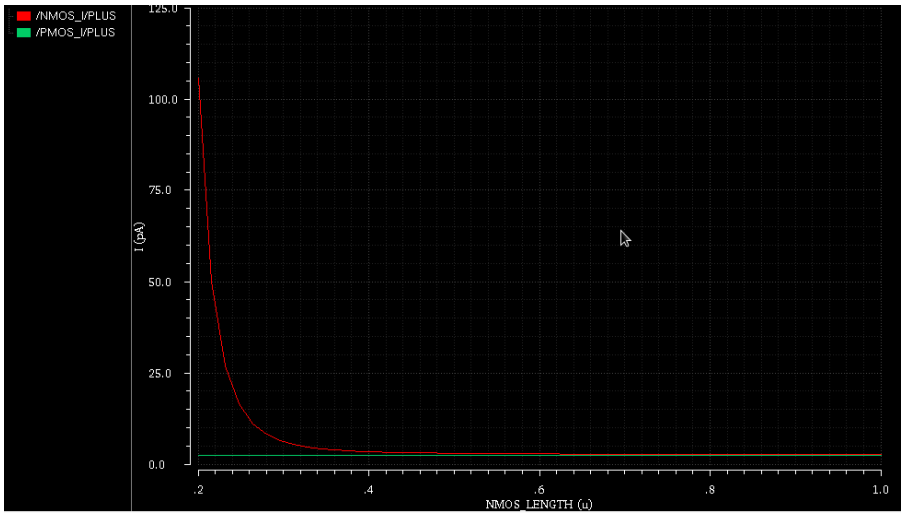

Figure 18: PMOS and NMOS leakage currents as a function of NMOS length

Figure 19: Incorrect NMOS and PMOS leakage currents as a function of PMOS length

Figure 20: Incorrect NMOS and PMOS leakage currents as a function of PMOS width

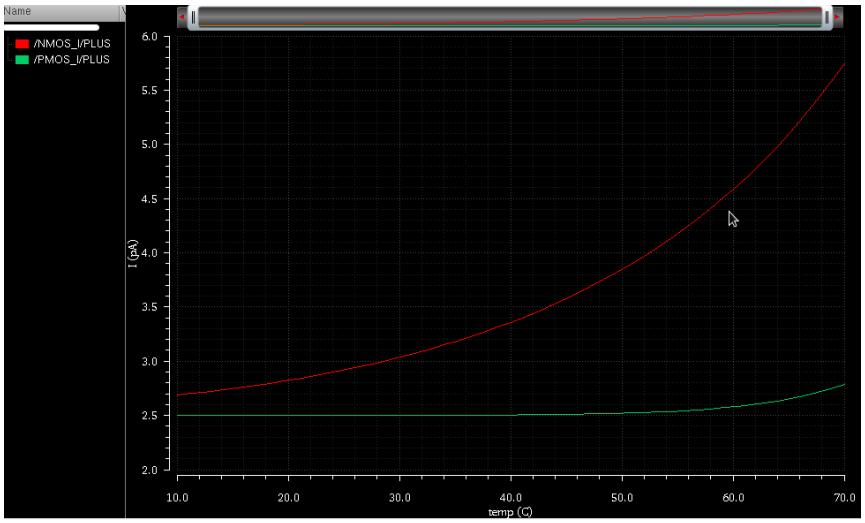

Figure 21: Incorrect NMOS and PMOS leakage currents as a function of temperature

The following simulations were provided by Disruptive Technologies as a complementary simulation following the failure of the simulation tool used in this report. The following parameters were provided:

Technology node: 0.18um
L=0.35u
W=0.5u
Temp=27^{o}C
VDD = 2.5V

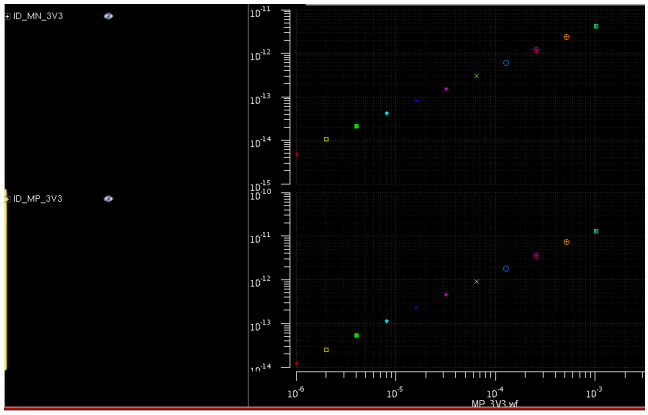

Figure 22: Leakage currents as a function of transistor width in a 0.18 CMOS process

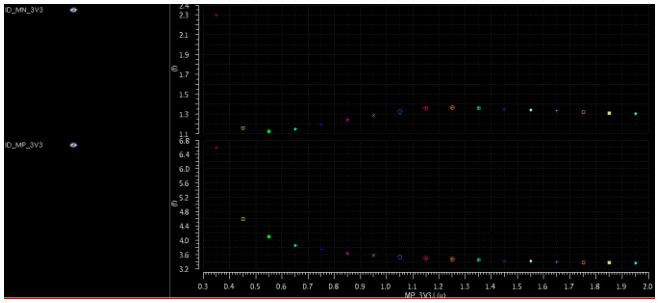

Figure 23: Leakage currents as a function of transistor length in a 0.18u CMOS process

