ENHANCING HARDWARE SECURITY USING SPIN TRANSFER TORQUE LOGIC

A thesis submitted to the faculty of
San Francisco State University
In partial fulfillment of
The requirements for
The Degree

Master of Science

In

Engineering of Embedded Electrical and Computer Systems

By

Darya Almasi

San Francisco, California

August 2015

CERTIFICATION OF APPROVAL

I certify that I have read Enhancing Hardware Security Using Spin Transfer Torque Logic by Darya Almasi, and that in my opinion this work meets the criteria for approving a thesis submitted in partial fulfillment of the requirement for the degree Master of Science in Embedded Electrical and Computer Systems at San Francisco State University.

Dr. Hamid Mahmoodi, Ph.D.
Associate Professor of Computer Engineering

Xiaorong Zhang, Ph.D.
Assistant Professor of Computer Engineering

Enhancing Hardware Security Using Spin Transfer Torque Logic

Darya Almasi
San Francisco, California
2015

Hardware security is becoming an increasing threat to fabless semiconductor industries due to split between fabless design businesses and IC fabrication foundries that are globally distributed all over the world.
Spin Transfer Torque (STT) is a promising technology for information storage in form of magnetic rather than existing charged based memory such as SRAM, DRAM and flash. This technology due to its non-volatility, ease of programming, scalability and standard CMOS compatibility, offers a unique opportunity for enhancing the hardware security in a power and performance efficient manner.
In this research, we propose a unique application for STTRAM and that is to realize reconfigurable logic using Look-Up table (LUT) based logic implementation. Implementing part of a logic circuit in a programmable form hides the identity of the hardware from hacker who may attempt to reverse engineer a product to get access to intellectual property of design.

I certify that the abstract is a correct representation of the content of this thesis.

_____ _____

Chair, Thesis Committee Date

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. HAMID MAHMOODI for the continuous support of my graduate study and brilliant research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisor and mentor for my graduate study.

Secondly, my warmest thank to my dear husband, for his continued and unfailing love, patience, support and understanding my persistence during my years of study.

Last but not least, I must express my profound gratitude to my parents for supporting and encouraging me spiritually throughout my education and my life. This accomplishment would not have been possible without them.

TABLE OF CONTENTS

List of Figures

Figures Page

LIST OF Tables

Tables Page

CHAPTER 1

1. Introduction:

1.1. Why Hardware Security?

Hardware is the foundation of all computational tasks. It is the processor of any information from input source, enabler of any algorithm, software or communication protocols. All the computation will eventually be carried out by hardware, namely the processor or the circuits. In some mission-critical applications (e.g. military and aerospace) several sophisticated electronics are being used. Potentially the sensitive electronics can fall into wrong hands. Thus, to protect such electronics there are enormous amount of interests for seeking solutions against skillful, well-equipped and well-funded hackers [1]. Today's protection techniques make it difficult to the powerful attackers; however, it is still possible maintaining different type of attacks to reverse engineer the design and further to derive critical information.

Reverse Engineering of an IC is a process of identifying its structure, design and functionality [3]. Reverse engineering can

- Identify the device technology used in the IC. For instance it was identified the Intel's Xeon processor use Tri-gate transistors [4][3].
- Extract the gate-level netlist of a baseband processor from Texas Instruments was extracted in [5][3].
- Infer the functionality. Reverse engineering on Apple's processor revealed the type of graphic processing units used in iphone 5 [6][3]

While reverse engineering serves several benefits, an attacker can misuse it to steal and/or pirate a design. One can use the readily available tools and techniques for reverse engineering. On identifying a device technology, one can fabricate similar devices. One can extract a gate-level netlist and use it to design one's own IC or illegally sell it as an Intellectual property (IP). Also, one can use the components extracted from competitor's products. This way one can reveal competitor's trade secrets. Because of these harmful effects, reverse engineering was listed as one of the serious threats to semiconductor industry [3].

In addition a significant security treat has emerged because of changes to chip fabrication introduced from the flattening of the once-vertical Integrated Circuit (IC) supply chain [2]. In the past, IC design and fabrication were typically handled by the same entity,

because the cost to build a foundry, though expensive, was a reasonable investment. However, decreasing feature size and time to market, coupled with demands for lower power, high-performance ICs, have made the cost required to establish a full-scale foundry prohibitive. In 2005, a full-scale 300-mm-wafer, 65-nm-process foundry cost $3 billion to build [2]. Very few companies can afford such an expense, thus driving the market to specialize. Intellectual Property (IP) vendors have emerged that specialize in creating functional units that they license to IC designers for their ASIC designs. IC design companies integrate third-party IP along with their own to create an IC design [2]. Finally, contract foundries harness economies of scale, as they spread the large capital required to build foundries among their clients. These contract foundries, originally driven by inexpensive labor, have established themselves throughout Asia. Since 1976, the market has dramatically changed, as Asia has increased its share of shipped semiconductors by 60%, according to the Semiconductor Industry Association [2].

This market shift has caused the US IC industry to change its business model. Large companies such as Texas Instruments and AMD that once fabricated their own ICs have gone fabless, and other fabless companies such as Qualcomm, Broadcom, and Nvidia have broken into the top 20 semiconductor sales leaders [2]. This paradigm shift has created a serious security threat because the in-house fabrication process, once monitored very closely, is now being outsourced to potentially untrusted facilities [2]. A recent survey on the semiconductor industry by Semiconductor Equipment and Materials International (SEMI) reports that 90% of companies have experienced IP infringement, with 54% reporting it as serious or extremely serious [2]. Adaptations to the horizontal supply chain are necessary to support the business model and protect the financial and intellectual rights of all the involved parties [2].

The concerns that arise from IC fabrication fall into three basic categories: metering, theft, and trust [2]. Each category broadly addresses whether extra ICs beyond the purchase order have been produced, whether unauthorized access to information has occurred, and whether the IC has been tampered with [2].

Figure 1 (Partially borrowed from [2]) shows the process flow for taking a design to an IC and in dark grey steps there is a possibility of reverse engineering. The IC designer begins by creating the RTL description of the IC, typically in a hardware description language (HDL) [2]. RTL synthesis then involves a series of EDA steps to produce a finalized netlist. First, the logic synthesis stage transforms the RTL design into a directed acyclic graph (DAG), $N = (V, E)$, where N is a logic network, V is a set of logic nodes, and E is the set of edges. The nodes in V represent a Boolean logic function, and the

interconnections represent the information flow [2]. Complex heuristics search for graph transformations that optimize the system's design goals to increase circuit quality. The optimized design is sent to the mapping stage, where the DAG's Boolean functionality is converted into primitive gates [2]. The placing-and-routing stage searches for the best locations and best connections between each component. Then, the design is exported to a layout-level geometry, in which large plaintext descriptions capture the physical polygons of the IC [2]. This exact blueprint, the layout geometry, is sent to the foundry, which legitimately makes modifications to the polygons to support the design. The fabrication process then continues through its many steps to create, test, package, and distribute the IC [2].

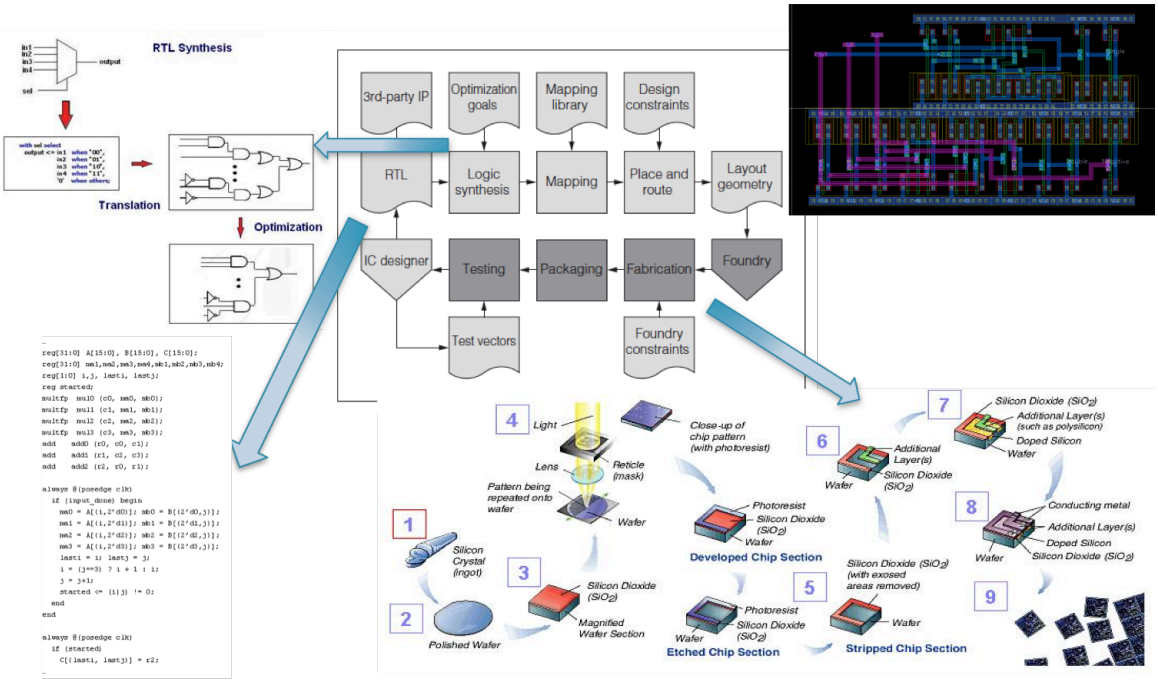


Figure 1, Process flow for IC fabrication, from RTL to IC

Our approach assumes there is a threat of a fabrication attacker external to both the IP creator and the IC designer, which follows directly from the widely popular horizontal contract model. The attacker enters after creation of the layout-level geometry but before IC fabrication. Also, the attacker has access to the layout level geometry and a set of test vectors, both given by the IC designer [2]. The attacker also has significant resources temporally, fiscally, and computationally, but they are finite [2]. These resources include advanced technical knowledge of IC design and fabrication, access to a foundry, ability to fabricate ICs, and advanced reverse-engineering tools [2].

The attacker's goal is IC piracy, which is motivated by either profit or acquisition of specialized functionality. In both cases, the attacker's potential benefit from piracy must outweigh its cost [2].

The standard horizontal IC supply chain is vulnerable in the fabrication phase because the entire IC design and specifications are transferred to the foundry without control over what the foundry will do with them [2]. The foundry is assumed to be well intentioned so that it will not make extra copies of the IC, steal information, or add a Trojan, but there are no guarantees [2].

Therefore hardware security is becoming an increasing threat and challenge to fabless semiconductor industries due to split between fabless design businesses and IC fabrication foundries that are globally distributed all over the world. We are proposing an emerging security technology for protection of intellectual property or design idea from stealing or reverse engineering.

1.2. Technology to Improve Hardware Security

Spin Transfer Torque (STT) is a promising technology for information storage. In this technology the information is stored in magnetic form that is non-volatile (STT-NV) and also much more scalable as compared to the existing charged based storage technologies such as SRAM, DRAM and flash. Moreover this technology is compatible with the standard CMOS technology, which is the mainstream technology for digital integrated circuits.

According to comparison STT type memory with other memories, currently, three general types of memory exist, each technology is suitable for specific applications: Static RAM (SRAM), Dynamic RAM (DRAM), and Flash memory. SRAM has excellent read and write speeds, but has a very large cell size (requiring 6 or more transistors per cell). The speed of SRAM makes it ideally suited for embedded applications, particularly cache memory, where performance is more important than memory density. SRAM is volatile, but requires very little active power for data retention [7]. DRAM is able to provide much better memory density through its use of a single transistor with a storage capacitor. However, charge tends to leak off of the capacitor, requiring a power hungry refresh cycle every few milliseconds. DRAM is typically used as the main system memory in a computer, where memory density and performance are more important than power consumption. Flash memory technologies are very attractive for mobile

applications where non-volatility and very high densities are required [7]. While Flash does have reasonably fast read access times, write speeds are very slow and endurance rates are very low (< 100; 000 cycles). To optimize for power, performance, and cost, a typical system must integrate all three types of memory. STT-MRAM promises to be a “universal memory", combining all of the advantages of SRAM, DRAM, and Flash. Such a memory would eliminate the need multiple application specific memories, improving system performance and reliability, while also lowering cost and power consumption [7].

Magneto resistive Random Access Memory (MRAM) is a technology that has existed in one form or another since the late 1970s. MRAM is based on the concept of using the direction of magnetization to store binary information, while exploiting magneto resistive properties for data retrieval [7]. The spin-torque transfer effect was first theoretically predicted and demonstrated by J. C. Slonczewski in 1996 [8], and has formed the basis of next generation MRAMs. STT-MRAM can scale down well, while reducing writing currents by more than a hundredfold [7]. The nonvolatile nature, low power, high performance, and memory density of STT-MRAM make it an excellent candidate for the first commercially available universal memory [7].

Therefore STTRAM has emerged as an alternative to conventional CMOS SRAM that offers significant leakage power reduction since the information is stored in the form of a programmable resistance represented by a Magnetic Tunneling Junction (MTJ) rather than by electron charge [9].

The main target application of STTRAM is for storage and the main targeted market is replacement of DRAM main memory and SRAM cache. The use of STTRAM and MTJ inside it has recently been explored for building low power programmable Look-Up Tables (LUT) used in Field Programmable Gate Arrays (FPGA) [9]. In this research, we propose this unique application for STTRAM and that is to realize reconfigurable logic using Spin Transfer Torque Look-Up table (STT-LUT) based logic implementation in which the LUTs are implemented using STTRAM technology on top of a NMOS transistor fabric realizing fixed logic.

Implementing part of a logic circuit in a programmable form hides the identity of the hardware from hacker who may attempt to reverse engineer a product to get access to intellectual property of design. During this way, we should quantify the impact of replacing logics with LUTs in terms of power, performance and area.

Thesis Outline

STT technology due to its non-volatility, ease of programming and CMOS compatibility, offers a unique opportunity for enhancing the hardware security in a power and performance efficient manner. Therefore we are using STTRAM technology in design of STT-LUT and our goal is to find the best options in such a complex circuit to replace these LUTs with one or combination of logic gates to secure the IP of design.

In comparison with Custom CMOS, logic gates in the form of LUT have much more overhead, because it has extra circuitry and more power and delay. But STTRAM allows us to lower the overhead because the good news is we only use this kind of memory in read operation and they have good characteristic in reading mode whereas they are no competitive in write operation in compare to other memory technologies. So after optimizing the design of STT-LUT, we specify one section of this report to characterize the power and delay of different fan-in STT-LUT in compare to complimentary CMOS.

Then we propose the novel idea for reducing power and delay overhead of STT-LUT, which it is Collapsing method in order to become competitive with custom CMOS. During implementing replacement of STT-LUT instead of logic gates in a circuit, we will face to different challenges such as cascading issue. We are trying to introduce this problem and some possible solutions for that.

Finally we implement all of mentioned method in a sample circuit from ISCAS Benchmark to evaluate power consumption and critical path delay for different approaches of collapsing. And we are showing that with collapsing method we can achieve a very good result in terms of power and delay while enhance the security of circuit deign.

Chapter 2

2. Background

2.1. STTRAM Memory Array

STTRAM consist of two elements NMOS transistor and MTJ. The typical architecture for STT-RAM memory cell is shown in figure2.

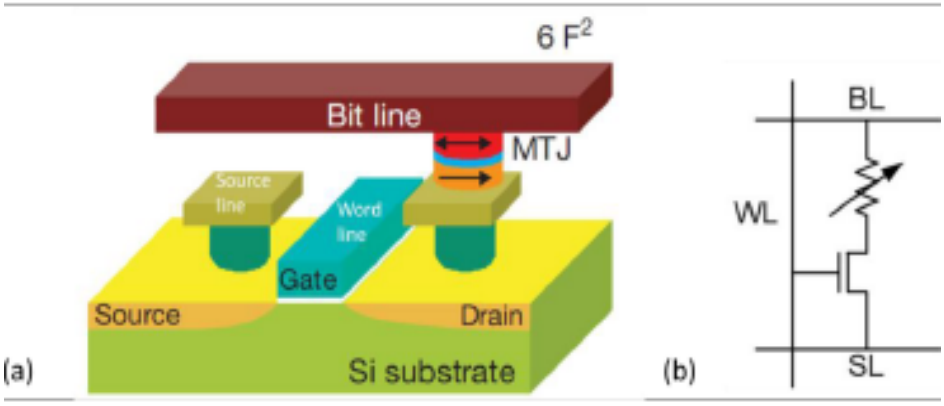


Figure 2, (a)basic structure of a STTRAM cell, (b)the equivalent schematic diagram [12]

As shown in figure 3, there are two kinds of MTJ, In-plane MTJ (IMTJ) where magnetization of ferromagnetic layers lies in the film plane, and perpendicular MTJ (PMTJ) where magnetization direction is perpendicular to the film plane. Recently, PMTJs are being vigorously developed [11].

