

# Resistive Computation: A Critique

Hamid Mahmoodi<sup>†</sup>, Sridevi Srinivasan Lakshminiparam<sup>†</sup>, Manish Arora<sup>\*</sup>, Yashar Asgari<sup>\*</sup>, Houman Homayoun<sup>‡</sup>,  
Bill Lin<sup>\*</sup> and Dean M. Tullsen<sup>\*</sup>

<sup>\*</sup>University of California, San Diego

<sup>†</sup>San Francisco State University <sup>‡</sup>George Mason University

**Abstract**—Resistive Computation was suggested by [6] as an idea for tackling the power wall by replacing conventional CMOS logic with Magnetic Tunnel Junction (MTJ) based Look-Up Tables (LUTs). Spin Transfer Torque RAM (STTRAM) is an emerging CMOS-compatible non-volatile memory technology based on Magnetic Tunnel Junctions as a memory bit [3]. The principal advantage of STTRAM is that it is leakage-resistant, which is an important characteristic beyond the 45nm technology node, where leakage concerns are becoming a limiting factor in microprocessor performance. Although STTRAM is a good candidate for replacing SRAM for on-chip memory, we argue in this article MTJ-based LUTs are unnecessarily expensive in terms of area, power, and performance when implementing fixed combinational logic that does not require the reprogramming ability provided by MTJs.

**Index Terms**—Resistive computation, magnetic-tunnel junctions, spin transfer torque RAM, MRAM, dynamic current-mode logic, leakage power.

## 1 INTRODUCTION

WITH the scaling of CMOS technology, leakage power has emerged as a major barrier to high performance computing circuits and high density SRAM arrays [4]. Spin Transfer Torque RAM (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 [3]. The use of MTJs has recently been explored for building low power programmable Look-Up Tables (LUT) used in Field Programmable Gate Arrays (FPGA) [9], [11]. In both memory and FPGA applications, re-configurability is a key requirement and exploits the programmability of MTJs. Hybrid CMOS-STTRAM FPGA solutions are particularly attractive because the write operation, which is a high power operation in STTRAM, happens very infrequently in FPGAs [9]. To fully exploit the benefits of MTJs in this type of FPGA, additional circuitry and optimizations were found to be necessary [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. 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 [10]. An alternative to this approach has been to realize logic in memory by using LUTs that are built based on MTJs [6]. A LUT, such as those used in FPGAs, offers programmability and includes write circuitry for changing the state of the MTJs. 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. In [6], such read-only MTJ-based LUTs are used to replace custom CMOS logic with the hope of achieving low power. In this article, we argue that for a fixed logic implementation in LUT circuits, the MTJs can be replaced by short or open circuits, and this replacement will always

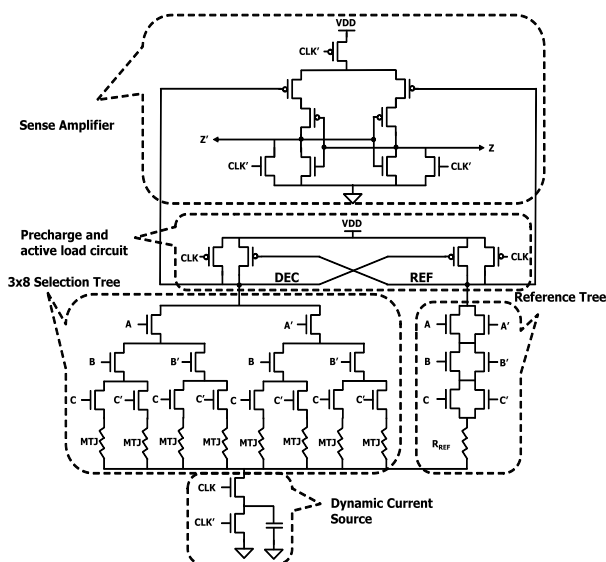


Fig. 1: 3-input MTJ-based LUT [6] [11].

improve the circuit power and performance. Moreover, we show that the leakage reduction of these LUT styles, which was mistakenly attributed to MTJs in [6], is in fact due to the stacking of transistors in this style. In fact, replacing the MTJs with short and open results in lower leakage too.

