Leveraging Spintronic Devices for Efficient Approximate Logic and Stochastic Neural Networks

Shaahin Angizi[†], Zhezhi He[†], Yu Bai[‡], Jie Han[*], Mingjie Lin[†], Ronald F. DeMara[†] and Deliang Fan[†]
[†] Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816
[‡] Department of Engineering and Computer Science, California State University, Fullerton CA, 92831
[*] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada
dfan@ucf.edu

ABSTRACT

ITRS has identified nano-magnet based spintronic devices as promising post-CMOS technologies for information processing and data storage due to their ultra-low switching energy, non-volatility, superior endurance, excellent retention time, high integration density and compatibility with CMOS technology. As for data storage, spintronic memory has been widely accepted as a universal high performance next-generation non-volatile memory candidate. As for information processing, spintronic computing remains complementary in its features to CMOS technology. In this paper, we present two innovative spintronic computing primitives, i.e. spintronic approximate logic and spintronic stochastic neural network, which both leverage the intrinsic spintronic device physics to achieve much more compact and efficient designs than CMOS counterparts. In spintronic approximate logic, we employ the intrinsic current-mode thresholding operation to implement an accuracy-configurable adder and further demonstrate its application in approximate DSP applications. In spintronic stochastic neural networks, we leverage the stochastic properties of domain wall devices and magnetic tunnel junction to implement a low-power and robust artificial neural network design.

ACM Reference Format:
Shaahin Angizi, Zhezhi He, Yu Bai, Jie Han, Mingjie Lin, Ronald F. DeMara and Deliang Fan. 2018. Leveraging Spintronic Devices for Efficient Approximate Logic and Stochastic Neural Networks. In *Proceedings of 2018 Great Lakes Symposium on VLSI (GLSVLSI '18).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3194554.3194618

1 INTRODUCTION

Nowadays, the insufficient ability of modern computing platforms to deliver simultaneously energy efficient and high performance computing solutions leads to a gap between meets and needs. Specially, owing to the boom in machine learning, artificial intelligence and internet of things areas, data analytics can not only rely on conventional computing methods. That's why recently approximate computing [3, 13] and brain-inspired computing [5] have drawn a lot of attentions.Within current Boolean logic and Complementary Metal Oxide Semiconductor (CMOS) based computing platforms, such gap will keep widening mainly due to limitations in devices and computing model. Of all nanoelectronic paradigms, spintronic devices have attracted significant attention over the past decade due to non-volatility, zero leakage current, high integration density, low standby power, and back end of line fabrication with CMOS technology [10]. However, these emerging devices often exhibit strong stochastic switching behaviors and suffer from large variations in both electrical characteristics and device reliability. Therefore, how to efficiently leverage the unique device properties of emerging spintronic devices to facilitate new computing tasks becomes a both intriguing and important research challenge. In this paper, we first present a majority gate design employing intrinsic current-mode thresholding operation of spintronic device to implement an accuracy configurable adder for approximate DSP applications. Moreover, we also show that a stochastic-based soft-limiting artificial neural network (S-ANN) can be efficiently designed for brain-inspired computing employing spintonic devices.

GLSVLSI '18, May 23–25, 2018, Chicago, IL, USA

ACM ISBN 978-1-4503-5724-1/18/05…$15.00
https://doi.org/10.1145/3194554.3194618

2 APPROXIMATE COMPUTING

2.1 Spin-TD

In this section, we present spintronic Threshold Device (Spin-TD) based on a composite device structure consisting of a Domain Wall Motion magnetic stripe (DWS) and Magnetic Tunnel Junction (MTJ). The device structure is shown in Fig. 1a [3, 10]. It consists of a thin and short ($2nm \times 20nm \times 50nm$) magnetic DWS connecting two fixed anti-parallel magnetic domains. When the electrons are injected into the lateral terminals (T1 or T2), they become spin-polarized and exert a Spin-Transfer Torque (STT) on the Domain Wall (DW) (i.e., the transition area between two domains). This spin-polarized current can move DW within DWS. A fixed small magnet and DWS beneath it form a MTJ to read the state of DWS. It is noteworthy that an MTJ consists of two ferromagnetic layers (a free layer and a fixed one as shown in Fig. 1a) with a tunneling oxide (commonly MgO) barrier sandwiched between them [10].

The fixed layer of sense MTJ in Spin-TD is very small ($20nm \times 20nm$). The magnetization of DWS can be identified anti-parallel (AP) or parallel (P) to the fixed layer by injecting a current (larger than critical current) along it from its terminals (T1 to T2) or vice-versa [10]. Hence, the Spin-TD can detect the polarity of current flow at its input node, acting as an ultra-low voltage and compact current comparator. The resistance states are binary, i.e. either high (corresponding to AP configuration) or low (corresponding to P configuration) and can be read employing the Spin-TD sense circuit.

