Integration of Threshold Logic Gates with RRAM devices for Energy Efficient and Robust Operation

Jinghua Yang, Niranjan Kulkarni, Shimeng Yu, Sarma Vrudhula
School of Computing, Informatics and Decision Systems Engineering
Arizona State University, Tempe, Arizona, 85281
Email: (jinghua.yang, nkulkar1, shimeng.yu, vrudhula)@asu.edu

***Abstract*—Differential mode threshold-logic gates can be programmed to compute complex logic functions within a single cell, resulting in significant reduction in area and power. However the circuit yield reduces if they are operated at low voltages. This paper describes a novel integration of RRAM with such threshold-logic gates to achieve robust, low voltage (0.6V for 65nm technology) and energy efficient computation of threshold-logic functions. Below 0.6V, we observed that the performance(and thereby, energy delay product) of conventional CMOS circuits degrades substantially compared to the proposed threshold-logic circuits. The improvement in performance and energy of the new circuit architecture are demonstrated while considering process variations in both the MOSFET and RRAM devices. For each threshold function implementable by threshold-logic gate, comparison of energy, delay and energy delay product with equivalent CMOS implementation is given. The advantages in area, energy and delay of threshold logic implementations over conventional CMOS logic gates is demonstrated by two commonly used functional components.**

I. Introduction

In this paper, we explore an alternate method of computing logic functions, which is based on using *threshold* functions as logic primitives. The main challenges in using threshold logic are related to *robustness* due to process variations, which is further exacerbated at low voltages. The main objective of this paper is to demonstrate that threshold logic cells can be made robust at low voltages and can indeed achieve significant improvements in area and energy-delay product when compared to conventional static CMOS logic. An important *long term* goal of this effort is to ensure that the design of the threshold logic primitives fit within existing design flows (i.e., conform to industrial design practices and standards for cell libraries) so that they can be processed by commercial design tools.

A Boolean function $f(x_1, x_2, \cdots, x_n)$ is a threshold function if there exist n weights $(w_1, w_2, \cdots, w_n)$ and a threshold T such that

$$f(x_1, x_2, \cdots, x_n) = \begin{cases} 1 & \text{if } \sum_{i=i}^{n} w_i x_i \geq T \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

$f(x_1, x_2, \cdots, x_n)$ is represented as $[w_1, w_2, \cdots, w_n; T]$, and without loss of generality, we assume that w_i and T are integers. Here are two examples of threshold functions: (1) $f(a, b, c) = a \vee bc \equiv 2a + b + c \geq 2 \equiv [w_a, w_b, w_c; T] = [2, 1, 1; 2]$; (2) $g(a, b, c, d, e) = ac(b \vee d \vee e) \vee de(a \vee bc) \vee ab(d \vee e) \equiv 2a+b+c+d+e \geq 4 \equiv [w_a, w_b, w_c, w_d, w_e; T] = [2, 1, 1, 1, 1; 4]$. However, $f(a, b, c, d) = ab \vee cd$, which appears similar to example (1), is not a threshold function.

Since the threshold function is a Boolean function, it can be realized by a network of conventional logic gates. We exclude such implementations, and define a *threshold logic gate* (TLG) as a non-decomposable primitive circuit that computes the weighted sum and evaluates the predicate in Equation (1) by a comparison of some physical quantity such as charge, current, or voltage. In the language of neural networks, a TLG is known as a perceptron [1].

The literature on threshold functions is extensive, dating back to the 1950s [2]. Restricting our attention to logic circuits, we note that the standard logic primitives (AND/OR) are threshold functions. Hence a network of TLGs may be viewed as a generalization of a conventional AND/OR network. However, TLGs can potentially be far more compact (fewer gates and fewer levels) than AND/OR networks, leading to equally large reductions in power and delay. However this can be realized only with efficient circuit implementation of a TLG.

More than fifty different implementations of TLGs have been reported [3], and the most promising, in terms of speed and power, are those that are based on *differential* mode logic [4]–[7]. A *differential threshold gate* (DTG) relies on clocked differential comparison between two banks of configurable conductance values, and therefore can be viewed as a combination of current and conductance-based threshold logic. However, the major drawbacks shared by nearly all threshold logic circuit architectures is their sensitivity to process variations and unsuitability for *low voltage* operation. In this paper we present a solution to these problems, and show that the potential advantages of DTGs, namely, smaller, faster, lower power and robust circuits are possible at low voltage.

This paper is organized as follows: **(1):** Section II describes the basic architecture of a differential threshold logic gate; **(2):** Section III demonstrates the poor robustness of DTGs at low voltages due to process variations, and shows how the cell can be augmented with a resistive network to achieve a tradeoff between a cell's functionality, the resistance value and the failure rate, and compares the resulting threshold gates and their CMOS equivalents in terms of energy, delay and area; **(3):** Section IV describes and justifies the use of oxide-based resistive RAMs (RRAM) for the resistive network; **(4):** Section VII compares implementations of two common

NANOARCH '14, Jul 8-10, 2014, Paris, France
 ISBN: 978-1-4503-2834-0/14/07…$15.00

circuit blocks constructed using DTGs and their pure CMOS equivalents.

