Preventing Design Reverse Engineering with Reconfigurable Spin Transfer Torque LUT Gates

Theodore Winograd[1], Hassan Salmani[2], Hamid Mahmoodi[3], Houman Homayoun[1]
[1]George Mason University
[2]Howard University
[3]San Francisco State University
[1]{twinogra@gmu.edu, hhomayou@gmu.edu}
[2]hassan.salmani@howard.edu, [3]mahmoodi@sfsu.edu

Abstract

This paper presents a rigorous step towards design-for-assurance by employing the non-volatile spin transfer torque magnetic technology to design reconfigurable Look-Up-Tables logic components (NV-STT-based LUTs). Further, we introduce a novel security-driven STT-CMOS hybrid design flow that ensures the functionality of NV-STT-based LUTs cannot be determined in any manageable time, rendering any design reverse engineering attack ineffective. In addition, the flow ensures that there is minimum or no impact on design parametric constraints including performance, power and area.

Keywords

Reverse-engineering; Security-driven design flow; STT-CMOS hybrid design;

1. Introduction

The integrated circuits (ICs) horizontal supply chain has posed significant challenges to hardware security assurance in various forms. Outsourcing design manufacturing provides significant opportunities for untrusted foundries for tampering, overproducing, and cloning. Even after releasing design to the market, the design can be subject to non-invasive reserve engineering, such as side-channel attacks, to obtain secret information during design operation or invasive reserve engineering to obtain detailed design implementation.

Current techniques for design reverse engineering have raised serious concerns in the IC design community, particularly when facing a very high-tech adversary. Hardware reconfigurability has been around for several years, primarily in the form of FPGAs. In [4], using embedded SRAM-based reconfigurable logic for application specific integrated circuits (ASIC) design obfuscation is investigated to limit hardware Trojan attacks. SRAM FPGAs and SRAM reconfigurable logic provide reconfigurability and potentially enhance security, but they are not practical for use in embedded systems where power and performance are major constraints. SRAM reconfigurable logic has large leakage power dissipation and is most suited for below GHz range operating frequency. Furthermore, they require an external non-volatile memory to keep reconfiguration bitstream, which becomes the source of vulnerability. In addition, they have a very high reconfiguration overhead.

To realize hardware security assurance, we introduce a novel design methodology that incorporates in custom CMOS circuits reconfigurability by employing the highly promising spin transfer torque (STT) magnetic technology to build look-up-tables (LUTs) logic components. A STT design is similar to a filed-programmable gate array (FPGA) design in functionality but with significantly higher speed running at GHz frequency, near zero leakage power, high thermal stability, highly integrative with CMOS and overall competitive with custom CMOS design in terms of performance and energy-efficiency. In addition, compared to the SRAM-based LUT in FPGA, the STT-based LUT is non-volatile, that is, there is no need to another flash memory (which could be a source of vulnerability) to store the configuration bits to load from on every power up.

To protect a design from design reverse engineering attacks after final product release, depending on the required level of security, we propose a novel algorithm to select and replace custom CMOS gates in circuit netlist with reconfigurable STT-based LUTs during design implementation. While an untrusted foundry may have access to the reconfigured design after its release to the market, the selection of custom CMOS gates for replacement is such that the untrusted foundry cannot determine the functionality of reconfigurable LUTs in any reasonable time. The selection algorithm will ensure that original design parametric constraints such as design performance will impact only minimally.

2. STT Technology

A. Overview of STT Technology

This paper uses on-die, run-time reconfigurable logic gates such as NAND and NOR to enhance the security of the design. The run-time reconfiguration is realized by a Look Up Table (LUT) based design, which uses highly promising new Spin Transfer Torque (STT) technology to store the LUT data. The new reconfigurable STT-LUT design not only enables run-time reconfiguration to vanish the logical property of the design, but also offers added advantage for complex blocks in terms of power, performance, lifetime reliability and thermal