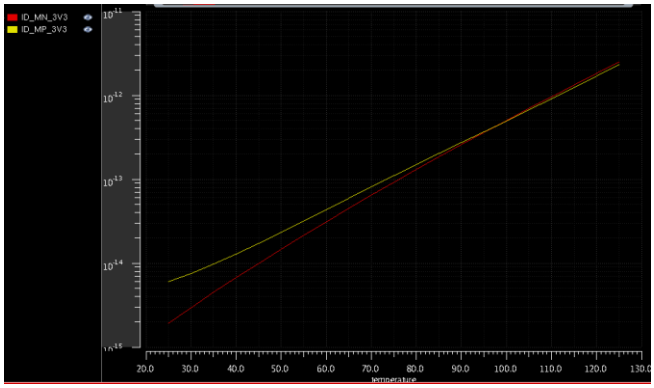

Figure 24: Leakage currents as a function of temperature in a 0.18u CMOS process

7. Discussion

7.1 Inaccurate models

In figure 18 one can see that when the length of the NMOS goes up, its leakage current goes down, which correlates well with formula 2. However, figures 19 and 20 shows that even a large change in the length and width of the PMOS transistor has almost no effect on the leakage current through it. Figure 21 also shows that a large increase in temperature is required before the leakage current of the PMOS transistor starts to increase. Other transistors pairs in the gdpk090 package were also tested, but all of them showed subthreshold characteristics which were not consistent with theory. This implies that the gdpk090 technology library is not intended for subthreshold leakage current simulation, and a more precise model set needs to be applied.

The inaccuracy of the simulation models is also why only a few cells were simulated. A precise technology library is required to make any such simulations useful in determining leakage power.

7.2 Method

Simulation is the most useful tool for determining leakage currents in memory cells. Calculations for single transistors with certain assumptions are possible (see formula 2), but once several transistors are connected in series and their source/drains are connected to the gates of other transistors, the state space quickly grows too big for any human to comprehend or analyze. Simple analyses can be performed, as proposed in section 5.1.7, but they do not offer much insight into the benefits or disadvantages of similar memory architectures such as flip-flops, SRAM and Latches.

The accuracy needed for simulation of leakage currents is a disadvantage, as the only tools available that meet the requirements are expensive and difficult to maintain. Precise transistor models provided by the chip manufacturer are required. In a research context, one can't always depend on the availability of these high-end tools and manufacturer-provided models, as was experienced in this report.

Even with high-end tools, the simulation of leakage currents in large circuits is a computing-power intensive effort. Efforts must be made to simplify the simulations without sacrificing too much accuracy.

7.3 Comparison based on analysis

The three first cells, SR-latch, D-latch, D-Flip-Flop and 6T-SRAM have resistance formulas which are almost identical. The only difference is that the 6T-SRAM's resistance increases more when $R_{off_{nmos}}$ increases than when $R_{off_{pmos}}$ increases, while for the SR-latch it is the opposite. The D-latch resistance assumes either one of the 6T or SR formulas for resistance depending on the D input. If it is possible to design for a default D-value when the cell is dormant, then the designer can choose the resistance formula which gives the most resistance.

The 4T SRAM Cell remains to be designed, but could potentially have less power leakage due to its nature of taking advantage of an already existing leakage current coming from the bit lines.

7.4 Externally acquired simulations

Figures 22, 23 and 24 show more believable results, with the PMOS transistor producing a leakage current greater than that of the NMOS at normal operating temperatures. In figure 23, the leakage current through the NMOS decreases until the length hits 500nm before increasing again, implying that an optimal NMOS gate length can be found to minimize leakage and area. This might also be true for other technologies.

The fact that the PMOS has a greater leakage current implies that care must be taken when designing CMOS circuits to use NMOS instead of PMOS whenever it is feasible.

8. Conclusion

The most important lesson to learn from this report is that things take time, unforeseen limitations can prevent any designer from completing his/her work. In this case the limitations of the Cadence

gdpk090 process technology library caused the subthreshold leakage current simulation results to be meaningless.

The pre-simulation analysis did however produce some useable results. When simplifying to a large extent, the four cells SR-Latch, D-Latch, 6T-SRAM and D-Flip-Flop are roughly equivalent. This lends credibility to the idea that the lowest area cell can be chosen without any penalty in increased leakage power. See table 6 for a quick area comparison of the 4 previously mentioned cells.

Cell name:	SR-Latch	D-Latch	6T-SRAM	D-Flip-Flop
Transistors per cell:	8	10	6	9

Table 6: Area considerations for 4 of the proposed memory cells.

8.1 Future work

The obvious work to be done is to simulate the cells using a technology library that properly simulates subthreshold currents.

The accuracy of the analysis of the peripheral circuits is questionable at best. Designing these circuits and measuring the areas of the peripheral circuitry is an important next step.

Modelling a transistor in the 'off' state as a simple resistance may not be viable. The inaccuracy of the technology library caused this simplification to be impossible to verify in simulation. More complicated transistor models may have to be developed.

The possibility of RRAM to be fabricated using a simple 0.18u CMOS technology needs to be evaluated.