II. Differential Threshold Gate Architecture

Figure 1 shows the schematic of a DTG. It consists of 5 main components: (1) a differential sense amplifier, which consists of two cross coupled NAND gates, (2) a SR latch, (3) two discharge devices, (4) left (**LIN**) and right (**RIN**) input networks, and (5) a network of resistors. DTG-n refers to a DTG with n inputs in the LIN and the RIN.

The DTG circuit is operated in two phases. In the *reset phase*, $clk = 0$, which activates the two discharge devices M_{11} and M_{12}, pulling nodes N_5 and N_6 low. These turn off M_7 and M_8, and turn on M_1 and M_4, resulting in both nodes N_1 and N_2 being set to logic 1. The SR latch is designed to keep its previous state during this phase. *Evaluation phase* takes place when $clk : 0 \rightarrow 1$. As a result, both N_5 and N_6 will begin a $0 \rightarrow 1$ transition. Now assume that the number of active pFETs in the LIN are greater than the number of active pFETs in the RIN. This ensures that the conductance of the LIN is greater than that of the RIN. The $0 \rightarrow 1$ transition on N_5 takes places before the same occurs on N_6. Consequently, M_1 turns off and M_7 turns on before the same happens with M_4 and M_8. As N_1 begins its $1 \rightarrow 0$ discharge through M_5 and M_7, it stops N_2 from discharging by turning off M_6, and turning on M_3 before M_8 is turned on. As a result N_2 returns to 1. The final state will be $N_1 = 0, N_2 = 1$, which sets output of the SR latch to 1. In summary, if the number of on-transistors in the LIN exceeds the number in the RIN, then the threshold inequality evaluates to *true* and output is a 1, otherwise the inequality is *false* and the output is 0.

Functionally, a DTG can be viewed as a *complex, multi-input* edge-triggered D-flipflop (D-FF). Whereas a D-FF computes the identity function ($Q = D$) on a clock edge, a DTG computes some threshold function $Q = f(x_1, x_2, \cdots, x_n)$ on the clock edge. The specific function is determined by how signals are connected to its inputs. In general, a DTG has a lower setup time than a D-FF while its *clock-to-Q* delay is comparable. A DTG also presents a lower input capacitance than a D-FF. Thus, if some or all of the cone of logic that ends at the input of a D-FF is a threshold function, then that threshold logic portion and the D-FF can be replaced by a single DTG gate, resulting in the significant elimination of logic, and reduction in the load and timing constraint presented by a DTG, and reduction of the size of the logic that feeds the DTG. All of these contribute to reducing the area and power when DTGs are judiciously incorporated in logic networks.

III. Low Voltage Operation

Standard cell layouts of DTG circuits based on Figure 1 (without the resistor network) were carried out for $n = 3, 5, 7, 9$, using a commercial 65nm LP process, with nominal $V_{dd} = 1.2V$. Optimal sizing of the input network, the sense-amp and the output latch were performed to minimize delay and minimize the circuit functional failures in the presence of process variations based on 100,000 Monte Carlo simulations.

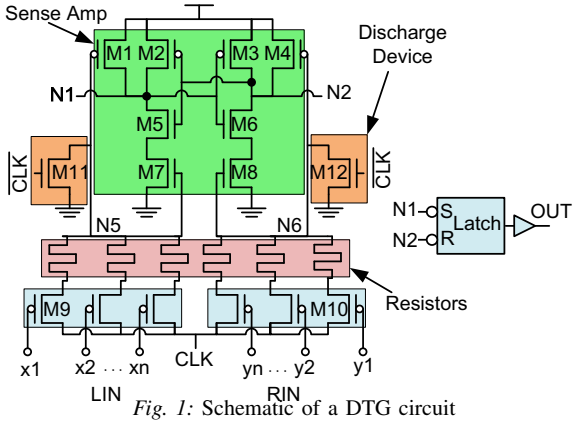

Fig. 1: Schematic of a DTG circuit

Unfortunately, failures begin to manifest as soon as the supply voltage is reduced below $1.08V$. This characteristic is shared by all DTG architectures. **Note:** The reason for the minimum voltage of $0.6V$ will be explained shortly.

