Hybrid VC-MTJ/CMOS Non-volatile Stochastic Logic for Efficient Computing

Shaodi Wang, Saptadeep Pal, Tianmu Li, Andrew Pan, Cecile Grezes, Pedram Khalili-Amiri,
Kang L. Wang, and Puneet Gupta
Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA
Corresponding Authors:{shaodiwang,puneet}@ucla.edu

***Abstract*—In this paper, we propose a non-volatile stochastic computing (SC) scheme using voltage-controlled magnetic tunnel junction (VC-MTJ) and negative differential resistance (NDR). The proposed design includes a VC-MTJ based true stochastic bit stream generator and VC-MTJ and NDR based stochastic adder, multiplier, register, which are experimentally demonstrated using 60nm VC-MTJ and CMOS NDR connected on die. These components are then used to realize FIR filter and AdaBoost (machine-learning algorithm). 3X - 37X energy advantage is shown for the proposed SC compared with CMOS binary arithmetic ASIC and SC designs.**

***Index Terms*—Non-volatile, voltage-controlled magnetic tunnel junction, negative differential resistance, stochastic computing**

I. Introduction

As CMOS technology is approaching its fundamental limitation, the improvement of traditional computing system faces many challenges including the conflicts among the limitation of memory bandwidth [1], performance, requirement of increased computing resources (e.g., number of cores) and stringent power and thermal constraints. Alternative accelerators and non-volatile memory (NVM) based computing are potential solutions to sustain the continuous technology development due to their low power, computing efficiency, and memory persistence [2–9] allowing efficient power gating.

As a class of accelerators, stochastic computing (SC) [10–12] intrinsically has great advantageous energy-efficiency due to very simple hardware implementation for logic operations such as addition and multiplication. Applications which doesn't rely on precise computation can potentially benefit from the use of SC in terms of energy efficiency, speed and high fault tolerance, e.g., digital signal processing applications, machine learning, and neural network [13]. Recently, early evaluation of SC designs using stochastic NVM like memristors [14–16] and all spin logics [17], have shown significant improvement in energy efficiency.

However, challenges exist in the adaption of SC designs into modern computing systems. For example, NVM based SC designs like [14] introduce additional memory read and write to feed and fetch data from CMOS logic unit, moreover, the endurance limitation and high write voltage are not compatible with on-chip system. SC designed by all spin logic [17] has reliability and efficiency concerns on the spin channels, and design challenges of stochastic bit streams generator (SBSG). CMOS SC uses Linear-feedback shift register (LFSR) to generate stochastic bit streams (SBS), which consumes high energy and offsets the benefit brought by SC [13].

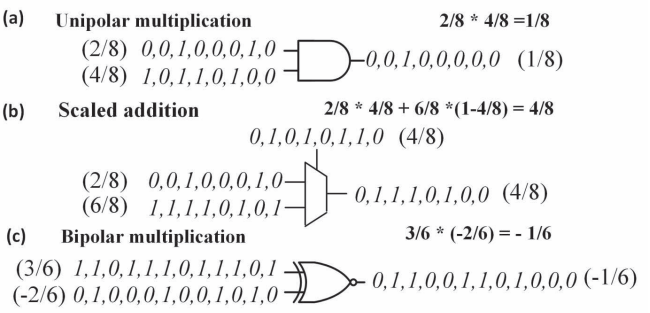


Fig. 1: Multiplication and addition using unipolar and bipolar encoded SBS. Unipolar coding represents decimal number ranging in [0,1], while bipolar coding is for decimal number in [-1,1]. The SC computations are bit-wise, where corresponding bits in two input SBS are operated using the AND, MUX, or XNOR gates.

In this paper, we propose a practical non-volatile (NV) computing system using stochastic logic built by voltage-controlled magnetic tunnel junctions (VC-MTJs) [18, 19] and CMOS based negative differential resistance (NDR). VC-MTJ is one of the fastest and lowest-power emerging NVM technology, where a state switching takes $< 1ns$, with switching energy $< 1fJ$. Unlike traditional NV logic and SC using NVM as additional data backup, where computation is still in CMOS logic units, the proposed system directly computes data on VC-MTJs, eliminating the need for the memory read and write and communication between NVM and CMOS logic. Thus, the proposed computing architecture is intrinsically fast and energy-efficient. In addition, VC-MTJ and NDR enable the design of robust SBSG, which generates truly random SBS with least design challenge. The NDR and MTJ based logic operations using 60nm in-house built VC-MTJs [20] and CMOS NDR are demonstrated.

Our contributions are summarized as follows.

- We proposed deterministic VC-MTJ write.
- WE proposed VC-MTJ and NDR based SC logic.
- We proposed a practical true SBSG design using VC-MTJ and NDR. The design is as precise as CMOS LFSR SBSG but consumes 55X less energy.
- We evaluate VC-MTJ and NDR based SC, which has 2-25X and 4-37X better energy-efficiency than binary ASIC and CMOS SC respectively for different applications and computing precisions.