9. References

[1] http://www.disruptive-technologies.com/ 10.12.2015

[2] Yang, Kim: A Low-Power SRAM Using Hierarchical Bit Line

and Local Sense Amplifiers, IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 40, NO. 6, JUNE 2005

[3] Khan, Beg: A New Area and Power Efficient Single Edge Triggered Flip-Flop

Structure for Low Data Activity and High Frequency Applications, Innovative Systems Design and Engineering, Vol.4, No.1, 2013

[4] Analysis and Design of Subthreshold Leakage Power-aware Ripple Carry Adder at Circuit-level Using 90nm Technology, International Conference on Intelligent Computing, Communication & Convergence, 2015

[5] Chenming Calvin Hu: Modern Semiconductor Devices for Integrated Circuits, 2010, chapter 7, page 265

[6] http://www.explainthatstuff.com/flashmemory.html 10.September 2015

[7] http://www.computerworld.com/article/2493603/data-center/everspin-ships-first-st-mram-memory-with-500x-performance-of-flash.html 14.12.2015

[8] https://en.wikipedia.org/wiki/Tunnel_magnetoresistance 13.09.15

[9] Dong, Wu, Sun, Xie, Li, Chen: Circuit and Microarchitecture Evaluation of 3D Stacking Magnetic RAM (MRAM) as a Universal Memory Replacement, Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE

[10] Desikan, Lefurgy, Keckler, Burger: On-chip MRAM as a High-Bandwidth, Low-Latency Replacement for DRAM Physical Memories, Department of Computer Sciences, Tech Report TR-02-47, The University of Texas at Austin

[11] https://en.wikipedia.org/wiki/Ferroelectric_RAM 13.09.2015

[12] Sheu, Chiang, Lin, Lee, Chen, Chen, Wu, Chen, Su, Kao, Cheng, Tsai: A 5ns Fast Write Multi-Level Non-Volatile 1K bits RRAM Memory with Advance Write Scheme, VLSI Circuits, 2009 Symposium on

[13] www.cs.auckland.ac.nz 10.08.2015

[14] Wang, Hamdi: Matching the Speed Gap between SRAM and DRAM, High Performance Switching and Routing, 2008. HSPR 2008. International Conference on

[15] Sakurai, Nogami, Sawada, Iizuka: Transparent-Refresh DRAM (TReD) Using Dual-Port DRAM Cell, Custom Integrated Circuits Conference, 1988., Proceedings of the IEEE 1988

[16] Erich Barke: Line-to-Ground Capacitance Calculation for VLSI, A Comparison, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, VOL. 7, NO. 2, FEBRUARY 1988

[17] Horowitz, Computer Systems Laboratory, Lecture 4, Stanford University, http://eia.udg.es/~forest/VLSI/lect.04.pdf