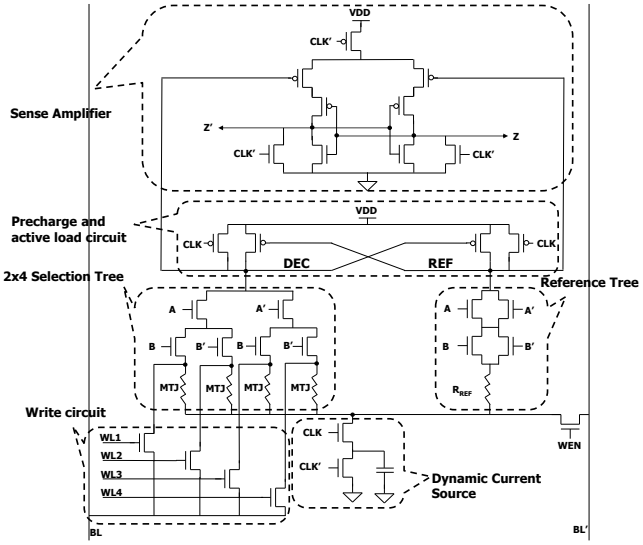


Figure 1. 2-input STT-LUT [10]

stability as compared to non-reconfigurable CMOS implementation.

STT provides i) approximately 4X higher integration density than conventional Static Random Access Memory (SRAM) [11, 17, 15, 21], ii) high retention times (even more than 10 years [9]), iii) high endurance (1016 writes, or 10 years of operation as L1 cache) [1], iv) near-zero leakage [18] with close to SRAM read performance, v) excellent thermal robustness 300oC, vi) soft error resilience, and vii) above all, STT cells are easy to integrate with the conventional CMOS fabrication process. To date, STT technology has been only used to break through the memory wall by implementing low-power, high-density on-chip memories. In our opinion, this is only a start of a new era. We expect that STT technology has transformative potential to impact logic design by offering a low overhead run-time reconfigurable platform that offers not only opportunities for power and performance improvements, but also enhanced security for sensitive blocks.

STT technology, for the first time, provides us the amazing opportunity to design reconfigurable logics that are on-die, comparable in performance to custom CMOS logic, and have low reconfiguration overhead. Existing Field Programmable Gate Arrays (FPGAs) cannot be used to design on-chip re-configurable logics since they are built using flash devices that do not integrate well with the conventional CMOS fabric. Moreover, the reconfiguration time is long in existing technologies. For example, typical partial reconfiguration time on Virtex 6 FPGA is in the order of tens of milliseconds [3]. An alternative would be to use SRAM based reconfigurable units, but they suffer from problems of scalability, high leakage, high sensitivity to variations, and soft errors [5, 7, 12, 20, 14]. Moreover, SRAM based reconfiguration is volatile and needs to be re-programmed on every power up and this demands a separate non-volatile storage such as a flash memory to store configuration bits.

B. Design of lookup table based reconfigurable logic in STT technology

In this paper we use the STT-LUT design proposed by Suzuki [10]. By loading different values in the LUTs, the reconfigurable fabric is able to implement various logic functions. Moreover, there is added security benefit because the content of the LUTs can be hidden to IC manufacturers or eliminated upon detection of a reverse engineering attempt. Moreover, the content of an LUT cannot be reverse engineered from its physical layout because of its generic and programmable nature. STT-NV technology utilizes Magnetic Tunnel Junctions (MTJ) to realize nonvolatile resistive storage. There have been several attempts to use MTJs for building logic circuits to exploit the leakage benefit of MTJs to reduce the circuit power [10]. However, due to the significant energy involved in changing the state of an MTJ, circuit styles that rely on changing the state of MTJs in response to input changes do not show any power and performance benefits [8]. An alternative to this approach has been to realize logic in memory by using LUTs that are built based on MTJs [2]. Resistive Computation [2] replaces conventional CMOS logic with Magnetic Tunnel Junction (MTJ) based Look-Up Tables (LUTs); it has been proposed for tackling the power wall. However in [2] STT-LUT design has mainly deployed in large functional unit blocks and its reconfiguration feature has not been explored.