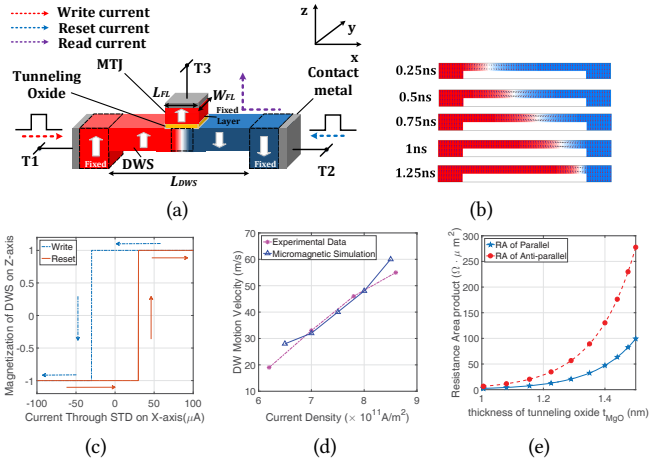

Figure 1: (a) Spin-TD structure, (b) Micro-magnetic simulation for the DW position, (c) Spin-TD transfer function and reset, (d) Simulated DW motion velocity vs. lateral current density, (e) Resistance-area product vs. the thickness of tunneling oxide in AP and P states.

Table 1: Device parameters used in simulation.

Symbol	Quantity	Values
α	Damping coefficient	0.02
K_u	Uniaxial anisotropy constant	$3.5 \times 10^5 J/m^3$
M_s	Saturation magnetization	$6.8 \times 10^5 A/m$
A_{ex}	Exchange stiffness	$1.1 \times 10^{-11} J/m$
P	Polarization	0.6
t_{MgO}	MgO thickness of MTJ	1.5 nm
$(L.W.t)_{DWS}$	DWS dimension	$50 \times 20 \times 2nm^3$

The threshold of Spin-TD, i.e. the minimum current magnitude required to switch the DWS magnetization (move DW from one end to the other end), is determined by the critical current density and DW velocity.

The transient micro-magnetic simulation of DW position (achieved from OOMMF [15]) is illustrated in Fig. 1b, using device dimension shown in Table 1, from 0.25 ns to 1.25 ns. Since the magnetization of DWS beneath the MTJ is fully switched at 1 ns, the Spin-TD intrinsic threshold (I_{th}) of this device can be considered $30\mu A$ within 1 ns corresponding to DW velocity of $\sim 50m/s$. Fig. 1c describes DWS magnetization switch corresponding to the applied current pulse (1 ns). A hysteresis effect can be observed due to DWM critical current density. We benchmarked the micro-magnetic simulation with the experimental data in [11] (the same nano-stripe width of 20nm is fabricated) and it shows a good match as shown in Fig. 1d. The MTJ is modeled using NEGF-LLG solution (non-equilibrium Green's function and Landau-Lifshitz-Gilbert equations) for spin to charge interface and calibrated with experimental data in [8, 11]. Resistance-area product vs. the thickness of tunneling oxide in AP and P states in this work considering a constant voltage of 50mV is plotted in Fig. 1e. For a 1 ns clock cycle, the oxide thickness in this work is chosen to be 1.5 nm that results in a total power dissipation of $\sim 1\mu W$ for the sensing circuit (including the clocking power). It is worth noting that in the sense circuit, the transient current with short duration (1 ns) and low magnitude ($\sim 2\mu A$) flows from T2 to T3, which will not disturb the state of DWS (domain wall position).

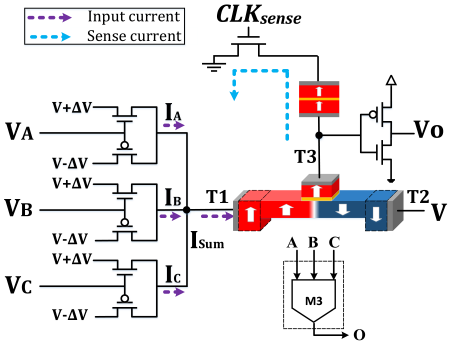

Figure 2: Hybrid Spin-CMOS 3-input majority gate.

2.2 Spin-CMOS majority gate

In this section, we present a highly-scalable spin-CMOS majority gate circuit design based on Spin-TD. The output of an n-input Majority Gate (MG) (n *is odd*) is determined by the majority of its inputs. For instance, the output is asserted to be logic value "1" only when more than $(\frac{n-1}{2})$ of the inputs are "1".

The proposed three-input MG circuit employing Spin-TD is shown in Fig. 2. The input terminal (T1) is connected to a network consisting of 3 pairs of NMOS-PMOS input transistors, in which all of the input transistors work as Deep Triode region Current Sources (DTCS) by applying $V + \Delta V$=550mV and $V - \Delta V$=450mV to the source and drain, respectively. The proposed circuit is controlled by two clock signals ($CLK_{compute}$ and CLK_{sense}) and each clock period is set to be 1 ns to synchronize with next stage circuits. Note that, T2 of Spin-TD is connected to a constant voltage of V=500mV and the voltage difference is ΔV=50mv, leading to an ultra-small voltage drop and correspondingly-low power consumption.

