# Spin-based reconfigurable logic for IoT applications 

Nikhil Rangarajan, Satwik Patnaik, Johann Knechtel, Ozgur Sinanoglu, and Shaloo Rakheja, Member, IEEE Department of Electrical and Computer Engineering, New York University, Brooklyn, NY, USA


#### Abstract

The accelerated proliferation of internet of things (IoT) devices in recent years has incentivized new paradigms in low-cost reconfigurable computing. Emergent devices, especially from the spin-domain, are promising in the IoT design-space owing to their area efficiency, lower power dissipation, and reconfigurability. In this paper, we design a polymorphic spinbased logic device for IoT applications by exploiting the giant spin-Hall effect (GSHE) in heavy metals. The GSHE device offers $\sim 600 \times$ reduction in area and $\sim 22.8 \%$ reduction in power dissipation over $45-\mathrm{nm}$ CMOS devices, while improving circuit modularity over CMOS FPGAs and reconfigurable computing platforms based on emergent devices.


Keywords-Reconfigurable computing, spin-domain, giant spinHall effect, polymorphic gates, FPGA

## I. Introduction

The rapidly growing IoT devices- and applications-space is anticipated to become one of the largest sectors of the electronics industry in the coming decade [1]. An estimated 26 billion devices will be connected to this omnipresent network by the year 2020, driving the market for such devices into a trillion dollar industry [2]. The versatility of these devices has really bolstered their ubiquity, as they have penetrated into various spheres of human life ranging from wearable electronics, wireless sensor-actuator systems, and smart appliances to smart healthcare, transport and communications [3]. Wireless sensor networks (WSN) and radio frequency identification (RFID) have become indispensable for remote sensing and identification in hazardous environments and workplaces like oil rigs, boilers, dams etc. Wearable consumer electronics like smart watches, fitness trackers, biochips, and health monitors have slowly diffused into our day-to-day lives. While virtually all IoT applications require (i) ultra-low power operation, (ii) low area footprint, and (iii) modularity or reconfigurability [1], the latency requirements for such systems are not very stringent. Recently, there has been a thrust towards reconfigurable and polymorphic systems based on novel devices for such IoT applications. Unique characteristics of emergent devices that make them an attractive option for IoT applications include non-volatile data retention, near-zero leakage, ultimate scalability, and ease of integration with existing CMOS technologies.

A CMOS-molecular (CMOL) reconfigurable FPGA architecture using hybrid circuits, composed of CMOS transistors, nanowire crossbar arrays, and 2-terminal molecular nanodevices as the latching switches, was proposed in Ref. [4]. This design exhibits competitive performance metrics: $\sim 0.22 \mathrm{~mW}$ total power, $110 \mu \mathrm{~m}^{2}$ area, and 1.3 ns delay for a 32-bit adder at 32 nm technology. However, these estimates do not include the contributions of the relatively slow nanodevice latching switches, which would increase the overall area,
power, and delay values. In Ref. [5], a semiconductor optical amplifier (SOA)-based ultra-fast reconfigurable photonic logic gate is presented. This photonic gate exploits four-wave mixing and cross wave modulation in the SOA to realize AND/NOR/NOT/XNOR functions, but requires an elaborate processing setup and, hence, is not very scalable. A hybrid logic circuit that uses memristive crossbar arrays functioning as the reconfigurable data routing network, fabricated on top of a CMOS transistor layer, was demonstrated in Ref. [6]. This hybrid memristor-CMOS architecture is able to implement AND/OR/NAND/NOR/NOT and D flip-flop functionalities, and offers significant benefits in terms of the power consumption and non-volatility of the memristor, but requires additional circuitry between the crossbar array and CMOS layer for faithful integration. The authors in Ref. [7] implement a spin-based logic design by integrating a magnetic tunnel junction (MTJ) on top of a giant spin-Hall metal layer, to realize AND/OR/NAND/NOR operations. However, in this case, the different logic functions are achieved by changing the switching threshold of the MTJ during manufacturing, and hence this logic is not reconfigurable or runtime polymorphic. The domain wall (DW) motion-based 5-terminal device in Ref. [8] is able to implement the full set of basic Boolean logic functions (except buffer), and seems promising for the purposes of IoT chips and circuits. The pitfall of this design is the requirement of 13 additional CMOS transistors (apart from the DW device) per gate, which exacerbates the power and area metrics.