B.1 Spin Transfer Torque Look Up Table (STT-LUT)

Figure 1 shows the schematic of a 2-input (4-bit) STT-LUT [10]. This is a dynamic circuit that operates in a precharge (CLK=0) and evaluate (CLK=1) fashion. For the read mode, the MTJ selection is performed via a pass-transistor decoder/mux (selection tree). To balance the transistor paths of the MTJs and the reference resistor (RREF), similar transistors are inserted above the reference resistor. When CLK goes high, the current provided by the dynamic current source is divided between the selected MTJ and the reference resistor, resulting in a current differential that is drained from the nodes DEC and REF. This current differential is converted to a low swing voltage differential on the nodes DEC and REF by the two cross coupled PMOSes. This voltage differential is then amplified by a sense amplifier to produce full swing differential outputs (Z and Z'). In the write mode, the write differential voltage is applied to the bitlines (BL and BL') and the right MTJ is selected by the world-line signals (WLi) and the write enable (WEN) is activated.

For high read performance and enhanced noise margin, greater difference between the low and high

resistances of the MTJ is desired. This resistance differential is quantified by the Tunnel Magneto Resistance (TMR), defined as:

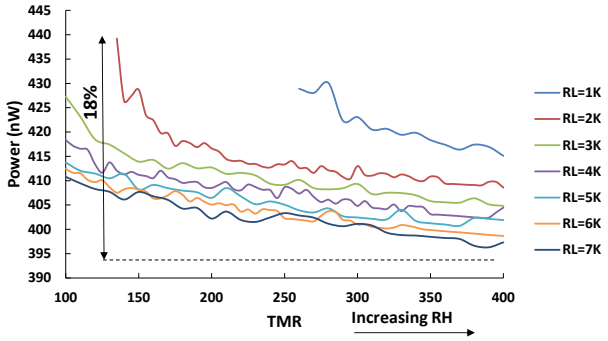
$$TMR = 100 \times (RAP - RP) / RP$$

where RP and RAP are the resistances of the MTJ in the parallel and anti-parallel states, respectively. TMR is technology parameter dependent on the MTJ geometries and materials.

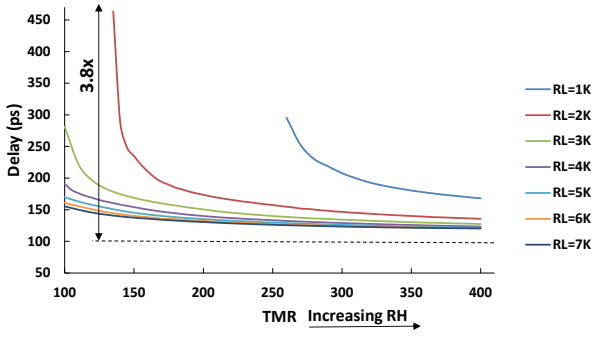
B.2) TMR Optimization

Higher TMR is desirable for read performance. We have simulated the read power and performance of a 2-input STT-LUT in a predictive 16nm CMOS technology node [19].

As shown in Figure 2, higher TMR results in reeducation in both read power and delay of the STT-LUT. Higher TMR results in large difference between the low and high resistances of the MTJ (Eq. (1)) and hence larger voltage differential produced at the input of the sense amplifier (Figure 1) in the read mode resulting in reduced delay. Moreover, larger TMR increases the average resistance of the pull-down network (due to increase in RH (RAP) and RREF that is about the average of RH an RL (RP)), and hence reduction in power dissipation.



(a)



(b)

Figure 2. (a) Power and (b) Delay of STT-LUT vs TMR. RH and RL denote the high and low resistances (RAP and RP) of the MTJ.

Achieving very high TMR has manufacturing limitations in terms of material choice and physical limitations. A survey of published literature on manufactured MTJs show TMR values ranging from 100 to 600%. For the rest of our analysis, we use TMR of 400% which is in the range feasible by manufacturing.

The reference resistor (RREF) can also be optimized for achieving best power and delay. A good starting point for RREF would be average of RH and RL, but it could be optimized. The effective resistance of the two networks, reference and selection trees (Figure 1), should be equal to achieve symmetric delay between the differential outputs. Moreover, the capacitance of the dynamic current source (Figure 1) can be jointly optimized with RREF to minimize the power and delay product of the STT-LUT. Table 1 shows the characterization of STT-LUTs with fan-ins ranging from 2 to 8. With increase in fan-in the active power and delay generally increase, but the leakage power remains very low and almost insensitive to the fan-in, since the leakage is limited by the stacking effect, and saturated through the NMOS pulldown network gated by the clocked transistor in the dynamic current source (Figure 1) [13, 16].