During the computation clock interval, the binary input voltages (VDD,GND) are applied at the gate of the input transistors, leading to input current flowing into (positive) or out of (negative) the connected Spin-TD. According to the principle of conservation of electric charge, the direction and magnitude of total current at intersection node depend on the algebraic sum of the input currents (I_A, I_B and I_C herein). This summation current (I_{Sum}) determines the position of DW as described in Section 2. By properly sizing the input transistors, the current flowing to T1 from each input branch is either $+30\mu A$ or $-30\mu A$ corresponding to input gate voltages as high ("1") or low ("0"), respectively. For instance, the input combination of (A,B,C)=(0,1,1) leads to (I_A,I_B,I_C)=($-30\mu A$, $+30\mu A$, $+30\mu A$) and the total current flowing into T1 is $+30\mu A$. Such current is equal to the threshold current of the Spin-TD and relocates the domain wall towards the T1 side, further resulting in the sense MTJ in an anti-parallel high resistance state. During the sense phase, when the CLK_{sense} is high, a voltage divider between Spin-TD's MTJ and a fixed reference MTJ is formed to sense the resistance state of spin-CMOS 3-input MG to produce reliable output voltage right after the inverter. In this case, the sensing circuit will generate a high output representing logic "1".

2.3 Spin-CMOS accuracy-configurable Adder

2.3.1 Functionality Analysis. A full adder (FA) is one of the most frequently-used components in arithmetic circuitry. In addition to its regular use for addition, it is employed in other arithmetic operations such as subtraction, multiplication, and division [3]. For

instance, multiplication has been implemented using successive additions. Moreover, FA is the key component and optimization target of many DSP algorithms. Hence, in order to obtain a high performance DSP system, we need to design energy efficient and low complexity adders [14]. While extensive work has been done in designing approximate adders [13, 17], the research efforts on accuracy-configurable approximate adders are limited. Let A, B, and C_{in} be inputs of an accurate full adder, the principle Boolean expression of Carry out (C_{out}) and accurate Sum (Sum_{acc}) of FA cell are as follows:

$$C_{out} = AB + AC_{in} + BC_{in} = M3(A, B, C_{in}) \tag{1}$$

$$Sum_{acc} = ABC_{in} + \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C_{in}} + A\bar{B}\bar{C_{in}} \tag{2}$$

Some Boolean expressions for Sum_{acc} and C_{out} of FA based on inverters and MGs have been reported in [4]. As can be seen in (1), C_{out} can be readily derived with a 3-input MG. Alternatively, Sum_{acc} can be obtained by using 3- and 5-input MG functions as:

$$\begin{aligned} Sum_{acc} &= ABC_{in} + (\overline{AB}.\overline{AC_{in}}.\overline{BC_{in}})(A + B + C_{in}) \\ &= M5(A, B, C_{in}, \overline{C_{out}}, \overline{C_{out}}) \end{aligned} \tag{3}$$

Table 2 shows the truth table of an FA. A close observation clarifies that six of eight outputs are correct if we make $Sum = \overline{C_{out}}$. Based on this observation, we propose a streamlined and cost-effective approximate FA circuit comprising one 3-input MG and one cascaded inverter. The approximate Sum output (Sum_{App}) of this adder is given by:

$$Sum_{App} = \overline{C_{out}} = \overline{M3(A, B, C_{in})} \tag{4}$$

2.3.2 Spin-CMOS Implementation. The proposed spin-CMOS implementation of the accuracy-configurable FA cell is shown in Fig. 3 consisting of two stages: Stage 1 to generate C_{out} and Sum_{app} and Stage 2 to generate Sum_{acc}. The first stage consists of a spin-CMOS MG realizing an approximate FA (App. FA) according to (1) and (4). As shown in Fig. 3, this circuit is designed with an appropriate fan-out for producing Sum_{App} output after one add-on inverter, while C_{out} is already achieved according to the Boolean expression in (1). Meanwhile, the $\overline{C_{out}}$ ($/Sum_{app}$) produced in Stage 1 is then connected to a similarly scaled input transistor network but with a $\frac{2w}{l}$ ratio to provide a double weighted current as expressed in (3). The double weighted current in conjunction with the sum of three primary inputs flow towards the T1 of the Stage 2's MG (realizing a 5-input MG as depicted in the logical schematic in Fig. 3). Consequently, the output voltage of this stage is Sum_{acc} realizing an accurate FA (Acc. FA). To provide the circuit with a proper and streamlined configurability, the wire connection between these two stages is regulated using a CMOS transmission

Table 2: Truth table for accurate and approximate FAs.