Appendix A: Cell simulation results

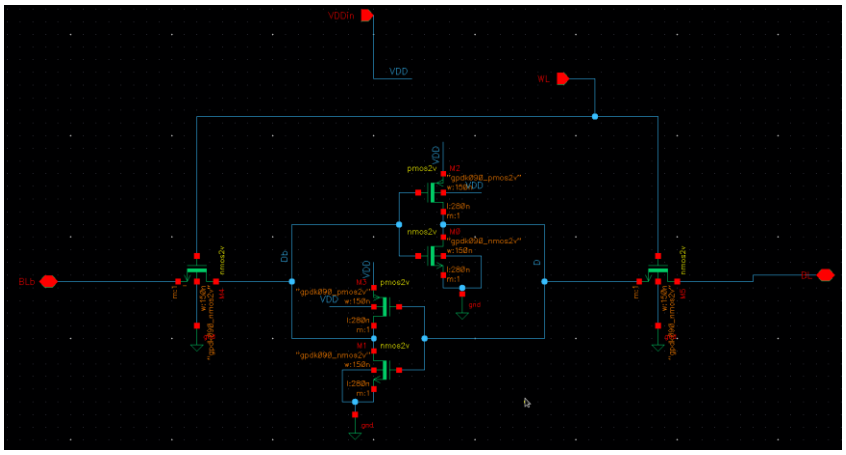

Figure 25: The 6T SRAM cell modelled in Cadence

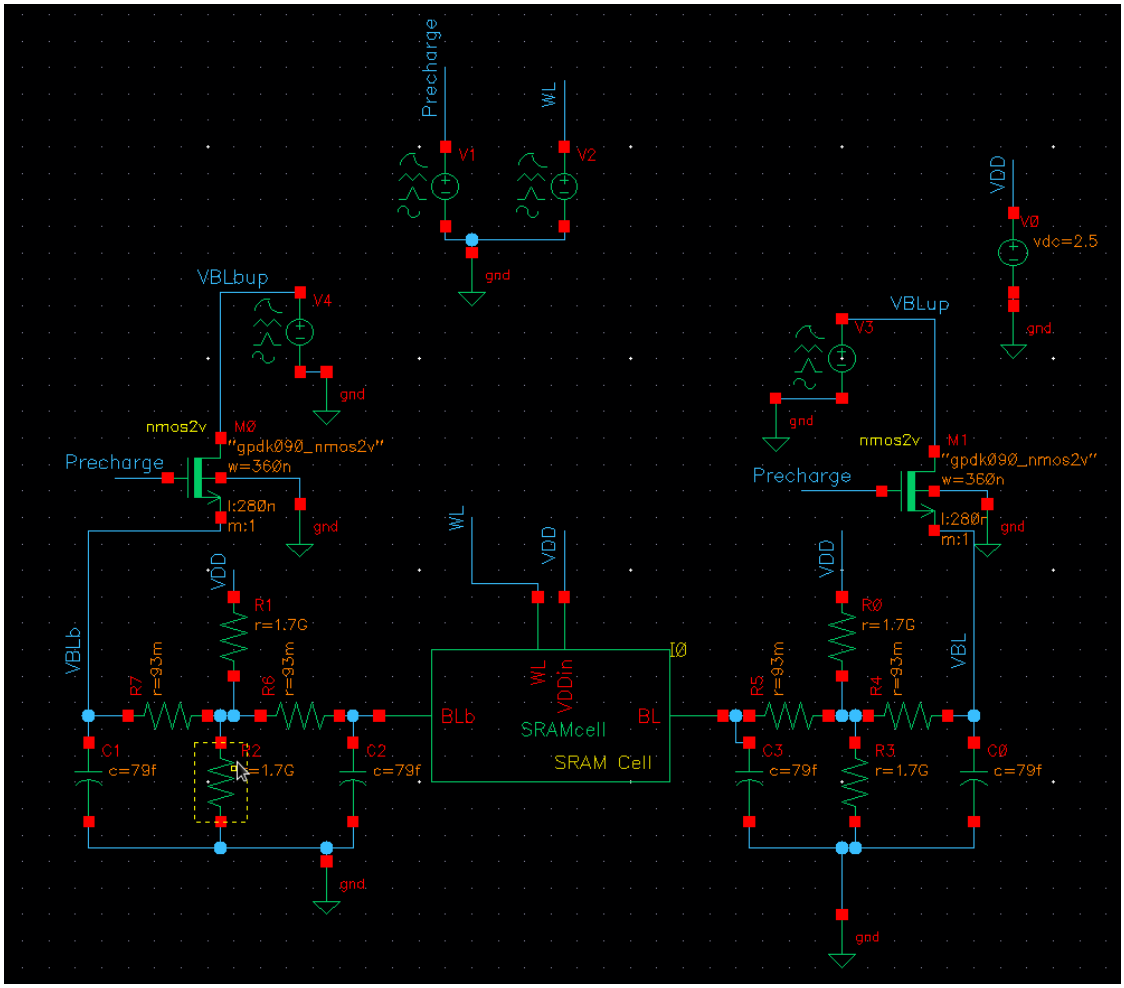

Figure 26: The 6T SRAM cell testbench

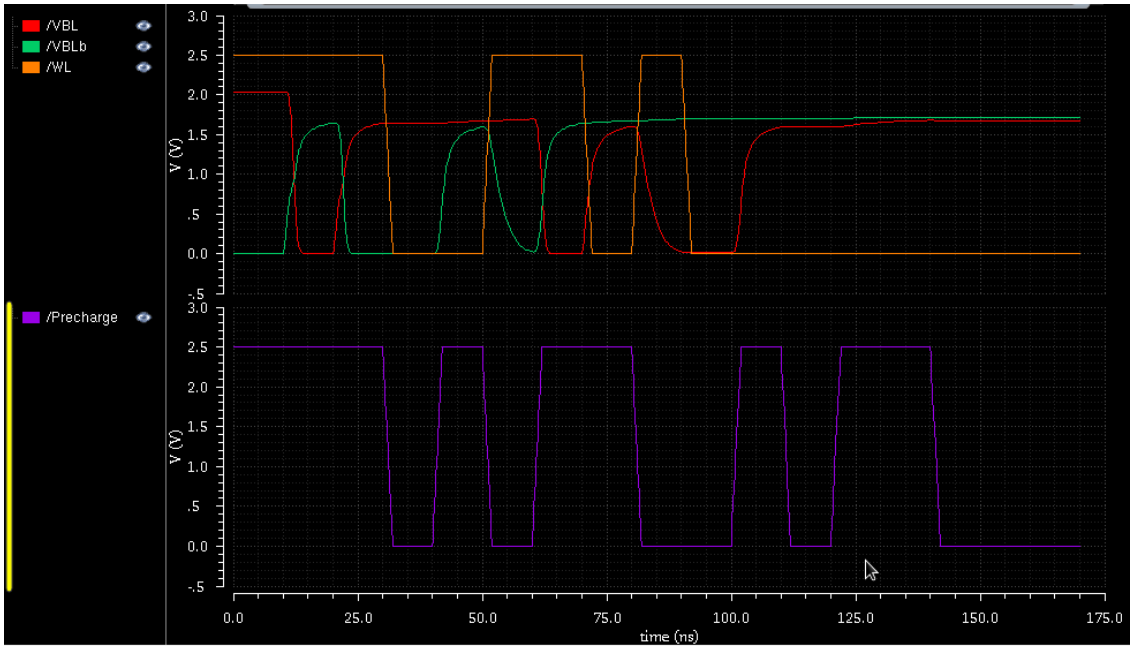

Figure 27: An operational simulation of the 6T SRAM cell

In figure 27, at time = 50ns and time=80ns, two reads are taking place. At time = 50ns, the VBLb (the bit line voltage) is pulled low by one of the inverters, signifying a logical '1' being stored. At time = 80ns the opposite bit line is pulled low, signifying a logical '0'.