In this section, we describe the necessity of the resistor network and how it helps low voltage operation of the DTG. Table I shows the results of 100,000 Monte Carlo simulations of DTG-7 (schematic only) at low voltages at nominal corner, $25\,^{\circ}\mathrm{C}$, with minimum size input transistors and the output buffer sized to match the minimum drive strength of standard D-FF in CMOS library. Table I shows significant functional failures at low voltages. The case $K{:}K-1$ represents the situation where there are K active transistors in the LIN and $K-1$ active transistors in the RIN, and vice-versa. The higher the value of K, the greater the number of input transistors and the complexity of the functions that can be implemented with a DTG. The number of functions with $K \leq 3$ is too small and provide no substantial advantage over conventional logic implementations. It is only with $K \geq 4$, do we see a significant compaction of logic and reduction in power with the use of DTGs.

We now examine how the resistor network shown in Figure 1 might help in improving the robustness of the DTG at low voltages. Figure 2 shows a simplified version of one of the input networks, with and without the resistor network. Consider Figure 2(a), which depicts the evaluation of the DTG. Let t_5 (t_6) denote the time when $N5$ ($N6$) reach the threshold voltage of M_7 (M_8), and t_{sen} denote the minimum time difference between t_5 and t_6 for the sense amplifier to correctly determine the output. Thus the DTG correctly computes the function if

$$\Delta t = \begin{cases} t_6 - t_5 \geq t_{sen} & \text{output = 1,} \\ t_5 - t_6 \geq t_{sen} & \text{output = 0.} \end{cases} \quad (2)$$

Let C_5 denote the total capacitance of node $N5$, which includes the gate capacitances of M_1, M_7, the drain capacitances of M_{11} and those of the transistors in the LIN. $N5$ and $N6$ are initially discharged to 0. When the clock rises from $0 \rightarrow 1$, $N5$ and $N6$ rise to V_{dd}. The active pFETs in the LIN and RIN immediately enter the saturation region, where current is $I_s = \mu_p C_{ox}(W/L)(V_{dd} - \mid V_{thp} \mid)^2$. Assuming that there are

VDD	Cases			
	2:1	3:2	4:3	5:4
0.7	33	713	3232	8478
0.65	253	2185	6697	13056
0.6	1500	5842	12153	19590

TABLE I: DTG failures in 100K MC simulations without resistive network

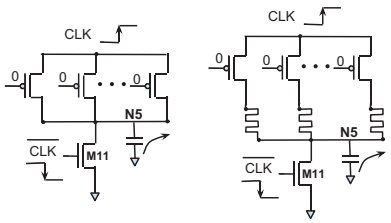

Fig. 2: Simplified input network of DTG

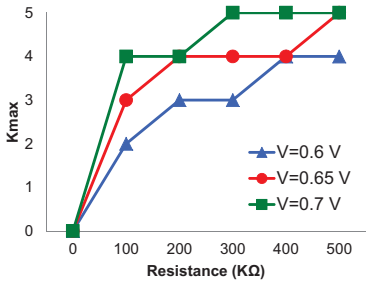

Fig. 3: DTG functionality vs R

K and $K-1$ active pFETS in the LIN and RIN, $C_5 = C_6 = C$, and $V_{th7} = V_{th8} = V_{thn}$, then t_5, t_6 and the corresponding Δt are approximately given by

$$t_5 \approx \frac{C_5 V_{th7}}{K I_s}, \quad t_6 \approx \frac{C_6 V_{th8}}{(K-1) I_s}, \quad \Delta t \approx \frac{C V_{thn}}{K(K-1) I_s}.$$

Note that the above approximate relations explain the trend in Table I for a fixed V_{dd}, and as K varies. Similarly for a fixed K and as V_{dd} decreases, failures increase because t_{sen} increases as a result of reduced discharge currents through the sense amplifier.

Figure 2(b) shows a resistance R_H in series with each pFET. If R_H is relatively very large, when the clock transitions from $0 \rightarrow 1$, most of the voltage drop will be across the resistor, and the small V_{ds} across the pFET will force it to operate in the linear region. In this case, pFET is very close to a linear resistor, with current $I \propto (V_{dd} - \mid V_{thp} \mid) V_{ds}$. If R_H is sufficiently large, then the resistance of the pFET, which is $R_{lin} \approx 1/(\mu_p C_{ox}(W/L)(V_{dd} - \mid V_{thp} \mid)$, is negligible relative to R_H. In this case t_5, t_6 and Δt are approximated by

$$t_5 = -\frac{R_H C_5}{K} \ln\left(1 - \frac{V_{th7}}{V_{dd}}\right), \quad t_6 = -\frac{R_H C_6}{K-1} \ln\left(1 - \frac{V_{th8}}{V_{dd}}\right),$$

$$\Delta t' = t_6 - t_5 \approx -\frac{R_H C}{K(K-1)} \ln\left(1 - \frac{V_{thn}}{V_{dd}}\right).$$

In a 65nm LP process, $I_s \approx 4.11 \mu A$, for a minimum size pFET, when $V_{dd} = 0.6V$. And typically, $V_{thn} = 0.4233V$, $V_{thp} = -0.43V$. For $R_H = 500K\Omega$, $\Delta t' \approx 6.12 \Delta t$, showing a substantial improvement in the robustness of a DTG at low voltages. To verify this, 100,000 Monte Carlo simulations were performed on DTG-7, considering the global variations and local mismatch in the CMOS devices, for various values of R_H. For each value of R_H, the maximum value of K in K:K-1 case that satisfied the robustness criterion of 99.99% successes was computed. Simulation results are shown in Figure 3.