Inputs			Acc. Outputs		App. Outputs			
A	B	C_{in}	C_{out}	Sum	C_{out}		Sum	
0	0	0	0	0	0	✓	1	✘
0	0	1	0	1	0	✓	1	✓
0	1	0	0	1	0	✓	1	✓
0	1	1	1	0	1	✓	0	✓
1	0	0	0	1	0	✓	1	✓
1	0	1	1	0	1	✓	0	✓
1	1	0	1	0	1	✓	0	✓
1	1	1	1	1	1	✓	0	✘

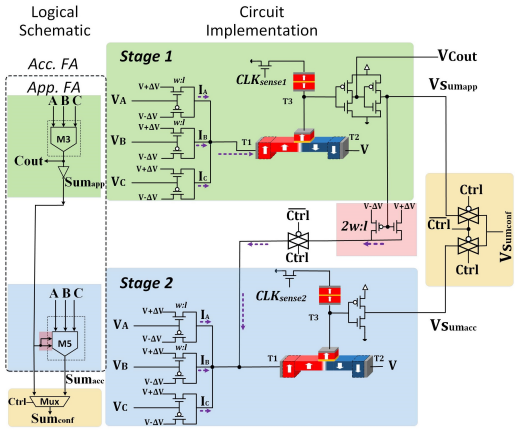

Figure 3: Logical schematic and circuit implementation of Spin-CMOS accuracy-configurable FA.

gate (TG). Furthermore, the sum outputs of both stages are laterally connected to a 2:1 CMOS multiplexer implemented utilizing two TGs to produce configurable sum (Sum_{conf}). Accordingly, the proposed spin-CMOS accuracy-configurable circuit operates in two different modes i.e. precision and approximation. In the precision mode, the control knob ($Ctrl$) is high, so the intermediate TG is ON and the double weighted current is routed to the second stage MG. Consequently, the circuit functions as an accurate adder since the second input of the multiplexer will be transmitted to the output ($Sum_{conf} = Sum_{acc}$). In the approximation mode, the $Ctrl$ is low and the double weighted branch is disconnected avoiding any switching activity in second stage. Therefore, the Stage 1's circuit works as a low power approximate adder when $Sum_{conf} = Sum_{app}$.

2.4 Performance

Comparison results between the proposed adder and previously published CMOS-, MTJ-, Spin Hall Effect (SHE)- and Domain Wall Motion (DWM)-based FAs are summarized in Table 3. Various metrics including the device count, total power consumption, and delay are considered for the comparison. Note that the accuracy-configurable circuit in this work is the only adder with the approximation configurability. For fair comparison, we have done fixed-voltage scaling to 180nm process node by using the appropriate scaling factor, which is $(1/S^2)$ for area and $(1/S)$ for energy [1]. The results clearly show that the proposed accuracy-configurable adder consumes less power than the other designs. For instance, 34.58% and 66% improvement in power consumption can be reported for the precision and approximation modes, respectively, over the best DWM-based FA design in [23]. In addition, compared to the recently-published work by Roohi et al. in [24], the proposed FA in precision mode can show ~ 12.7× and 2.3× smaller power and delay, respectively.

We expect that leveraging the proposed accuracy-configurable adder could provide limited accuracy loss for improvements in other circuit metrics such as power and speed while implementing image processing applications. To examine this, we take widely-used Discrete Cosine Transform (DCT)/ Inverse DCT (IDCT) as an image compression algorithm as an example. We use the approximation mode of the proposed accuracy-configurable FA only in the LSBs

Table 3: Comparison of FA designs.

Designs	Device count	Power	Delay	Config.
CMOS [21]	42 MOSs	71.1μW+ 0.9nW	2200ps	No
MTJ-based [21]	34 MOSs + 4 MTJs	2100 μW + 0nW	10200ps	No
SHE-based [24]	23 MOSs + 3 SHEs	710μW + 0nW	7000ps	No
LPM DWM [23]	20 MOSs + 4 MTJs + 2 DWSs	85μW+ 0nW	877ps	No
Prop. FA in prec. mode	28 MOSs+ 4MTJs+ 2DWSs	55.6μW+ 0nW	3000ps	Yes
Prop. FA in approx. mode	28 MOSs+ 4MTJs+ 2DWS	28.9μW+ 0nW	2000ps	Yes

of adders in a 20-bit DCT-IDCT architecture while exploiting the precision mode in MSBs.