Table 1. Power and delay characterization of STT-LUTs (TMR=400% (RL=20KΩ and RH= 100KΩ), Clock frequency=500MHz, Vdd=0.7V, Temperature=100˚C, Process: 16nm CMOS). PDP=Power Delay Product

Fan-in	R_{REF} (KΩ)	C (fF)	Active Power(nW)	Delay (pS)	PDP (fJ)	Leakage Power(nW)
2	32.6	3	378.4	108.8	0.407	4.37
3	30.6	3	337.3	178.4	0.601	3.97
4	28.5	3	328.5	251.3	0.825	3.76
5	64.7	5	443.4	250.4	0.111	3.66
6	62.9	5	437.8	316.9	0.138	3.68
7	61	7	500.6	293.2	0.146	3.87
8	55	9	553.6	303.8	0.168	4.34

B.3) Comparison with custom CMOS

Table 2 shows the simulation results of the STT-LUT and static custom CMOS circuit styles for logic gates of various complexity implemented in a predictive 32nm technology. All the results are normalized to the corresponding results for a static CMOS implementation. It is clear from the results that for small logic gates, the STT-LUT style shows considerable overhead as compared to the custom CMOS implementation; however, as the circuit complexity increases this overhead reduces. The delay overhead is also less for high fan-in NOR gates that their static CMOS implementation would require a series connection of PMOSes in their pull-up networks. PMOS transistors tend to be slower than NMOS transistors and since the STT-LUT style uses less number of PMOS transistors, its benefit is more noticeable for implementation of such logic gates.

Another observation from Table 1 is that the LUT style shows less power overhead for higher data activity (α). This is due to the dynamic nature of the STT-LUT style that increases its switching activity making it a better fit for high data activity applications. Note that the power and delay of the STT-LUT is independent of the logic it is programmed to implement (i.e. its data content) and

also independent of its input data activity. The power and delay of the STT-LUT only depends on its fain-in (number of inputs).

The leakage power of the STT-LUT style is lower than the custom CMOS except for high fan-in NAND and NOR gates. In high fan-in static CMOS NAND (NOR) gates, there is a long chain of series connected NMOS (PMOS) transistors that suppresses leakage via the transistor stacking effect. However, this leakage advantage for such static CMOS gates will disappear if those gates are implemented using cascade of lower fan-in gates for performance reasons. Therefore we can argue that for low fan-in (4-input or less) standard logic gates, the STT-LUT style implementation offers less leakage.

Table 2. Comparison of circuit style alternatives (α: output switching activity).

Gate	Metric	MTJ Based	Static CMOS
NAND2	Delay	6.46	1
	Active Power(α=10%)	90.35	1
	Active Power(α=30%)	30.12	1
	Standby Power	0.48	1
	Energy per Switching	58.36	1
NAND4	Delay	4.49	1
	Active Power(α=10%)	76.73	1
	Active Power(α=30%)	25.57	1
	Standby Power	0.96	1
	Energy per Switching	34.45	1
NOR2	Delay	4.85	1
	Active Power	80.2	1
	Active Power(α=30%)	26.73	1
	Standby Power	0.51	1
	Energy per Switching	38.89	1
NOR4	Delay	3.06	1
	Active Power(α=10%)	24.25	1
	Active Power(α=30%)	8.08	1
	Standby Power	1.06	1
	Energy per Switching	7.42	1
XOR2	Delay	4.95	1
	Active Power(α=10%)	22.45	1
	Active Power(α=30%)	7.48	1
	Standby Power	0.13	1
	Energy per Switching	11.11	1
XOR4	Delay	4.18	1
	Active Power(α=10%)	90.06	1
	Active Power(α=30%)	30.02	1
	Standby Power	0.04	1
	Energy per Switching	37.64	1

3. Security and STT Technology

While an untrusted foundry can create an unresolved netlist containing STT-based LUTs from a submitted GDSII by a design house, the foundry can (in)directly obtain a configured design after design release. The foundry may then attempt to determine the functionality of STT-based LUTs by applying test patterns to both the configure design and the unresolved netlist. Figure 3 presents our novel security-driven design flow to prevent design reverse engineering. While it is fully compatible with the common-practice VLSI design flow, the proposed flow aims to introduce security in the early design stages to prevent design reverse engineering with no or minimum impact on design parametric constraints.