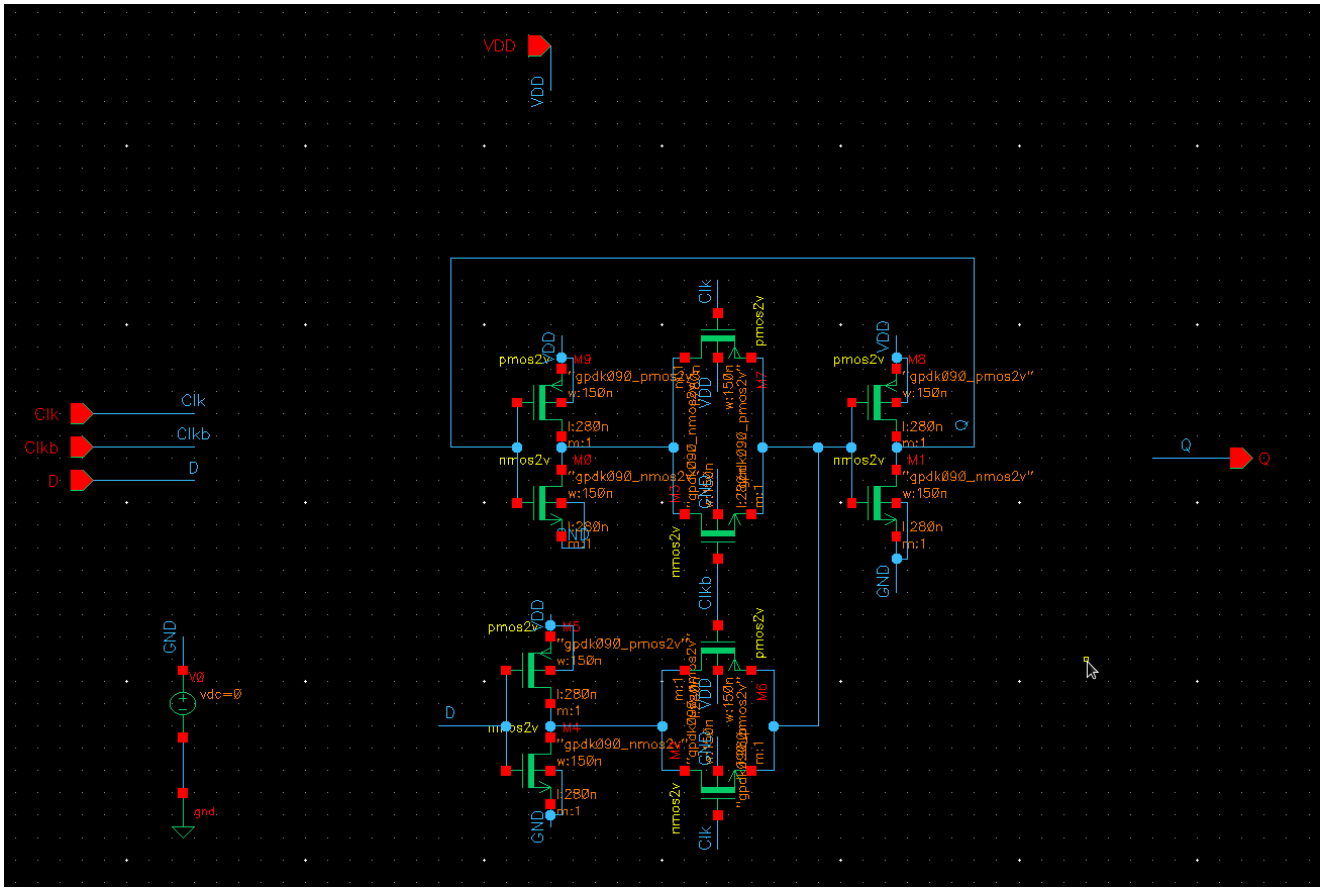

Figure 28: The gated D -Latch cell modelled in Cadence

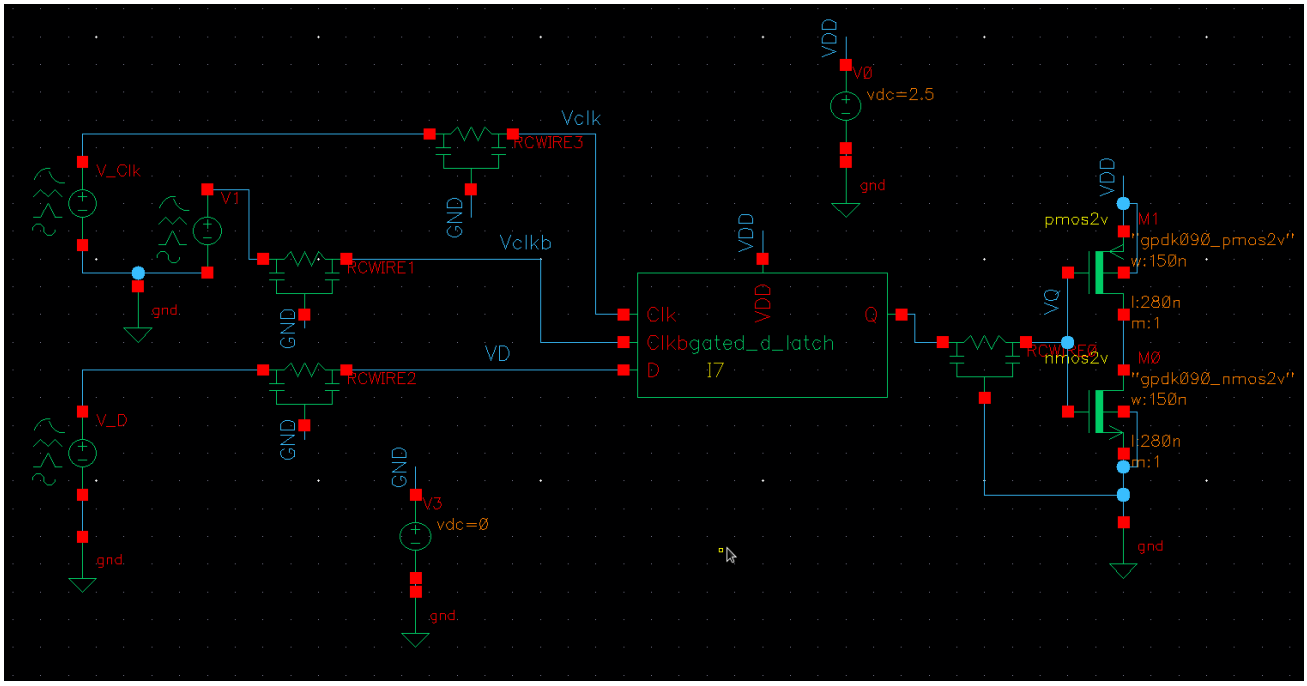

Figure 29: The testbench for the gated D-latch

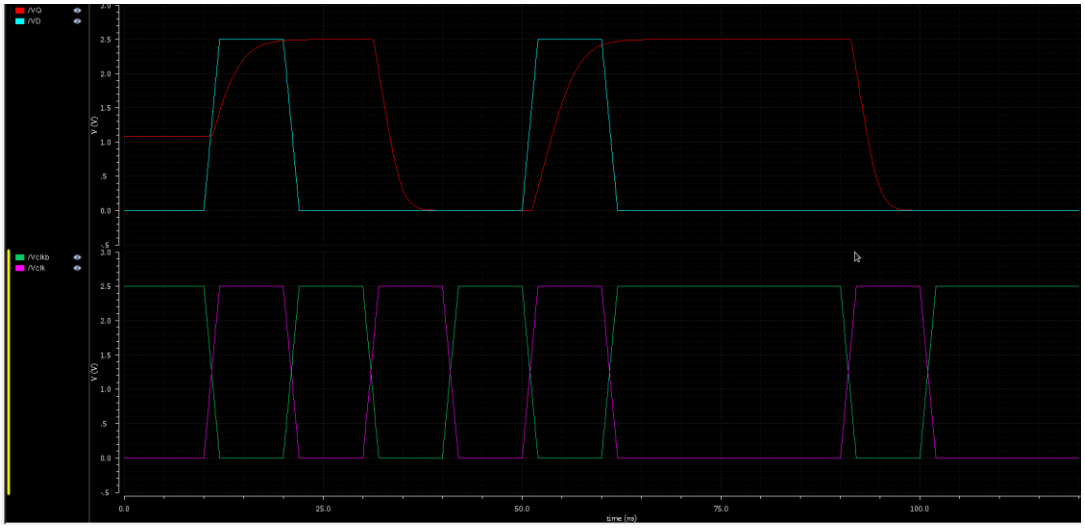

Figure 30: The simulation results for the gated D-latch

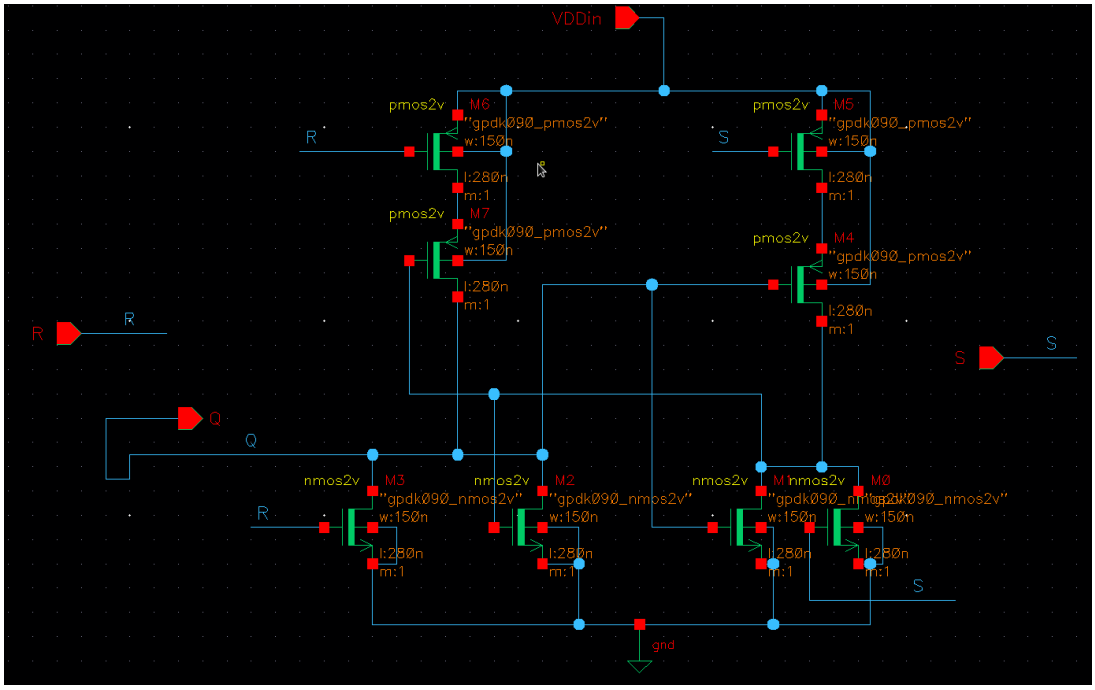

Figure 31: The SR-Latch modelled in Cadence

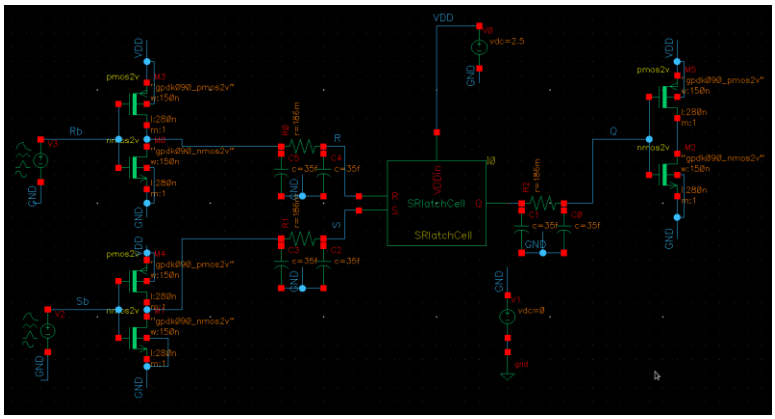

Figure 32: The testbench for the SR-Latch

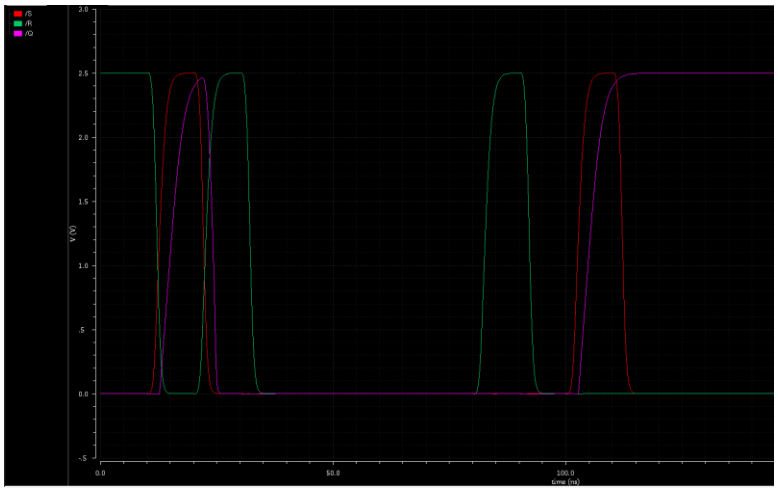

Figure 33: The simulation results for the SR-Latch

Appendix B: SRAM circuit model

This is a model developed to emulate the effects of other cells in the SRAM cell array.

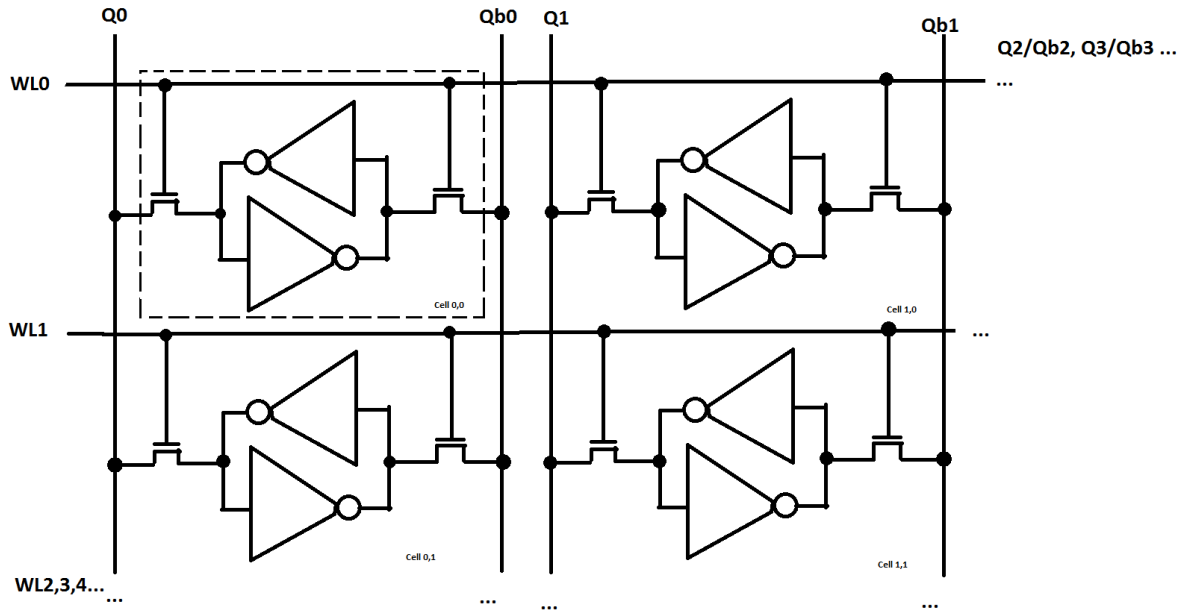

Figure 34: The SRAM cell seen in the context of the SRAM cell array