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

This paper is outlined as follows. In Section II, we introduce SC, VC-MTJ, NDR, and the interaction of NDR and VC-MTJ. In Section III, we describe operations in the VC-MTJ SC design. In Section IV, we describe the SBSG built by VC-MTJ and NDR. In Section V, we evaluate the proposed SC for finite impulse response (FIR) filter and Adaboost design [21], and compared with CMOS binary and CMOS SC designs. The paper is concluded in Section VI.

II. VC-MTJ and NDR in SC

SC [10–12] uses the fraction of "1"s in an SBS to represent a fraction number. Two common encoding methods are unipolar and bipolar. For a unipolar encoded SBS, the fraction of "1"s is the represented number, e.g., 6 "1"s out of 8 bits is 0.75. By contrast, in a n-bit bipolar encoding, a number with m '1"s represents $m/n - 1/2$. SC computation using SBS is bit-wise. As shown in Fig. 1, the multiplication of unipolar SBS is implemented by an AND gate, while that of bipolar SBS is implemented by an XNOR gate. Scaled addition is commonly used in SC instead of normal addition for hardware simplicity, which is implemented by a MUX with an selection SBS ($4/8$ in Fig. 1b) for both unipolar and bipolar encoding..

Due to the bit-wise computing nature, SC is robust to most hardware failures and soft errors, where limited bit false flips are tolerable. Moreover, parallelism of SC is straightforward that can be done by duplicating logic functions. Nevertheless, the hardware resource linearly increases with application precision, which is determined by SBS length. In addition, SC computation inherently has non-deterministic output, e.g., up to $3 \cdot 10^{-4}$ output variation in a 1024-bit XNOR operation. The variation is higher for operands close to 0.5, and hence can be mitigated by avoiding using such numbers [13].

A. VC-MTJ

VC-MTJ is a resistive memory device, whose resistance is determined by the magnetization directions of two ferromagnetic layers. The direction of one layer is fixed (with respect to reference layer) while the other one can be switched (referred to free layer). Low and high resistance states (LRS and HRS) are obtained by changing the magnetic directions into parallel (*P* state) or anti-parallel (*AP* state) respectively. The resistance difference is quantified by tunnel magnetoresistance ratio (TMR, defined as $(R_H - R_L)/R_L$), where VC-MTJ's TMR of over 150% has been demonstrated.

VC-MTJ is a uni-polar memory technology, which is switched by one-directional voltage pulse. Fig. 2 shows the voltage-controlled magnetic anistropy (VCMA) effect in VC-MTJs. The energy barrier (E_B) separates two stable states of the free layer magnetization (pointing up and down). When a positive voltage is applied across the VC-MTJ, E_B decreases due to VCMA effect, and as voltage reaches V_c (about 1V), precessional switching is fully activated when the energy barrier is close to 0. The magnetization direction spins fast ($< 1ns$ for 180^o degree) in precessional switching. If a long pulse is applied, the magnetization is finally aligned with horizontal external magnetic field after several turns.

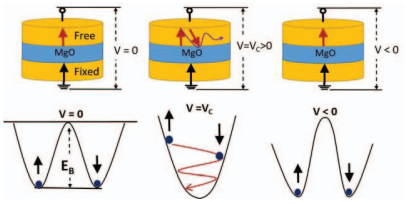


Fig. 2: VCMA-induced precessional switching. A positive (negative) voltage on a VC-MTJ reduces (increases) the energy barrier separating the two magnetization states. $V = V_c$ leads to a full energy barrier reduction, and precessional switching is conducted by magnetic field provided by buried layer. Same write pulse is used for the symmetric two switching directions.

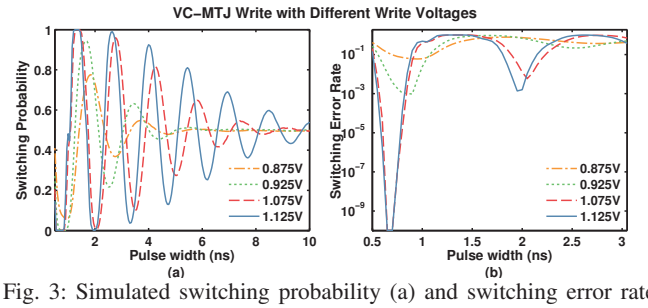


Fig. 3: Simulated switching probability (a) and switching error rate (b) as functions of pulse width for different write voltages using an experimentally verified model in [22, 23].

1) VC-MTJ write: A successful switching alternates the MTJ state by controlling pulse width to the half of switching cycle. A switching error may occur if applied voltage is not sufficient or the pulse width mismatches the processional switching cycle [22, 24]. Fig. 3 illustrates the switching behavior of VC-MTJ. With appropriate write voltage (e.g., 1.075V to 1.125V), a switching can be completed in 700 to 800 ps pulse with error rate $< 10^{-10}$. The switching probability converges to 0.5 with long pulse, where other resistive NVM with stochastic write converge to 1, e.g., memristor and spin-transfer-torque MTJ. The convergence to 0.5 is the key for efficient SBSG design (section IV). The convergence is faster with lower voltage (e.g., the 5ns convergence time for 0.875-0.925V) for that damping field is not fully removed.