Along with the design constraints and the target CMOS technology node, the design security requirements and the STT technology library information are passed to the standard VLSI design flow. The design flow is continued with circuit implementation and then the logic synthesis. In the heart of flow, there exists CMOS gate selection and replacement where our novel parametric-aware dependent selection algorithm takes the synthesized gate-level netlist and carefully selects a number of CMOS gates to replace them with equivalent STT-based LUT (missing gates) implementation. Selection and replacement of CMOS gates are such that it makes it impossible to determine missing gates in any reasonable time. Contrary to some other work like [6] and [4], the security-driven hybrid STT-CMOS hybrid design flow also ensures that the design parametric constraints are not violated or only minimally impacted.

Given a timing path p containing k number of gates tt such that $tt = \{g_i \mid g_i \in gates\ defined\ in\ the\ technology\ library\}$,

the parametric-aware dependent selection method selects only a few number of gates on p and replaces them with reconfigurable STT-based LUTs. The selection of gates for replacement is random to avoid any biased decision; otherwise, determining the functionality of missing gates would be easier for an attacker. As untouched gates on p make determining missing gates possible through creating the truth table for each missing gate, all gates driving and driven by the untouched gates that are not on p are also replaced with reconfigurable units.

To resolve missing gates, an attacker may try several approaches including machine-learning attacks, SAT-based attacks, and brute-force attacks. When a STT-CMOS hybrid circuit is subject to a machine learning attack, the attacker tries to generate a partial truth table for each missing gate to narrow possible candidates (classification) and then selects select one of the candidates (predication). The machine learning attack render ineffective to determine the functionality of missing gates when the parametric-aware dependent selection is being applied because it is not possible to even generate partial truth tables. As a result, a more plausible approach for the attacker is to launch a brute force. Considering M is the number of missing gates, I is inputs accessible driving missing gates, P is the number of possible gates for each missing gate, and D is the depth of circuit, the number of required clock cycles to determine missing gates in a brute-force attack (Nbf) is

$$N_{bf} = 2^I \times P^M \times D \quad (1)$$

Equation 1 shows the exponential relationship between the number of required clock cycles and the number of missing gates and the number of inputs driving missing gates.

Table 3. The percentage of power, performance and area overhead after introducing STT-based LUT units.

Circuit	Performance degradation	Power overhead	Area overhead	Number of	size
s641	1.00	8.45	4.98	9	287
s820	2.37	5.08	1.34	2	289
s832	7.75	1.92	0.51	1	379
s953	4.55	8.03	2.38	5	395
s1196	0.00	7.95	2.64	7	508
s1238	4.45	8.13	2.73	7	529
s1488	6.7	8.18	3.47	11	657
s5378a	1.50	9.80	6.88	98	2779
s9234a	0.00	9.83	3.24	82	5597
s13207	0.00	8.21	2.60	111	7951
s15850	0.00	6.04	1.78	85	9772
s38584	0.00	5.13	1.56	166	19253

While both the machine-learning attack and the brute-force attack is ineffective, the attacker may execute the SAT-based attack to determine the functionality of missing gates. The possible candidates per STT-based LUT is not limited to a small number of gates. A 2-input STT-based LUT can realize 6 meaningful 2-input gates consisting of AND, NAND, OR, NOR, XOR, XNOR gates. 3-/4-input STT-based LUTs can also implement more than 12 meaningful gates. To exacerbate the situation for SAT-based attacks, a 4-input STT-based LUT and a 3-input STT-based LUT can be also used to implement 3-/2-input gates and 2-input gates, respectively, with connecting unused inputs of STT-based LUTs to some signals in the circuit to expand search space for SAT-based attacks. Furthermore, we can realize complex functions, such as $(A.(B \oplus C)) + D$, using a STT-based LUT instead of implementing only one simple gate. With incorporating these measures, the SAT-based attacks would also render ineffective to determine the missing gates in any reasonable time as the size of search space is significantly large even with inserting a moderate number STT-based LUTs.