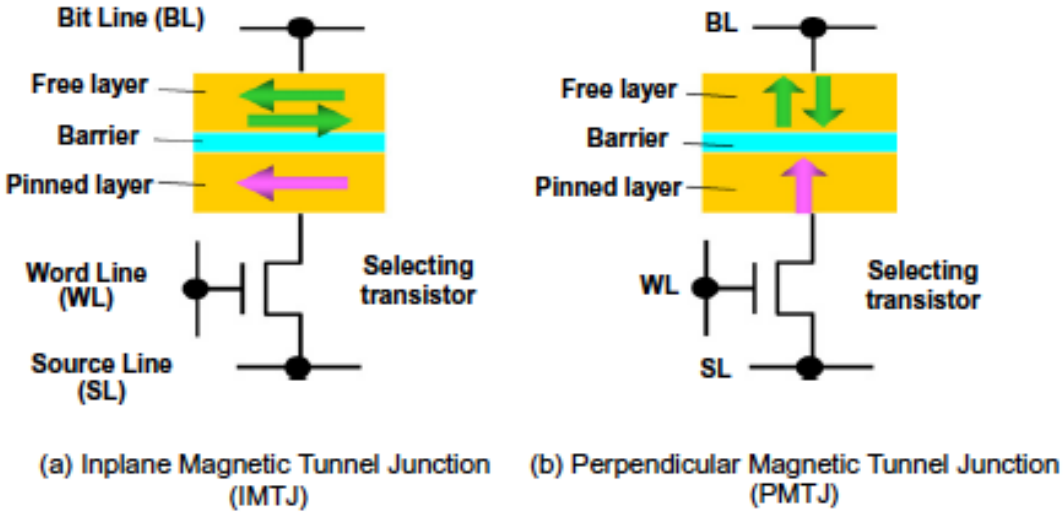


Figure 3, Memory cell structure with MTJs [11]

2.2. MTJ and TMR

As shown in Figure 4a an MTJ has two ferromagnetic layers (FL) and one oxide barrier layer (BL). The resistance of MTJ depends on the relative magnetization direction (MDs) of the two FLs. The mechanism of fetching and storing the data in this type of memory in order to do the reading and writing purposes, are based on magnetic fields of MTJ whether they are parallel or anti-parallel and as shown as Figure 4b two states of parallel and anti-parallel are defined by charge current direction, which is induced through STT switching. Figure 5 shows scanning electron microscope (SEM) photo of an MTJ and figure 5 shows the sketch of basic MTJ structure [2].

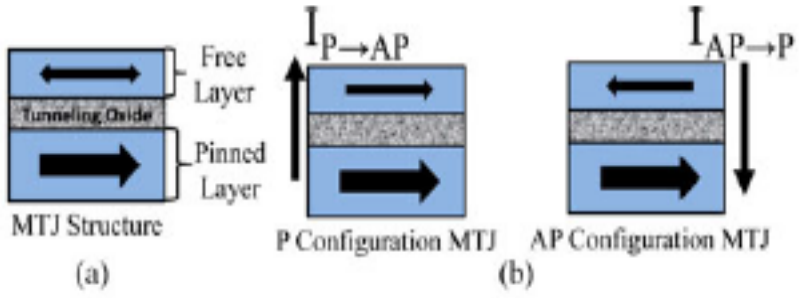


Figure 4, (a) Structure of a magnetic tunnel junction. (b) Charge current directions to induce spin-transfer torque switching [10]

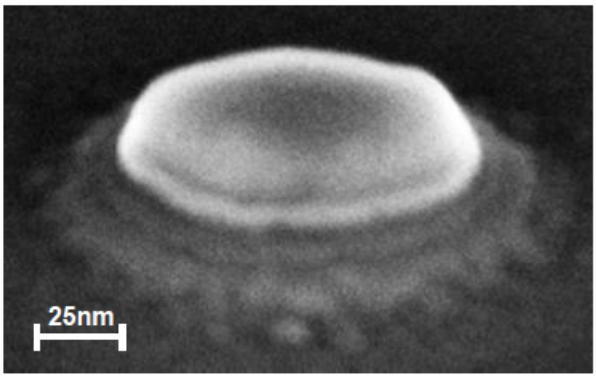


Figure 5, SEM photo of an MTJ [7]

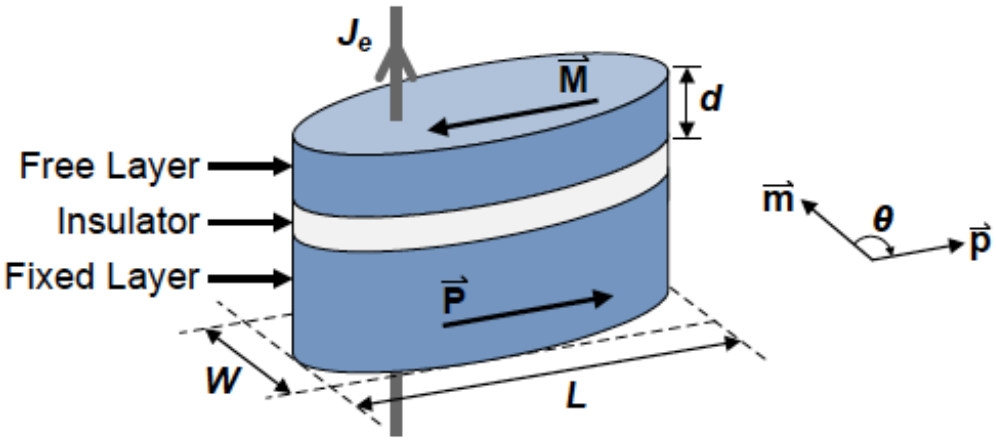


Figure 6, the sketch of basic MTJ structure [7]

As shown in figure 6, M is the magnetization of the free layer, P is the magnetization of the pinned layer or fixed layer, Je is the current density and θ is the angle between M and P.

The resistance of MTJ depends on the relative orientation of magnetization in two ferromagnetic layers. This resistance change is called the tunnel magneto-resistance (TMR) ratio, which is defined by

$$TMR = \Delta R/R = ((RAP - RP) / RP) \text{ X } 100$$

Where RAP and RP are the resistances for anti parallel (AP) and parallel (P) magnetization configuration between the two ferromagnetic films, respectively, RAP is

considered as high resistance (RH) and RP is considered as low resistance (RL) of MTJ. Normally, the magnetization direction of one ferromagnetic layer, called a pinned layer, is fixed by exchange interaction with an adjacent anti-ferromagnetic (AFM) layer, called a free layer, can freely rotate [11].

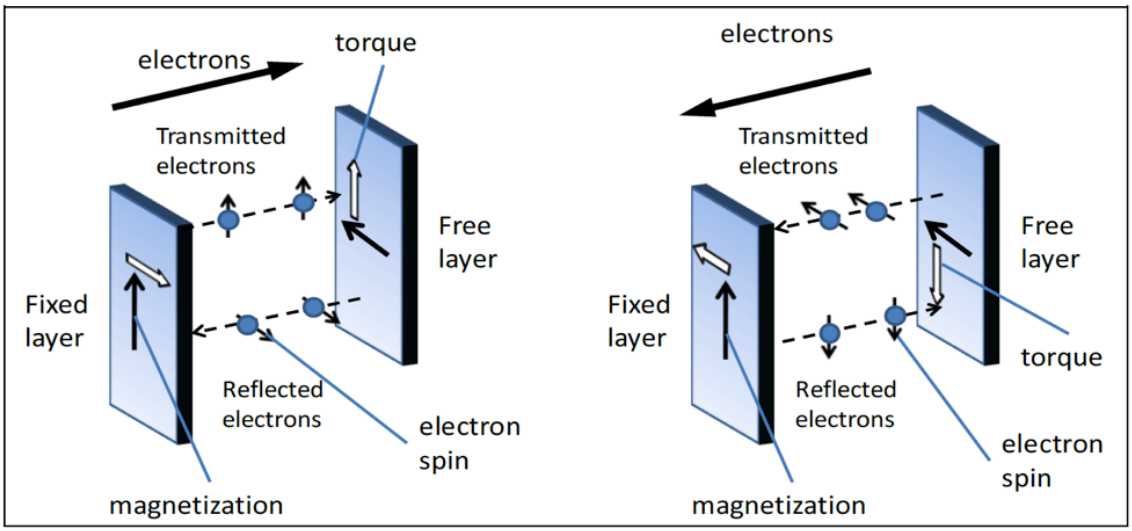


Figure 7, the direction of magnetization and spins of electron [12]

As figure 7 shows, when current flows from the free layer to fixed layer, the s-band electrons will be spin-polarized in the direction of magnetization of the fixed layer. This is the first spin filtering. The second spin filtering, when the electrons reach the free layer, s-d exchange interaction occurs. The electrons will align themselves along the magnetization of the free layer. So the spin will start to process around the magnetization of free layer [12].

Since the precession is averaged over all electrons, transverse components of spin angular momentum become zero since the electrons are out of phase. Due to conservation of spin angular momentum, the transverse components of the electron spins will be absorbed and transferred to magnetization of free layer. Therefore, the same interaction also exerts a torque on the magnetization of the fixed layer. This torque effect is commonly known as spin transfer torque (STT) [12].

Although the minority-spin electrons, with respect to the free layer, will be reflected back to the fixed layer, the magnetization of the fixed layer will not change because this torque is not strong enough [12].

Similar situation happens when the electrons move from the free layer with one exception. The torque exerted by the electrons that process around the magnetization of the fixed layer are insufficient to switch the magnetization. The minority-spin electrons, with respect to the fixed layer, are reflected back to the free layer. These electrons apply torque that enough to switch the magnetization of the free layer antiparallel to the fixed layer. The strength of the torque is normally expressed as the magnitude of current density [12].

2.3. Read and Write Operation of Memory

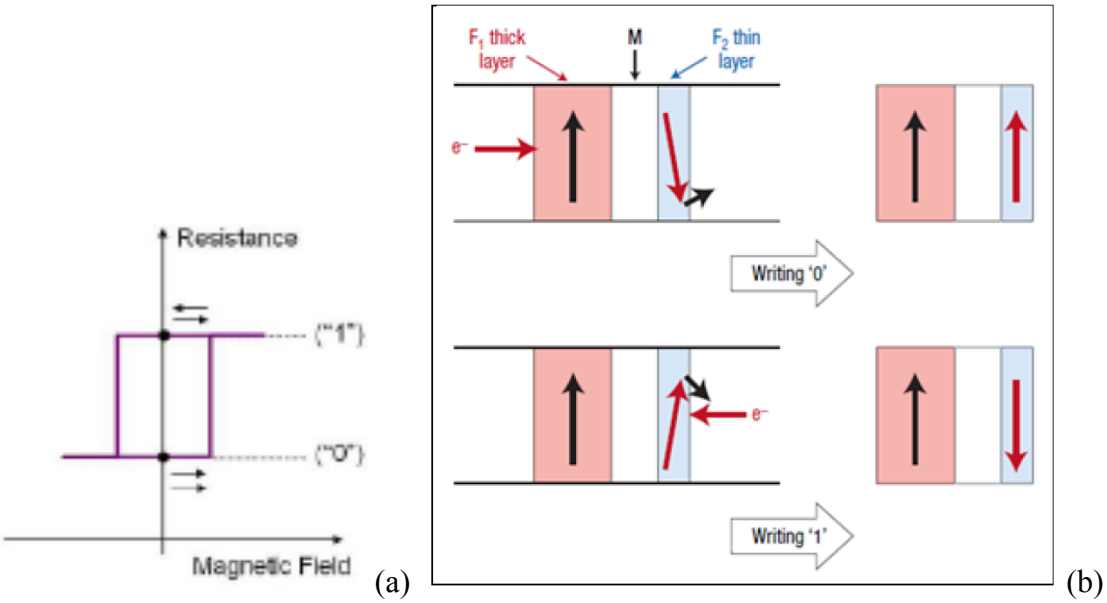


Figure 8, MTJ functions that cause STTRAM reading and writing operations [12]

In figure 8 based on the magnetization fields, read and write operations are shown. So in (a), the "1" situation is anti parallel and has higher resistance with higher voltage output. Whereas the "0" situation is parallel and has lower resistance and therefore lower voltage output.

And for reading operation, we follow this process:

Read operation => word line is selected => a voltage is applied to the bit line => current density of magnitude is less than the switching current

In side (b) of the figure 8, write operation 0 and 1 are shown as this process:

Write operation => a bit is selected by selecting the word line => either the bit line or the source line of a selected column is positively biased =>

Current density larger has critical density.

Writing "0" is AP => P and writing "1" is P => AP and this is the worst case, so the current in this situation is critical current (figure 9) [13].

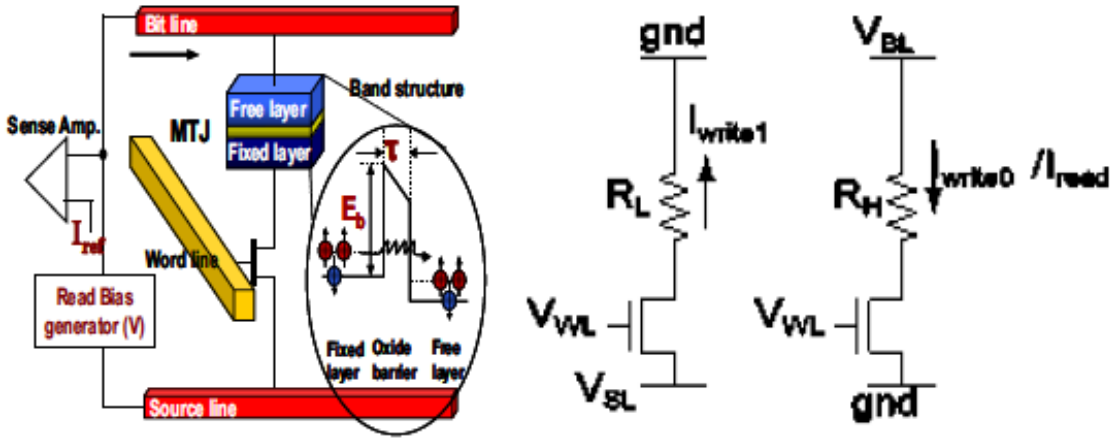


Figure 9, STTRAM read and write operations [13]

Based on our model of MTJ from [14], we simulate one cell of STTRAM and the wave view of result is shown in figure 10, which clearly shows different read 0, read1, write 0, write 1 operations. It is obvious that the writing 1 operation has high power consumption because of such a big critical current (347uA). But hybrid CMOS-STTRAM FPGA solutions are particularly attractive because the write operation, which is a high power operation in STTRAM, happens very infrequently in FPGA [17].

Tunnel Magneto Resistance (TMR)

- Effective for read operation
- The resistance of MTJ is determined by magnetic orientation
- Higher TMR is preferred for a reliable read operation
- Generate larger signal difference between two states

$$TMR = \frac{RH-RL}{RL} * 100$$

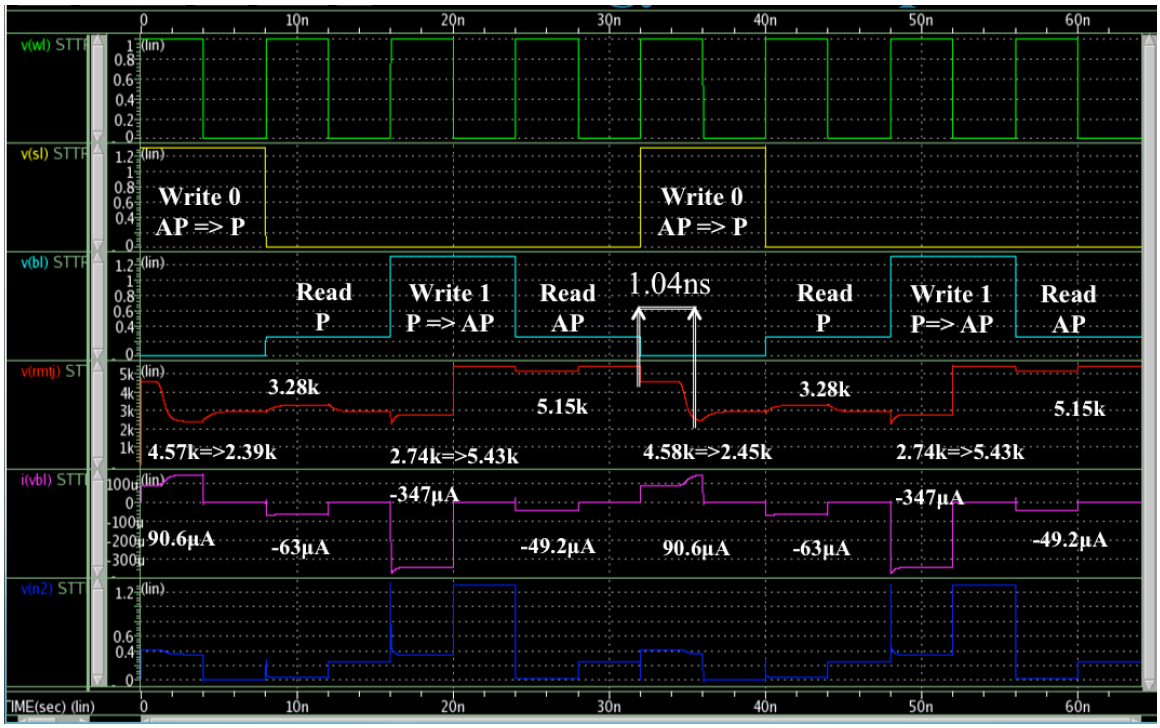


Figure 10, wave view result of simulation of one cell of STTRAM

2.4. STT-LUT and its application

The use of MTJs has recently been explored for building low power programmable Look-Up Tables (LUT) used in Field Programmable Gate Arrays (FPGA) [17], [18]. In both memory and FPGA applications, re-configurability is a key requirement and exploits the programmability of MTJs [9].