The replacements of MTJs with short or open circuits will also create opportunities for simplification of the LUT circuit for additional performance improvement. By replacing an MTJ that is at a high state with an open circuit, the transistors in the path above it can also be eliminated, and by shorting the low state MTJs, some paths could possibly be merged, resulting in a reduced number of transistors.

## 2 RESISTIVE COMPUTATION VIA MTJ-BASED LUTS

Fig. 1 shows the schematic of a 3-input MTJ-based LUT that was used in [6]. This LUT was obtained by eliminating the write circuitry from the original design in [11]; hence,

it is a read-only LUT. The read-only nature of this design requires that the state of the MTJs be initialized during manufacturing. An MTJ is selected by using the pull-down NMOS selection tree, and the current of the dynamic current source is divided between the selected MTJ and the reference resistor, resulting in a low swing differential voltage on nodes DEC and REF during the evaluation phase when the clock (CLK) is high. This low swing voltage is then amplified using a sense amplifier stage to achieve full voltage swing outputs (Z and Z').

### 3 REDUCING CIRCUIT COMPLEXITY

In principle, the resistance of the reference tree (Fig. 1) should be in between the high and low values of the resistance of the selected path on the selection tree. Moreover, the larger the difference between these two resistances, the greater are the robustness and performance of the circuit against noise and process variations. Hence, we can argue that the higher the resistance of the high state of the MTJ or the lower the resistance of the low MTJ state, the better are the performance and reliability of the circuit. Therefore, we propose to replace a high state MTJ with an open circuit (infinite resistance) and a low state MTJ with a short circuit (zero resistance). This will enhance the current differential between the left and the right hand side trees (Fig. 1), resulting in reduced delay. Moreover, the open path will eliminate the current of the path, resulting in reduced switching and leakage power. The shorted path, on the other hand, will slightly increase the leakage of the path. However, this increase is not as much as the reduction obtained by opening a high resistance path, because the current is limited by the chain of the transistors in the path. Also, the current is limited by the NMOS clocked transistor that acts like a power gating switch, and the leakage is not very sensitive to the MTJ resistance. Fig. 2 shows the plots of power, delay, and energy for LUT sizes ranging from 2 inputs to 8 inputs. Each LUT is examined under four scenarios of high and low MTJ resistance states (RH and RL). This data is obtained for the cases where 50% of the MTJs are at the high state, and the remaining 50% at the low state. Simulations are performed in a 32nm predictive technology [1], where the expected RH and RL values are at 6.25K $\Omega$  and 2.5K $\Omega$ , respectively [6]. It is evident that replacing a high state MTJ (RH) with an open circuit and low state MTJ (RL) with short is beneficial in all aspects of power, delay, and energy. The power and leakage benefit will be more substantial when more MTJs are at the high state. After replacing a high state MTJ with an open circuit, the transistors above it in the selection tree can be eliminated, resulting in area reduction (Fig. 2(e)).

### 4 ADDITIONAL CIRCUIT SIMPLIFICATIONS

After the replacement of MTJs with short or open circuits, the NMOS pull-down tree can be optimized to minimize the number of transistors and hence further improve the performance. The results presented in the previous section do not include these optimizations. Consider the example of a 3-input AOI gate shown in Fig. 3. Fig. 3(a) shows the implementation using the original MTJ-based LUT. The AOI functionality is realized by manufacturing the three rightmost MTJs at the low resistance state (RL) and the rest at the high resistance state (RH). Fig. 3(b) shows the LUT after the replacement of each RH with an open circuit