Thanks to the voltage (electric field) induced switching, VC-MTJ is generally designed with thick MgO layer, which leads to high resistance ($> 200k\Omega$) and hence reduces write leakage current and energy. Every VC-MTJ switching consumes about $1fJ$, which results in the lowest energy among existing NVM [18]. Please note that, with low write current, switching effect induced by current (e.g., STT) is minimized, creating symmetric switching for $HRS \rightarrow LRS$ and $LRS \rightarrow HRS$.

2) VC-MTJ read: A VC-MTJ read signal uses the reversed bias direction of the write (i.e., positive voltage is applied on the cathode), which increases energy barrier and device

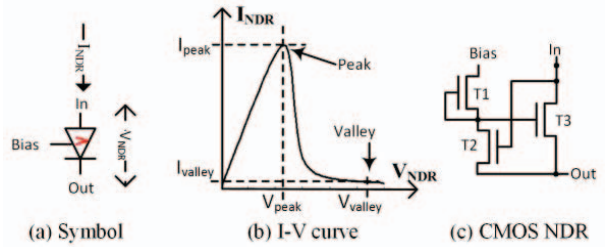


Fig. 4: (a) The symbol, (b) CMOS design, and (c) I-V characteristic of an negative differential resistance.

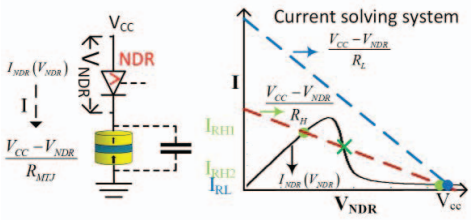


Fig. 5: (a) Series connection of a VC-MTJ and NDR device. (b) NDR (solid line) and VC-MTJ current (dashed lines) as a function of voltage drop on NDR (V_{NDR}). Three possible solutions exist for HRS, where the left and right ones are stable, while the middle one is unstable. In the beginning of our designs, the V_{NDR} always starts from 0 which guarantees the solution starts at the left one (high current). For the LRS, only one low current solution exists.

stability (Section 2), leading to non-destructive read [25].

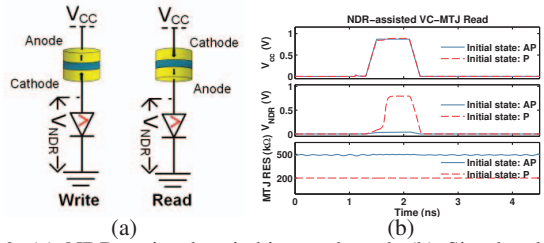
B. Negative Differential Resistance and VC-MTJ

In the proposed non-volatile stochastic computing. negative differential resistance (NDR) [26] works as logic elements which directly interact with VC-MTJ based registers. An NDR element with a 3T CMOS circuit and corresponding I-V characteristics are shown in Fig. 4. This NDR circuit has three terminals (IN, OUT, and BIAS). With a non-zero voltage (0.8V in our design) on the BIAS, the current through IN and OUT (I_{NDR}) behaves like Fig. 4b, where a negative differential resistance exists between the peak and valley. When 0 voltage is on BIAS, the NDR behaves as an OFF transistor. This circuit is capable of low V_{peak} (< 0.1V) and highly tunable I_{peak} and peak-to-valley current ratio (PVR). We have demonstrated this structure experimentally using NMOS transistors on a single die. We are able to tune the peak current from the nA to μA range with a peak voltage of 0.25V while achieving PVR in excess of 1,000.

As is seen in Fig. 5, by placing an NDR element in series with a VC-MTJ and choosing I_{peak} between R_H and R_L lines of the VC-MTJ, a high current is obtained (the left green solid dot) when VC-MTJ is in HRS and a close-to-0 current is obtained (right blue solid dot) otherwise. This feature enables logic operations using MTJ in the proposed SC.

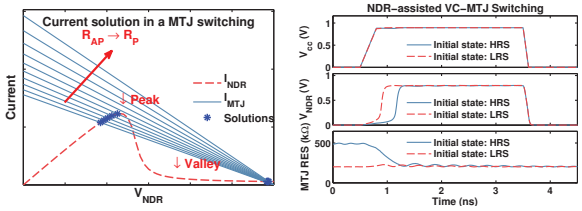
1) Non-destructive VC-MTJ read with NDR: Given that a reversed voltage increases the energy barrier of VC-MTJ and stabilizes the state rather than destroys it, a high voltage (V_{CC}) is allowed for read. In a NDR-assisted read, the NDR is serially connected with the VC-MTJ as shown in Fig. 6a. We simulated the read process in Fig. 6a with 0.9V V_{cc}, the V_{NDR} has full voltage difference for HRS and LRS.