Fig. 4a shows the output quality for the base case and five different degrees of approximations in PSNR. It can be seen that by increasing the approximation degree from the base case to 8 LSBs, the PSNR only drops by 2.93 dB. The power consumption of the DCT-IDCT circuit is evaluated using Synopsys Design Compiler for both pure-CMOS and spin-CMOS circuits as depicted in Fig. 4b. For pure-CMOS and spin-CMOS circuits, a Verilog code describing the truth table in Table 2 is considered for implementing the approximate adder based on existing and developed cell libraries, receptively, which is then used in 8-12 LSBs of a 20-bit DCT-IDCT architecture. Simulation results show that for all cases the power dissipation of the proposed spin-CMOS architecture is less than the CMOS counterpart. Evidently, by changing the degree of approximation, the power consumption of the entire system is changed. For instance, 31.33% power saving is obtained for the spin-CMOS architecture with 12 approximate LSBs in comparison with the base case, although the output quality is degraded to a PSNR of 23.75 dB. In a similar scenario, 8 approximate LSBs provide power saving of 20.4%, although the output quality is slightly degraded to 30.82 dB.

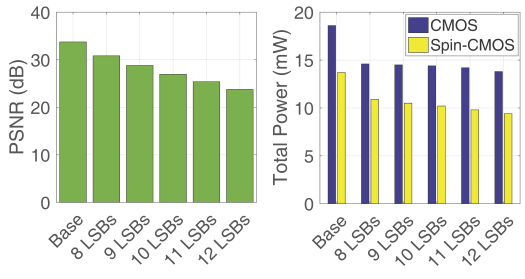

Figure 4: (a) Output quality comparison of different approximations, (b) Power consumption comparison of CMOS and spin-CMOS DCT-IDCT.

3 STOCHASTIC ARTIFICIAL NEURAL NETWORK (S-ANN)

3.1 Architecture of S-ANN

In this section, a stochastic-based ANN using Magnetic Tunnel Junction (MTJ) and domain wall device is presented [5], which has three main motivations. First, current emerging devices can not simply be considered as an alternative to replace CMOS due to large device variations. Thus, a new computation paradigm which embraces and exploits physical characteristics of spintronic devices instead of diminishing or circumventing them is essential. Second, stochastic computing offers much simpler logic operation such as multiplications and additions compared to expensive logic in deterministic method. For example, deterministic multiplications can be replaced with a simple AND operations of two random bit streams. Third, experimental results show that computation in the stochastic domain is much more robust than the deterministic method. The stochastic-based Artificial Neural Network (S-ANN) employs multiple controlled random bit streams instead of weighted sum operation in a deterministic ANN. Mathematically, we model the neuron and synapse function as the following equation:

$$Y = f(\sum X_i \oplus W_i - P_{T_i}) \tag{5}$$

where Y is the neuron output bit stream, X_i and W_i denote the i^{th} input bit stream and its corresponding synapse weighting random bit stream, respectively. In addition, P_{T_i} denotes a threshold in stochastic bit stream and f is the stochastic neuron transfer function. The stochastic soft-limiting function is modeled as:

$$f(v) = \begin{cases} 1 & \text{if} \quad v \geq T_i \\ P_{i-1} & \text{if} \quad T_{i-1} < v < T_i \\ \cdots & \cdots \quad \cdots \\ P_1 & \text{if} \quad T_1 < v < T_2 \\ 0 & \text{if} \quad v < T_1 \end{cases} \tag{6}$$

where v is the weighted sum of inputs in a stochastic bit stream, $T_{(1,\cdots,i)}$ is stochastic threshold range, $P_{(1,\cdots i)}$ is output probability. The training of S-ANN utilizes a conventional training algorithm, i.e. the backpropagation training algorithm. Such training process begins with initial weights, which are chosen randomly. The network processes training data and inputs to the weights and functions in the hidden layers. The resulting outputs are compared with the desired outputs. If there is an error, it then propagates back through the system, causing the system to adjust the weights for application and for the next data that needs to be processed. In training process, the deterministic sigmoid function is selected. Thus, the training process of S-ANN will not add any additional concerns. The training process of S-ANN does not involve in any stochastic and non-linear behavior within the S-ANN system. Once the training process is done, the probable stochastic approximations are applied to trained ANN, such as stochastic computing synthesis [2] and stochastic finite state machine [16]. Currently, most of stochastic ANNs employ finite state machine to approximate learned transfer function [26]. The transfer function is approximated by stochastic state machine using Markov chain theory. In Markov chain theory, each state is connected serially, and the transition of different states is decided by the input, which converts to the stochastic pulse stream. If the incoming pulse is '1' then the state will move forward; otherwise, it will keep at the current state. It is the probability of '1' appearing in the stochastic pulse streams. However, in the S-ANN design, we employ the DW device to approximate state and to integrate the input bitstream. It is feasible to use deterministic learning approach for this non-deterministic system because in S-ANN learning and processing are distinct and approximated by the stochastic algorithm. Fig. 5 shows an architecture of S-ANN. Instead of digital deterministic value, the S-ANN propagates random bit stream.