There have been attempts to use MTJs for building logic circuits with the hope of exploiting the leakage benefit of MTJs in order to reduce circuit power [9]. However, due to the significant energy involved in changing the state of an MTJ, circuit styles that rely on changing the state in response to input changes do not show any power and performance benefit [15]. An alternative to this approach has been to realize logic in memory by using LUTs that are built based on MTJs [16]. A LUT, such as those used in FPGAs, offers programmability and includes write circuitry for changing the state of the MTJs [9]. However, if the LUT is used for implementing fixed combinational logic, there is no need for the write circuitry, so it can be eliminated to simplify the circuit [9]. In [16], such read-only MTJ-based LUTs are used to replace custom CMOS logic with the hope of achieving low power and delay.

Therefore by programming the content of LUT we can implement any arbitrary logic function like NAND, NOR, etc. or any combination of logic gates that have any truth table.

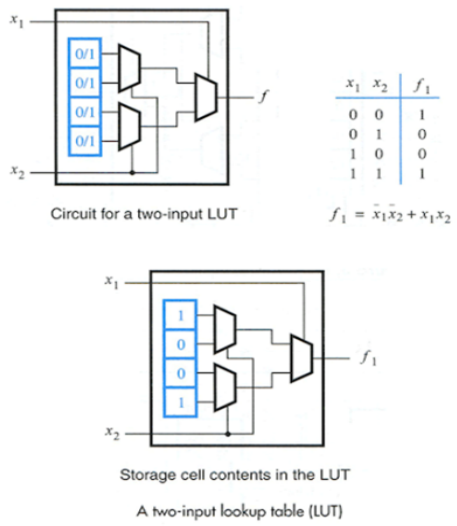


Figure 11, implementation of XNOR gate with LUT

For example in Figure 11, an implementation of XNOR gate is shown, it is obvious that for execution function like 2 input XNOR, we need to program 4 MTJ of STTRAM cells for storing 4 different state of XNOR gate and two inputs are our address data for selecting different logic states. For programing such these memory bits we need external write circuit to enable writing to memory cells.

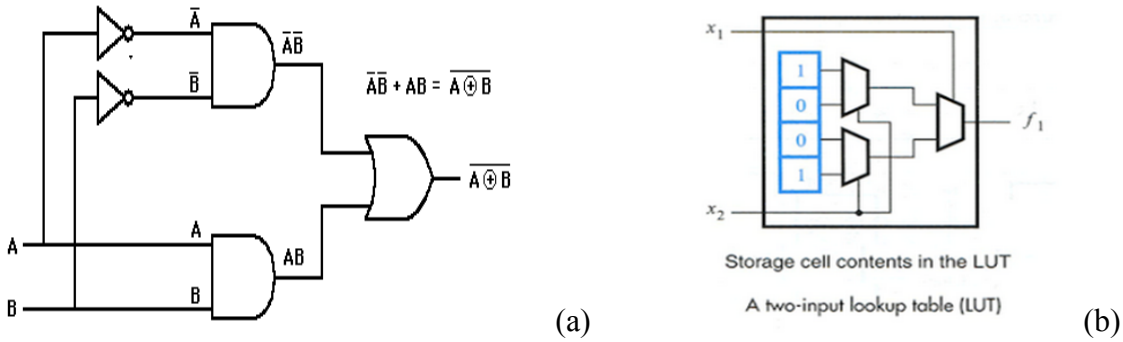


Figure 12, (a) Conventional Design of XNOR which is non secure whereas (b) LUT Based Design that can implement XNOR logic which is Secure design

Based on figure 12, this is the comparison between complimentary gate design and new LUT based design, so we can conveniently replace new LUT instead of old gate and claim that, now we have secure design and no one can get access to understand which gate it is.

LUT using STTRAM technology is illustrated as Figure 13.

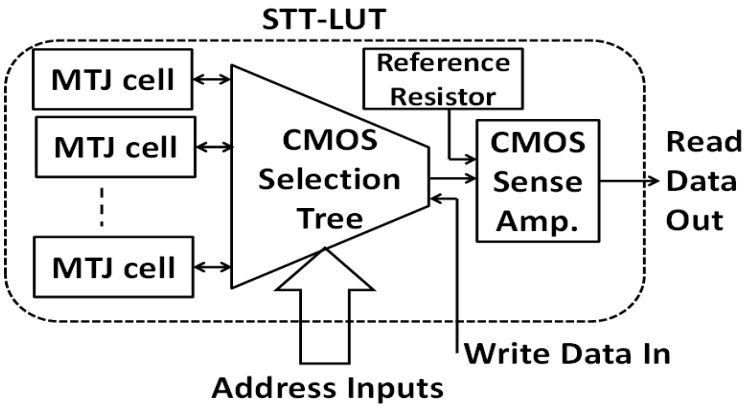


Figure 13, LUT design, general concept

MTJ cells are storage of LUT, which can be programmable via write data inputs circuit. Based on data address inputs, one of the STTRAM branches is selected. Then Sense Amplifier (SA) compares the resistor of selected MTJ with reference resistor to define that whether is high resistance (RH) or low resistance (RL) and converts RH and RL to actual high and low voltages.

Chapter 3

3. Implementation

3.1. STT-LUT Types

Based on our research there are different design styles of Look-Up-Table, a current sensing scheme based STT-LUT (CSM STT-LUT) and a voltage sensing scheme based STT-LUT (VSM STT-LUT), and come up with a conclusion as to which is the best design, in terms of power, delay and PDP (power delay product). And then we are comparing the selected LUT with complimentary CMOS in 16 nm CMOS technology to achieve our goal, which is using this technology in enhancing the hardware security.

Voltage sensing mode STT-LUT (VSM STT-LUT)

In the voltage-sensing mode LUT design Figure 14, we make use of a voltage sensing amplifier and resistors in conventional CMOS logic. A particular resistance is selected through the NMOS pull down selection tree, and the dynamic source current is divided across the selected resistance and the reference resistor which results in a low swing differential voltage across the nodes DEC and REF when the clock (CLK) is high [9]. When sufficient voltage difference in generated across the two sides of the sense amplifier, it turns on and latches onto the value stored in the LUT. The low signal voltage gets amplified across the sense amplifier to full swing outputs of Z and Z' [9].

Current Sensing mode STT-LUT (CSM STT-LUT)

In this LUT design, we make use of current sensing amplifier along with the resistors Figure 15. In this design the sense amplifier demonstrates high speed during reading. The sensing begins when the clock (CLK) switches which places the sense amplifier in the metastable state [19]. Sense amplifier is made up of two cross-coupled inverters, which contain PMOS and NMOS [19]. The resistance that is present modulates the pull down transistors when the CLK signal is low [19]. Then the amplifier reaches one of its two stable states depending on the resistance difference between the LUT block and the reference block [19]. The amplifier stores the result in the positive half cycle [19]. The sensing in this sense amplifier to be faster when compared to voltage sensing mode [19]. In this scheme the CLK high period is 25% of the time period, which is sufficient for sensing [19].

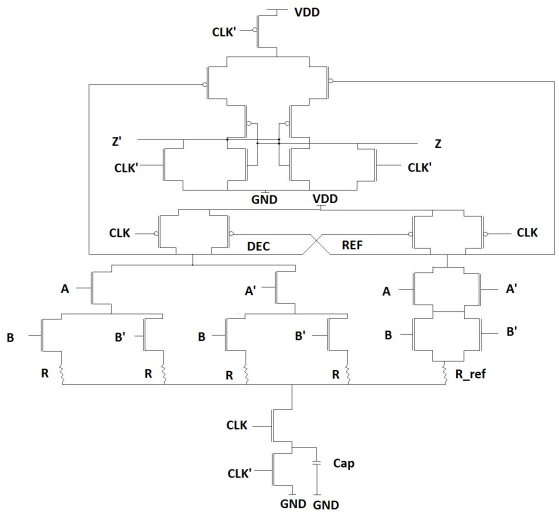


Figure 14, Voltage sensing mode based STT-LUT (VSM STT-LUT) [9] [16]

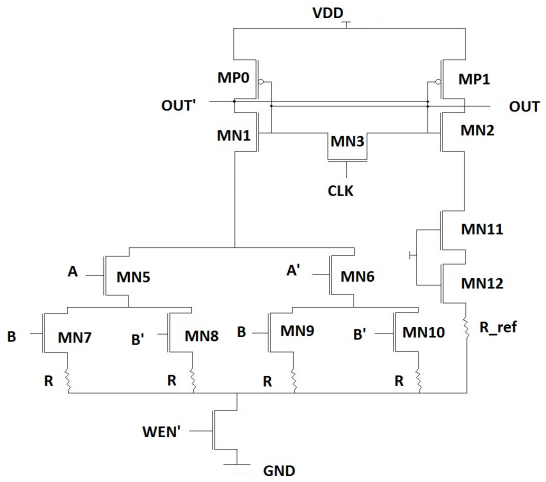


Figure 15, Current sensing mode STT-LUT (CSM STT-LUT) [19]

Based on our design simulation with H-spice, our results are shown on table 1, and we use PTM library for our CMOS technology [20]. We swept the reference resistor and capacitor with a Perl scripting language in a defined range of RH to RL, to gain the min power delay product (PDP) and then choose that RRef and Cap. RH and RL are defined by designer for programming the MTJs in circuit and tested to find out the RH and RL failure values. The other metrics are the result of our simulation with Perl scripting language.

	VSM STT-LUT	CSM STT-LUT
R Ref	6k	11k
Cap	3femto	-
RH	10k	10k
RL	4k	4k
Delay high-low	TPZ=126.41ps	TPH=161.54ps
Delay low-high	TPZBAR=121.01ps	TPL=120.56ps
Power	581.2556nw	666.0159nw
Standby Power	221.1656nw	13.6548nw
PDP	73477.1016 e-21 J	107592 e-21J
RH fail	7k	5k
RL fail	6.9k	5k
Delta_R_margin	2.9k	1k

Table 1, comparison between CSM STT-LUT and VSM STT-LUT

It is obvious from the results that voltage sensing based LUT has lower delay and active power in compare to current sensing based STT-LUT, however the standby power of the CSM STT-LUT is low compared to VSM STT-LUT because the number of PMOS and hence leakage is lower. So we conclude that VSM STT-LUT is a better candidate design in terms of low power and more performance.

3.2. STT-LUT Design

Figure 16 shows the voltage sensing STT-LUT in detailed transistor level schematic with Write circuit.

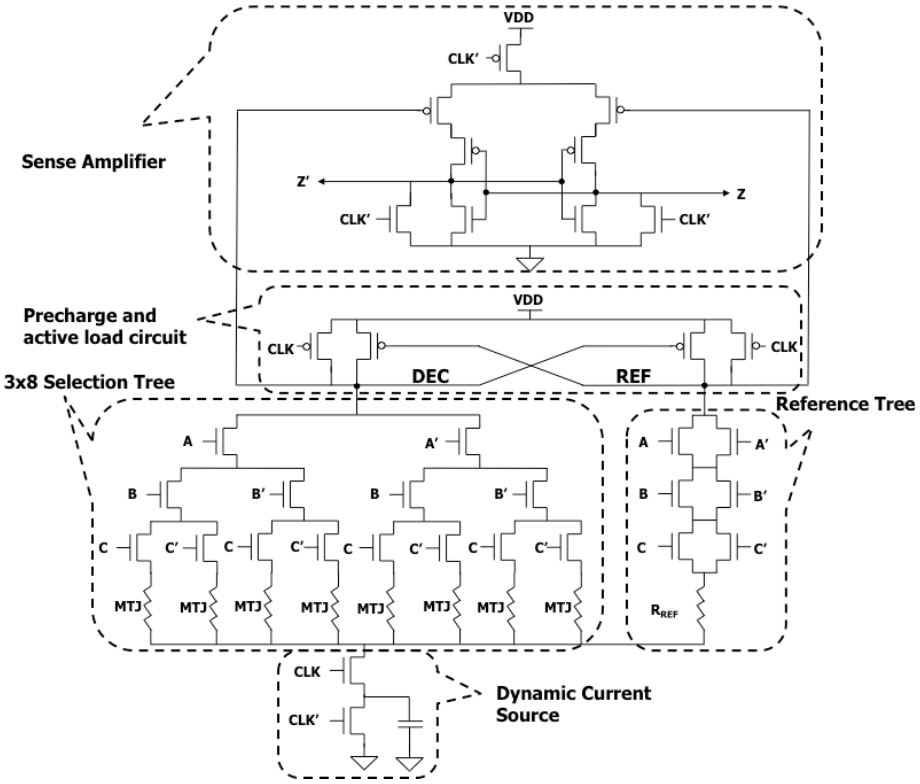


Figure 16, three inputs MTJ based STT-LUT

Simulating such a this circuit is possible in HSPICE tool, all the efforts is done to optimize design in terms of lower power, lower power and therefore minimum power delay product (PDP), the specification of the design is as follows and the result sample of LUT 2 inputs wave view that has the function of NAND gate is shown in Figure 17;

Length of transistor is the technology that we are using in our design=> L=16nm

Width of NMOS transistor => Wnmos=30nm

Width of PMOS transistor => Wpmos=60nm

VDD = 0.7v

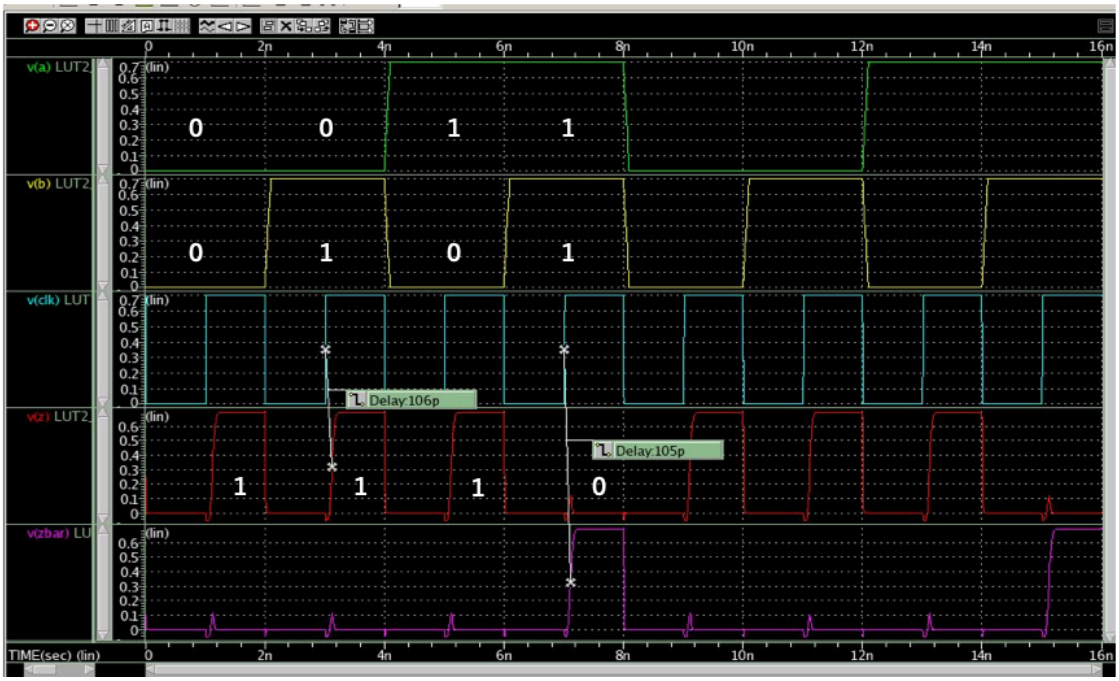
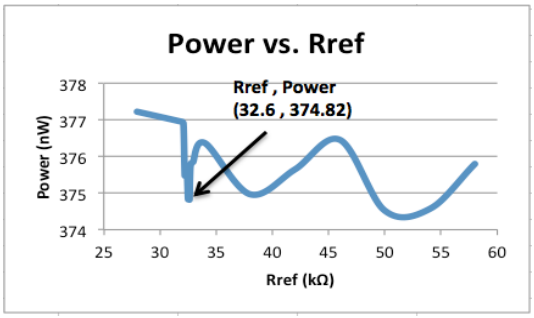
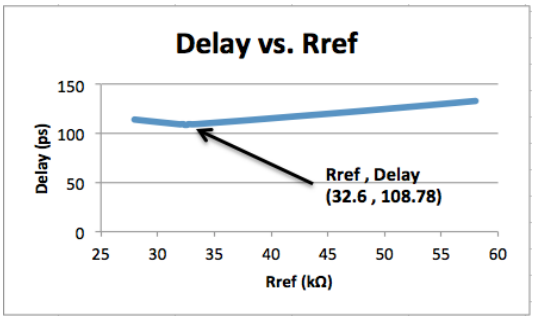