In this paper, we propose a spin-based reconfigurable logic family by exploiting the runtime polymorphism of the giant spin-Hall effect (GSHE) switch [9, 10]. The GSHE switch is capable of implementing all eight Boolean operations (INV/BUF/AND/OR/NAND/NOR/XOR/XNOR) as well as sequential latches and flip-flops using a single device. It does not need additional CMOS circuitry to achieve this, and is able to switch between the different logic functionalities dynamically on-the-fly. This peculiar feature coupled with its low area and power characteristics make it an ideal candidate for IoT devices. Consider, for instance, a remotely deployed sensoractuator system (Fig. 1) that senses environmental signals periodically, stores them in a built-in memory, and actuates a response once a certain threshold has been crossed. This can be very efficiently implemented using the GSHE logic with integrated spin transfer-torque (STT) MRAM memory, on a single chip. The circuit is initially configured to function as a sensor that samples and stores the incoming stimulus, and then reconfigured to implement the actuator circuit, which responds according to the stored data. Such a design would result in tremendous gains in area, power, and lifetime of the remotely deployed system. We note here that in GSHE logic, the gates
themselves morph to perform multiple functions, which is in contrast to conventional CMOS FPGA architectures that have a reconfigurable routing fabric. The GSHE logic could also have a huge potential for future 5 G mobile communication systems to implement load-dependent or protocol-dependent dynamically morphing systems.


Fig. 1: Representative reconfigurable sensor-actuator system. The circuit, initially configured as a sensor, first stores the incoming signal in spin-transfer torque (STT) magnetic random access memory (MRAM). It then reconfigures to perform the actuation.

The main contributions of this paper are:
(1) To highlight the implementation of the various Boolean logic functions using the GSHE switch,
(2) To analyze the dynamic reconfigurability of the GSHE logic with a pertinent TOY example, and to evaluate the area, power, and delay metrics of the implemented GSHE logic,
(3) To compare the performance metrics of benchmark
circuits implemented using GSHE logic against those of CMOS-based implementations and other polymorphic gates proposed in prior works.
The remainder of this paper is organized as follows. Section II describes the construction and operation of the GSHE switch. In section III, we present the implementation of combinational and sequential logic elements using the GSHE switch. Section IV demonstrates a full adder circuit constructed with GSHE gates, which dynamically reconfigures to function as a full subtractor. Finally, section V presents the performance metrics of ISCAS' 85 combinational benchmarks using GSHE logic, which are then compared against prior art.

## II. GSHE SWITCH MODEL

The GSHE switch, illustrated in Fig. 2a, relies on the phenomena of spin-Hall effect and magnetic dipolar coupling to implement Boolean logic functions. A giant spin-Hall layer (shown in blue in Fig. 2a), typically composed of a heavy metal (HM) like W, Pt or Pd, results in spin accumulation of opposite polarities on the lateral surfaces of the layer, on the injection of a charge current (orange arrow) [10]. The direction of spin accumulation is orthogonal to the direction of the injected charge current, with the polarization of the spin current being orthogonal to both of them. In Fig. 2a, the charge current is assumed to be along $\hat{x}$, while the spin accumulation and polarization directions are along $\hat{y}$ and $\hat{z}$, respectively. The polarized spin current imparts a torque to the magnetization vector of the free Write (W) nanomagnet (shown in peach in Fig. 2a) through the STT mechanism. The W nanomagnet is magnetically coupled to the $\operatorname{Read}(\mathrm{R})$ nanomagnet (also in peach) via their mutual dipolar coupling. We consider a negative coupling, which means that in equilibrium, the magnetizations of W and R nanomagnets are anti-parallel to each other. Hence, when the STT acts on the W nanomagnet to switch it from one stable state to the other, the R nanomagnet will also switch, but in the opposite direction. Once the


Fig. 2: (a) Construction and operation of the GSHE switch. A charge current (orange arrow) supplied to the heavy metal layer produces a spin current, which results in STT-induced switching of the W nanomagnet. Dipolar coupling with the W nanomagnet switches the R nanomagnet, and logic read-out is achieved via an MTJ arrangement on the R side. The GSHE switch in this figure is laid out horizontally for better clarity, but will be fabricated as a vertical stack. (b) Realizing inverter and buffer operations with the GSHE switch. Polarity of voltages on top of the fixed nanomagnets decides the direction of the output electrical current and, therefore, the output logic state.
information is transferred from the W nanomagnet to the R nanomagnet, an output electrical current is generated via an MTJ stack using two fixed nanomagnets (shown in dark green in Fig. 2a) sitting atop the R nanomagnet. Since the two fixed nanomagnets are configured in an anti-parallel fashion, the final magnetization state of the R nanomagnet will be parallel to one of the fixed nanomagnets and anti-parallel to the other. Voltages of opposite polarities are applied to contacts on top of each of the fixed nanomagnets. The MTJ in which the R nanomagnet is parallel to the fixed nanomagnet ( $V^{-}$in Fig. 2a) will offer a lower resistance path for the output electrical current. By reversing the polarities of the MTJ supply voltages, the direction of the output electrical current can be flipped. The direction of the electrical current determines the logic state; therefore, interchanging the voltage polarities $\left(V^{+}\right.$and $\left.V^{-}\right)$ changes the operation of the switch from an inverter to a buffer (or vice versa), as shown in Fig. 2b.