The plot shows a significant improvement in the robustness of a DTG with the addition of resistors in series with the input network. At $V_{dd} = 0.6V$, with minimum $400K\Omega$ resistor, all threshold functions with a 4:3 combination of active devices in the LIN and RIN can be implemented. Restricted to DTG-7, this constitutes a total of 29 functions. The set of functions implementable is actually larger because the subset of DTG-9, DTG-11 and DTG-13 functions with worst-case $4:3$ combinations can also be implemented. To implement all DTG-9 functions (additional 42 functions) the V_{dd} has to be increased to 0.65, and R_H to $500K\Omega$. The important take-away from this discussion is that R_H can be reduced and compensated by increasing the supply voltage.

In general, lowering the supply voltage will significantly increase the delay. This is true for both conventional CMOS logic and DTGs. However, reducing the supply voltage of a DTG requires larger resistances in the input networks. To realize R_H in the range of a few $100K\Omega$, as required by DTGs, in a CMOS process is impractical. Fortunately, emerging memory technologies offer a solution. In the following section, we propose the use of oxide based resistive random access memory (RRAM) devices as resistors for the DTG. As demonstrated in [8], excellent and stable resistance values were achieved for $R_H = 500K\Omega$. At this value, the minimum V_{dd} that met the robustness criteria for DTGs was 0.6V. This is the reason for using $V_{dd} = 0.6V$ as the minimum supply voltage for the DTGs.

IV. Resistor Network

The oxide-based resistive random access memory (RRAM) technology [9] is an emerging candidate for next-generation non-volatile memory (NVM). Here we use RRAM as a CMOS compatible nano-scale resistor. For our application, its resistance need only be set once. Hence, technically speaking we are using an RRAM as a RROM. However to avoid confusion we will continue to refer to it as an RRAM. Used in this way, an RRAM has excellent scalability (¡10 nm) and good retention (>10 years). Other NVM candidates such as spin-torque-transfer magnetic random access memory (STT-MRAM) [10] and phase change memory (PCM) [11] can also be used. However, they have some undesirable features: the resistance of STT-MRAM is relatively low (a few kilo Ohm), and PCM has a well-known time-dependent resistance drift (even without voltage stress) problem.

One of the concerns about RRAM is the resistance variation from device to device. This is largely related to the manufacturing process control. Materials engineering such as multi-layer oxide design can restrict the resistance variation to ¡10% standard deviation around the medium off-state resistance of $500K\Omega$ [8]. It can be expected that as the RRAM technology matures, the manufacturing yield and process variation will be further improved. The second concern is the time-dependent

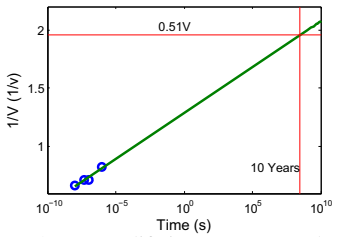

Fig. 4: RRAM lifetime vs stress voltage

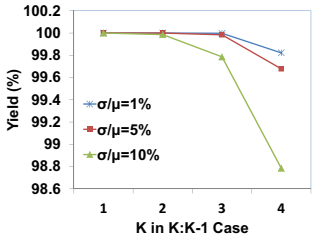

Fig. 5: DTG-7 yield with RRAM

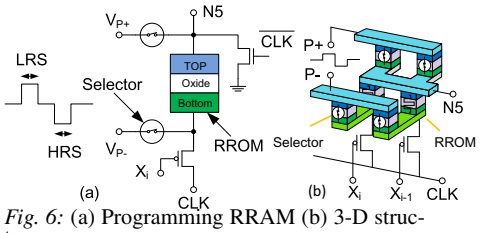

Fig. 6: (a) Programming RRAM (b) 3-D structure

resistance drift under voltage stress. There is a well-known exponential voltage-time relationship in the switching dynamics of RRAM [12]: the switching time exponentially depends on the applied voltage. To ensure a lifetime of at least 10 years at low voltage stress, we employed an RRAM compact device model [13] to study the dynamics of the resistance drift. Extrapolating from the experimental data shown in [8], and using $1/E$ model, we can see from Fig. 4 that the lifetime of RRAM device is 10 years under continuous voltage stress of 0.51V. The voltage drop across RRAM in a DTG is much less than 0.51V, ensuring significantly longer lifetime.