Figure 17, wave view result of HSPICE simulation of LUT that has the functionality on NAND gate

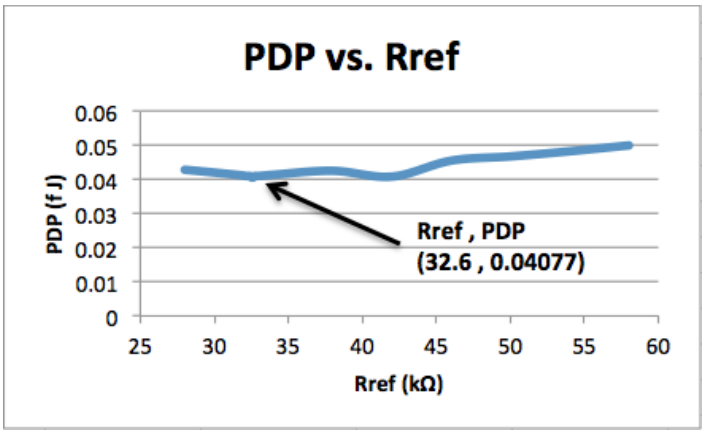
Reference resistor value is the average between high resistance (RH) and low resistance (RL), but the average resistance is not the optimized number to have minimum power, delay and also PDP. Therefore we need to repeat simulation many times for find the minimum power, delay and PDP. So now we feel a need to write a script for sweeping different number of reference resistors and reference capacitors. Figure 18 is showing graphs of power, delay and PDP for LUT 2 inputs as a result of running PERL script for sweeping Rref with fixed number of reference capacitor as C=3 (Femto Farad) and the RH=100k, RL=20k and PDP=400%.



(a)



(b)



(c)

Figure 18, results of power, delay and PDP for different Rref as running PERL

We find out from these graphs that the best Rref for minimized the PDP is Rref=32.6k. We can do this for all different fan-in LUT. Another benefit that we can get from tuning reference resistor, is having almost same low to high delay from 50% of clock signal to 50% of output z and 50% of clock signal to 50% of output zbar.

3.3. Investigating the Impact of TMR on Power and Delay of STT-LUT

Upon completing the necessary simulations under Hspice software, plots were made to observe how modifying TMR values affect the overall power and delay of the circuit. So efforts are made to investigate the trend of impact of changing TMR on power and delay of STT-LUT. For simplifying the simulation results we keep Reference resistor as average of RH and RL. Figure 19, 20 and 21 are the graphs that shows the data collected of Power vs. TMR, Delay vs. TMR and Power Delay Product (PDP) vs. TMR.

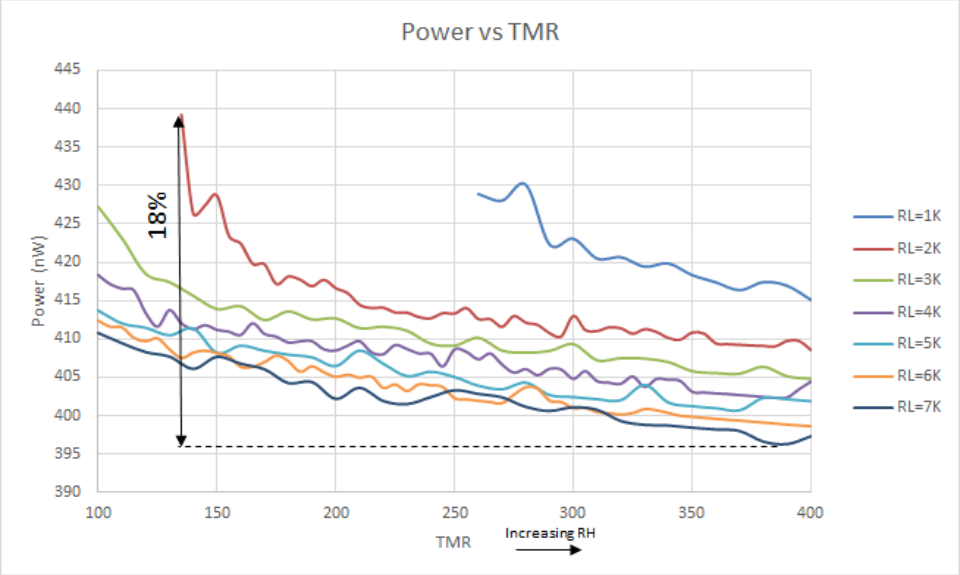


Figure 19, Observing power dissipated in the circuit as a function of increasing TMR for constant low resistance (RL)

According to initial expectation, as TMR increases, the overall power consumption by the overall circuit decreases. The reason can be explained by simply using Ohm's law (R=V/I). The resistance is inversely proportional to the current; as the resistance increases the current decreases. In turn, as current decreases the power also decreases.

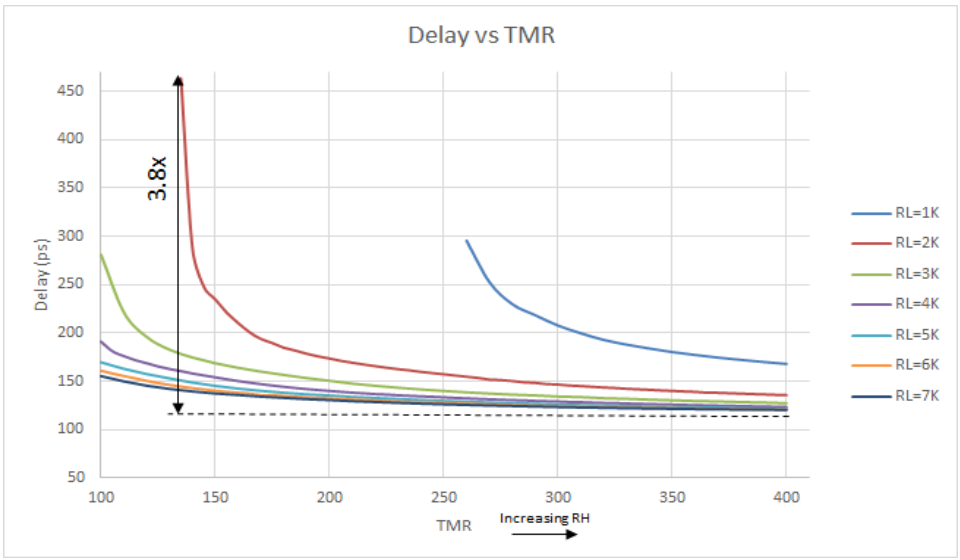


Figure 20, observing the delay as a function of increasing TMR for constant low resistance (RL)

In observing the delay, as the TMR increases the delay also decreases but then remains as the same level. As TMR increases, it yields a larger difference in high and low resistances (RH-RL), which results in a larger voltage across the MTJ cell. In other word, the voltage in proportional to resistance difference, ($\Delta V \alpha \Delta R$ => RH-Rref = RH-((RH+RL)/2 => ΔR=(RH-RL)/2 = (TMR*RL)/2), therefore when the TMR increases voltage difference in going to increase. But the voltage difference cannot exceed the supply voltage (0.7v) and reaches a saturation point. This saturation point limits the amount of current that flows through the MTJ and in turn limits the delay between clock and output.

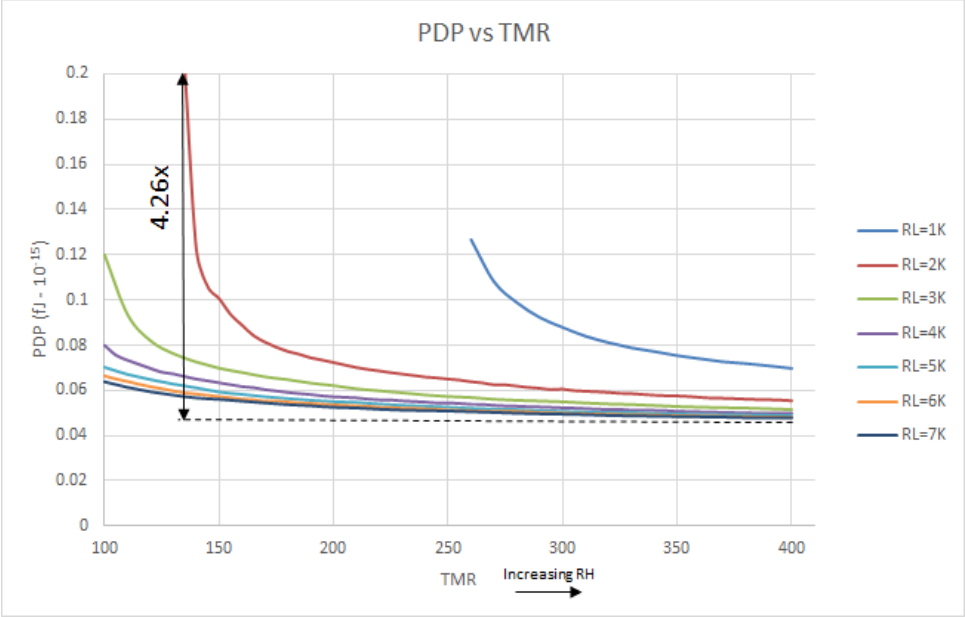


Figure 21, observing Power Delay Product (PDP) as a function of increasing TMR for constant low resistance (RL)

Because of delay trend, as TMR increases the power delay product decreases until it reaches the saturation point. Based on these results, we reached the conclusion that it would be best to have a TMR at the highest possible value, which results in lower power and delay. However, there are limitations related to manufacturing processes, therefore it would not be possible to reach very high TMR values.

We did lots of reviewing different papers to do feasibility study and find reasonable range of published TMR that is possible to manufacture. Figure 22 is shown the published TMR values based of different technical papers in almost 15 years.

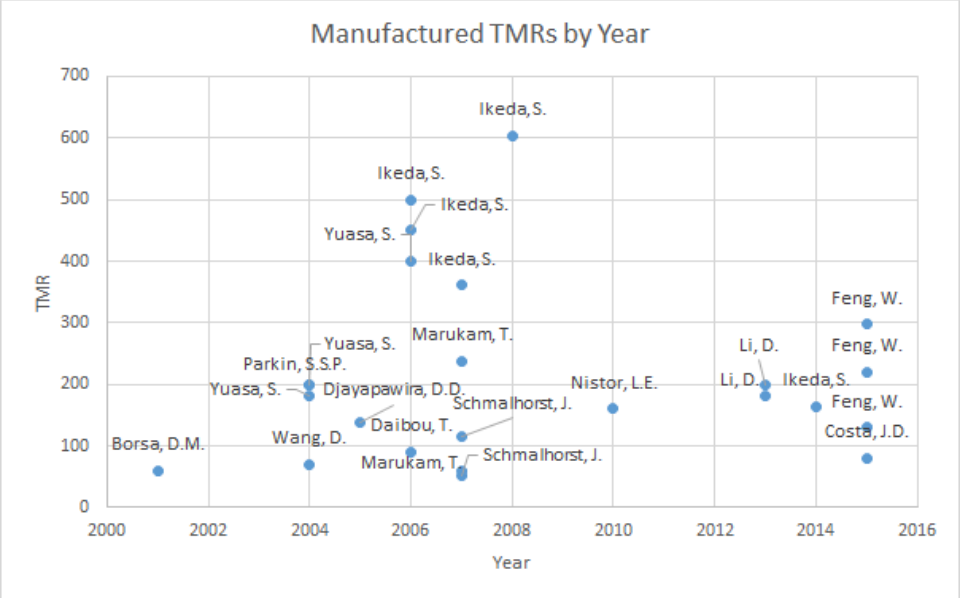


Figure 22, Manufactured MTJ during years and their TMR value

Based on published TMR among 15 years, the TMR of 600 is the highest value reported. Due to modified variables of MTJ (TMR, write current, reliability) there is no increasing trend. And high TMR applicable, reconfiguration rarely occurs.

Based on our result, we have chosen the most feasible TMR value of 400% from the results mentioned above, reference resistor (Rref) is optimized to find minimum power delay product as shown in the graph below. By using Perl scripting language, the value of 32.6(kΩ) was found to be the most effective reference resistor (Rref). Now we are ready to characterize different fan-in STT-LUT with the specification that we define for optimized STT-LUT design.

3.4. Characterizing the Power and Delay Overhead of STT-LUT in Compare to Custom CMOS

After finding the feasible and optimized TMR of 400% and RH=100k and RL=20k, we begin to optimize the reference resistor and reference capacitor for different fan-in STT-LUT, and measure the active power, standby power, max low to high delay between clock to z and clock to zbar. Then after that we can compare our result with complimentary CMOS. Table 2 is showing the result of STT-LUT 2 inputs to 8 inputs and table 3 is showing result of Custom CMOS 2 input to 4 inputs. Then in oreder to compare these numbers we can draw graphs.

	RH	RL	TMR %	Rref (k)	cap ref (f)	Power (nw)	tpz delay (ps)	tpzbar delay (ps)	PDP (fj)	Stand by power (nw)
LUT2	100	20	400	32.6	3	378.4	108.76	108.78	0.407	4.37
LUT3	100	20	400	30.6	3	337.3	178.41	178.4	0.601	3.97
LUT4	100	20	400	28.5	3	328.5	251.05	251.3	0.825	3.76
LUT5	100	20	400	64.7	5	443.4	250.4	200.9	0.111	3.66
LUT6	100	20	400	62.9	5	437.8	316.9	233.1	0.138	3.68
LUT7	100	20	400	61	7	500.6	293.2	208.6	0.146	3.87
LUT8	100	20	400	55	9	553.6	303.8	191.2	0.168	4.34

Table 2, result of optimized STT-LUT from 2 inputs to 8 inputs

Activity of logic gates is total number times output of gate switches divide by number of clock cycle observed. Since the equation of power is $P= \alpha.f.c.v^2$. Therefore power of custom CMOS gate is dependent to activity of them in the circuit, but delay of logic gate is not dependent to activity. Furthermore, LUT is also independent of activity, because it is a dynamic design and LUT operates every cycle since it is clocked and therefore it doesn't matter if input switches or not, output is switching every cycle. So table 3 is showing different custom CMOS gates with their delay result and power result for different activity.

	Delay (ps)	Power (nw)	Power (nw)	Power (nw)	Power (nw)
		α=100%	α=50%	α=30%	α=10%
Inv	3.41	164.8	130.9	102.1	68.04
NAND2	10.44	262.3	203.05	145.3	66.8
NOR2	8.1	265.2	182.2	163.4	118.2
XOR2	13.08	472.7	390.7	371.5	304.8
AND2	17.3	497.2	367.3	305.6	208.4
NAND3	14.3	424.8	319.2	204.9	67.6
NOR3	10.58	471.2	289.8	224.4	93.7
AND3	24.8	755.9	553.3	431.4	218.7
NAND4	22.9	568.6	417.7	262.6	74.1
NOR4	41.3	1120.6	737.3	623.5	394.7
AND4	28.07	885.2	607.39	432.7	160.4

Table 3, Custom CMOS results in term of delay and power based on different activity of them (10%, 30%, 50%, 100%)

When the activity of gate is 100%, it means that output is switching at the clock rate. When the activity of gate is 50%, it means that output is switching every 2 cycle of clock. When the activity of gate is 30%, it means that output is switching every 3 cycle of clock or in other word, after 2 clock cycle, output is switching at the third clock cycle. And finally when the activity of gate is 10%, it means that output is switching every 10 cycle of clock, or after 9 cycles then output is going to switch in the 10^{th} cycle.

