[Abstract](#_52l4o5y4kl3d)

[Background](#_japvh0vkamk6)

[Traditional Memory - Fernando](#_h9zrcnaf1m19)

[Spin Transfer Torque (How it works) - Sergio](#_6ebh0h1nc8c)

[Look Up Table - Andrea (slide 11~14)](#_me07d3wswt91)

[Experiment](#_7dio9knxa7z2)

[The netlisting - Brandon](#_h25vsw37vm0f)

[Individual results - Janine](#_ta4lwavjs4hc)

[Multiple Gates - Janine](#_atpst27druyg)

[Conclusion](#_8ijaqkq9duwo)

[Conclusion](#_h48d9h4erpxx)

[References](#_gvr4r5tsxaty)

# Abstract

Spin Transfer Torque Random Access Memory (STTRAM) is a promising technology for information storage in the form of magnetic fields rather than existing charge-based memories like static random-access memory (SRAM), dynamic random-access memory (DRAM) and flash. Charge-based memory is notorious for its volatility or its constant use of power. STTRAM technology is advantageous for its non-volatility, complementary metal oxide semiconductor (CMOS) compatibility, and programmability and hardware security through look-up tables (LUTs). The contents of LUTs may be programmed and then simulated in a software called HSPICE so it can implement any arbitrary logic function. However, LUTs have significantly more transistors than logic gates, so the amount of delay and power consumption will inevitably increase. The purpose of this paper is to maximize the number of LUTs implemented while minimizing delay and power consumption by replacing specific logic gates with LUTs. At first, all logic gates of a 4-bit adder were individually replaced. From the data extracted, certain gates produced high delay and power consumption. The next step is to take gates with no to little penalties and map them with their respective LUTs at once. By discovering the optimal combination, we will be able to minimize the disadvantages of SRAM, DRAM, and flash such as extreme power consumption while being non-volatile and programmable.

# Background

## Traditional Memory - Fernando

* 1. SRAM DRAM and Flash
     1. SRAM
        1. Volatile (loses memory when cut off from power source)
        2. Large cell size
           1. Limited memory
        3. Built into CPU
           1. Used for cache memory due to its great access speeds
     2. DRAM
        1. Memory stored in capacitor
           1. Stores data by charging capacitor
           2. Requires power to constantly charge capacitor since charge leaks out(power leakage)
        2. Smaller than SRAM therefore has more memory density
           1. A single cell is made of one transistor and one capacitor
        3. Read and write speeds are slower than SRAM
     3. Flash
        1. Slow read/write speeds
           1. Slower than DRAM
        2. Limited endurance compared to SRAM and DRAM
           1. Finite read/write cycles
        3. Non-volatile
           1. Can store data without connected to power
           2. Can be used as permanent storage
  2. All custom CMOS design
     1. What is it
        1. Logic gates (show picture of logic gates)
           1. Made of transistors, which are tiny switches that allow or block off a signal
           2. Combination of logic gates form circuits designed with specific tasks in mind
        2. Receiving a specific output based on what and how many inputs given
           1. Output is called truth table
     2. Can easily be reversed engineered
        1. Pirates can rob company technology
        2. Big security issues
  3. STT(Transition)
     1. Proposed phenomenon will amend problems
        1. Has fast read and write speeds, non-volatile, small cell size therefore dense, and durable

## Spin Transfer Torque (How it works) - **Sergio**

* 1. Magnetic Tunneling Junction
     1. Composition
        1. Two ferromagnetic layers, one called fixed the other called free.
           1. Fixed has a fixed magnetic field while the free layer’s magnetic field can be changed.
           2. Because the free layer is smaller, it can freely flip orientations
     2. read/write operation
        1. Depending on the orientation of the two magnetic layers, we can read either a 0 for an “off” state and a 1 for an “on” state
        2. Parallel state is 0(low resistance), anti parallel state is 1(high resistance)
        3. Low for read,high for write/critical write current
  2. Physics behind STT
     1. Low current to write a 0(transfer electrons change the magnetic field direction of the free layer)
     2. High current to write 1(needs a larger amount of reflected electrons)
  3. Why we want this?
     1. Advantages
        1. Non-volatile, compatibility, programmability
           1. Since information is stored in magnetic fields rather than charge based, the data will be preserved even when power is cut off.
     2. Faster startup time if used in memory technology
     3. Security
        1. Prevents reverse engineering
        2. Harder to steal and pirate designs