Figure 5 shows the yield of a RRAM based DTG-7 circuit in the presence of process variations. The yield calculation is based on 100,000 Monte Carlo simulations, which includes variations in both transistors and the RRAMs. The mean RRAM R_H value is $500k\Omega$ and the simulations were carried out for $\sigma/\mu = 1\%$, 5% and 10%. The simulations indicate that for high circuit yields with $K = 4$ the required σ/μ should be no more than 5% which is expected in near future.

The RRAM devices need to be initially programmed to their high resistance state (HRS) only once after fabrication, and the programming circuitry for doing this has to be part of the DTG. Note that RRAM devices are in the top metal layer and do not contribute to the silicon area. The schematic and its 3D structure are shown in Fig. 6. Two selection elements [14] are connected on both bottom and top electrodes. In Fig. 6 (a), the top electrode is connected to one selector and node N5 of the DTG, and the bottom electrode is connected to another selector and pFETs in the input network. The CLK is first set to 1 and a large positive forming voltage pulse is applied, to set the RRAM device to a low resistance state (LRS). Following this, the CLK is set to 0 and a large negative reset pulse is applied which resets the RRAM device to the high resistance state (HRS).

Fig. 6 (b) shows the 3-D arrangement of the different elements. Of the 3 pairs of pillars, the first and the last pair serve as selectors while the middle pair are the RRAM resistors. Although they all have the same structure, the selectors are designed to ensure that the programming does not affect the normal operation.

V. Realizing Threshold Functions using DTG

Threshold functions are a proper subset of unate functions. Without loss of generality, we can assume that they are also positive i.e. all the weights are positive integers. In this section we show how a DTG can be configured to realize a given positive threshold function $f(z_1, z_2, \cdots, z_m)$. The *configuration* simply involves connecting the $z's$ and their complements to the gates of transistors in the LIN and RIN. There are many possible ways to do this assignment. Here describe a specific signal assignment.

A DTG will be viewed as having n data inputs $(x_1, \cdots, x_n)$, and will be denoted as DTG-n. Internally, all the gates in the RIN will be driven by $(x_1, \cdots, x_n)$, and all the gates in the LIN will be driven by their compliments $(x'_1, \cdots, x'_n)$. This signal assignment is referred to as *complementary signal assignment* (CSA).

To ensure that the number of ON transistors in the LIN and RIN are never equal, n must be odd. This is because if n were even, and if r were active in the LIN then $n-r$ will be active in the RIN. Hence if $r = n/2$, an equal number of transistors will be active in the LIN and RIN. Since the LIN and RIN are complementary, for the output to be 1, (see Figure 1) just over 1/2 (or more) of the transistors in the LIN must be active. That is, the function is 1 if and only if $(n+1)/2$ or more of the inputs are 1. Hence with n being odd, a DTG-n with this signal assignment (all input gates driven by distinct x_i), implements the threshold function defined by

$$\text{DTG-n:} \equiv x_1 + x_2 + \cdots + x_n \geq (n+1)/2. \tag{3}$$

Consider an arbitrary threshold function $f(z_1, z_2, \cdots, z_m)$ defined by $w_1 z_1 + w_2 z_2 + \cdots + w_m z_m \geq T$, that is to be realized by DTG-n. Clearly if $T > (n+1)/2$, then f cannot be implemented by DTG-n. Hence $T \leq (n+1)/2$. Let $D = (n+1)/2 - T$ and $W = \sum_{i=1}^{m} w_i$.

Now to see how signals can be mapped to the inputs of DTG-n to realize $f(z_1, z_2, \cdots, z_m)$, replicate z_i, w_i times, for $1 \leq i \leq m$ in the definition of f.

$$\underbrace{z_{1,1} + \cdots + z_{1,w_1}}_{w_1 z_1} + \cdots + \underbrace{z_{m,1} + \cdots + z_{m,w_m}}_{w_m z_m} \geq T.$$

$$z_{1,1} + \cdots + z_{1,w_1} + \cdots + z_{m,1} + \cdots + z_{m,w_m} + D \geq \frac{n+1}{2}. \tag{4}$$

Therefore from (3) and (4), the second condition on DTG-n to be able to realize $f(z_1, z_2, \cdots, z_m)$ is $W + D \leq n$, or $W - T \leq (n-1)/2$. Given a DTG-n, if $f(z_1, z_2, \cdots, z_m)$ can be realized, then from (4) the assignment of signals can be done as follows: (1) assign D of the inputs of DTG-n to '1'; (2) for each i, $1 \leq i \leq m$, assign w_i inputs of DTG-n to the signal z_i; (3) connect any remaining inputs of DTG-n to '0'.