## III. Complex logic using the GSHE switch

## A. Implementation of NAND / NOR / AND / OR logic

NAND/NOR gates are implemented with the GSHE switch using the setup shown in Fig. 3a. Three current domain signals, $A, B$ and $X$, are fed into the input terminal of the GSHE switch. While $A, B$ are the primary inputs, input $X$ is a tiebreaking signal, which is required to implement an even-input gate, as seen from the truth tables in Fig. 3b. By choosing $X=+I$, the gate exhibits the functionality of a two-input NAND gate, while $X=-I$ transforms the gate into a two-input NOR gate. In general, $n$-input logic gates can be constructed directly using the same setup shown in Fig. 3a if $X= \pm(n-1) I$. Alternately, two-input GSHE gates may be cascaded to implement complex gates with a higher fan-in.


| A | B | X | I (total) | Out |
| :---: | :---: | :---: | :---: | :---: |
| $-\mathbf{I}$ | $-\mathbf{I}$ | $-\mathbf{I}$ | $-\mathbf{I I}$ | $+\mathbf{I}$ |
| $-\mathbf{I}$ | $+\mathbf{I}$ | $-\mathbf{I}$ | $-\mathbf{I}$ | $+\mathbf{I}$ |
| $+\mathbf{I}$ | $-\mathbf{I}$ | $-\mathbf{I}$ | $-\mathbf{I}$ | $+\mathbf{I}$ |
| $+\mathbf{I}$ | $+\mathbf{I}$ | $-\mathbf{I}$ | $+\mathbf{I}$ | $-\mathbf{I}$ |


| $\mathbf{A}$ | $\mathbf{B}$ | $\mathbf{X}$ | $\mathbf{I}($ total $)$ | Out |
| :---: | :---: | :---: | :---: | :---: |
| $-\mathbf{I}$ | $-\mathbf{I}$ | $+\mathbf{I}$ | $-\mathbf{I}$ | $+\mathbf{I}$ |
| $-\mathbf{I}$ | $+\mathbf{I}$ | $+\mathbf{I}$ | $+\mathbf{I}$ | $-\mathbf{I}$ |
| $+\mathbf{I}$ | $-\mathbf{I}$ | $+\mathbf{I}$ | $+\mathbf{I}$ | $-\mathbf{I}$ |
| $+\mathbf{I}$ | $+\mathbf{I}$ | $+\mathbf{I}$ | $+\mathbf{I}$ | $-\mathbf{I}$ |

(b)

Fig. 3: (a) Configuration of GSHE switch for NAND/NOR operation. (b) Realization of NAND/NOR and their respective truth tables.

Changing the direction of current $X$ during operation will dynamically reconfigure the gate from NAND to NOR or
vice versa, on-the-fly. Here, $I$ is formulated in terms of the critical current of the W nanomagnet, which will be used for calculating the performance metrics of the device in section IV. To implement AND/OR gates, the polarities of the voltages on the fixed nanomagnets in the MTJ stack must be flipped $\left(V^{+/-} \rightarrow V^{-/+}\right)$.

## B. Implementation of XOR / XNOR logic

To realize XOR logic, the tie-breaking signal $X$ is eliminated. One of the primary inputs (say $A$ ) is applied as a current domain signal, while the other primary input and its complement $(B$ and $\bar{B})$ are applied as voltages to the $\times$ and - fixed nanomagnets in the MTJ stack. The truth table for XOR operation is shown in Fig. 4. Here, for instance, applying a current of $-I$ as the input $A$ (first row of the truth table) sets the W nanomagnet in $\times$ orientation, and the R nanomagnet in - orientation. Applying $B$ and $\bar{B}$ on the $X$ and $\bullet$ fixed nanomag-

| A | W | R | B | B' | Out |
| :---: | :---: | :---: | :---: | :---: | :---: |
| -I | * | - | * | - | -I |
| -I | * | - | $\bullet$ | * | +I |
| +I | - | $\times$ | * | $\bullet$ | +I |
| +I | - | $\boldsymbol{x}$ | - | $\boldsymbol{*}$ | -I |

Fig. 4: Truth table for XOR operation. $\times$ (into the plane) corresponds to logic 0 and • (out of the plane) corresponds to logic 1. nets, respectively, will result in an output current flowing into the device $(-I)$. This is because $\bar{B}$ is applied on the fixed nanomagnet that is parallel to the R nanomagnet. The XOR gate can be converted to an XNOR gate by interchanging $B$ and $\bar{B}$ on the voltage terminals.