## Look Up Table - **Andrea (slide 9~12)**

* 1. Logic gate in custom CMOS:
     1. Most logic gates are made from (Complementary metal–oxide–semiconductor) CMOS transistors and wires. The physical identity of CMOS logic gates cannot be altered once manufactured.
     2. a collection of logic gates with 1 to 2 input signals...
     3. Take NAND gate for instance, NAND gate is the negation of AND gate. The NAND gate in CMOS logic is composed of two PMOS transistors connected in parallel at the upper part and two NMOS transistors connected in series at the lower part.
     4. A truth table…
  2. STT based Look Up Table (STT-LUT) // structure
     1. Since our design is static implementation, STT-LUT is composed of a number of multiplexers (MUX) and STTRAM.
     2. Each STTRAM cell consists of two major components: MTJs and access transistors, which is CMOS circuit. The MTJs are used as storage cells to store logic “0” or “1” by the magnetic formation while the CMOS circuits are used to provide the switching current density to the MTJs and to select a particular memory cell.
     3. a MUX is a combinational circuit in which a set of selection inputs determines a specific input signal to be selected as output signal. The number of MUX-based LUT selector signals sets a limit on numbers of input signals; in other words, 2n bit content requires n numbers of selector signals.
  3. Logic Implementation using Look-Up-Table (LUT)// How and why we use LUT
     1. Since each LUT works as a reconfigurable function, we can map any logic function by programming the content of LUT.
     2. Take XNOR gate for instance: the function output of XNOR gate with 2 input variables is 1 0 0 1 respectively. By programming the content of LUT, we can devise a XNOR logic gate.
     3. LUT fan-in sets a limit on numbers of input variables; in other words, 2n bit content requires n numbers of input variables.
     4. LUT has only one output.
  4. The pros and cons of STT-LUT implementation:  
     The substitution of STT-LUTs with logics has its advantages and disadvantages. The 3 major benefits are:
     1. > Security: The reconfigurability of STT-LUT not only makes reverse engineering difficult but also neutralizes the threat of unauthorized hardware fabrication.   
        > Non-volatility: Since information is stored in magnetic fields rather than in charged based technology , the data will be preserved even when power is cut off.   
        > Compatibility: STT-LUT uses CMOS circuit for write and read access.
     2. While the advantages of STT-LUT is promising, to make STT-LUT much competitive, researches are focusing on addressing the issues of:  
        > power consumption  
        > Delay overhead  
        > Area.

# Experiment

## The netlisting - **Brandon**

* 1. Research Goal
     1. Design a hybrid circuit that consists of both custom CMOS and MTJ-LUT
     2. Find an optimal circuit that maximizes benefits of both CMOS and MTJ-LUT and minimizes penalties
     3. Use of a 4-bit adder to test hybridization
  2. 4-bit adder
     1. An adder is a circuit and its purpose is to take three inputs and produce outputs called a sum and a carry. The more complex the circuit, the more bits per input can be used
        1. Contains 30 gates (show picture of the full adder)
           1. A mix of NAND, AND, NOR, NOT, XOR
     2. Subcircuits (show examples of what subcircuits in cdesigner and code looks like)
        1. Subcircuits are essentially a smaller part of a circuit
        2. Subcircuits that we used were created with cdesigner and optimized by our mentor and professor
        3. Mention the subcircuit libraries we used
           1. Libraries contain all the subcircuits
  3. HSPICE
     1. Software used to run simulations and extract data
        1. What we call “netlisting” since we call the files we run netlists
  4. Mapping
     1. We replace logic gates with their respective LUTs and extract their delay and power consumption
  5. Transition: individual mapping and the all custom cmos
     1. Different LUT mappings on the Adder. We did one with all custom cmos and replaced one gate at a time to identify which gates with LUT implementation does not cause delay overhead.

## Individual results - **Janine**

1. Data
2. Figures: gate schematic (gates with no delay, critical path, and highlighted gates with delay from non critical path)
3. Bar plot: both active power in one plot (include standby power?), make one for pdp, make one for gates with delay penalty - I’ll plot the normalized results(?)
4. Analysis
   1. Critical output and changes in pathway

## Multiple Gates - Janine

* 1. Three algorithms were used to determine optimal selections of gates
     1. Independent Selection
        1. Gates that were not along the same branch are chosen to be mapped
     2. Dependent Selection
        1. Gates that are along the same branch are chosen to be mapped
     3. Parametric Aware Dependent Selection
        1. Gates that do not severely impact delay are chosen to be mapped regardless of dependency

# Conclusion

## Conclusion

* 1. Talk about what research we did and what we looked for
  2. Summarize results from mapping multiple gates using the algorithm

## References

**Notable Points**

Our focus on this research is to replace logic gates with LUTs and determine which specific individual gates and combination of gates produce the best results in terms of the least delay and least power consumption. Our method to approach

For LUTs, we use a subcircuit library for implementation. Subcircuits are basically smaller circuits that are a part of and used in a much larger, complex circuit. Within the subcircuit library we used, it includes the MTJ subcircuits, multiplexer (mux) subcircuits, and LUT subcircuits.

There are two MTJ subcircuit files; One represents the MTJ as a whole, which shows its parameters while the other contains the netlisting of the latches, which shows the arrangement of components in the MTJ.

The mux files are multiplexers. Multiplexers consist of an “n” number of data inputs, one or more select inputs (which is determined by 2n) and one output. The select input controls which data input is to appear at the output. Thus a multiplexer allows us to choose what input we want to use.

The mux is a component of an LUT, which is dependent on the amount of input.

The LUT files are used with a its respective mux whose select lines are the inputs of the LUT and whose inputs are constants.