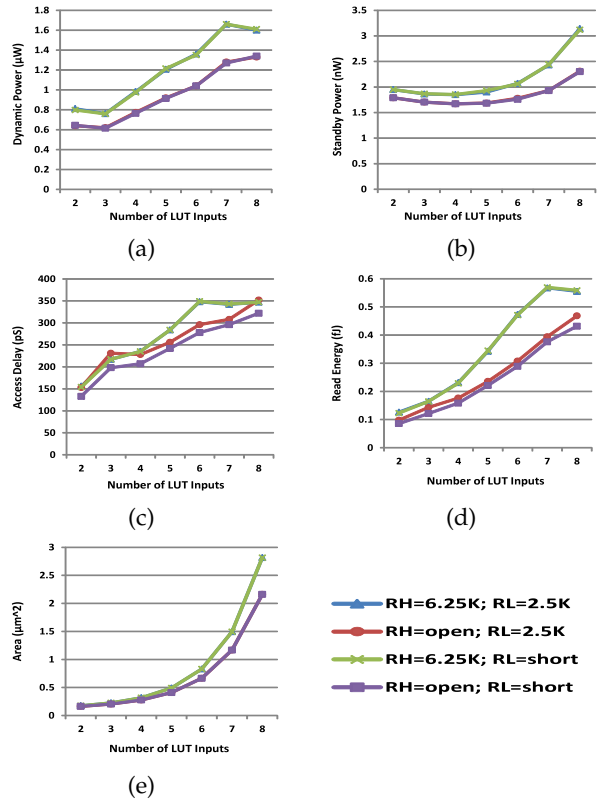


Fig. 2: Power, leakage, performance, and area results of LUTs with high and low state MTJs (RH, RL), replaced with open and short, respectively.

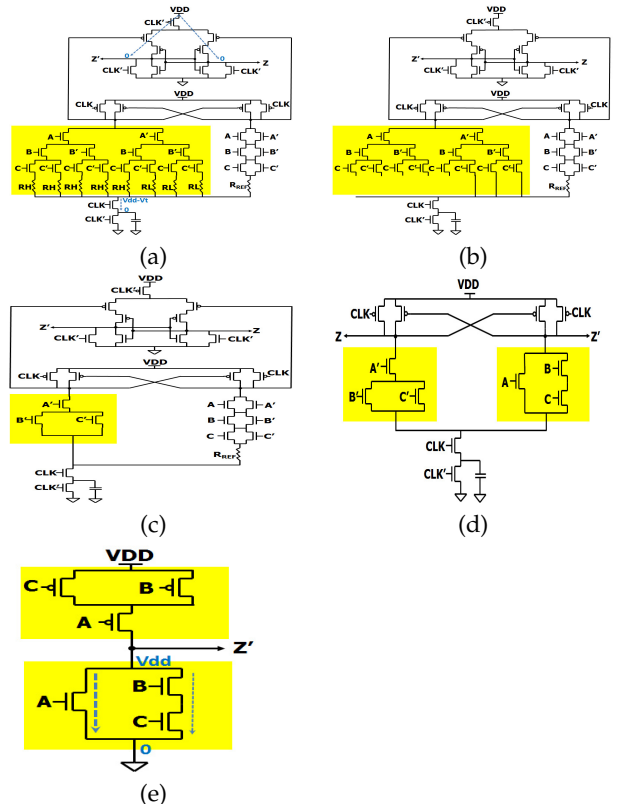


Fig. 3: Implementation of logic function  $Z=A+BC$  in (a) MTJ-Based LUT (b) MTJ-less LUT (c) Minimized tree LUT (d) DCML (e) Static CMOS.

and each RL with a short circuit (MTJ-less LUT). As can be seen, the NMOS selection tree can now be reduced in size by eliminating the open and/or redundant paths, which results in the circuit shown in Fig. 3(c) (minimized tree LUT). Replacing the reference tree with the dual of the minimized tree, and removing the sense-amp, will result in the already known Dynamic Current Mode Logic (DCML) style [2] which has low voltage swing outputs (Fig. 3(d)).

The DCML circuit style was originally presented in [2] and compared against static CMOS. It was shown in [2] that while this style shows better speed than static CMOS due to its dynamic operation, its power consumption is higher except for very complex functions such as high fan-in XOR gates. Hence in this paper, we will not repeat this comparison and only focus on comparing the original MTJ LUT style Fig. 3(a), the MTJ-less LUT style Fig. 3(b), and the simplified MTJ-less (min-tree) LUT style Fig. 3(c), with the static CMOS Fig. 3(e).

Table 1 shows the simulation results of the above circuit styles for logic gates of various complexity implemented in a predictive 32nm technology [1]. All the results are normalized to the corresponding results for a static CMOS implementation. The capacitance of the dynamic current supply of the MTJ-based LUT style (Fig. 1) is optimized to achieve the minimum Power-Delay Product (PDP). The same capacitance is kept for other logic styles.