Example 1: Consider the threshold function $f(a,b,c) = a \vee bc \equiv 2a + b + c \geq 2$. It is easily verified that DTG-7 can implement this function since $T = 2$, $W = 4$, $D = (7+1)/2 - 2 = 2$. $T < (7+1)/2$ and $W - T < 3$. The CSA results in $x_1 = a$, $x_2 = a$, $x_3 = b$, $x_4 = c$, $x_5 = 1$, $x_6 = 1$, $x_7 = 0$. Internally, the transistors in the LIN will be driven by $(a', a', b', c', 0, 0, 1)$, and the transistors in the RIN will be driven by $(a, a, b, c, 1, 1, 0)$. If $a = 1$ then at least 4 of the 7 transistors in the LIN will be active, and no more than 3 of the transistors in the RIN will be active, resulting in an output of 1. **Note** that this function also be realized by a DTG-5. The two different implementations will have different delay, power and robustness characteristics.

Example 2: Consider $f(a,b,c,d,e) = ab \vee (a \vee b)(cd \vee ce \vee de) \equiv 2a + 2b + c + d + e \geq 4$. This can be implemented by DTG-7 and no smaller. Since $D = 0$, there are no inputs driven by constants, and the CSA leads to $x_1 = a$, $x_2 = a$, $x_3 = b$, $x_4 = b$, $x_5 = c$, $x_6 = d$, $x_7 = e$.

Let L denote the number of active transistors in LIN and R denote the same for RIN. It can be shown (and experimentally verified) that the worst case robustness condition will be for an input vector (i.e. $x's$) that results in a unit difference in conductance i.e. $(L - R) = 1$ with the largest number of active transistors $(L + R)$ is maximum. Incidentally, among all the cases for which $(L - R) = 1$, the one that maximizes $(L + R)$ results in the least delay. Note that it is impossible to avoid $(L - R) = 1$ for any signal assignment.

Justification of CSA: The CSA has three important characteristics that justify its use. First, it maximizes $(L+R)$ resulting in the fastest possible DTG. Secondly, there is only case for which $(L - R) = 1$ and the circuit can be optimized only for this case. Third, irrespective of an input vector $(x's)$, the total number of OFF transistors in the LIN and RIN is always n. This is important because it results in a constant load on the clock input regardless of the input vector.

VI. Cell Comparison of Energy, Delay and Area

Circuits implemented using conventional CMOS logic gates only will be denoted as CCL where those employing DTGs will be referred to as hybrid circuits (they may include inverters). Table II compares CCL and hybrid implementations of each threshold function.

For CCL circuits, each of the functions was synthesized using a commercial 65nm standard cell library. Since DTG is equivalent to an edge-triggered flipflop, each of CCL implementations contain a D-flipflop at the output of the function. The synthesized implementations were simulated using HSPICE for 100 random vectors having 30% switching activity for each primary input. The simulation corner was nominal, $25\,^{\circ}$C, and $V_{dd} = 0.6V$. The average energy-delay product (EDP) of each circuit is shown in the table. The delay of each circuit was determined by applying the critical input vector and reducing the clock period until the function ceases to simulate correctly in SPICE. The DTG configuration contains RRAM with $500K\Omega$ resistances.

		Energy X Delay		Ratio	CCL
Function	**Boolean Expression**	**CCL**	**DTG**		**Area**
1111111;4	abcd + abce + ··· (35 terms)	102.2	12.45	8.2	138.4
111111;4	abcd + abce + ··· (15 terms)	90.4	15.45	5.9	119.2
211111;4	abc + abd + ··· (15 terms)	71.1	12.37	5.7	95.2
21111;4	abc + abd + ··· (10 terms)	51.8	12.41	4.2	72
111111;3	abc + abd + ··· (20 terms)	48.8	13.56	3.6	79.2
11111;3	abc + abd + ··· (10 terms)	43.5	13.54	3.2	53
11111;4	abcd + abce + abde + acde + bcde	37.1	14.8	2.5	36.4
22111;4	ab + (a+b) (cd +de + ce)	29.4	12.33	2.4	20.4
11111;2	ab + bc + ··· (10 terms)	31	13.78	2.2	48
31111;4	a (b+c+d+e) + bcde	25.9	12.36	2.1	37
2211;4	ab+ acd +bcd	29.5	15.47	1.9	31.8
3111;4	a(b+c+d)	23.4	12.27	1.9	27.8
21111;3	a(b+c+d+e) + bcd + bce + bde + cde	23.7	13.48	1.8	35
1111;2	ab + ac + ad + bc + bd + cd	23.4	13.85	1.7	34.8
2111;4	a(bc+bd+cd)	20.5	12.11	1.7	23.8
2211;3	ab + (a+b) (c+d)	21.4	13.48	1.6	26.8
1111;3	abc + abd + acd + bcd	23.9	16.79	1.4	34.8
2111;2	a+bc+cd+bd	19.6	13.79	1.4	26.8
111;2	ab+bc+ac	24.1	17.48	1.4	23.6
2111;3	a(b+c+d)+bcd	23.5	16.99	1.4	32.8
211;2	a+bc	18	13.97	1.3	19.6
211;3	a(b+c)	16.5	13.36	1.2	13.6
111;1	a+b+c	15.3	13.63	1.1	21.6
1111;1	a+b+c+d	14.5	13.61	1.1	21.8
11;2	ab	14.3	13.64	1	13.4
1111;4	**abcd**	**13.4**	**14.31**	**0.9**	**18.8**
11;1	**a+b**	**12.8**	**13.76**	**0.9**	**13.4**
3111;3	**a+bcd**	**15.4**	**16.92**	**0.9**	**17.8**
111;3	**abc**	**11.9**	**16.02**	**0.7**	**16.6**
MEAN		**30.9**	**14.1**	**2.3**	**39.8**
STDEV		**22.6**	**1.6**	**1.7**	**31.7**