## C. Implementation of $D$ latch and flip-flop

The GSHE switch can also implement sequential logic, like the D latch shown in Fig. 5. Here, the clock signal $(\underline{\phi})$ and its complement ( $\bar{\phi}$ ) are applied at the voltage terminals of the MTJ, and reversing their polarities allows one to transform between a positive and negative latch. A D flip-flop is implemented by connecting two such GSHE switches in series, in a master-slave fashion.


Fig. 5: Implementation of a $D$ latch using GSHE logic, and the corresponding truth table.

## IV. Runtime polymorphism of GSHE LOGIC

In this section, we discuss the implementation of majority logic function using the GSHE switch. The majority logic is used to construct a full adder circuit. The reconfigurability of the GSHE switch is exploited in the transformation of the full adder to a full substractor circuit without any modifications at the layout level. Finally, we arrive at the performance metrics of individual GSHE gates and use those metrics to calculate circuit-level metrics for the full adder, which are then compared with the full adder implementation in prior works.

## A. Majority logic with GSHE switch

The GSHE switch can directly implement a 3-input majority logic gate, for which the truth table is shown in Fig. 6. The three inputs are applied as current signals at the input terminal, while the control signal $X$ is eliminated. Voltage $V^{+}$is applied on the fixed nanomagnet oriented along $\times$, and $V^{-}$is applied on the fixed nanomagnet along $\bullet$. As before, the polarity of the voltage signals on the MTJ stack and the magnetization state of the R nanomagnet determines the direction of the output electrical current and, therefore, the Boolean logic function realized by the gate.

| A | B | C | I (total) | W | R | V ${ }^{+}$ | V- | Out |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| -I | -I | -I | -3I | $\times$ | - | * | - | -I |
| -I | -I | +I | -I | * | - | $\times$ | - | -I |
| -I | +I | -I | -I | $\times$ | - | $\times$ | - | -I |
| -I | +I | +I | +I | - | * | * | - | +I |
| +I | -I | -I | -I | * | - | * | - | -I |
| +I | -I | +I | +I | - | * | * | - | +I |
| +I | +I | -I | +I | - | * | $\times$ | - | +I |
| +I | +I | +I | +3I | - | * | $\times$ | - | +I |

Fig. 6: Truth table of majority gate implemented with the GSHE switch.

## B. Full adder to full subtractor transformation

Full adders and subtractors are essential elements of logic design and are pervasively used in any integrated circuit, including IoT devices. The ability to morph between these two functions as needed will be advantageous in an IoT chip in terms of area savings. In conventional FPGAs, a circuit configured to run as a full adder maybe transformed into a full subtractor by re-routing the interconnects around the gates. However, in GSHE logic, the gates themselves can be changed without the need for reconfigurable interconnects, which opens up new possibilities in the FPGA and reconfigurable hardware design-space. While we do not use reconfigurable interconnects in this TOY example, the integration of polymorphic GSHE gates with such dynamic routing fabric would drastically increase the degrees of freedom of reconfigurability.

Full adder and full subtractor functions can be realized using two XOR gates and one Majority gate as follows [8]:
(a) Full adder

$$
\begin{equation*}
\text { Sum }=A \oplus B \oplus C, \quad \text { Carry }=\operatorname{MAJ3}(A, B, C) \tag{1}
\end{equation*}
$$

(b) Full subtractor

$$
\begin{equation*}
\text { Diff }=A \oplus B \oplus C, \quad \text { Borrow }=\operatorname{MAJ3}(\bar{A}, B, C) \tag{2}
\end{equation*}
$$

We use the configuration shown in Fig. 7 to implement these circuits. Gate $X 3$ functions as a buffer for the full adder and as an inverter to achieve the full subtractor, and changing the voltage polarities on $X 3$ allows one to dynamically change between the two circuits. We note that this implementation is essentially different from CMOS-based adder-subtractor circuits, which are not reconfigurable and require $1.5 \times$ the number of gates. Whereas, using low-power GSHE logic, one can achieve both functions at a fraction of that area and power.