Figure 23 is showing the comparison between LUTs and Custom CMOS gates in terms of power that in case of Custom CMOS is dependent to activity. And Figure 24 is a bar chart that shows comparison between LUT and Custom CMOS gates in terms of maximum low to high delay.

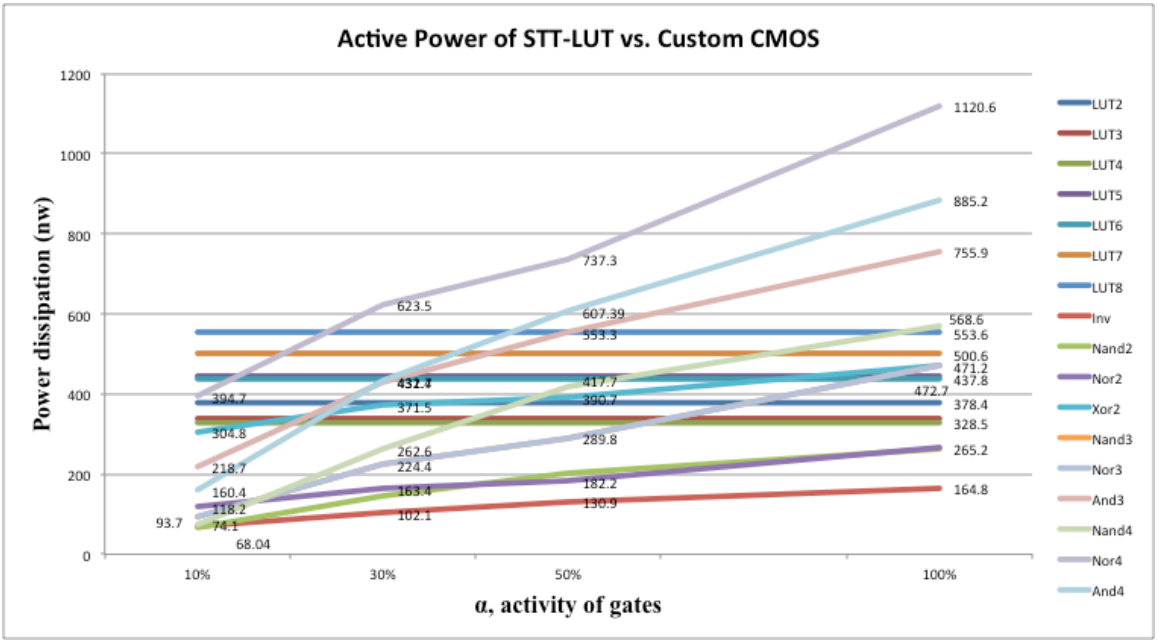


Figure 23, Active power comparison between STT-LUT and Custom CMOS gates

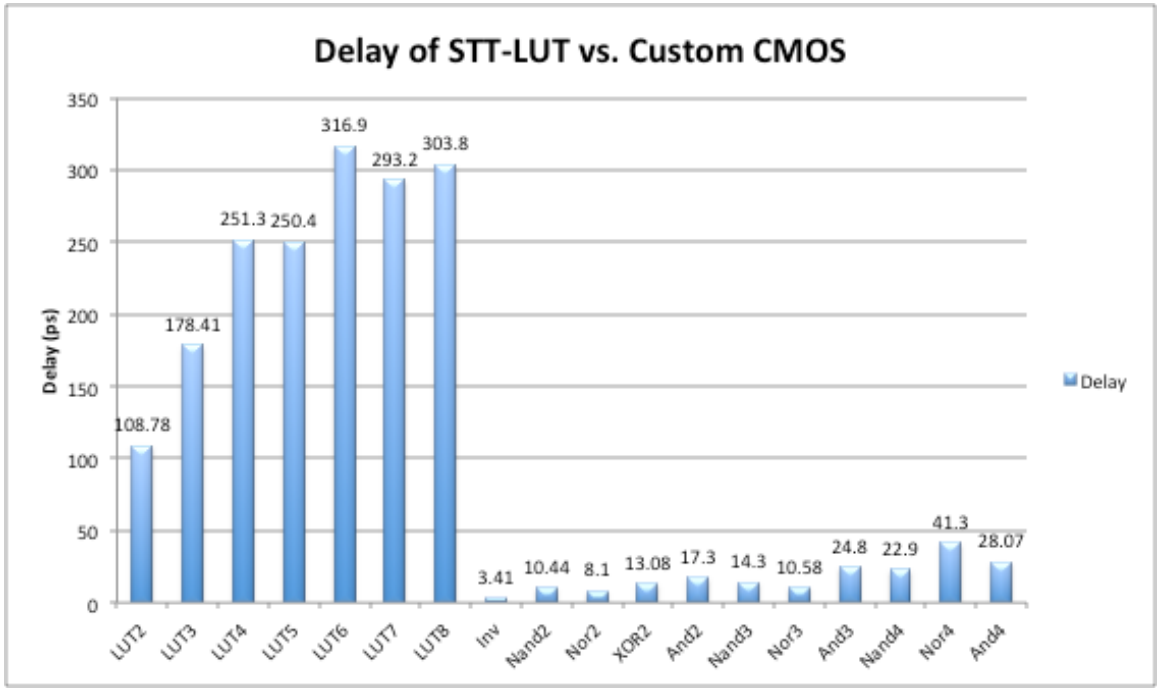


Figure 24, delay comparison between STT-LUT and Custom CMOS

By studying and analyzing of these charts, we understand that our STT-LUT design is kind of competitive with Custom CMOS especially in high fan-in gates. But in terms of delay we are facing to too much difference when comparing STT-LUT and Custom CMOS. In result we have much more delay overhead and still some power overhead and our challenge in now how we can lower this much of overhead in order to be able to replace STT-LUT instead of custom CMOS logic gates.

3.5. Reducing power and delay overhead with proposing idea of Collapsing

Since we are facing power and delay overhead when we compare STT-LUT and custom CMOS, we have to lower as much as possible this overhead for the purpose of replacing STTLUT instead of one CMOS gate or combination of gates, because we are using STT-LUT technology for implementing reconfigurable block for security issue.

We are proposing that if we combine some logic gates that this subset of gates follows constraints, then use STTLUT instead of it, we get reasonable result of power and delay of the circuit. We call this idea, The Idea of Collapsing. And we are trying to prove that by collapsing number of gates that has single output and no internal node as output, we can lower power and delay overhead.

Therefore possible-collapsing options of circuit are grouping on logic gates that their internal nodes are not used, as other input of gates and it should have single output. Then this possible option of collapsing can be replaced by STT-LUT and the number of inputs of the subset of gates is number of inputs of STT-LUT.

After finding all possible collapsing options, then we can define many approaches. One of the approaches we can use always is replacing all individual gates with same fan-in LUT. Another one is replacing all logic gates with one LUT. We can claim that this approach is the most secure on. Then based on our collapsing options other approaches will be defined. Then we need to do analysis to get the power and delay of the circuit.

In general, we can divide types of analysis into two category, Analytical analysis and Numerical analysis. In analytical analysis we measure everything manually with the result of power and delay of LUT and Custom CMOS gates. The most important part that it should be considered carefully is the measuring of activity of Custom CMOS gates because in terms of power CMOS gates are dependent to activity. Therefore for analytical analysis of CMOS we need to measure the probability of switching of each gate from 0 to 1 and 1 to 0 and then calculate the overall power of circuit based on activity of each gate. And for delay of Custom CMOS we only need to add delay of each COMS gate in critical path. Analytical analysis is easy for LUT because delay and power of LUT are independent to the functionality of the gate. So for measuring overall circuit power of LUTs we only need to add all the amount of power for different subset after collapsing. And for circuit delay we need to add delay of collapsing LUT in critical path.

Whereas in Numerical analysis, in order to find activity of CMOS gate simulation is needed. We should setup environment for finding activity in custom CMOS by

generating random binary inputs, then simulate the Verilog or System Verilog design with VCS and then analyze the output data with the help of script to find how many time output changes its state from 0 to 1 or reverse.

For better understanding of how to measure activity of gate and probability of output switching from 0 to 1 or reverse, we need to review these concepts from one of our reference books, Digital Integrated circuit, Jan M. Rabaey [21].

The sources of power consumption in a complementary CMOS are problematic. Many of these issues apply directly to complex CMOS gates. The power dissipation is a strong function of transistor sizing (which affects physical capacitance), input and output rise/fall times (which affects the short-circuit power), device thresholds and temperature (which affect leakage power), and switching activity. The dynamic power dissipation is given by α (0=>1)*CL*VDD^2*f. Making a gate more complex mostly affects the switching activity α (0=>1), which has two components: a static component that is only a function of the topology of the logic network, and a dynamic one that results from the timing behavior of the circuit—the latter factor is also called glitching.

Logic Function— The transition activity is a strong function of the logic function being implemented. For static CMOS gates with statistically independent inputs, the static transition probability is the probability p_0 that the output will be in the zero state in one cycle, multiplied by the probability p_1 that the output will be in the one state in the next cycle:

$$\alpha\,(0 => 1) = P_0 * P_1 = P_0 * (1 - P_0)$$

Assuming that the inputs are independent and uniformly distributed, any N-input static gate has a transition probability that corresponds to

$$\alpha\,(0 => 1) = \frac{N_0}{2^N} * \frac{N_1}{2^N} = \frac{N_0 * (2^N - N_0)}{2^{2N}}$$

Where N0 is the number of zero entries and N_1 is the number of one entry in truth table. And the total activity is $\alpha = \alpha\,(0 => 1) * \alpha\,(1 => 0) = 2\,\alpha\,(0 => 1)$.

For clarifying collapsing idea and its analytical analysis we want to do an example as follows:

Figure 25 is the sample circuit for implementing collapsing idea and measures the overall power and delay of critical path.

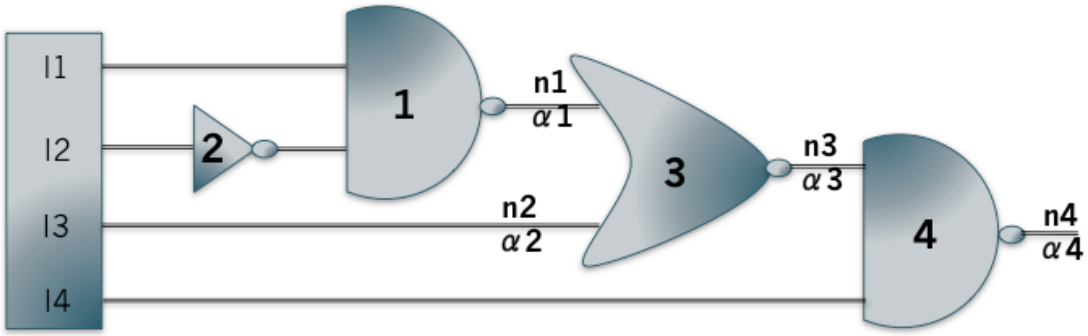


Figure 25, Sample circuit for implementing collapsing idea

Custom CMOS: Based on the function of the gate we can analyze that how is the probability of switching and therefore the activity of gate in circuit

1: NAND2: α1 = 0.3

2: INV: α2 = 0.5 P = 470.51nw

3: NOR: α3 = 0.3 Tp = 21.03ps

4: NAND2: α4 = 0.1

One example of analytical analysis:

Since n1 is the output of NAND gate then the probability of output to be 0 is ¼ and the probability of output to be 1 in ¾.

α1 (0 => 1) = P (n1 = 0) * P (n1 = 1) = ¼ * ¾ = 3/16

α1 (1 => 0) = P (n1 = 1) * P (n1 = 0) = ¾ * ¼ = 3/16

α (n1) = 2 * α1 (0 => 1) = 3/8 ~ 0.3 => α (n1) = 30%

α3 (0 => 1) = P (n3 = 0) * P (n3 = 1) => α (n3) = 2 * α3 (0 => 1)

α (n3) = P (n3 = 0) * P (n3 = 1) * 2

α (n3) = P (n3 = 1) * (1 - P (n3 = 1)) * 2

P (n3 = 1) = P (n2 = 0) * P (n1 = 0) = ½ * ¼ = 1/8

α (n3) = 1/8 * 7/8 * 2 = 0.21 ~ 0.3 α (n3) = 30%

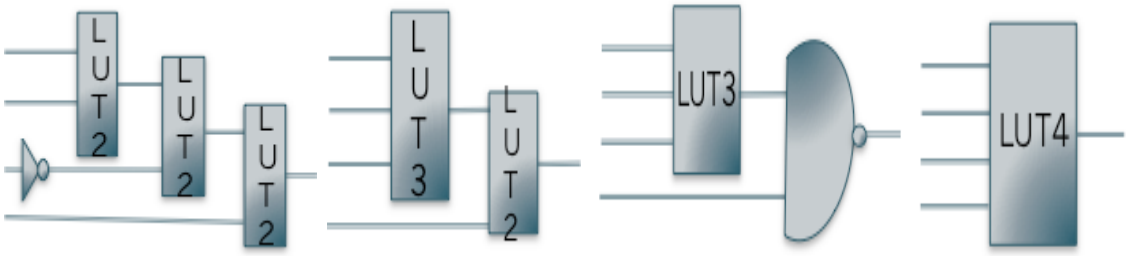


Figure 26, different approaches of replacing LUT with CMOS gates

(a)

Approach1,

Replacing all the gates with relative LUT

(b)

Approach 2:
Replacing NAND-1, INV-2 and NOR-3 with LUT3 and NAND-4 with LUT2

(c)

Approach 3:
Replacing NAND-1, INV-2 and NOR-3 with LUT3

(d)

Approach 4:
Replacing all gates with LUT4

	LUT Approach1	LUT Approach2	LUT Approach 3	LUT Approach4	Custom CMOS
Power	1367.59nw	793.93nw	435.19nw	460.52nw	470.51nw
Delay	339.75ps	306.59ps	202.77ps	238.93ps	21.03ps

Table 4, Result of power and delay of 4 approaches compare to Custom CMOS

Based on Figure 26, we are showing 4 different approaches of replacing the LUT instead of CMOS logic gates and then we can calculate power and delay in Table 4. Due to our results, we can prove that the idea of collapsing which is replacement of part of a circuit with same size LUT in approach 3 of this example, can reduce both power and delay overhead in compare to other approach of replacing gates.

3.6. ISCAS Benchmark Example

Now after proving the collapsing idea with a sample circuit, we should move forward and analyze the same process but for such a more complex circuit. For having the best circuit are using ISCAS benchmark circuits. These circuit designs are open source and we get the Verilog code if design from [22]. We have chosen one of the circuits with 3 stages, it is from 74X series and the number is ISCAS 74283. The design belongs to 4-bit fast adder circuit.