It is clear that the MTJ-based LUT style is not competitive with the MTJ-less styles irrespective of the circuit complexity or the metric. The MTJ-less LUT outperforms the MTJ-based LUT in all metrics and for all logic gates and the simplified (min-tree) LUT shows even better results. The power results for the LUT styles included the clock power as well as the sense amplifier power. Also, we notice that the LUT styles have maximum output switching activity of 200% irrespective of the data switching pattern. That is because in every cycle the differential output of the LUT style will make two sightings (one high-to-low in the precharge phase and one low-to-high in the evaluation phase), irrespective of the input pattern. This high switching activity results in excessively high active power for the LUT styles at low output switching activity factors. The static CMOS is the fastest logic style and shows the lowest active power consumption. The static CMOS style shows better energy per output switching results as compared to the MTJ-based LUT style.

Except in the cases of high fan-in NAND and NOR gates, the LUT styles show lower standby leakage than the static CMOS counterpart. In the high fan-in NAND and NOR gates, there is a long chain of series connected transistors that suppress the leakage significantly due to the stacking effect [8]. In these cases, the LUT styles show more leakage due to extra leakage on the sense amplifier stage. On the other hand, high fan-in NAND and NOR gates are delay inefficient when implemented in single stage in static CMOS [7], and therefore, this leakage advantage of static CMOS will disappear if these gates are implemented in a multi-stage fashion by cascading NAND/NOR gates of lower fan-in.

Why are the LUT styles low leakage? In [6], low leakage property of the MTJ based LUTs is attributed to the use of MTJs. However, our results clearly show that MTJs play no useful role in leakage reduction. In fact, by eliminating the MTJs (replacing them with short or open circuits), the leakage is further reduced (Table 1). The leakage reduction observed

for LUT circuits, as compared with static complementary CMOS style, is because of the stacking of transistors [8] that occurs due to the addition of the clocked transistors in the NMOS pull-down (dynamic current supply) and the clocked PMOS in the sense amplifier.

To better understand this, consider the static CMOS circuit that implements  $Z'=A+BC$  with the leakage paths shown for the input state  $A=B=C=0$  in Fig. 3(e). There are two leakage paths as shown in Fig. 3(e). Notice that the NMOS A leaks significantly because it has maximum drain to source voltage of  $V_{dd}$ . The same circuit in LUT style with its leakage paths is shown in Fig. 3(a). The leakage paths are shown again assuming  $CLK=A=B=C=0$  in the standby mode. Notice  $CLK'=V_{dd}$  in the standby mode, and hence the lower NMOS in the dynamic current supply is ON and discharging the capacitance to zero volts. The leakage of the logic stage is now limited to a single clocked NMOS leakage regardless of the complexity of the logic function, and this leakage is reduced because this transistor has less drain-to-source voltage ( $V_{DS}=V_{dd}-V_t$ ). The threshold voltage ( $V_t$ ) drop is caused by the NMOS evaluation tree (or reference tree), because the NMOS transistors cannot charge their source voltage beyond the gate voltage ( $V_{dd}$ ) minus the threshold voltage [7]. The leakage of the sense amplifier is also small because of the stacking effect (two OFF PMOSes in series).

## 5 CASE STUDY: 3-BIT ADDER EXAMPLE

We use a 3-bit adder as a test case to compare the alternative circuit styles presented in the previous sections. The 3-bit static CMOS adder was implemented using a ripple-carry scheme with a first-stage half adder followed by two full adders. The half adder and full adders were implemented in the static complementary CMOS style. The MTJ and MTJ-less LUT styles were implemented by using four LUTs to produce each output from the six inputs. The first output bit is only a function of the least significant bits of the input operands and hence realized by a 2-input LUT. The second output bit is a function of the two least significant bits of the input operands and hence implemented by a 4-input LUT. The last sum output bit and carry output bit are functions of all the inputs and hence implemented using 6-input LUTs. Since sum outputs are not on the critical path (i.e. carry generation path), the capacitance of the dynamic current sources for the sum LUTs are minimized for low power, whereas the capacitance of the carry LUT is optimized for minimum power-delay product. Sense amplifiers are preset at the outputs of each LUT to produce full-swing outputs and their power and delay overhead is counted.