2) Deterministic VC-MTJ write with NDR: The VC-MTJ has symmetric switching between two resistance states, which uses same pulse in VC-MTJ based memory. To switch an



(a) (b)

Fig. 6: (a) NDR-assisted switching and read. (b) Simulated waveforms of a NDR-assisted read.



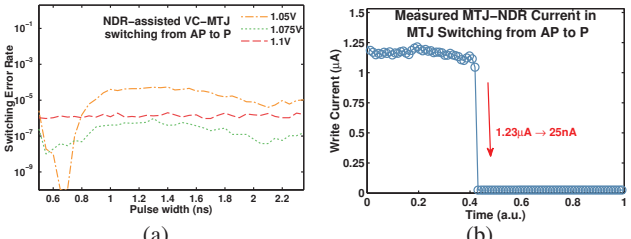
(a) (b)

Fig. 7: (a) The current solution change of serially connected NDR and VC-MTJ with VC-MTJ switching from high resistance to LRS. (b) The simulated VC-MTJ write curves for two directions. The $HRS \rightarrow LRS$ switching is assisted by NDR, in which voltage on VC-MTJ turns to 0 automatically upon resistance decreases to read NDR peak current. The $LRS \rightarrow HRS$ switching is prohibited by NDR.

VC-MTJ deterministically without NDR, a read operation is required firstly, and then the subsequent switching pulse is waived if the MTJ is already in target state [22, 27].

With the assistance of NDR, deterministic write from HRS to LRS is achieved by serially connecting VC-MTJ and NDR like Fig. 6a. The NDR-MTJ current solution changes when an VC-MTJ switching from HRS to LRS as shown in Fig. 7a. For a HRS-to-LRS switching, MTJ-NDR starts at the high-current solution when a write voltage is applied, then the current increases with decreasing VC-MTJ resistance, but suddenly drops to I_{valley} after reaching NDR peak current (i.e., VC-MTJ resistance line passes over the NDR peak). Reversely, if VC-MTJ is initially in LRS, when a voltage is applied, the NDR quickly goes to and stays at the valley region (in about 0.1ns) with most voltage dropped on it, and the remaining voltage on VC-MTJ cannot trigger precessional switching. **This indicates that the NDR can precisely control write pulse in programming VC-MTJ from HRS to LRS and prohibits the switching from LRS to HRS, which creates a deterministic write for HRS to LRS.** The write process is simulated in Fig. 7b. The switching error rate for the deterministic write is illustrated in Fig. 8a. Compared with Fig. 3, NDR significant relaxes the precise pulse width requirement. In addition, with NDR, the switching rate from LRS to HRS is prohibited to $< 10^{-10}$ (the simulation accuracy), which is lower than the SC natural computing error rate in SC (~ 0.0001).

We have experimentally demonstrated the NDR's functionality on 60nm VC-MTJs with NMOS built NDR. The 60nm VC-MTJ's LRS and HRS resistance are 240 $k\Omega$ and 320 $k\Omega$ respectively. As Fig. 8b shows, when the VC-MTJ switches, the current suddenly drops from 1.2 μA to 25 nA with



(a) (b)

Fig. 8: (a) Switching error rate of NDR-assisted VC-MTJ write from HRS to LRS. The $LRS \rightarrow HRS$ switching rate is $< 10^{-10}$, which is not shown. (b) Experimentally measured MTJ-NDR current change in a VC-MTJ switching from HRS to LRS.

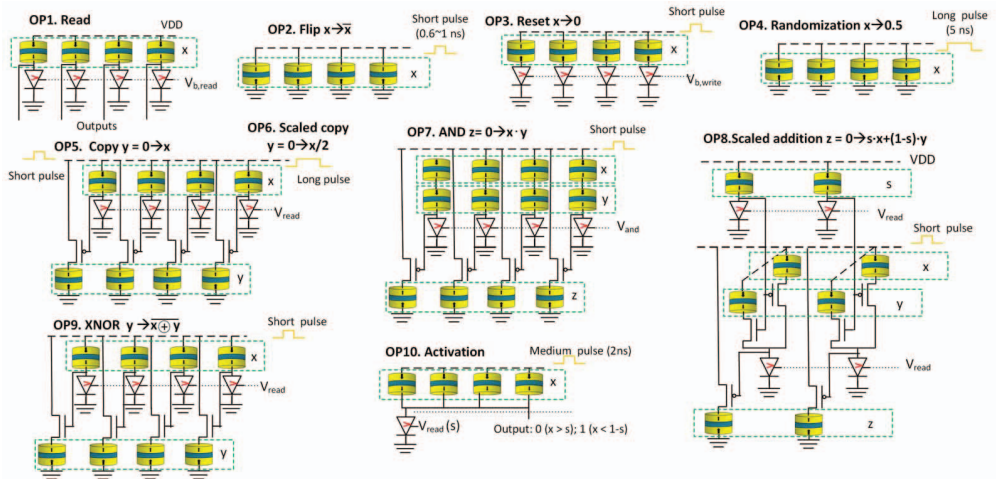


Fig. 9: VC-MTJ and NDR built SC logic operations. Where a long low-voltage pulse is used to randomize VC-MTJ states, and a short high-voltage pulse is used to switch VC-MTJ states. Every VC-MTJ array stores an SBS. All VC-MTJs in an array are computed simultaneously for throughput and design efficiency purpose. Please note that the XNOR gate directly changes the array Y's data to $\overline{X \oplus Y}$.

voltage on VC-MTJ changing 64X. This current termination demonstrates the deterministic NDR-assisted write, and the possibility to output large read-out voltage swing .