Fig. 7: Dynamic reconfiguration of full adder to full subtractor and vice versa using GSHE XOR, INV/BUF and MAJ3 gates.

## C. Performance of GSHE gates and comparison of full adder

To quantify the performance of ISCAS' 85 benchmark circuits implemented with GSHE switches in section V, we first evaluate the delay, power, and area metrics of an individual GSHE switch. The geometrical and material parameters of the GSHE switch used for analysis in this paper are given in Table I. To deterministically switch the state of the GSHE device, a critical spin current of $I_{S}=20 \mu \mathrm{~A}$ is required (mathematical details can be found in Ref. [10].) Even though the switching process is deterministic, the delay for magnetization reversal is stochastic and has a distribution as depicted in Fig. 8a. The average delay in an ensemble of 100,000 simulations is 1.55 ns. While constructing a large circuit using GSHE logic, some of the gates in the critical path would be faster than the average case, while some would have a delay greater than 1.55 ns . Hence, we assume 1.55 ns as the delay of each gate in the critical path for the purpose of evaluating circuit-level delay metrics. This translates to circuit-level speeds of typically a few 10's to 100 's of MHz , which, even though slower than CMOS circuits, is perfectly acceptable for IoT devices.


Fig. 8: Obtaining delay and power of the GSHE switch.

Figure 10b illustrates the circuit representation of the GSHE switch, used for calculating its power dissipation. The total power of the GSHE switch is given as

$$
\begin{equation*}
P_{G S H E}=\frac{V_{o u t}^{2}}{r}+\left(V_{S}-V_{o u t}\right)^{2} G_{P}+\left(V_{S}+V_{o u t}\right)^{2} G_{A P} \tag{3a}
\end{equation*}
$$

TABLE I: Specifications of the GSHE switch, used for calculating its metrics [10].

| Parameter | Value |
| :---: | :---: |
| Volume | $(28 \times 15 \times 2) \mathrm{nm}^{3}$ |
| Saturation magnetization <br> $\left(M_{s}\right)$ | $10^{6} \mathrm{~A} / \mathrm{m}(\mathrm{W})$ |
| $5 \times 10^{5} \mathrm{~A} / \mathrm{m}(\mathrm{R})$ |  |
| Uniaxial energy density <br> $\left(K_{u}\right)$ | $2.5 \times 10^{4} \mathrm{~J} / \mathrm{m}^{3}(\mathrm{~W})$ |
| $5 \times 10^{3} \mathrm{~J} / \mathrm{m}^{3}(\mathrm{R})$ |  |
| Critical spin current (deterministic <br> switching) | $20 \mu \mathrm{~A}$ |
| Resistance area product (RAP) | $1 \Omega \mu \mathrm{~m}^{2}[11]$ |
| Tunneling Magnetoresistance $(\mathrm{TMR})$ | $170 \%[11]$ |
| $G_{P}$ | $420 \times 10^{-6} \mathrm{~S}$ |
| $G_{A P}$ | $155.6 \times 10^{-6} \mathrm{~S}$ |
| Spin-Hall angle $\left(\theta_{S H}\right)$ | 0.4 |
| Internal gain of HM $(\beta)$ | 6 |
| Thickness of HM $\left(t_{h m}\right)$ | 1 nm |
| Resistivity of HM | $5.6 \times 10^{-7} \Omega \mathrm{~m}$ |

$$
\begin{equation*}
V_{o u t}=\frac{I_{S} r}{\beta} ; \quad V_{S}=\frac{I_{S}}{\beta}\left[\frac{1+r\left(G_{P}+G_{A P}\right)}{G_{P}-G_{A P}}\right] \tag{3b}
\end{equation*}
$$

where $V_{S}$ is the magnitude of voltage applied to the MTJ stack, $G_{P}$ and $G_{A P}$ are the parallel and anti-parallel conductances of the MTJ stacks, $r$ is the resistance of the giant spin-Hall HM, and $\beta=\frac{I_{\text {spin }}}{I_{\text {elec }}}=\theta_{S H}\left(\frac{w_{n m}}{t_{h m}}\right)$ is the internal gain of the heavy metal. Here, $w_{n m}$ is the width of the nanomagnets and $t_{h m}$ is the thickness of the HM layer. $G_{P}$ and $G_{A P}$ are calculated from the TMR and RAP values given in Table I as $420 \mu \mathrm{~S}$ and $155.6 \mu \mathrm{~S}$, respectively. Considering a HM layer 46 nm long, the thickness and resistivity of the HM in Table I yield $r \sim 1 k \Omega$. These values result in a total power of $0.2125 \mu \mathrm{~W}$ for the GSHE switch, from (3a). Note that this power is for a spin current equal to the critical current of $20 \mu \mathrm{~A}$.