Since we get access to Verilog code, we need to synthesis the design to get gate level netlist. We use Verilog Compiler simulator (VCS), DVE tools and synthesis Synopsys tools. Figure 27 is the capture of Verilog code; Figure 28 is hierarchical synthesized schematic; Figure 29 is ungrouped gate level netlist; Figure 30 is fragment of area report, figure 31 is fragment of qor report and Figure 32 is the fragment of timing report.

```
module Circuit74283 (C0, A, B, S, C4);
  input[3:0]    A, B;
  input         C0;
  output[3:0]   S;
  output        C4;
  TopLevel74283 Ckt74283 (C0, A, B, S, C4);
endmodule /* Circuit74283 */

/*********************************************************************************/

module TopLevel74283 (C0, A, B, S, C4);
  input[3:0]    A, B;
  input         C0;
  output[3:0]   S;
  output        C4;
  wire[3:0]     GB, PB, AxB;
  wire[3:0]     C;
  GP_Module GP_Mod1(A, B, GB, PB, AxB);
  CLA_Module CLA_Mod2(GB, PB, C0, C, C4);
  Sum_Module Sum_Mod3(AxB, C, S);
endmodule /* TopLevel74182 */

/*********************************************************************************/

module GP_Module(A, B, GB, PB, AxB);
  input[3:0]    A, B;
  output[3:0]   GB, PB, AxB;
  wire[3:0]     P;
  nor PBgate0(PB[0], A[0], B[0]);
  nand GBgate0(GB[0], A[0], B[0]);
  not Pgate0(P[0], PB[0]);
  and AxBgate0(AxB[0], GB[0], P[0]);
  nor PBgate1(PB[1], A[1], B[1]);
  nand GBgate1(GB[1], A[1], B[1]);
  not Pgate1(P[1], PB[1]);
  and AxBgate1(AxB[1], GB[1], P[1]);
  nor PBgate2(PB[2], A[2], B[2]);
  nand GBgate2(GB[2], A[2], B[2]);
  not Pgate2(P[2], PB[2]);
```

Figure 27, fragment of original Verilog code of ISCAS 74283

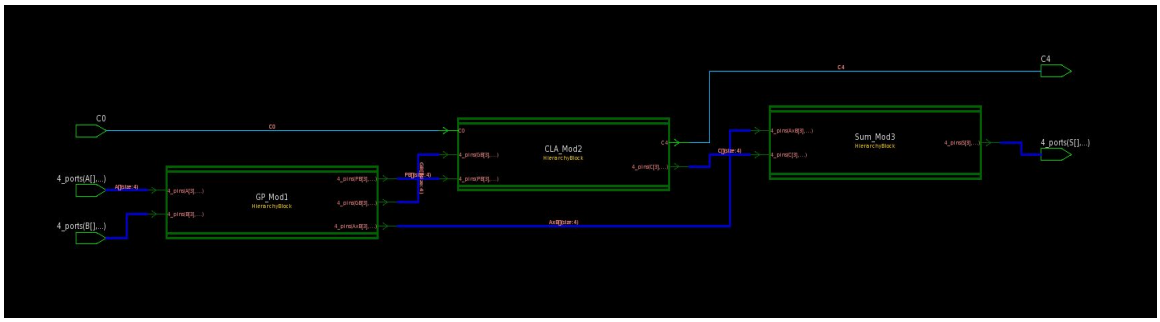


Figure 28, hierarchical synthesized schematic of ISCAS 74283

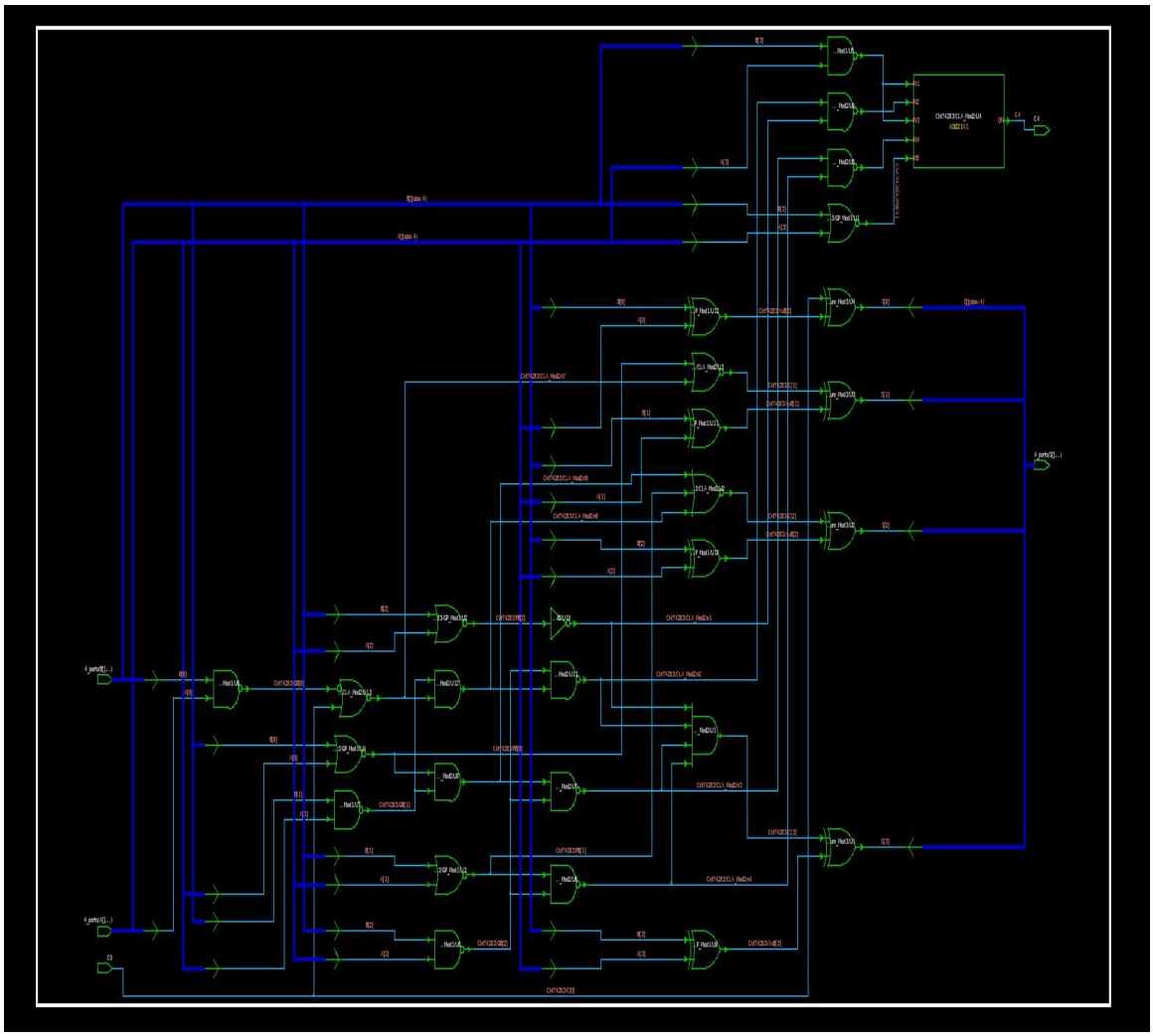


Figure 29, Ungrouped synthesized gate level netlist schematic of ISCAS 74283

```
****************************************
Report : area
Design : Circuit74283
Version: I-2013.12-SP2
Date   : Thu Jul 30 16:54:27 2015
****************************************

Information: Updating design information... (UID-85)
Library(s) Used:

    saed90nm_max (File: /packages/process_kit/generic/generic_90nm/
updated_Oct2008/SAED_EDK90nm/Digital_Standard_Cell_Library/synopsys/models/
saed90nm_max.db)

Number of ports:                           14
Number of nets:                            30
Number of cells:                            3
Number of combinational cells:              0
Number of sequential cells:                 0
Number of macros/black boxes:               0
Number of buf/inv:                          0
Number of references:                       3

Combinational area:                248.185003
Buf/Inv area:                        5.530000
Noncombinational area:               0.000000
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                   248.185003
Total area:                 undefined
1
```

Figure 30, Fragment of area report of circuit

```
Timing Path Group (none)
-----------------------------------
Levels of Logic:               6.00
Critical Path Length:          1.49
Critical Path Slack:         uninit
Critical Path Clk Period:       n/a
Total Negative Slack:          0.00
No. of Violating Paths:        0.00
Worst Hold Violation:          0.00
Total Hold Violation:          0.00
No. of Hold Violations:        0.00
-----------------------------------


Cell Count
-----------------------------------
Hierarchical Cell Count:          3
Hierarchical Port Count:         46
Leaf Cell Count:                 29
Buf/Inv Cell Count:               1
Buf Cell Count:                   0
Inv Cell Count:                   1
CT Buf/Inv Cell Count:            0
Combinational Cell Count:        29
Sequential Cell Count:            0
Macro Count:                      0
-----------------------------------


Area
-----------------------------------
Combinational Area:      248.185003
Noncombinational Area:     0.000000
Buf/Inv Area:              5.530000
Total Buffer Area:             0.00
Total Inverter Area:           5.53
```

Figure 31, Fragment of qor (quality of design) report

```
Startpoint: A[0] (input port)
Endpoint: S[3] (output port)
Path Group: (none)
Path Type: max

Point                                                          Incr       Path
-----------------------------------------------------------------------------
input external delay                                           0.00       0.00 r
A[0] (in)                                                      0.00       0.00 r
Ckt74283/GP_Mod1/A[0] (GP_Module)                              0.00       0.00 r
Ckt74283/GP_Mod1/U8/QN (NAND2X0)                               0.10       0.10 f
Ckt74283/GP_Mod1/GB[0] (GP_Module)                             0.00       0.10 f
Ckt74283/CLA_Mod2/GB[0] (CLA_Module)                           0.00       0.10 f
Ckt74283/CLA_Mod2/U13/Q (ISOLANDX1)                            0.23       0.33 f
Ckt74283/CLA_Mod2/U12/Q (AND2X1)                               0.23       0.56 f
Ckt74283/CLA_Mod2/U11/QN (NAND2X0)                             0.13       0.69 r
Ckt74283/CLA_Mod2/U1/Q (AND4X1)                                0.40       1.09 r
Ckt74283/CLA_Mod2/C[3] (CLA_Module)                            0.00       1.09 r
Ckt74283/Sum_Mod3/C[3] (Sum_Module)                            0.00       1.09 r
Ckt74283/Sum_Mod3/U1/Q (XOR2X1)                                0.40       1.49 r
Ckt74283/Sum_Mod3/S[3] (Sum_Module)                            0.00       1.49 r
S[3] (out)                                                     0.00       1.49 r
data arrival time                                                         1.49
-----------------------------------------------------------------------------
(Path is unconstrained)
```

Figure 32, fragment of timing report

Now everything is ready for analyzing all possible collapsing options. We know the critical path and we do analytical analysis to find the overall power of different approaches and delay of critical path for it.

Figure 33 is showing the detail of gate level netlist of our design under test.

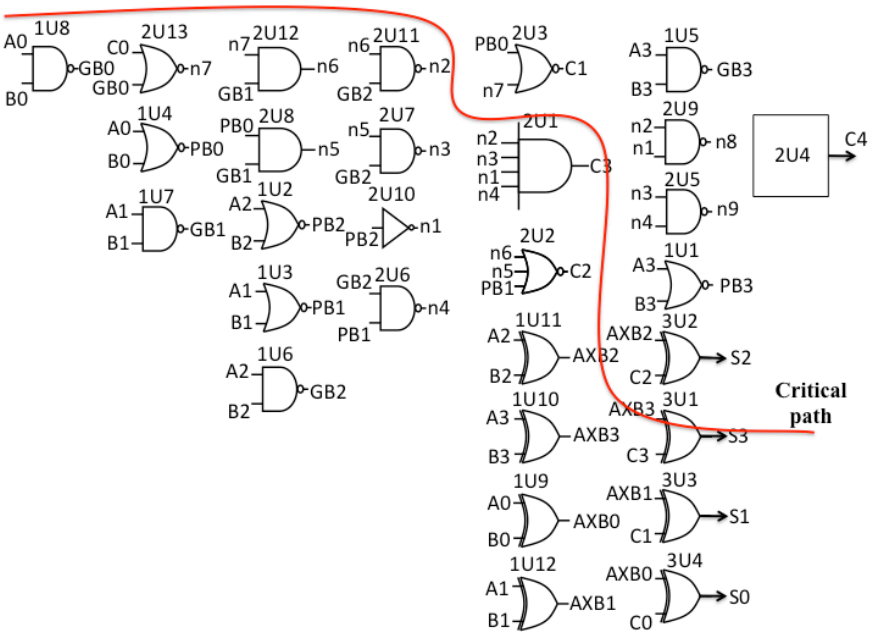


Figure 33, detailed gate level netlist of ISCAS benchmark circuit 74283

Now in this stage we need to carefully find all possible collapsing options. The constraints are first the subset of gates should have single output, second any internal node of subset shouldn't use as input of other gates. All possible collapsing options are as follows, in Figure 34, 35, 36.

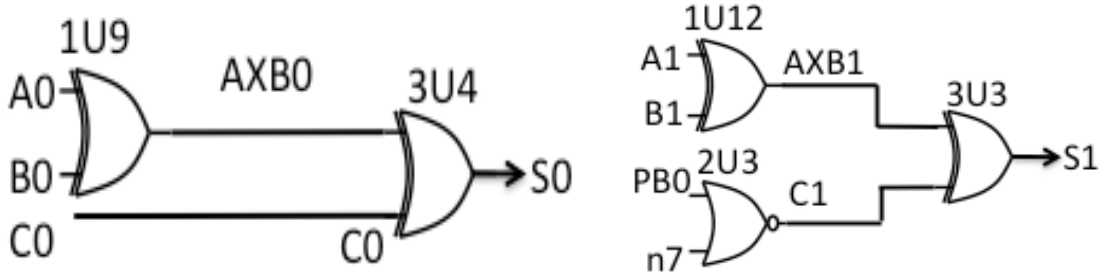


Figure 34, Collapsing options 1 and 2

1U9-3U4 => LUT3x1 ***1U12-2U3-3U3 => LUT4x1***

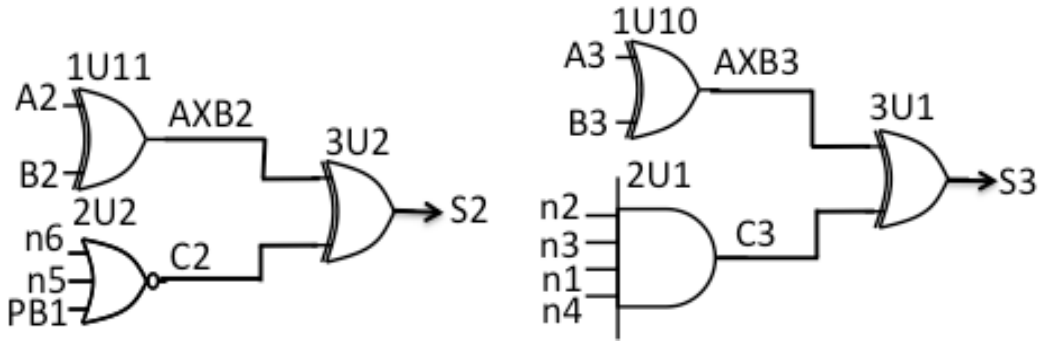


Figure 35, Collapsing options 3 and 4

1U11-3U2-2U2 => LUT5x1 ***1U10-2U1-3U1 => LUT6x1***

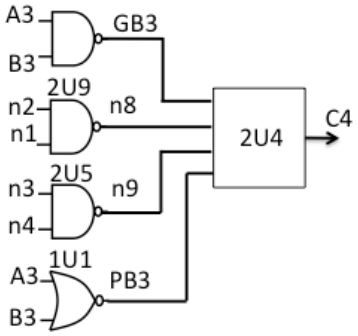


Figure 36, collapsing option 5

1U5-2U9-2U5-1U1-2U4 => LUT6x1

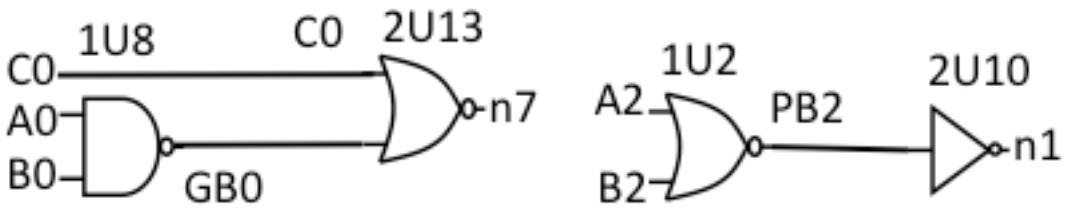


Figure 37, collapsing options 6 and 7

1U8-2U13 => LUT3x1 ***1U2-2U10 => LUT2x1***

Now we are ready to develop different approaches for replacing possible collapsing options with CMOS gates. Approaches are as follows:

- **Approach #1: all CMOS gates are replaced by same fan-in LUT except inverter**

This is the one of the most secure approaches (Figure 38).

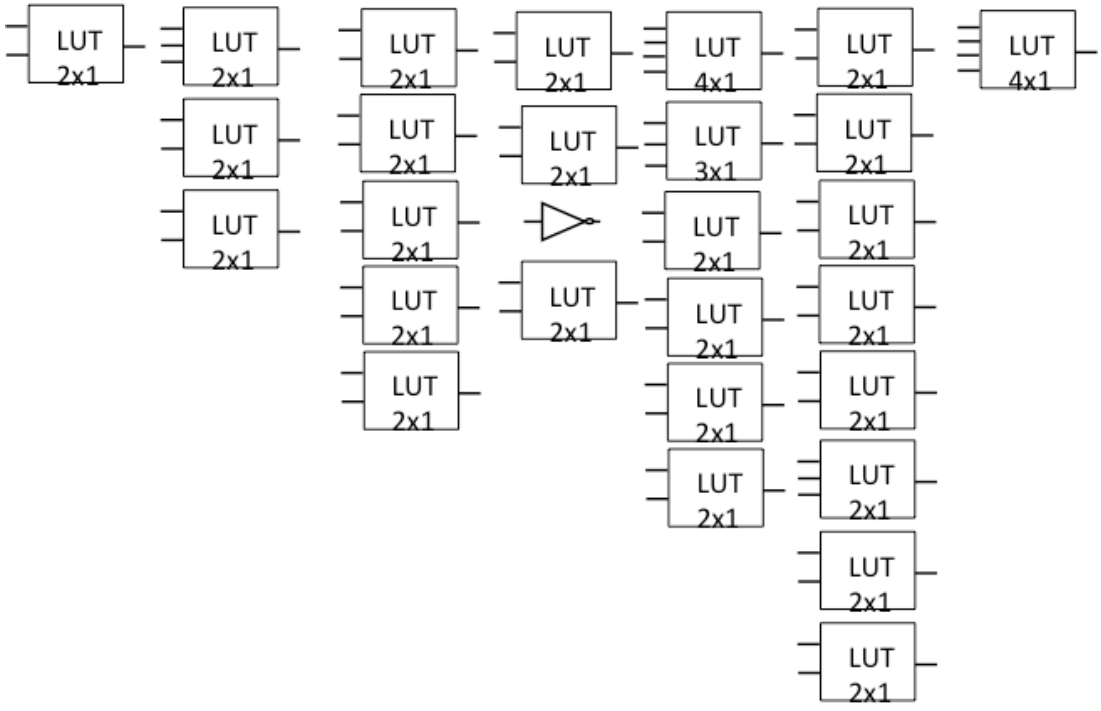


Figure 38, Approach #1: all CMOS gates are replaced by same fan-in LUT except inverter

25 LUT2x1 + 1LUT3x1 + 2LUT4x1 + 1 Inv (α = 30%)

Total power = **10.556uw**

Delay of data path = **795.2ps**

- **Approach #2: all possible collapsing options, and remaining gates are replaced by same fan-in LUT**

This is another high security approach (Figure 39).