3.2 Stochastic Switching of MTJ Devices

Numerous experimental results have shown that spintronic devices exhibit complex switching behaviors due to the shifting of their

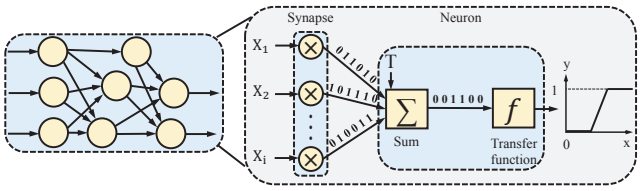

Figure 5: Architecture of stochastic neuron in S-ANN.

intrinsic magnetic moment (spin) of electrons. For example, the spin-torque switching characteristic of MTJ device is highly stochastic and exhibits a well-defined probability as shown in Fig. 6b. Several recent works have realized that the MTJ's switching probability (P_{sw}), depends on its intrinsic switching current and a thermal stability parameter (Δ). The thermal stability parameter Δ is modeled as $\Delta = E_u/k_BT$, E_u, k_B, and T are uni-axial magnetic anisotropy energy, Boltzmann's constant, and temperature, respectively. For example, if an initial state of MTJ is given as parallel state, a write current I_w applied on MTJ device with a pulse duration t can lead to a state switching under certain probability P_{sw}. This switching probability is defined as $P_{sw} = 1 - \exp(-t/\tau_p)$, where τ_p is the switching time constant. Some recent works show how to control the switching probability P_{sw} by changing the applied pulse duration and amplitude [12]. Therefore, using the applied pulse duration and amplitude, the switching probability can be concisely formulated as $P_{sw}(I) = 1 - \exp(-\frac{t}{\tau_p}\exp(-\Delta(1 - I/I_{c0})))$, where I_{c0} is the critical switching current at 0 K. In conclusion, the certain switching probability of a given MTJ device can be modeled by controlling the critical current I_c and the duration of applied pulse current τ_p. In Fig. 6a, experimental and analytical results of switching probability are plotted [22, 25].

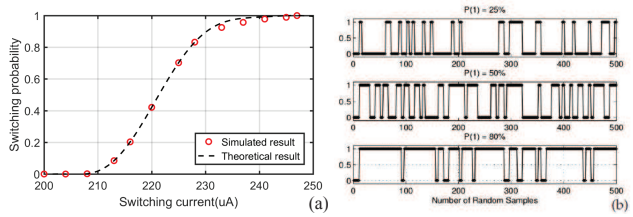

Figure 6: (a) Experimental and analytical results of switching probability [9, 22, 25], (b) SPICE simulation results of random signal generation.

3.3 Stochastic Synapse and Neuron

Fig. 7 shows the stochastic synapse design based on the reconfigurable random sample generator with one MTJ device and one DW device. The key idea of the stochastic synapse is to designing random sample generator by exploiting the stochastic switching behavior of an MTJ device at different input currents under a fixed pulse duration. The S-ANN architecture can be operated by two modes. In configuration mode (Fig. 7a), the proper DW position is programmed according to pre-computed stochastic weights. The required writing current to MTJ is generated upon DW resistance. During the operational mode shown in (Fig. 7b), depending on the applied input either a logic "0" or "1" value, the applied voltage V_c is written into the MTJ on the right hand. However, in order to sense the DW position, a vertical current density must be smaller than its critical value to avoid the shifting of DW position. Thus, a PMOS transistor to amplify a small input current into a larger output current is essential. In Fig. 7 c, d, e, the simulation results have shown that an input small sensing current ($< 30\mu A$) to PMOS transistor terminal can be effectively boosted into a larger current at the output with a supply voltage($V_{cc} = 450mV$). In this paper, we employ the conditional perturbation scheme [7] to generate random sample and achieve a bit rate 2.7 times faster and consume switch energy 6 times lower than conventional MTJ-based random number generator method.

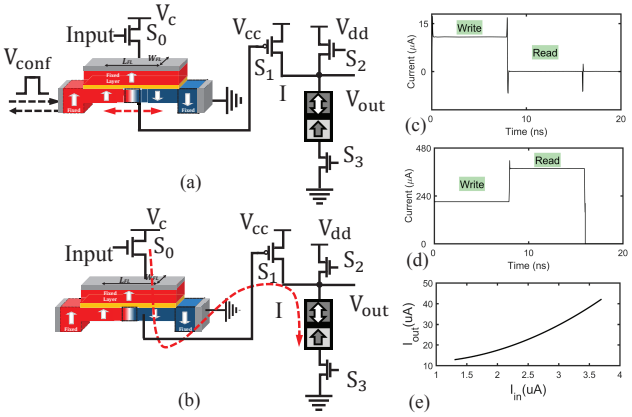

Figure 7: Circuit design of random bit stream generation. (a) Configuration mode, (b) Operational mode. Red curves depict signal directions. (c) Simulation of the current passed through DWS in write and read operation (d) Simulation of the current passed through MTJ, in write procedure, the input current is generated from a DW device and amplified by a PMOS transistor, (e) Output current.