C. Reliability of NDR-assisted Write and Read

The PVR and variation of peak current are trade-off in NDR design optimization, which are mainly affected by transistor threshold voltage (V_t) variation. The peak current is usually only sensitive to one transistor's V_t. For the PVR of 10, 10% peak current change is seen with 25 mV V_t shift. With error rate of 10^{-3} allowed for a 256-bit SC, $\sim$ 100 mV V_t variation can be tolerated, which is beyond 5σ of V_t change [28].

III. VC-MTJ Based Operations in Stochastic Computing

Addition, subtraction, and multiplication are the basic stochastic logic operations [10–12]. Other operations including division are derived from addition and multiplication, which are usually not as efficient as addition and multiplications.

The VC-MTJ SC uses the same logic operations as CMOS for SC computing, including AND for unipolar coding multiplications, MUX for scaled addition, XNOR for bipolar multiplication (see Fig. 1). However, the contributions in this paper are MTJ-NDR based SC logic gates and registers, where SBS are directly stored and computed in VC-MTJs. We have designed logic operations including AND, MUX, and XNOR, SBS generation, threshold function (e.g., activation used in machine learning application), and other operations allowing VC-MTJ to move data like dynamic flip flop (DFF) i.e., copy and reset. In the remaining of the paper, HRS is recognized as 1 state, while LRS is 0 state for simplicity. The designed logic operations are shown in Fig. 9. All operations are based on the experimentally demonstrated VC-MTJ switching [20], NDR-assisted write and read (see Section II).

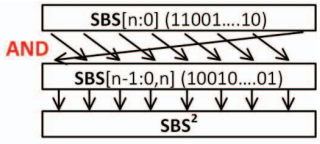


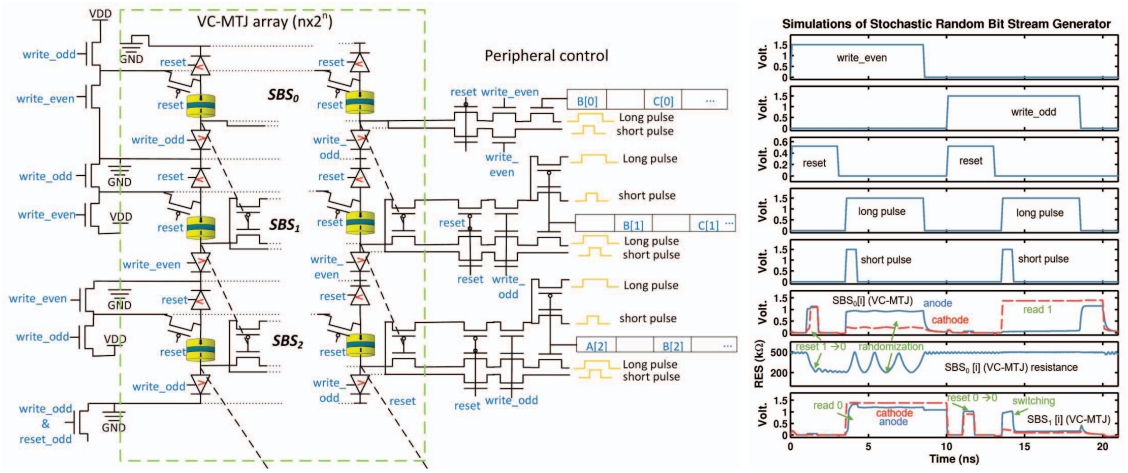
Fig. 10: Removing correlation using shuffle operation for SBS^2.

OP 1-3 are based on simple write and read as explained in section II. OP4 generates a random bit stream with 50% of "1"s, based on the observation in Fig. 2 that a long write pulse leads to an VC-MTJ switching probability to 50%. The convergence to 50% probability takes about 5ns for write voltage between 0.875 V and 0.925 V. To accomplish OP 5-9, a reset must be performed firstly on the output VC-MTJ array. OP5 (OP6) combines read and flip (randomization) operations, which results in a copy (scaled copy) operation. In OP7, NDR is sized or biased to differentiate the highest series resistance combination from two VC-MTJs. Two serially connected 1-state VC-MTJs (in array x and y) allow NDR to stay at low voltage bias, thus the PMOS is turned on, and a short pulse passes to switch the corresponding output MTJ to 1. When two read VC-MTJs are in other states, NDR is in the valley, drops most voltage, turns off the PMOS, so that output MTJ stays 0 state. This design completes an AND operation. OP8 is a scaled addition with a selection array (array s) selecting corresponding data from array x and y to be copied to z array. OP9 is a NDR XNOR gate, the VC-MTJ in array y is flipped for corresponding "0" in x. OP10 is an activation function for machine learning applications. When the number of 1-state VC-MTJs is over a threshold, "1" is output. It needs a large sized NDR (e.g., with 2 mA peak current) for judging the parallel resistance of a VC-MTJ array. The proposed logic operations are significantly cheaper than corresponding CMOS ones. One NDR contains only three minimal sized transistors, whereas a CMOS DFF has 14-20 transistors.