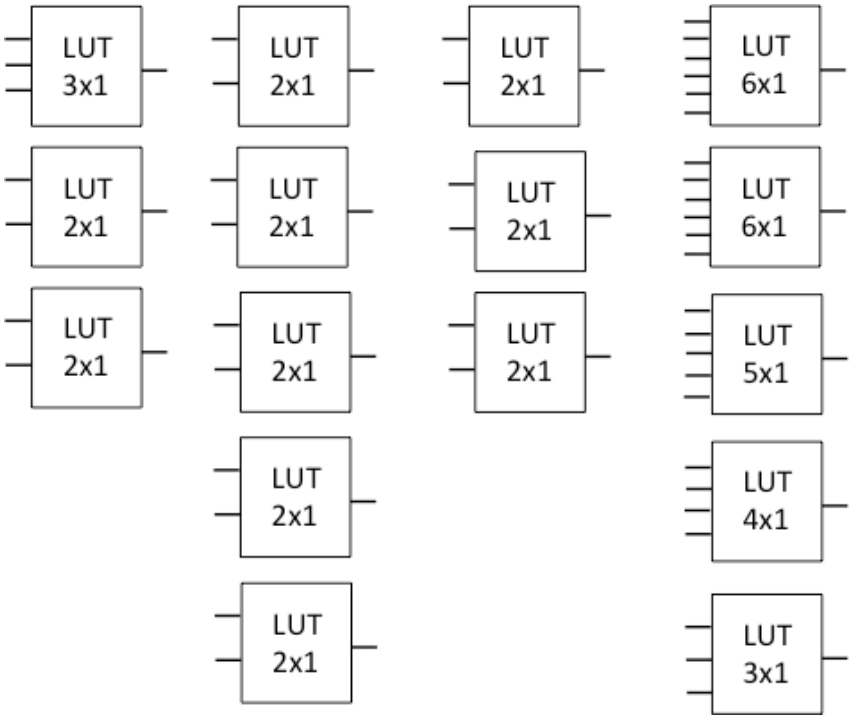


Figure 39, Approach #2: all possible collapsing options, and remaining gates are replaced by same fan-in LUT

Total power = **6.106uw**

Delay of data path = **715.4ps**

- **Approach #3: all possible collapsing options, and remaining gates keep as Custom CMOS (Figure40)**

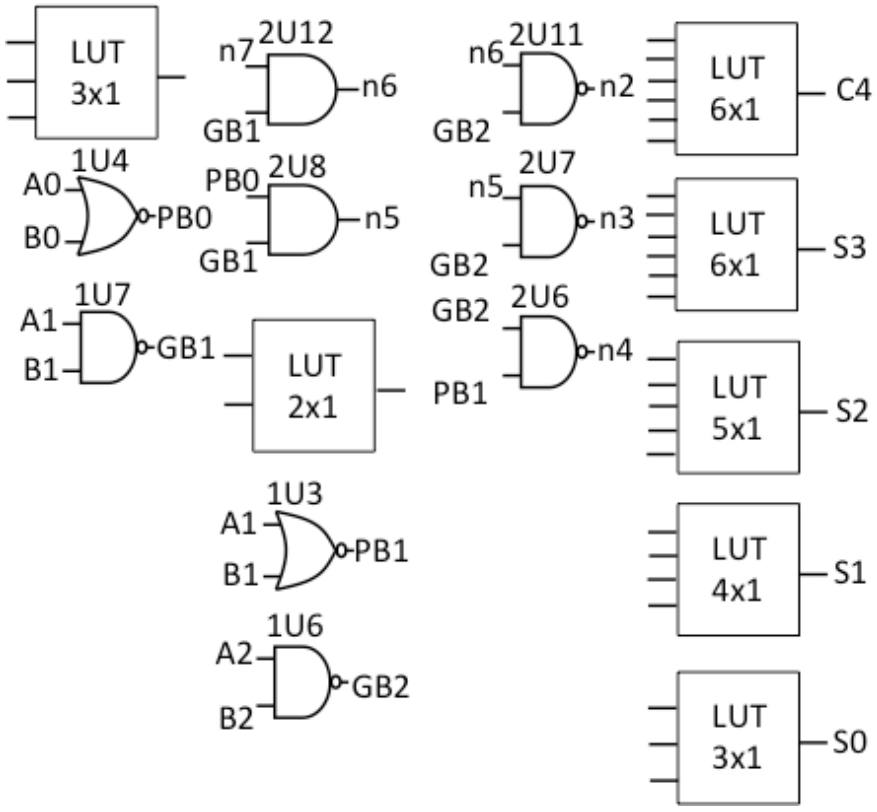


Figure 40, Approach #3: all possible collapsing options, and remaining gates keep as Custom CMOS

Total power = **4.363uw**

Delay of data path = **525.74ps**

- **Approach #4: Three random LUT are replaced instead of Custom CMOS gates (figure 41)**

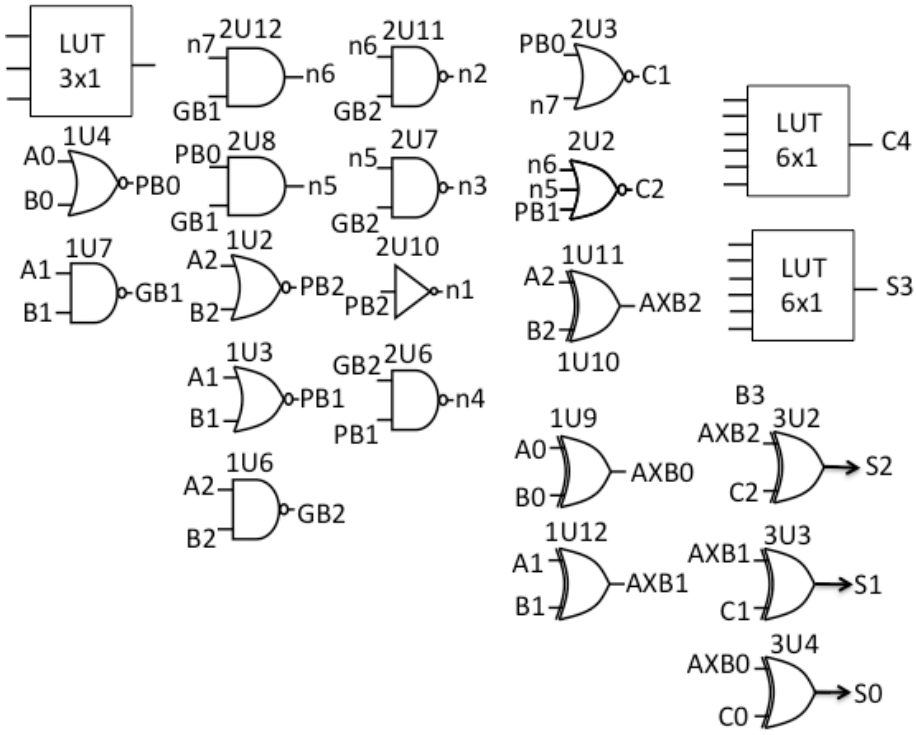


Figure 41, Approach #4: Three random LUT are replaced instead of Custom CMOS gates

Total power = **5.834uw**

Delay of data path = **505.74ps**

Approach #5: any collapsing options are replaced instead of Custom CMOS gates but not in critical path (Figure42)

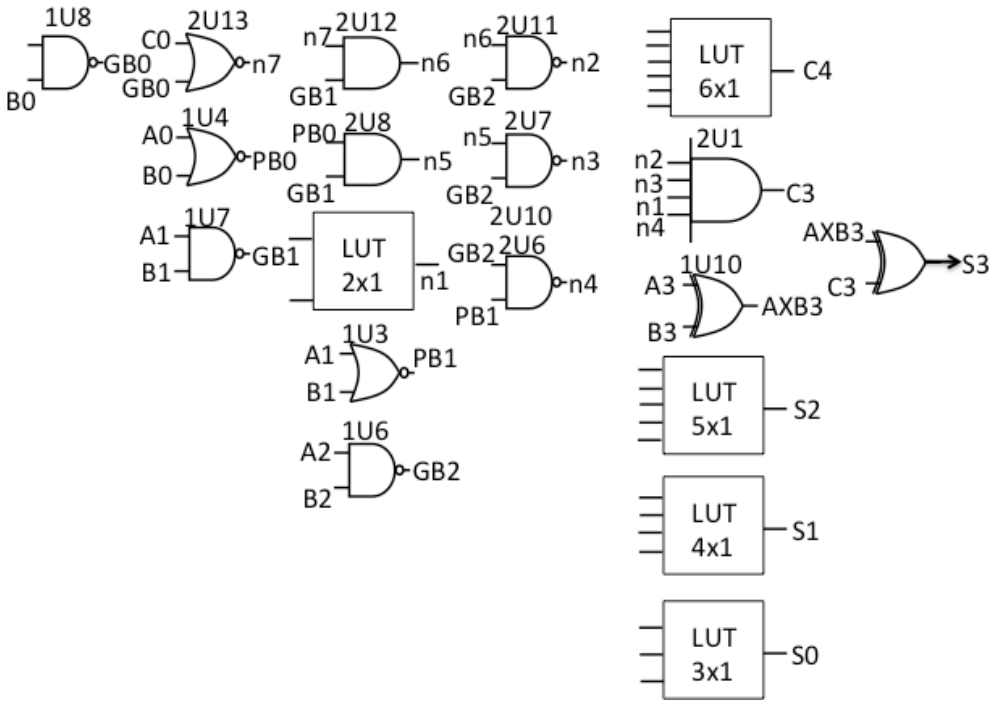


Figure 42, Approach #5: any collapsing options are replaced instead of Custom CMOS gates but not in critical path

Although the delay of critical path is equal to Custom CMOS, but with this replacing we create new critical path, therefore we need to measure delay of each path.

Total power = **5.026uw**

Delay of data path = 87.43ps X

Delay of new critical path = **365.8ps**

Custom CMOS analytical analysis

As we discuss before for any CMOS gate in order to find power we need to find the activity of gates or how many time output switches. Figure 43 is one example of how we measure activity of 4 inputs And gate.

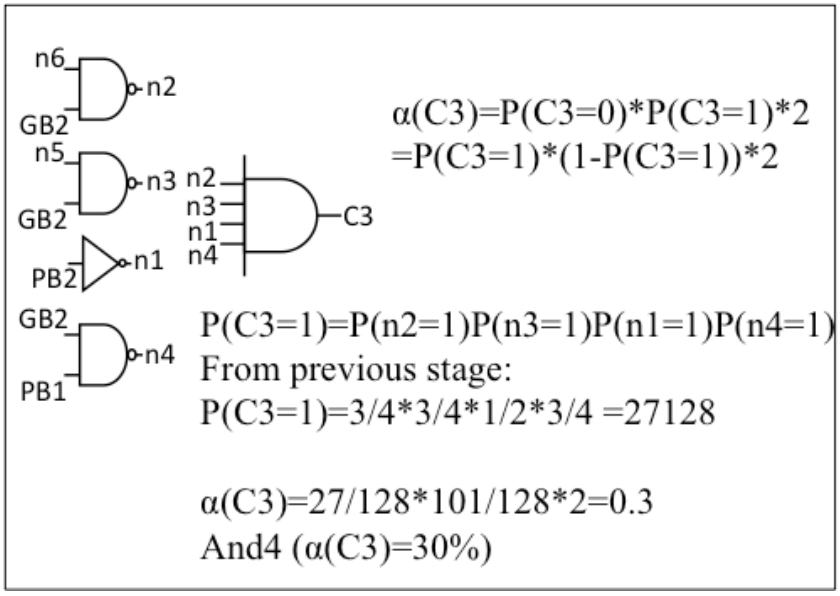


Figure 43, example of how to measure activity of 4 inputs AND gate in the circuit

7 Nand2 (α = 30%) + And4 (α = 30%) + Nand2 (α = 10%) + Nand2 (α = 50%) + 5 Nor2 (α = 30%) + Nor2 (α = 50%) + Nor3 (α = 50%) + 2And(α = 30%) + INV (α = 30%) + 4Xor2(α = 50%) + 2Xor2(α = 10%) + 2Xor2(α = 30%) + 2U4(α = 30%)

Total power = **7.596uw**

Delay of data path = **87.43ps**

Among analytical analysis we face to a problem of Inter-signal Correlations. The evaluation of the switching activity is further complicated by the fact that signals exhibit correlation in space and time. Even if the primary inputs to a logic network are uncorrelated, the signals become correlated or "colored", as they propagate through the logic network [21]. We aware of this issue but since this makes our analysis more complex, we prefer to ignore it at this stage and for future work we can solve this problem with Numerical analysis based on simulation in order to find activity of each logic gates.

Comparison Table between different approaches of collapsing options with Custom CMOS (Table 5)

Approach	Power (uw)	Delay (ps)
#1: all CMOS gates are replaced by same fan-in LUT except inverter	10.556	795.2
#2: all possible collapsing options, and remaining gates are replaced by same fan-in LUT	6.108	715.4
#3: all possible collapsing options, and remaining gates keep as Custom CMOS	4.363	525.74
#4: Three random LUT are replaced instead of Custom CMOS gates	5.834	525.74
#5: any collapsing options are replaced instead of Custom CMOS gates but not in critical path	5.026	395.8
CUSTOM CMOS	7.596	87.43

Table 5, comparison between different approaches of collapsing with Custom CMOS

Results and observations:

- We find out that the lowest power is apprach#3, which all possible collapsing options, and remaining gates keep as Custom CMOS.
- The first two approaches that either if all CMOS gates are replaced by same fan-in LUT or all possible collapsing options, and remaining gates are replaced by same fan-in LUT are the most secure designs. Because each gates or subset of possible collapsing gates are replaced by LUT. And among these 2, the second approach is more efficient in terms of power and delay.
- We did a good try of lowering the delay by replacing LUT in non-critical path, but in this way we are making new critical path. In compare to others this approach has the lowest delay in compare to Custom CMOS.
- In complex circuit when the delay of critical path is greater than LUT2, then maybe the delay overhead is avoided.

Chapter 4: Conclusion

- For the purpose of enhancing hardware security, among existed types of STT-LUT, voltage-sensing LUT has better characteristic in terms of power and delay.
- More optimization efforts for STT-LUT design is exploited by finding the feasible and efficient TMR and tuning Reference resistor.
- By comparing characteristics of LUT and complimentary CMOS we are facing to power and delay overhead. Although we are aware of this much overhead, but we know that for our goal, which is hardware security, there might be a trade off. But we define our approach to lower the overhead as much as possible and solve all the issues related to it.
- Collapsing idea is a way that we can lower power and delay overhead. For lowering the power we need to replace as much as collapsing options with LUTs and keep other gates as Custom CMOS. But if we also replace remaining gates with LUT, we have the benefit of highest security level. For lowering the delay we have to replace any possible collapsing options in non-critical path. And there is chance to avoid delay overhead in more complex circuits.

Chapter 5: Future Challenges and future work

5.1. Cascading Issue

Cascading issue is corresponded to a problem when we cascade many LUTs together and it is because of clock is coming at the same time to different LUTs while they have not received the previous stage output as their inputs. And this will definitely cause failing because clock trigger the sense amplifier while the input didn't come yet. As shown in Figure 44 cascading of LUTs are illustrated.