Fig. 9: Layout of GSHE gates for a vertically stacked structure - the read unit is built on top of the write unit for better coupling.

Increasing the spin current from this value will result in an improved delay for the device, but at the cost of higher power consumption. The layout of the GSHE switch, illustrated in Fig. 9, was created according to the design rules for beyond-CMOS devices, formulated in [12] in units of maximum misalignment length $\lambda$. The area of the GSHE switch as obtained from this layout is $0.0016 \mu \mathrm{~m}^{2}$.

We compare the performance of the GSHE full adder with the DW motion-based implementation presented in Ref. [8], when both designs are running at 500 MHz . To achieve a speed of 500 MHz for the GSHE full adder, each of the gates in Fig.

7 must have a delay of 1 ns . Hence, to reduce the GSHE switch delay from 1.55 ns to 1 ns , the input spin current is increased to $40 \mu$ A (Fig. 10a). From Fig. 10b, the power consumption of the GSHE full adder is $5 \times$ lower than that of the DW motionbased design in Ref. [8], at 500 MHz . Further, the ability of the GSHE device to implement circuits without the requirement of additional CMOS peripherals leads to significant area savings when compared against Ref. [8].

(a) PDF of delays of the GSHE (b) Comparison of total power and device switch at input spin current of count metrics of the full adder design. $40 \mu \mathrm{~A}$.

Fig. 10: Performance evaluation of the GSHE full adder implementation and comparison with Ref. [8]. The green and purple bars in (b) represent the count of DW devices and MOS transistors, respectively.

## V. Benchmarking

In this section, we compare the area, power, and delay (APD) metrics of benchmark circuits implemented using three techniques: (i) GSHE switches, (ii) CMOS (used in conventional FPGAs), and (iii) DW switches in Ref. [8] We utilize 10 benchmark circuits from the ISCAS'85 combinational benchmark suite, whose details are mentioned in Table II. All simulations are carried out using Cadence RTL compiler at 45 nm technology node, utilizing the typical process corner. For a fair comparison, we first obtain the metrics (APD) for the CMOS implementation of a given benchmark, and assume that the GSHE logic version of that benchmark would be constructed using a one-to-one replacement of the CMOS gates in the circuit with GSHE switches.
The APD values of the 10 benchmarks for CMOS, Ref. [8] and our proposed approach are outlined in Table III. Compared to the CMOS implementation, our approach scales particularly well for area, with an average area savings of $\sim 600 \times$ across all benchmarks used in this work. With respect to power consumption, our GSHE logic achieves an average reduction of $22.79 \%$ when compared to CMOS-based implementations. However, the power savings are considerably higher when compared to Ref. [8], where, on an average, we achieve a reduction of $\sim 41 \times$. For calculating the delay, we dump the critical path of every design under consideration, and keep a tally of all the logic gates that make up the critical path. This allows us to estimate the delay when the same circuits are constructed with GSHE switches. For example, 10 CMOS logic gates in the critical path would result in a delay of $\sim 15.5$ ns in the GSHE logic version of the circuit (as delay of each GSHE gate is 1.55 ns from section IV. C).

When accounting for the delay, it is not surprising to note that designs utilizing GSHE switches are slower than their

TABLE II: Characteristics of benchmarks from ISCAS' 85 combinational suite.

| Benchmark | Inputs | Outputs | Gate Count | Functionality |
| :---: | :---: | :---: | :---: | :---: |
| c 432 | 36 | 7 | 160 | 27-channel interrupt controller |
| c 499 | 41 | 32 | 202 | 32-bit single-error-correcting (SEC) |
| c 880 | 60 | 26 | 383 | 8-bit Arithmetic Logic unit (ALU) |
| c 1355 | 41 | 32 | 546 | 32-bit SEC |
| c 1908 | 33 | 25 | 880 | 16-bit error detector/corrector |
| c 2670 | 233 | 140 | 1,193 | 12-bit ALU and controller |
| c 3540 | 50 | 22 | 1,669 | 8-bit ALU |
| $\mathrm{c5315}$ | 178 | 123 | 2,307 | 9-bit ALU |
| c 6288 | 32 | 32 | 2,416 | 16x16 multiplier |
| c 7552 | 207 | 108 | 3,512 | 32-bit adder/comparator |