The VC-MTJ and NDR based logic operations are simulated with 60nm verified VC-MTJ model [22, 23] and 45nm SOI library. The energy and delay are listed in Table I.

TABLE I: Simulated energy per bit and delay of VC-MTJ-NDR based logic operations. Interconnect and fan-out load is considered.

	read	flip	reset	rand	copy
Energy (fJ)	1.01	2.57	1.79	16.5	3.34
Delay (ns)	0.7	1	1	5	1
	scaled copy	AND	addition	XNOR	activation
Energy (fJ)	15.2	5.23	3.74	3.53	1.16
Delay (ns)	5	1.8	1.2	1	5



(a) (b)

Fig. 11: (a) The schematic of a pipe-lined SBS generator with binary fraction input A[n-1:0], B[n-1:0], and C[n-1:0]. n VC-MTJ arrays (every one stores an SBS) are divided into even and odd groups based on the index. At every cycle (10ns), one group is written according to the read-out of the other group. Output SBS is generated every two cycles because of the pipe-line. (b) Simulated waveforms of two-cycle SBS generation. The MTJs in SBS_0 are written in the first cycle ($write\_even$: high, while the MTJs in SBS_1 are written in the second cycle ($write\_odd$: low) with the read-out from SBS_0.

An SC operation with two correlated SBS would result in an unwanted false output. Thus eliminating correlations between computed SBS is important to maintain SC accuracy. This can be solved by shuffling the bits of a stochastic number. One example in Fig 10 shows how a simple shuffle/shift removes SBS correlation between the input and output. Please note that this correction can be done by simply changing the interconnects, which results in no hardware overhead.

IV. Stochastic Bit Stream Generator

SBSG has been recognized as the bottleneck of previous SC works [13, 17]. CMOS LFSR based generator consumes high energy, while other stochastic memory based generators, e.g., memristor [14] and STt-MTJ [29], suffer from challenges of creating precise bias voltage for accurate switching probability. Utilizing the operations in Section III, we propose a practical true SBSG with VC-MTJ and NDR. This SBSG does not have precision limitation and design challenge.

As explained in Section II and III, when a long pulse is applied on VC-MTJ, its switches to 1 with 50% probability. We utilize this feature to translate n-bit binary fraction floating number IN[n-1:0] to SBS[2^n-1:0]. One example is illustrated in Fig. 12. The process starts from the least significant bit IN[0]. If IN[0] is 1, the first array SBS_0 is randomized to 0.5, otherwise to 0. Then upon the next bit, if IN[1] is 0, an $scaled\_copy$ (i.e., OP6 in Fig. 9, where the "1"s have 50% to be copied to the next SBS) is performed from SBS_0 to SBS_1,

Binary input: IN[2:0] = **.101** (5/8, decimal)

Step	Input		Operations	Decimal value	SBS_0	SBS_1	SBS_2
1			Reset SBS_0: SBS_0[i] = 0	SBS_0 = 0	00000000		
2	IN[0]	**1**	**Randomization**: SBS_0[i]=random	SBS_0 = IN[0]*1/2	01101001		
3			Reset SBS_1: SBS_1[i]=0	SBS_1 = 0	01101001	00000000	
4	IN[1]	**0**	**Scaled copy**: if SBS_0[i]=1, then SBS_1[i]=random	SBS_1 = SBS_0/2 = IN[0]*(1/2)² + IN[1]* (1/2)	01101001	01001000	
5			Reset SBS_2	SBS_2 = 0		01001000	00000000
6	IN[2]	**1**	**Copy_and_rand**: if SBS_1[i]=1, then SBS_2[i]=1, else SBS_2[i]=random	SBS_2 = (1-SBS_1)/2 + SBS_1 =IN[0]* (1/2)³ + IN[1]*(1/2)² + IN[2]*(1/2)		01001000	01101011 SBS output

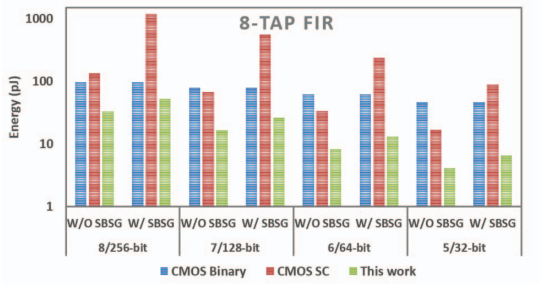
Fig. 12: An SBS generation example. The input 0.101 (binary) is translated to SBS (01101011).

and otherwise a $copy\_and\_rand$ is performed (i.e., the "1"s in SBS_0 are copied to SBS_1, the remaining "0"s in SBS_1 are then randomized). The $scaled\_copy$ obtains half of the origin SBS number, while the $copy\_and\_rand$ obtains half of the origin SBS number plus 0.5. In other words, the previous MTJ array is half copied to the current array, whether a 0.5 is added depends on corresponding bit in IN. This process continues and obtains output at step n (e.g., n=8 for a 256-bit SBS) The generation can be pipe-lined such that an SBS is generated at every clock cycle. In addition, there is no correlation between consequent SBS.