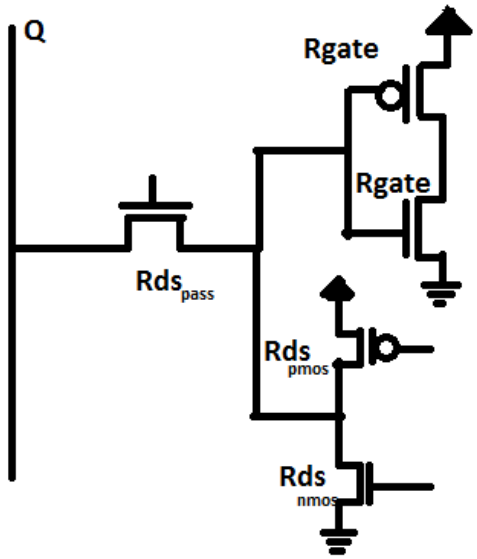

Figure 35: The SRAM cell if one is looking into the cell from the bit line

As each cell is connected by a pass transistor to its corresponding bit line Q and Qb, it is a good idea to model the effect of other cells connected to the bit lines, as well as the effect of the bit lines themselves.

A regular wire is usually modelled using the π-model, but the bit lines are connected to many pass transistors distributed evenly across the bit line. The pass transistors add capacitance to the wire, as well as a connection to GND and VDD caused by drain-source leakage.

If one were to look into a cell from the bit line's perspective when the circuit is inactive, it would look like figure 35. The pass transistor would be in the cut-off region and one of the transistors in the bottom inverter would be in the saturation region and the other in cut-off. If one assumes that the gate resistance of a transistor is much higher than the drain-source resistance during cut-off, the current from the bit line to the upper inverter is negligible. If one also assumes that the current through the transistor in the saturation region is much greater than the current through the one in cut-off, the simplified view from the bit line becomes as shown in figure 36. The Rds_{pass} resistance is the cut-off resistance of the pass transistor and the $Rds_{nmos/pmos}$ is the Resistance from drain to source in the saturation region of the transistor. The value stored in the cell determines which transistor is in cut-off.