To implement the stochastic neuron, multiple-phase pumping circuit is presented and depicted in Fig. 8b. Differ from deterministic DW device, the spin memristor DW device is employed [27]. In stochastic neuron, three key parameters: $th1$, $th2$, and H are defined. $th1$ and $th2$ denote the starting point and ending point of neural signal transformation, while $\Delta = \frac{H}{th1-th2}$ denotes the slope of a signal change. While a changing magnitude of current input to the stochastic neuron, the DW of DW1 becomes to move. When the I_{in} exceeds the critical of DW1, DW at DW1 starts to move. Finally, the last DWn will start to move, when it receives a signal from its previous stages. Therefore, one signal V_{in} is delivered to the last DW device (DWn) and can be amplified depending on the (DW_2, ..., DW_n). Firstly, we model the function of input current and DW movement x as $x_i = f(I_i)$. Using experimental results from DW devices, the function f is approximated as a linear function. Thus, the current passed to the next stage is equal to $\frac{V_{c1}(B\cdot x_1)+C}{A}$, where x_1 and v_{c1} are defined as DW movement and supply voltage in DW1, respectively. Consequently, the second stage output current is equal to $\frac{V_{c2}(B\cdot x_2)+C}{A}$, where x_2 is defined as DW movement in DW2 and equal to $x_2 = f(\frac{V_{c1}(B\cdot x_1)+C}{A})$. The output current at the last DW device can be calculated according to the equation on above. Thus, representing a tri-layered DW structure, the output current at the third DW device is amplified by factor $V_{c3} \cdot V_{c2} \cdot V_{c1}$, approximately.

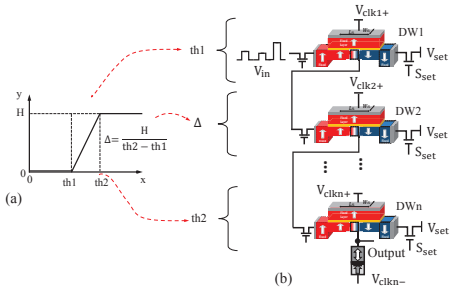

Figure 8: (a) Transfer function of ANN neuron, (b) Architecture of stochastic soft-limiting transfer function neuron.

3.4 Performance

In order to analyze the performance of the S-ANN, we select a well-known neural network task for handwritten digits recognition (MNIST). The MNIST dataset has 60K training samples and 10K testing samples. The simulation process contains two steps: training and pattern matching. In training process, the topology, synapse weights, and neural transfer functions of the S-ANN are obtained using the standard software and open resource code. Afterwards, the S-ANN is built up and predetermined input voltages. We summarize trade-off energy efficiency and recognition performance. In Fig. 9a and b, we present trade-off energy and inaccuracy between S-ANN and conventional CMOS-based stochastic ANN [6, 16, 18, 26]. Each experimental result is repeated ten times in order to minimize random errors. The inaccuracy calculation is based upon comparing the stochastic output with the corresponding probability. Compare to the other stochastic neural networks in CMOS, the S-ANN has high recognition rate. There are three reasons that consider as accuracy degradation: 1) the scaled addition of bit-stream 2) the Inter-stream correlation 3) transfer function FMS-approximation. To improve the accuracy of stochastic ANN, we use KCL summation to overcome scaled addition issue, MTJ-based random number generator to avoid inter-stream correlations, and emerging device soft-limit approximation to reduce the approximation error. In energy comparison, the S-ANN using MTJ-based true random number generator saves 8-10X energy compared with synthesized CMOS-based LFSR [16, 19, 20, 26]. In Fig. 9a and b, we compare the energy and inaccuracy with different bitstream sizes.

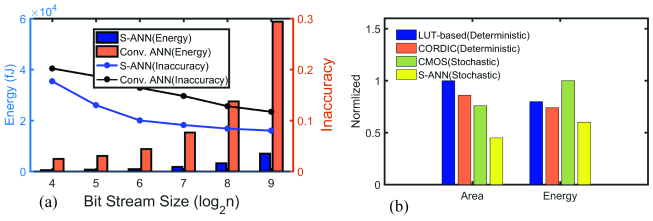

Figure 9: (a) Energy and inaccuracy comparison with different bitstream sizes. (b) Area and Energy comparison at same accuracy level.

4 CONCLUSION

In this paper, we presented two innovative spintronic computing primitives, i.e. spintronic approximate logic and spintronic stochastic neural network, which both leverage the intrinsic spintronic device physics to achieve much more compact and efficient designs than CMOS counterparts. In spintronic approximate logic, we employed the intrinsic current-mode thresholding operation to implement an accuracy configurable adder and further demonstrate its application in approximate DSP applications. In spintronic stochastic neuromorphic computation, we leveraged the stochastic properties of domain wall devices and magnetic tunnel junction to implement a low-power and robust artificial neural network design.

REFERENCES

[1] Zia Abbas and Mauro Olivieri. 2014. Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells. *Microelectronics Journal* 45, 2 (2014), 179–195.