TABLE III: Area (A), Power (P), and Delay (D) comparison for selected benchmarks. Reference [8] does not provide area estimates. However, their device dimensions and the need for peripheral CMOS circuits implies that their area is considerably larger than the proposed GSHE logic.

| Benchmark | CMOS |  |  | Ref. [8] |  |  | Proposed GSHE logic |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $\mathbf{A}\left(\mu \mathbf{m}^{2}\right)$ | $\mathbf{P}(\mathbf{m W})$ | $\mathbf{D}(\mathbf{n s})$ | $\mathbf{A}\left(\mu \mathbf{m}^{2}\right)$ | $\mathbf{P}(\mathbf{m W})$ | $\mathbf{D}(\mathbf{n s})$ | $\mathbf{A}\left(\mu \mathbf{m}^{2}\right)$ | $\mathbf{P}(\mathbf{m W})$ | $\mathbf{D}(\mathbf{n s})$ |
| c 432 | 129.011 | 0.036 | 2.087 | - | 1.0 | 26 | 0.261 | 0.034 | 37.2 |
| c 499 | 232.218 | 0.074 | 1.797 | - | 1.2 | 15 | 0.299 | 0.039 | 20.1 |
| c 880 | 290.472 | 0.077 | 1.784 | - | 2.1 | 22 | 0.531 | 0.071 | 24.8 |
| c 1355 | 234.346 | 0.083 | 2.072 | - | 2.0 | 14 | 0.301 | 0.039 | 17.1 |
| c 1908 | 283.822 | 0.084 | 2.328 | - | 3.6 | 21 | 0.451 | 0.059 | 29.4 |
| c 2670 | 459.116 | 0.128 | 1.821 | - | 5.6 | 20 | 0.824 | 0.109 | 24.8 |
| c 3540 | 856.521 | 0.263 | 2.799 | - | 8 | 32 | 1.626 | 0.216 | 34.1 |
| c 5315 | $1,073.842$ | 0.245 | 2.538 | - | 11 | 28 | 1.936 | 0.257 | 31 |
| c 6288 | $1,936.481$ | 0.795 | 5.371 | - | 75 | 58.9 | 3.394 | 0.45 | 63.6 |
| c 7552 | $1,122.254$ | 0.308 | 3.325 | - | 24 | 72.2 | 1.869 | 0.248 | 32.6 |

CMOS counterparts; on an average, the GSHE circuits are about $\sim 12 \times$ slower. As the primary requirement of IoT circuits hinges on smaller area and low-power, we believe the increased delay of GSHE logic would not be a prohibitive factor in the design of such circuits. The benchmarks listed in Table II can be used at a frequency of $15-58 \mathrm{MHz}$ for the GSHE logic versions, which is lower than the operating frequency for CMOS-centric deployment ( $186-556 \mathrm{MHz}$ ), but it is critical to note that these GSHE-based circuits are far more area and power efficient.

## VI. CONCLUSION

This paper introduces a novel spin-based reconfigurable logic design for IoT applications, by utilizing the polymorphic GSHE gates. These gates, which leverage the giant spinHall effect in heavy metals, are capable of implementing various combinational as well as sequential logic functions using a single device, and exhibit the ability to dynamically morph between different gates on-the-fly. Compared to existing CMOS-based and other spin-based reconfigurable systems, GSHE devices exhibit significant area and power savings, while operating at 10 's -100 's of MHz speeds that are ideal for IoT circuits. The possibility of integrating these spin-based polymorphic gates with conventional reconfigurable interconnects used in FPGAs opens up new avenues in the field of reconfigurable computing.

## AcKnowledgements

This work was supported in part by the Semiconductor Research Corporation (SRC) and the National Science Foundation (NSF) through ECCS 1740136. The authors also thank the funding support from the MRSEC Program of the National Science Foundation under Award Number DMR-1420073.