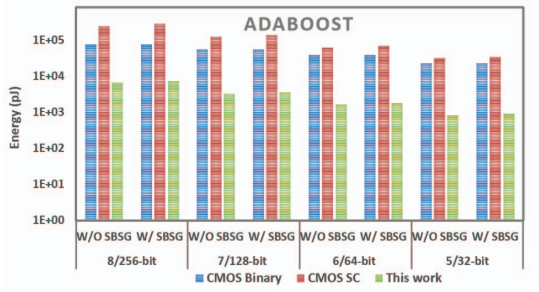
The schematic of an pipe-lined SBS generator is shown in Fig. 11a. Every VC-MTJ array stores an SBS. The VC-MTJ arrays are divided into even and odd groups according to the index. At every cycle, write operations are performed in one group using the data read from the other group and input binary fraction number A, B, C. The $write\_even$ and $write\_odd$ take turn to select group to read or write. A reset operation (controlled by $reset$ signal) and an SBS array operation ($scaled\_copy$ and $copy\_and\_rand$) are performed in every clock cycle. Input A, B, C are shifted in the registers in sequence. Output is generated every two clock cycles.

Two-cycle SPICE simulated waveforms are shown in Fig 11b. In the first cycle, $write\_even$ is high, and an VC-MTJs at SBS_0 is firstly reset and then randomized because input B[0] is 1. In the second cycle, $write\_odd$ is high, and a reset pulse on 0-state VC-MTJ at SBS_1 is prohibited by the NDR, and then the VC-MTJ is switched by a $copy$ operation since its corresponding VC-MTJ at SBS_0 is in 1.

In the proposed SBSG, every bit generation involves $9n$ transistors, whereas the CMOS LFSR generator involves n DFFs (12-20 transistors in a DFF), many computing logic gates, and a n-bit comparator (about $4n$ logic gates). Thanks to the efficiency of VC-MTJ and the generation scheme, the VC-MTJ based SBSG saves 55X energy compared with the synthesized CMOS LFSR based generator (see Section V).



(a)



(b)

Fig. 13: (a) Computing energy of 8-tap FIR for fix-point width from 5-bit to 8-bit (32-bit to 256-bit for uni-polar encoded SC). (b) Computing energy of 32-classifier Adaboost with 32-pixel input image for fix-point width from 5-bit to 8-bit (32-bit to 256-bit for bipolar encoded SC). The wire activity is 0.375 for both CMOS binary and SC designs, and 1 for VC-MTJ and NDR based SC design. Energy are shown for two categories: energy with SBSG (W/ SBSG) and energy without SBSG (W/O SBSG).

V. Evaluation

We evaluate the proposed VC-MTJ SC designs for FIR and Adaboost [21] (a machine learning algorithm commonly used for face detection). We synthesize CMOS binary logic designs with fixed-point width from 5-bit to 8-bit and corresponding CMOS SC implementation with SBS width from 32-bit to 256-bit. CMOS LFSR based generators are also synthesized for SBS bandwidth of 32 bits to 256 bits. The VC-MTJ and NDR based SC are simulated using HSPICE with experimentally verified VC-MTJ model [22, 23] and 45nm SOI CMOS library. As is illustrated in Fig. 13, VC-MTJ SC shows 3X to 25X advantageous energy-efficient compared with CMOS binary designs. SC is more efficient in low-precision designs and less efficient in high-precision designs, because SC design cost linearly scales with precision, while binary design logarithmically scales. The energy-benefit of the proposed SC against CMOS binary designs is better in Adaboost (12-25X) than in FIR (3X to 7X), because Adaboost relatively contains more adders and multipliers. The computation energy (without SBS generation) of CMOS SC is slightly lower than CMOS binary for low-precision applications. However, the advantage disappears when including the energy of the inefficient LFSR based generator, which is also observed in [13]. Please note that, the comparison here is for computing one output from high-activity designs. The energy benefit of the proposed SC is expected to further increase for low-activity applications, where the non-volatility of VC-MTJ allows for immediate power gating with least energy overhead.

VI. Conclusion

In this paper, we proposed a practical NV SC and a truly random SBSG built by VC-MTJ and NDR. The functionality of the NV SC logic gates is based on experimentally demonstrated NDR-assisted VC-MTJ write and read. The proposed SBSG design consumes 55X lower energy than CMOS LFSR based SBSG. For applications including FIR and Adaboost, the proposed SC is 3-25X and 4-37X more energy-efficient compared with CMOS binary and CMOS SC designs respectively.