Table 2 shows the results of the 3-bit adder implemented in the alternative logic styles. All results are normalized to the corresponding results for the static CMOS style. The active power results are measured by applying random input stimuli. It is observed again that MTJs provide no advantage and irrespective of the metric used MTJ-less and simplified (minimum tree) LUTs offer better results. The only advantage of the MTJ-based LUT over CMOS is less standby power, and that is not caused by MTJs as explained before. In fact the standby power is further reduced in the MTJ-less and min-tree LUTs. Overall static CMOS style is the best solution, except in terms of leakage power. The delay of the minimum tree LUT style is better than static CMOS due to its dynamic operation. By using static CMOS circuits instead

of MTJ-based LUTs, area is reduced by a factor of 3.89X, active power by a factor of 5.2X, and delay by a factor of 2.84X. These results clearly indicate that there is no benefit associated with the use of MTJs for implementing fixed logic and the leakage advantage is not attributed to the use of MTJs and rather the stacking of transistors created in this logic style.

## 6 OTHER ISSUES WITH MTJ-BASED LUTS

Dynamic logic styles are generally less robust and more susceptible to noise and Process, Voltage, and Temperature (PVT) variations as compared to the static CMOS counterpart [5], and the LUT styles are no exception. The last row in Table 2 shows the normalized delay sensitivity to threshold voltage ( $V_{th}$ ) variation. The results are obtained by applying 30 mV of inter-die shift in  $V_{th}$ . The results show that the LUT styles are more sensitive to process variations. The comparative voltage sensitivity results are expected to be similar to the  $V_{th}$  sensitivity results because the performance of a circuit (or current of a transistor) depends on  $V_{dd}$ - $V_{th}$ , and therefore, a circuit with higher  $V_{th}$  sensitivity is expected to also show higher voltage ( $V_{dd}$ ) sensitivity. The temperature sensitivity will depend on the combined effect of the temperature sensitivities of MTJ resistance and transistor performance. Robustness is one of the major limitations for use of dynamic logic styles in general and this issue gets worse in nano-scale due to increased process variations [5]. Besides reliability issues, dynamic logic styles are not supported by electronic design automation tools in an automated design flow and this further limits the usefulness of MTJ-based LUT styles.

## 7 CONCLUSIONS

We have shown that there is no advantage associated with the use of MTJs for realizing fixed logic in read-only STTRAM-based LUTs. In fact, a custom solution based on the static CMOS easily outperforms the MTJ-based LUTs in all metrics except leakage power. The leakage power saving of MTJ-based LUT is not attributed to the MTJs and rather the stacking of transistors in this logic style, and in fact further leakage reduction is observed by replacing the fixed MTJs with short and open circuits. The use of MTJs is viable for large memory arrays (STTRAM) and writable LUTs that are needed for implementing reconfigurable functional units and FPGAs. Existing research shows the benefits of MTJs in STTRAM and reprogrammable FPGAs.

## REFERENCES

- [1] Predictive technology models. <http://ptm.asu.edu/>.
- [2] M. Allam and M. Elmasry. Dynamic current mode logic (dycml): a new low-power high-performance logic style. *Solid-State Circuits, IEEE Journal of*, 36(3):550–558, mar 2001.
- [3] T. Andre et. al. A 4-mb 0.18- $\mu\text{m}$  1t1mtj toggle mram with balanced three input sensing scheme and locally mirrored unidirectional write drivers. *Solid-State Circuits, IEEE Journal of*, 40(1):301–309, jan. 2005.
- [4] S. Borkar. Design challenges of technology scaling. *IEEE Micro*, 19(4):23–29, July 1999.
- [5] C. Cornelius, F. Grassert, S. Koppe, and D. Timmermann. Deep submicron technology: Opportunity or dead end for dynamic circuit techniques. In *VLSI Design, 2007. Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on*, pages 330–338, 2007.