[2] Armin Alaghi and John P Hayes. 2012. A spectral transform approach to stochastic circuits. In *ICCD*. IEEE.

[3] Shaahin Angizi et al. 2017. Composite spintronic accuracy-configurable adder for low power Digital Signal Processing. In *ISQED), 2017*. IEEE, 391–396.

[4] Mostafa Rahimi Azghadi, O Kavehie, and Keivan Navi. 2012. A novel design for quantum-dot cellular automata cells and full adders. *arXiv preprint arXiv:1204.2048* (2012).

[5] Yu Bai et al. 2017. Stochastic-Based Synapse and Soft-Limiting Neuron with Spintronic Devices for Low Power and Robust Artificial Neural Networks. *IEEE TMSCS* (2017).

[6] Bradley D Brown and Howard C Card. 2001. Stochastic neural computation. II. Soft competitive learning. *IEEE TC* 50 (2001), 906–920.

[7] Won Ho Choi et al. 2014. A Magnetic Tunnel Junction based True Random Number Generator with conditional perturb and real-time output probability tracking. In *IEDM*. https://doi.org/10.1109/IEDM.2014.7047039

[8] Xuanyao Fong et al. 2011. KNACK: A hybrid spin-charge mixed-mode simulator for evaluating different genres of spin-transfer torque MRAM bit-cells. In *SISPAD*. IEEE, 51–54.

[9] Xuanyao Fong et al. 2014. Generating true random numbers using on-chip complementary polarizer spin-transfer torque magnetic tunnel junctions. In *72nd DRC*. https://doi.org/10.1109/DRC.2014.6872318

[10] Xuanyao Fong et al. 2016. Spin-transfer torque devices for logic and memory: Prospects and perspectives. *IEEE TCAD* 35, 1 (2016), 1–22.

[11] S Fukami et al. 2013. 20-nm magnetic domain wall motion memory with ultralow-power operation. In *IEDM*. IEEE, 3–5.

[12] Akio Fukushima et al. 2014. Spin dice: A scalable truly random number generator based on spintronics. *Applied Physics Express* 7 (2014), 083001.

[13] Vaibhav Gupta et al. 2011. IMPACT: imprecise adders for low-power approximate computing. In *17th IEEE/ACM ISLPED*. IEEE Press, 409–414.

[14] Jie Han and Michael Orshansky. 2013. Approximate computing: An emerging paradigm for energy-efficient design. In *ETS*. IEEE, 1–6.

[15] http://math.nist.gov/oommf/. [n. d.]. ([n. d.]).

[16] Yuan Ji et al. 2015. A hardware implementation of a radial basis function neural network using stochastic logic. In *DATE, 2015*. 880–883.

[17] Honglan Jiang et al. 2017. A review, classification, and comparative evaluation of approximate arithmetic circuits. *ACM JETC* 13 (2017).

[18] H. Li et al. 2006. A Stochastic Digital Implementation of a Neural Network Controller for Small Wind Turbine Systems. *IEEE TPE* (Sept 2006). https://doi.org/10.1109/TPEL.2006.882420

[19] Peng Li et al. 2012. The synthesis of complex arithmetic computation on stochastic bit streams using sequential logic. In *ICCAD*. ACM.

[20] Z. Li et al. 2016. DSCNN: Hardware-oriented optimization for Stochastic Computing based Deep Convolutional Neural Networks. In *ICCD*. https://doi.org/10.1109/ICCD.2016.7753357

[21] Shoun Matsunaga et al. 2008. Fabrication of a nonvolatile full adder based on logic-in-memory architecture using magnetic tunnel junctions. *Applied Physics Express* 1, 9 (2008), 091301.

[22] N. Onizawa et al. 2014. Analog-to-stochastic converter using magnetic-tunnel junction devices. In *NANOARCH*. 59–64. https://doi.org/10.1109/NANOARCH.2014.6880490

[23] Arman Roohi et al. 2016. A tunable majority gate-based full adder using current-induced domain wall nanomagnets. *IEEE TMAG* 52, 8 (2016), 1–7.

[24] Arman Roohi, Ramtin Zand, et al. 2017. Voltage-based concatenatable full adder using spin Hall effect switching. *IEEE TCAD* 36 (2017).

[25] A.F. Vincent et al. 2015. Spin-Transfer Torque Magnetic Memory as a Stochastic Memristive Synapse for Neuromorphic Systems. *Biomedical Circuits and Systems, IEEE Transactions on* (2015), 166–174.

[26] Shaodi Wang et al. 2017. Hybrid VC-MTJ/CMOS non-volatile stochastic logic for efficient computing. In *Proceedings of the Conference on Design, Automation & Test in Europe*. European Design and Automation Association, 1442–1447.

[27] X. Wang and Y. Chen. 2010. Spintronic memristor devices and application. In *DATE 2010*. 667–672. https://doi.org/10.1109/DATE.2010.5457118