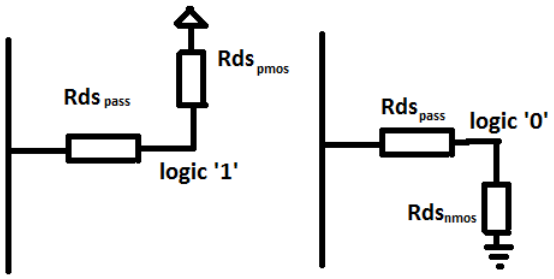

Figure 36: A simplified view from the bitline looking into the SRAM cell

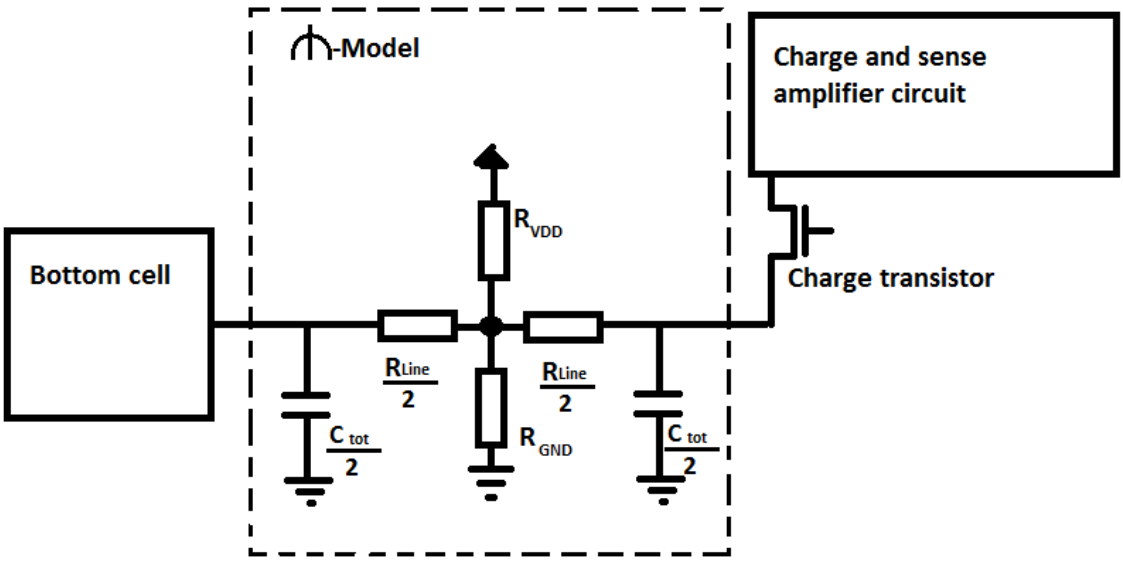

Figure 37: The phi model being used in the SRAM cell array

To solve the problem of a series of resistances either to VDD or GND along the bitline, a new model is presented in this paper, named the Ψ-model. See figure 37. C_{tot} is the sum of the total capacitance of the wire plus the capacitances to ground (in this case the drain capacitance of the pass transistor) of each branch. R_{line} is the same as in the π-model. R_{VDD} is found by finding the resistance to VDD

from the bit line in every branch, and then solving the problem of N parallel resistances to combine the resistances into one resistance. R_{GND} is found similarly.

Calculating R_{line} ,R_{VDD} ,R_{GND} and C_{tot} in the SRAM circuit

In all calculations the transistors have a default size of W = 180 nm and L = 100 nm.

R_{line} – Assuming an array of 200 cells in height and 400 cells in width, forming 1kB of memory, the length of the wire running down to the bottom cell is the height of a cell * 200. If the height of one cell is approximately 12 times the minimum feature size, then the length of the wire in a 90nm technology is L = 12 * 200 * 90nm = 216 µm.

The formula for the resistance of a copper wire is *R = P * L / A*, where *P is the resistivity of copper and A is the cross-sectional area of the wire. In a 90nm technology, A is assumed to be A = 140nm * 140nm = 1.96 * 10^{-14} m^2. P is 1.68* 10^{-8}. The resulting resistance becomes* **R_{line} = *186mΩ.***

C_{tot} – for capacitance of a wire, equation 2 in [16] is used with the simplification that the height from substrate is the same as the height of the wire. The permittivity ε in silicon is calculated to be approximately 1*10^{-10} F/m. The capacitance per metre for the wire when h = t = w is then $C = 4\varepsilon$ = 4*10^{-10} F/m. The capacitance of the wire is then 4* 10^{-10} * 216 µm = 86 fF.

For drain capacitance a rule of thumb [17] is used which estimates a drain capacitance of 2 fF per µm of transistor width. The total drain capacitance is then 0.18µm * 2 fF * 200 transistors = 72fF.

The total capacitance **C_{tot} = 158 fF**

R_{VDD} and **R_{GND} –** In figure 36, one can see that with a few simplifications, each cell represents either a resistance to ground or to VDD, depending on the logic value of the cell. Assuming a 50-50% distribution of 1's and 0's, there are 100 parallel resistances to VDD of value $RDS_{pass,off}$ + $RDS_{pmos,on}$ and similarly there are parallel 100 resistances to ground of value $RDS_{pass,off}$ + $RDS_{nmos,on}$. $RDS_{pass,off}$ is the dominant value and was measured in simulation to be $RDS_{pass,off}$ = 170GΩ.

$$\frac{1}{R_{VDD}} = \sum_{i=1}^{100} \frac{1}{170G\Omega} \text{ [11]}$$

R_{VDD} = R_{GND} = 1.7GΩ.