- [6] 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, New York, NY, USA, 2010. ACM.
- [7] A. C. Jan M. Rabaey and B. Nikolic. *Digital Integrated Circuits*. Prentice Hall, 2003.
- [8] S. Narendra et. al. Scaling of stack effect and its application for leakage reduction. In *Proceedings of the 2001 international symposium on Low power electronics and design*, ISLPED '01, pages 195–200, New York, NY, USA, 2001. ACM.
- [9] 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.
- [10] 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.
- [11] 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.

TABLE 1: Comparison of circuit style alternatives ( $\alpha$ : output switching activity).

Gate	Metric	MTJ Based LUT (Fig 3a)	MTJ Less LUT (Fig 3b)	Min Tree LUT (Fig 3c)	Static CMOS (Fig 3e)
NAND2	Delay	6.46	5.19	4.69	1
	Active Power( $\alpha=10\%$ )	90.35	51.65	51.9	1
	Active Power( $\alpha=30\%$ )	30.12	17.22	17.3	1
	Standby Power	0.48	0.45	0.45	1
	Energy per Switching	58.36	26.8	24.34	1
NAND4	Delay	4.49	2.99	2.61	1
	Active Power( $\alpha=10\%$ )	76.73	43.86	43.71	1
	Active Power( $\alpha=30\%$ )	25.57	14.62	14.57	1
	Standby Power	0.96	0.86	0.83	1
	Energy per Switching	34.45	13.11	11.4	1
NAND8	Delay	2.49	1.50	1.40	1
	Active Power( $\alpha=10\%$ )	34.53	14.75	14.32	1
	Active Power( $\alpha=30\%$ )	11.51	4.91	4.77	1
	Standby Power	8.05	5.23	3.79	1
	Energy per Switching	8.59	2.21	2	1
NOR2	Delay	4.85	3.86	3.52	1
	Active Power ( $\alpha=10\%$ )	80.2	53.35	53.15	1
	Active Power( $\alpha=30\%$ )	26.73	17.78	17.71	1
	Standby Power	0.51	0.48	0.48	1
	Energy per Switching	38.89	20.59	18.7	1
NOR4	Delay	3.06	2.01	1.77	1
	Active Power( $\alpha=10\%$ )	24.25	13.98	14.18	1
	Active Power( $\alpha=30\%$ )	8.08	4.66	4.72	1
	Standby Power	1.06	0.95	0.93	1
	Energy per Switching	7.42	2.8	2.5	1
NOR8	Delay	1.51	0.90	0.84	1
	Active Power( $\alpha=10\%$ )	17.1	7.37	7.1	1
	Active Power( $\alpha=30\%$ )	5.7	2.45	2.36	1
	Standby Power	10.83	7.03	5.14	1
	Energy per Switching	2.58	0.66	0.59	1
XOR2	Delay	4.95	4.03	3.89	1
	Active Power( $\alpha=10\%$ )	22.45	17.45	17.5	1
	Active Power( $\alpha=30\%$ )	7.48	5.81	5.83	1
	Standby Power	0.13	0.12	0.12	1
	Energy per Switching	11.11	7.03	6.8	1
XOR4	Delay	4.18	3.17	2.95	1
	Active Power( $\alpha=10\%$ )	90.06	73.25	71.18	1
	Active Power( $\alpha=30\%$ )	30.02	24.41	23.72	1
	Standby Power	0.04	0.04	0.04	1
	Energy per Switching	37.64	23.22	21	1
XOR8	Delay	3.12	2.55	2.10	1
	Active Power( $\alpha=10\%$ )	63.93	55.06	34.06	1
	Active Power( $\alpha=30\%$ )	21.31	18.35	11.35	1
	Standby Power	0.03	0.02	0.01	1
	Energy per Switching	19.94	14.04	7.15	1

TABLE 2: Comparison of 3-bit adder results in alternatives styles.

Metric	MTJ-Based LUT	MTJ-Less LUT	Min-Tree LUT	Static CMOS
Delay	2.84	2.13	0.86	1.00
Active Power	5.2	4.25	3.87	1.00
Standby Power	0.17	0.15	0.14	1.00
PDP	14.77	9.05	3.33	1.00
Area	3.89	3.89	1.67	1.00
Delay sensitivity to $V_{th}$ variation	1.16	1.22	1.32	1.00