## References

[1] D. Blaauw, D. Sylvester, P. Dutta, Y. Lee, I. Lee, S. Bang, Y. Kim, G. Kim, P. Pannuto, Y.-S. Kuo et al., "Iot design space challenges: Circuits and systems," in VLSI Technology (VLSI-Technology): Digest of Technical Papers, 2014 Symposium on. IEEE, 2014, pp. 1-2.
[2] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," Business Horizons, vol. 58, no. 4, pp. 431-440, 2015.
[3] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in Frontiers of Information Technology (FIT), 2012 10th International Conference on. IEEE, 2012, pp. 257260.
[4] D. B. Strukov and K. K. Likharev, "Cmol fpga: a reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices," Nanotechnology, vol. 16, no. 6, p. 888, 2005.
[5] G. Berrettini, A. Simi, A. Malacarne, A. Bogoni, and L. Poti, "Ultrafast integrable and reconfigurable xnor, and, nor, and not photonic logic gate," IEEE Photonics Technology Letters, vol. 18, no. 8, pp. 917-919, 2006.
[6] Q. Xia, W. Robinett, M. W. Cumbie, N. Banerjee, T. J. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, D. B. Strukov et al., "Memristor- cmos hybrid integrated circuits for reconfigurable logic," Nano letters, vol. 9, no. 10, pp. 3640-3645, 2009.
[7] Y. Zhang, B. Yan, W. Wu, H. Li, and Y. Chen, "Giant spin hall effect (gshe) logic design for low power application," in Design, Automation \& Test in Europe Conference \& Exhibition (DATE), 2015. IEEE, 2015, pp. 1000-1005.
[8] F. Parveen, Z. He, S. Angizi, and D. Fan, "Hybrid polymorphic logic gate with 5-terminal magnetic domain wall motion device," in VLSI (ISVLSI), 2017 IEEE Computer Society Annual Symposium on. IEEE, 2017, pp. 152-157.
[9] S. Datta, S. Salahuddin, and B. Behin-Aein, "Non-volatile spin switch for boolean and non-boolean logic," Applied Physics Letters, vol. 101, no. 25, p. 252411, 2012.
[10] N. Rangarajan, A. Parthasarathy, N. Kani, and S. Rakheja, "Energy-efficient computing with probabilistic magnetic bits-performance modeling and comparison against probabilistic cmos logic," IEEE Transactions on Magnetics, 2017.
[11] H. Maehara, K. Nishimura, Y. Nagamine, K. Tsunekawa, T. Seki, H. Kubota, A. Fukushima, K. Yakushiji, K. Ando, and S. Yuasa, "Tunnel magnetoresistance above $170 \%$ and resistance-area product of $1 \omega(\mu \mathrm{~m}) 2$ attained by in situ annealing of ultra-thin mgo tunnel barrier," Applied physics express, vol. 4, no. 3, p. 033002, 2011.
[12] D. E. Nikonov and I. A. Young, "Overview of beyond-cmos devices and a uniform methodology for their benchmarking," Proceedings of the IEEE, vol. 101, no. 12, pp. 2498-2533, 2013.
$\mathbf{N}$ ikhil Rangarajan is a PhD student at the department of electrical engineering, New York University, New York, NY, USA. He has an M.S. in electrical engineering from New York University. His research interests include spintronics, nanoelectronics and device physics. He is a student member of IEEE and he can be reached at nikhil.rangarajan@ nyu.edu.
$\mathbf{S}$ atwik Patnaik is a PhD student at the department of electrical engineering, New York University, New York, NY, USA. He is interested in security and trust for hardware and low power VLSI design, and has an M.Tech. in computer science from Indian Institute of Information Technology, Gwalior, India. He is a student member of IEEE and is available at sp4012@nyu.edu

J ohann Knechtel is a Postdoctoral Associate at New York Univeristy Abu Dhabi. Previously, he was with Masdar Institute of Technology, UAE (201516), TU Dresden, Germany (2010-14, PhD in Computer Engineering), as well as a visiting researcher with CUHK, China (2012) and UMich, USA (2010). His research is on Electronic Design Automation, particularly in the context of emerging technologies and hardware security. Dr. Knechtel is a member of IEEE (knechtel@ieee.org).

O zgur Sinanoglu is an Associate Professor of electrical and computer engineering at New York University Abu Dhabi. He obtained his PhD in Computer Science and Engineering from University of California San Diego. He has industry experience at TI, IBM and Qualcomm, and has been with NYU Abu Dhabi since 2010. Prof. Sinanoglu's research interests include reliability and security of VLSI circuits, where he has around 160 conference and journal papers, and 20 issued and pending US Patents. His email id is ozgursin@nyu.edu.

S haloo Rakheja is an Assistant Professor of electrical and computer engineering with New York University, Brooklyn, NY, USA, where she works on nanoelectronic devices and circuits. Prior to joining NYU, she was a Postdoctoral Research Associate with the Microsystems Technology Laboratories, Massachusetts Institute of Technology, Cambridge, USA. She obtained her M.S. and Ph.D. degrees in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA. She is a member of IEEE and her email id is shaloo.rakheja@nyu.edu.