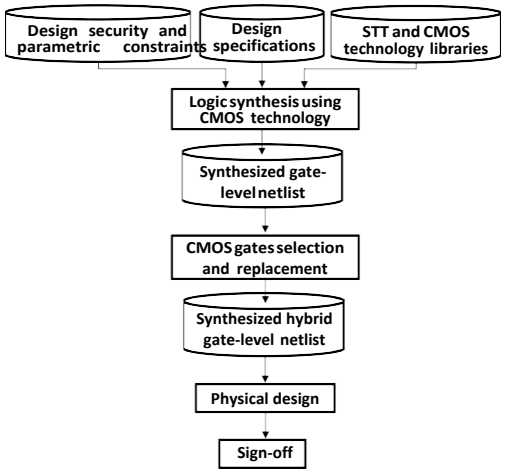


Figure 3. Our novel security-driven STT-CMOS hybrid design flow.

4. Results

To evaluate the effectiveness of functional reconfigurability against design reverse-engineering, the proposed security-driven STT-CMOS hybrid design flow is applied to several ISCAS '89 benchmarks.

Table 3 shows the impact on performance, power, and area after introducing STT-based LUT units to the selected benchmarks. While the first column of table indicates the name of circuits, the second, third, and forth columns present the relative performance degradation after applying the parametric-aware selection algorithm on the original circuits. While the circuit sizes rages from about 300 to 20,000 gates, the results indicate that the performance degradation is less or none after applying the parametric-aware selection algorithm as all STT-based LUTs are not placed on a single I/O path. Furthermore, with increasing the size of the circuit, the algorithm is provided a larger pool of gates and timing paths; therefore, STT-based LUTs are fairly distributed and a very few STT-based LUTs are located on a single timing path. The results in Table 3 signify that the relative performance degradation is almost zero for larger circuits. The results imply that for large industrial circuits the impact of STT-based LUT units on circuit performance will be very negligible.

In Table 3, we also present the relative power overhead and the number of replaced gates after applying the parametric-aware selection algorithm. With increasing the size of the circuits, more number of gates are generally chosen for replacement. On the other hand, the power overhead is considerably reduces when the size of the circuit increases. For example, s641 benchmark only consists of 287 gates and only 9 gates are replaced with STT-based LUTs. Due to the small size of the circuit, the power overhead is relatively high, i.e. 8.45%. On the opposite, s38584 benchmark consists of 19,253 gates, and 166 gates are replaced. While there is a considerable increase in the number of replaced gates, these incur only a small power overhead, i.e. 5.13%. The last column of Table 3 indicates the number of gates in the circuits excluding the number of flip-flops. Columns 8 to 10 of Table 3 presents the percentage of incurred area overhead. The results clearly indicate that the area overhead significantly reduces with increasing the size of the circuit. Collectively analyzing results in Table 3 reveals that with increasing the size of the circuit, it is possible to insert more number of STT-

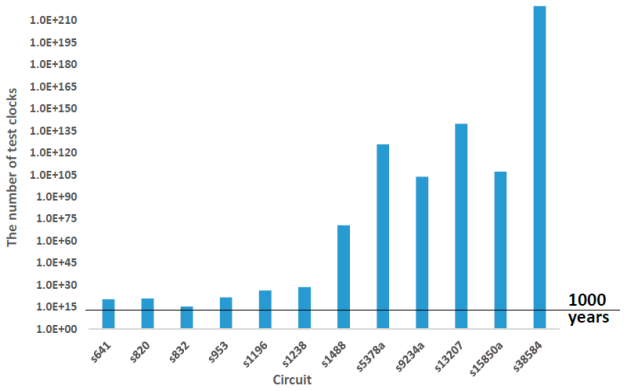


Figure 4. The number of possible required test clocks to determine the functionality of missing gates.

based LUTs with no or very negligible impact on performance, power, and area.

Figure 4 shows the number of possible required test clocks to determine the missing gates using the machine learning attacks. The results signify that even for small circuits the number of required test clocks for the parametric-aware selection is significantly high so that it would take more than 1000 years to correctly resolve a STT-CMOS hybrid circuit using modern testing equipment. For example, the analysis of s38584 benchmark shows that with introducing only 166 STT-based LUTs using the parametric-aware selection algorithm, about 6.07E+219 test clocks are required to determine their functionality while there is only about 5.13% increase in power consumption, 1.56% increase in area, and 0% performance degradation.