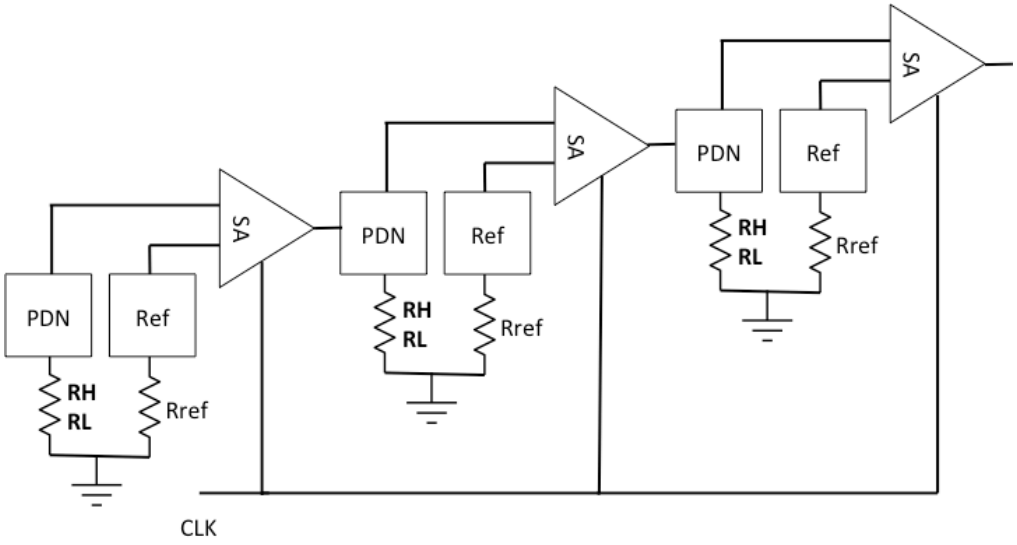


Figure 44, cascading of 3 LUT 2 inputs

For the purpose of test a sample circuit in order to present cascading issue, we are simulating very simple circuit of cascading 4 LUT2 input that have inverter functionality as shown in Figure 45.

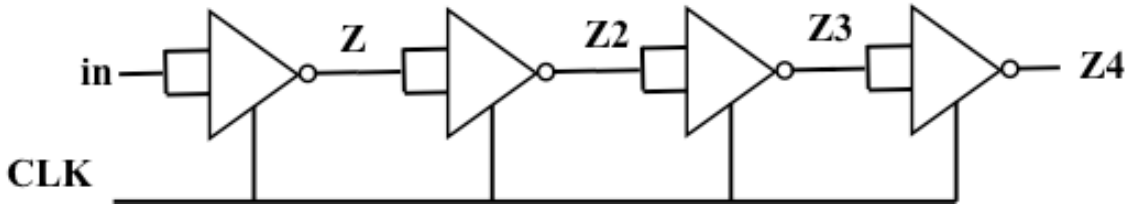


Figure 45, simple circuit for testing if we have cascading issue

As shown in Figure 46 after simulating the circuit, all the internal nodes as outputs of every stage inverters are illustrated. And it is obvious that even after first clock cycle the second inverter output is failing, this is exactly what we call cascading issue.

Figure 46, wave view of cascaded 4 LUT, to illustrate cascading issue

Solution for cascading issue

1. Delayed Clock with reasonable number of even inverter

The first solution is delayed clock, it means that for clock of its stage we need to delayed the original clock by extra circuit of even number of inverters to make buffers. As shown in Figure 47, we delayed the original clock of each LUT for each stage in order to solve the cascading issue and all the LUTs are working properly

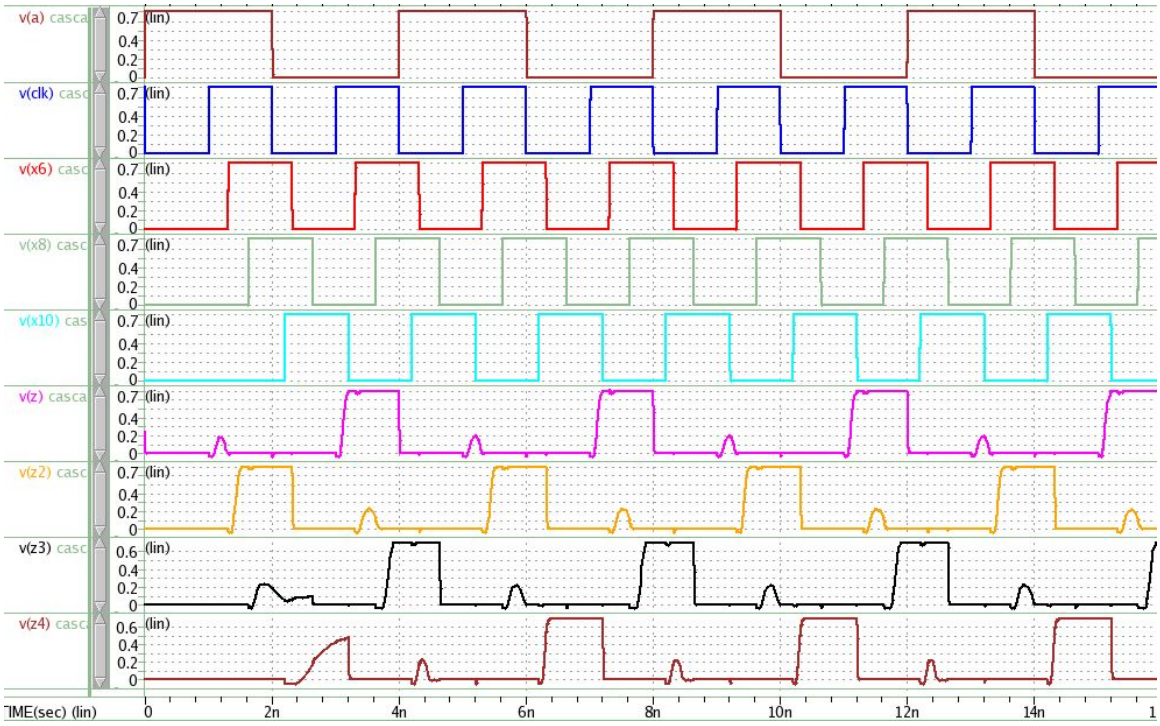


Figure 47, wave view result of cascaded 4LUT after generating delayed clock for each LUT

2. Having LUTs in boundary of each stage

With help of Flip flop, which is one master latch and one slave latch the data is kept until next positive edge of clock. We can combine master latch logic gate for replacement with LUT and then output of LUT is edge triggered and can retain itself. With this approach we are solving the timing issue because the LUTs are only in boundaries.

So we are applying limitation to ourselves to only have LUTs in boundaries and this will limit us lowering the security level, because Gates in between of boundaries are unsecured. Figure 48 is showing haw we can replace LUT in boundaries instead of combination of logic gates and master latch, and then this LUT is followed by slave latch. Now our new LUT is positive edge triggered. And now the output of new LUT can retain itself.

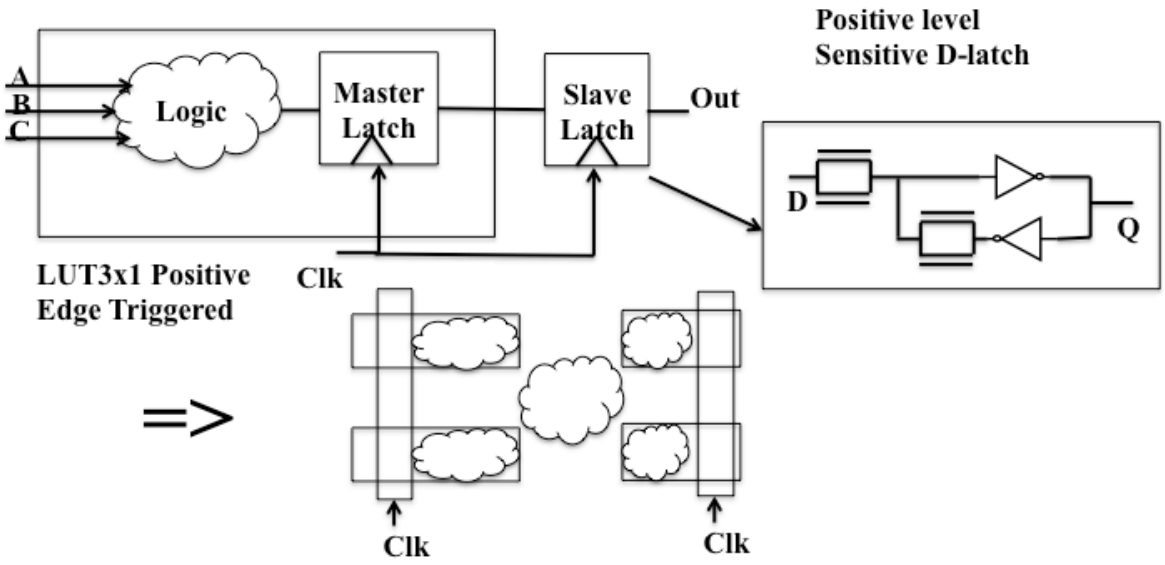


Figure 48, replacing LUT in boundaries of circuit and add flip flop to make LUT positive edge-triggered

3. Using self clocked LUT design

We need to add extra circuit in order to generate clock at the right time when the inputs are coming (Figure 49). But we are applying extra overhead, but in high fan-in LUT it's possible because the extra overhead in negligible.

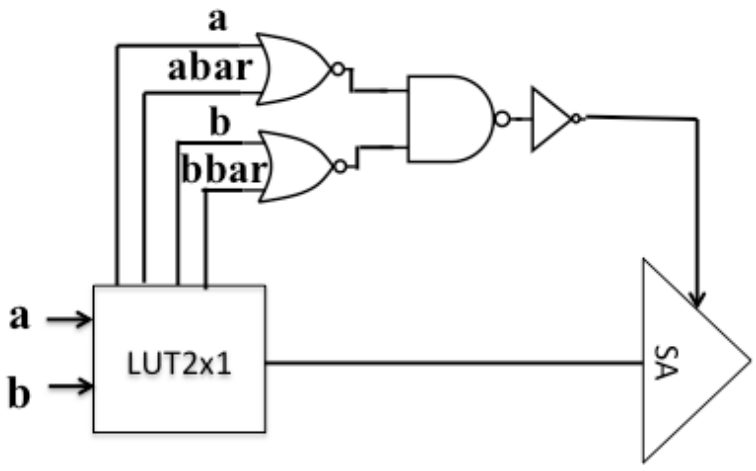


Figure 49, self clocked LUT design that can generate clock same time as the inputs come

4. Using CLOCKLESS LUT design

As shown in Figure 50, the design without clock is with the same concept but the output of inverter between selection resistors of and reference resistor will trigger the sense amplifier. The problem of this solution is too much standby power, because when all the inputs are zero, there is path from VDD to resistors and the ground, so we have high leakage current and then high standby power consumption.

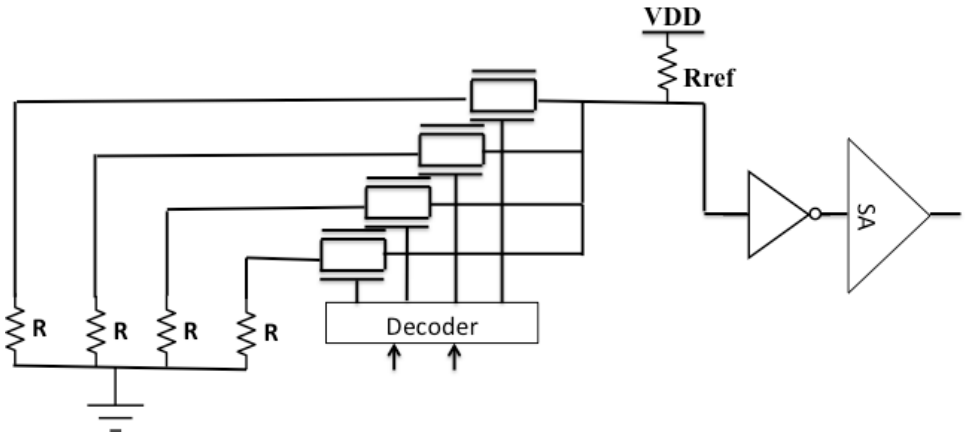


Figure 50, Design of LUT without clock

5.1. Numerical Analysis based on Simulation

For solving problem of input correlation of logic gates, in order to find activity of gate, we need to do numerical analysis for more accurate power estimation.

We can use system Verilog code for generating random inputs and then find the activity of CMOS gates.

5.2. Developing an algorithm for applying Collapsing

Based on optimization that we get from collapsing idea, in future we need to develop an algorithm for insertion to more complex and large circuits.

Algorithm should be able to find all possible subset of gates that are candidate to be possible option of collapsing.

6. References:

[1] Mohammad Rezaeirad, “A vanishing sensitive electronics”, George Mason University, Not published

[2] Alex Baugarten, Akhilesh Tyagi and Joseph Zambreno, “Preventing IC piracy using reconfigurable logic barriers”, IEEE Design & Test of Computers, 2010

[3] Jeyavijayan Ranjendran, Michael Sam, Ozgur Sinanoglu, Ramesh Karri, “Security Analysis of Integrated Circuit Camouflaging”, ACM SIGSAC conference, 2013

[4] Chipworks, “Intel‘s 22-nm Tri-gate Transistors Exposed,” http://www.chipworks.com/blog/technologyblog/2012/04/23/intels-22-nm-trigate-transistors-exposed/, 2012.

[5] R. Torrance and D. James, “The state-of-the-art in semiconductor reverse engineering,” in the Proc. of IEEE/ACM Design Automation Conference, pp. 333–338, 2011.

[6] ExtremeTech, “iPhone 5 A6 SoC reverse engineered, reveals rare hand-made

custom CPU, and tri-core GPU,”

http://www.extremetech.com/computing/136749-iphone-5-a6-soc-reverseengineered-

reveals-rare-hand-made-custom-cpu-and-a-tri-core-gpu.

[7] Richard William Dorrance, a master thesis, “Modeling and Design of STT-MRAMs”, University of California, Los Angeles, 2011.

[8] J. C. Slonczewski, “Current-Driven Excitation of Magnetic Multilayers," Journal of Magnetism and Magnetic Materials, vol. 159, no. 1-2, 1996.

[9] Hamid Mahmoodi, Sridevi Srinivasan Lakshmipuram, Manish Arora, Yashar Asgarieh, Houman Homayoun, Bill lin and Dean M. Tullsen, “Resistive Computation: A Critique,” Computer Architecture letters, vol.13, no.2, pp.89,92, 2014.

[10] Xuanyao Fong, Sri Harsha Choday and Kaushik Roy, “Bit-Cell Level Optimization for Non-volatile Memories Using Magnetic Tunnel Junctions and Spin Tranfer Torque switching”, IEEE transactions on Nano technology, Vol.11, No. 1, 2012.

[11] T.Kawahara, K. Ito, R.takemura, H.Ohno. 'Spin-transfer torque Ram technology: Review and prospect" Elevier, 2012

[12] Iong Ying Loh, "Mechanism and Assessment o Spin Transfer torque (STT) based memory", Master degree thesis, Massachusetts Institute of Technology, September, 2009.

[13] Subho Chatterjee et. al, "A Methodology for Robust, Energy Efficient Design of Spin-Torque-Transfer RAM Arrays at Scaled Technologies", ICCAD'09, November 2–5, 2009.

[14] https://nanohub.org/resources/19264, By Deepanjan Datta, Behtash Behin-Aein, Sayeef Salahuddin, Supriyo Datta, "Quantitative Model for TMR and Spin-transfer Torque in MTJ devices".

[15] F. Ren and D. Markovic. True energy-performance analysis of the mtj-based logic-in-memory architecture (1-bit full adder). Electron Devices, IEEE Transactions on, 57(5):1023 –1028, may 2010.

[16] X. Guo, E. Ipek, and T. Soyata. Resistive computation: avoiding the power wall with low-leakage, stt-mram based computing. In Proceedings of the 37th annual international symposium on Computer architecture, ISCA '10, pages 371–382, USA, 2010. ACM.

[17] S. Paul, S. Mukhopadhyay, and S. Bhunia. A circuit and architecture codesign approach for a hybrid cmos-sttram nonvolatile fpga. Nanotechnology, IEEE Transactions on, 10(3):385–394, 2011.

[18] D. Suzuki et. al. Fabrication of a nonvolatile lookup-table circuit chip using magneto/semiconductor-hybrid structure for an immediate-power-up field programmable gate array. In VLSI Circuits, 2009 Symposium on, pages 80 –81, june 2009.

[19] Weisheng Zhao. Eric Belhaire, Claude Chappert and Pascale Mazoyer, "Spin Transfer Torque (STT)-MRAM-Based Runtime Reconfiguration FPGA Circuit", IEF, ACM Transactions on Embedded Computing Systems, Vol.9, No. 2, Article 14, 2009.

[20] Predictive Technology Model, http://www.ptm.asu.edu/.

[21] Jan M. Rabaey, " Digital integrated circuits", Second edition, ISBN-13: 978-0130909961

[22] http://web.eecs.umich.edu/~jhayes/iscas.restore/benchmark.html