TABLE II: Comparison of functions

The column labeled Ratio denotes the ratio of the energy-delay product (EDP) of CCL to hybrid. Two important things to note are : (1) the energy-delay product of hybrid cells is almost independent of the function, switching activity and input vectors. Therefore the standard deviation of EDP is much less for hybrid circuits than for CCL; (2) except for four functions (shown in bold) where CMOS circuits have a slightly better energy-delay product, hybrid circuits show a consistent and significant improvement in EDP. These four functions are mostly small AND/OR functions. However even for these four functions, as the switching activity increases, the hybrid implementations of these functions start to show improvement over corresponding CCL counterparts. Significant reduction in silicon area was also achieved. Note that the hybrid implementations of all the functions are the worst case implementations as they employ DTG-7 for all functions. While many of which can actually be implemented with smaller DTG (DTG-5 and DTG-3) which would further reduce area and EDP. The total area each of hybrid function using DTG-7 and 7 inverters was $29.12\mu m^2$. However it should be noted that every function doesn't need all 7 inverters especially if inverter inputs are driven by constants. Similarly multiple inverters driven by the same signal can be merged. Comparing to the average area of CMOS circuit which was $39.77\mu m^2$, it is a 26.7% reduction. When DTGs replace threshold logic cones driving D-flipflops, the area savings will actually be greater because the input capacitance that DTG cell exhibits compared to the CCL counterpart is significantly reduced (approximately 30% for larger circuits). Synthesis tools can take advantage of this to reduce the size of logic that feeds DTGs in a ASIC design, providing additional reductions in area and power, without any performance degradation.

VII. Circuit Implementations and Comparison

In this section, we will show how circuit block implementations can benefit by using RRAM based DTG cell library.

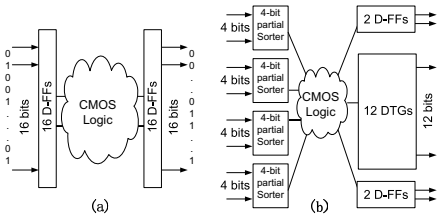

Fig. 7: A 16 inputs 1-bit sorter

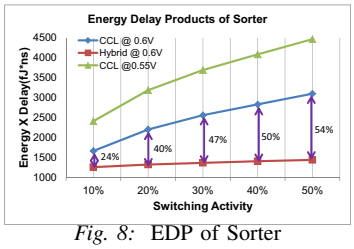

Fig. 8: EDP of Sorter

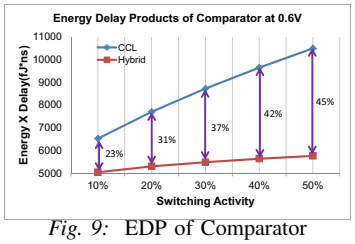

Fig. 9: EDP of Comparator

The library includes DTG-3, DTG-5 and DTG-7 cells. Two different implementations of each circuit were created using 65nm commercial library : one using CCL and the other using DTGs. The maximum operating frequency of each circuit was determined using SPICE simulation. To estimate the energy consumption of the clock tree, optimally sized clock buffers are included for both CCL and hybrid circuits. CCL implementations were created using Cadence RTL Compiler while the network of DTG cells were interconnected manually.

A. 16-input Single bit sorter

Figure 7 shows the structure of a 16-input single bit sorter. The sorter has 16 1-bit inputs and 16 1-bit outputs. The sorter is especially useful in parity and instruction control circuits and all symmetric functions. Both circuits are two stage pipelines. For the hybrid circuit, the first stage is implemented by four 4-input sorters, each of which is implemented by four DTG-7 gates. The second stage is implemented by 12 DTGs, 4 D-flipflops and CMOS logic cells, by suitably replacing the remaining CMOS logic and flipflops with DTGs. The peak frequency of CCL sorter at 0.6V is 125 MHz while the hybrid is 167 MHz. Figure 8 shows the energy-delay product of both circuits over different input switching activities. As the switching activity increases, CCL power increases because more nets toggle and there is greater glitching. On the other hand, DTGs have constant energy irrespective of the input switching activity. Hence the hybrid circuit shows a larger improvement in EDP especially for high switching activity applications. Typical switching activities are between 20 and 30%. For these, the hybrid design shows approximately 40% improvement in the EDP. Finally, the hybrid sorter ($943\mu m^2$) was 18% smaller than the CCL ($1159\mu m^2$).