References

[1] Kyungsu Kang, Luca Benini, and Giovanni De Micheli. "A high-throughput and low-latency interconnection network for multi-core clusters with 3-d stacked l2 tightly-coupled data memory". *Proc. VLSI-SoC*. IEEE. 2012.

[2] Xiaoxiao Liu et al. "RENO: A high-efficient reconfigurable neuromorphic computing accelerator design". *Proc. DAC*. IEEE. 2015.

[3] Zheng Li et al. "An overview on memristor crossabr based neuromorphic circuit and architecture". *Proc. (VLSI-SoC)*. IEEE. 2015.

[4] Mohsen Imani, Abbas Rahimi, and Tajana S Rosing. "Resistive configurable associative memory for approximate computing". *Prof. DATE*. IEEE. 2016.

[5] Ibrahim Kazi et al. "A ReRAM-based non-volatile flip-flop with sub-V T read and CMOS voltage-compatible write". *Proc. NEWCAS*. Ieee. 2013.

[6] Wang Kang et al. "Reconfigurable codesign of STT-MRAM under process variations in deeply scaled technology". *IEEE TED* (2015).

[7] Wang Kang et al. "Spintronics: Emerging ultra-low-power circuits and systems beyond MOS technology". *JETC* (2015).

[8] Arne Heittmann and Tobias G Noll. "Limits of writing multivalued resistances in passive nanoelectronic crossbars used in neuromorphic circuits". *GLVLSI*. ACM. 2012.

[9] Fabien Clermidy et al. "Resistive memories: Which applications?" *Prof. DATE*. European Design and Automation Association. 2014.

[10] Brian R Gaines. "Stochastic computing systems". *Advances in information systems science*. Springer, 1969.

[11] Weikang Qian and Marc D Riedel. "The synthesis of robust polynomial arithmetic with stochastic logic". *Proc. DAC. 45th ACM/IEEE*. IEEE. 2008.

[12] Bradley D Brown and Howard C Card. "Stochastic neural computation. I. Computational elements". *IEEE Transactions on computers* (2001).

[13] Kyounghoon Kim et al. "Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks". *Proceedings of the 53rd Annual Design Automation Conference*. ACM. 2016.

[14] Siddharth Gaba et al. "Stochastic memristive devices for computing and neuromorphic applications". *Nanoscale* (2013).

[15] Alexander Stotland and Massimiliano Di Ventra. "Stochastic memory: memory enhancement due to noise". *Physical Review E* (2012).

[16] Stefan Slesazeck et al. "Physical model of threshold switching in NbO 2 based memristors". *RSC Advances* (2015).

[17] Rangharajan Venkatesan et al. "Spintastic: spin-based stochastic logic for energy-efficient computing". *Proc. DATE*. IEEE. 2015.

[18] P Khalili Amiri et al. "Electric-field-induced thermally assisted switching of monodomain magnetic bits". *Journal of Applied Physics* (2013).

[19] Yoichi Shiota et al. "Induction of coherent magnetization switching in a few atomic layers of FeCo using voltage pulses". *Nature materials* (2012).

[20] C Grezes et al. "Ultra-low switching energy and scaling in electric-field-controlled nanoscale magnetic tunnel junctions with high resistance-area product". *Applied Physics Letters* (2016).

[21] Gunnar Rätsch, Takashi Onoda, and K-R Müller. "Soft margins for AdaBoost". *Machine learning* (2001).

[22] Shaodi Wang et al. "Comparative Evaluation of Spin-Transfer-Torque and Magnetoelectric Random Access Memory". *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* (2016).

[23] C. Grezes et al. "Write Error Rate and Read Disturbance in Electric-Field-Controlled MRAM". *IEEE Magnetics Letters* (2016).

[24] Shaodi Wang et al. "MTJ variation monitor-assisted adaptive MRAM write". *Proceedings of the 53rd Annual Design Automation Conference*. ACM. 2016.

[25] H. Lee et al. "Source Line Sensing in Magneto-Electric Random-Access Memory to Reduce Read Disturbance and Improve Sensing Margin". *IEEE Magnetics Letters* (2016).

[26] Shaodi Wang et al. "Tunneling Negative Differential Resistance-Assisted STT-RAM for Efficient Read and Write Operation". *IEEE Transactions on Electron Devices* (2017).

[27] H. Lee et al. "Design of a Fast and Low-Power Sense Amplifier and Writing Circuit for High-Speed MRAM". *Magnetics, IEEE Transactions on* (2015).

[28] Shaodi Wang et al. "Evaluation of digital circuit-level variability in inversion-mode and junctionless FinFET technologies". *IEEE Transactions on Electron Devices* (2013).

[29] Lirida Alves de Barros Naviner et al. "Stochastic computation with Spin Torque Transfer Magnetic Tunnel Junction". *New Circuits and Systems Conference (NEWCAS), 2015 IEEE 13th International*. IEEE. 2015.