5. References

[1] S Giordano, et.al. "Thermal effects in magnetoelectric memories with stress-mediated switching", Journal of Physics D: Applied Physics, 46(32):325002, 2013.

[2] X. Guo, E. Ipek, and T. Soyata. "Resistive computation: avoiding the power wall with low-leakage", STT-MRAM based computing. Proceedings of the 37th annual international symposium on Computer architecture, page 371382, 2010.

[3] Siew-Kei Lam, et. a. "Exploiting FPGA-aware merging of custom instructions for runtime reconfiguration", International Workshop on Reconfigurable Communication-centric Systems-on-Chip, 2012.

[4] B. Liu and B. Wang. "Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks", In DATE 2014.

[5] A. Makosiej, et. al. "CMOS SRAM scaling limits under optimum stability constraints", International Symposium on Circuits and Systems (ISCAS), 2013.

[6] J. Rajendran and et al. "Security analysis of integrated circuit camouflaging", In ACM CCS 2013.

[7] Homayoun, Houman, Mohammad Makhzan, and Alex Veidenbaum. "ZZ-HVS: Zig-zag horizontal and vertical sleep transistor sharing to reduce leakage power in on-chip SRAM peripheral circuits", Computer Design, ICCD 2008.

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

[9] Clinton W Smullen, et. al. "Relaxing non-volatility for fast and energy-efficient STT-RAM caches", In High Performance Computer Architecture (HPCA), 2011.

[10] D. Suzuki and et. al. "Fabrication of a nonvolatile lookup-table circuit chip using magneto/semiconductor hybrid structure for an immediate power-up field programmable gate array", VLSI Circuits, 2009.

[11] Weisheng Zhao, et. al. "Spin transfer torque (STT)-MRAM-based runtime reconfiguration FPGA circuit", ACM TECS 2009.

[12] Homayoun, Houman, and Amirali Baniasadi. "Reducing execution unit leakage power in embedded processors", Embedded Computer Systems: Architectures, Modeling, and Simulation. Springer Berlin Heidelberg, 2006. 299-308

[13] Shammagari, Adarsh Reddy, Hamid Mahmoodi, and Houman Homayoun. "Exploiting STT-NV technology for reconfigurable, high performance, low power, and low temperature functional unit design", Proceedings of the conference on Design, Automation & Test in Europe, 2014.

[14] Tran, L. N., Kurdahi, F. J., Eltawil, A. M., & Homayoun, H. (2013, January). "Heterogeneous memory management for 3D-DRAM and external DRAM with QoS", In Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific IEEE.

[15] Page, Adam and Kulkarni, Amey and Mohsenin, Tinoosh, "Utilizing Deep Neural Nets for an Embedded ECG-based Biometric Authentication System", IEEE Biomedical Circuits and Systems Conference, 2015.

[16] Ashammagari, Adarsh Reddy, et al. "Reconfigurable STT-NV LUT-based functional units to improve performance in general-purpose processors", Proceedings of the 24th edition of the great lakes symposium on VLSI. ACM, 2014.

[17] Strikos, N., Kontorinis, V., Dong, X., Homayoun, H., & Tullsen, D. "Low-current probabilistic writes for power-efficient STT-RAM caches", In Computer Design (ICCD), 2013.

[18] M. Rasquinha, et. al. "An energy efficient cache design using spin torque transfer (STT) RAM", International Symposium on Low-Power Electronics and Design (ISLPED), pages 389-394, 2010.

[19] Predictive Technology Models. Online: http://ptm.asu.edu.

[20] BanaiyanMofrad, A., Homayoun, H., Kontorinis, V., Tullsen, D., & Dutt, N. "REMEDIATE: A scalable fault-tolerant architecture for low-power NUCA cache in tiled CMPs", IEEE IGCC 2013.

[21] Kulkarni, A. and Pino, Y. and French, M. and Mohsenin, T. "Real-Time Anomaly Detection Framework for Many-Core Router through Machine Learning Techniques", ACM J. Emerg. Technol. Comput. Syst, 2016.