Fig. 8 shows the energy-delay product of CCL is even worse at lower voltages. Therefore even though voltage of DTGs cannot be scaled down as much as CCL circuits, the EDP of DTG is still much lower.

B. 128-bit Comparator

The second circuit implemented was a 128-bit comparator designed as a 4-stage pipeline. The hybrid comparator consists of a hierarchy of several 8-bit comparators. The peak frequency of CCL comparator is 222 MHz while that of the hybrid is 250 MHz. Fig. 9 shows the energy-delay product of these two circuits as a function of switching activity. The hybrid comparator required 19% less area than the CCL ($5615\mu m^2$ vs $6822\mu m^2$), and a 31-37% improvement in EDP over CCL for switching activities of 20% to 30%.

VIII. Conclusions

We demonstrated the use of differential mode threshold gates (DTGs) as complex cells to realize significant reductions in area and energy-delay product. Conventionally differential mode circuits are not robust at low voltages. We have shown how we can circumvent this problem by integrating threshold gates with emerging RRAM memory technology and improved the robustness of the DTGs at low voltages. Logic functions implemented using the proposed DTG can result in significant improvement in area and energy-delay product.

IX. Acknowledgement

We acknowledge the support in part by the NSF I/UCRC Center for Embedded Systems award 0856090, and NSF PFI-BIC award 1237856. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] K.-Y. Siu, V. Roychowdhury, and T. Kailath, *Discrete Neural Computation: A Theoretical Foundataion*, ser. Information and Systems Sciences Series, T. Kailath, Ed. Englewood Cliffs, New Jersey: Prentice Hall, 1995.

[2] S. Muroga, *Threshold Logic and Its Applications*. New York: WILEY-INTERSCIENCE, 1971.

[3] V. Beiu, J. M. Quintana, and M. J. Avedillo, ""VLSI implementations of threshold logic - a comprehensive survey",," *IEEE Transactions on Neural Networks*, vol. 14, no. 5, pp. 1217–1243, 2003.

[4] M. J. Avedillo, J. M. Quintana, A. Rueda, and E. Jiménez, ""A low-power CMOS threshold-gate",," in *Electronics Letters*, vol. 31, 1995, pp. 2157–2159.

[5] J. A. H. López, J. G. Tejero, J. F. Ramos, and A. G. Bohórquez, ""New types of digital comparators",," in *International Symposium on Circuits and Systems*, vol. 1, 1995, pp. 29–32.

[6] M. Padure, S. Cotofana, C. Dan, S. Vassiliadis, and M. Bodea, ""A new latch-based threshold logic family",," in *International Semiconductor Conference*, vol. 2, 2001, pp. 531–534.

[7] R. Strandberg and J. Yuan, ""Single input current-sensing differential logic (SCSDL)",," in *International Symposium on Circuits and Systems*, vol. 1, 2000, pp. 764–767.

[8] Z. Fang, H. Yu, X. Li, N. Singh, G. Q. Lo, and D. L. Kwong, "$HfO_x/TiO_x/HfO_x/TiO_x$ multilayer-based forming-free rram devices with excellent uniformity," *Electron Device Letters, IEEE*, vol. 32, no. 4, pp. 566–568, 2011.

[9] H. S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. Chen, and M.-J. Tsai, "Metal oxide rram," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 1951–1970, 2012.

[10] J.-G. Zhu, "Magnetoresistive random access memory: The path to competitiveness and scalability," *Proceedings of the IEEE*, vol. 96, no. 11, pp. 1786–1798, 2008.

[11] H. S. P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, "Phase change memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, 2010.

[12] S. Yu, Y. Wu, and H.-S. Wong, "Investigating the switching dynamics and multilevel capability of bipolar metal oxide resistive switching memory," *Applied Physics Letters*, vol. 98, no. 10, pp. 103 514–103 514–3, 2011.

[13] X. Guan, S. Yu, and H.-S. Wong, "A spice compact model of metal oxide resistive switching memory with variations," *Electron Device Letters, IEEE*, vol. 33, no. 10, pp. 1405–1407, 2012.

[14] Y. Deng, P. Huang, B. Chen, X. Yang, B. Gao, J. Wang, L. Zeng, G. Du, J. Kang, and X. Liu, "Rram crossbar array with cell selection device: A device and circuit interaction study," *Electron Devices, IEEE Transactions on*, vol. 60, no. 2, pp. 719–726, 2013.