Work-in-Progress : A Scalable Stochastic Number Generator for Phase Change Memory Based In-Memory Stochastic Processing

Supreeth Mysore Shivanandamurthy, Ishan G Thakkar, Sayed Ahmad Salehi
Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA
supreethms@uky.edu, igthakkar@uky.edu, SayedSalehi@uky.edu

EXTENDED ABSTRACT

Stochastic computing based Processing-In-Memory (PIM) architectures (e.g., [1]) can provide massive parallelism with higher energy-efficiency, for implementing complex computations in main memory. However, stochastic computing arithmetic requires random bit streams generated by stochastic number generators (SNGs), which account for significant area and energy consumption. Moreover, SNGs' numerical precision needs improvement to reduce errors in stochastic computations [1]. Thus, low numerical precision and high implementation overheads of SNGs can offset the benefits of adopting stochastic computing in PIM architectures. In this paper, we exploit the inherent stochasticity of Phase Change Memory (PCM) cells to design a scalable and area-energy efficient SNG for PCM-based stochastic PIM architectures. Our designed SNG can achieve up to ~300× lower area and up to ~250× lower energy consumption with better numerical precision, compared to the Linear Feedback Shift Register (LFSR) based conventional SNG from [2].

KEYWORDS : Phase Change Memory(PCM), Stochastic Number generator(SNG), Processing-In-Memory(PIM).

1 BACKGROUND: PHASE CHANGE MEMORY

A PCM cell embeds a small volume of chalcogenide material $Ge_2Sb_2Te_5$ (GST) [3], which can be programmed into two different states (i.e., crystalline/SET state and amorphous/RESET state) with dramatically different electrical resistance [3]. The amorphous (RESET) state represents a binary "0", while the crystalline (SET) state represents a "1". The resistance of PCM cells in the SET state (R_{SET}) follows a normal distribution in the kΩ range [4], whereas the resistance of PCM cells in the RESET state (R_{RESET}) follows a normal distribution in the MΩ range [4], as shown in Fig. 1-(B). Fig. 1-(A) shows a PCM cell array where PCM rows share sense amplifiers (S/As). To read a PCM row, a read current pulse I_{READ} is passed through each PCM cell. Consequently, the resistance of each PCM cell (R_{SET} or R_{RESET}) generates a voltage drop ($V_{SET} = R_{SET}\times I_{READ}$ or $V_{RESET} = R_{RESET}\times I_{READ}$), which is compared against a reference voltage (V_{REF}) in each cell's respective S/A (Fig, 1-(A)), to distinguish the cell as storing a '1' (if $V_{SET}<V_{REF}$) or a '0' (if $V_{RESET}>V_{REF}$). Typically, for a PCM array, I_{READ} and V_{REF} are judiciously designed such that V_{REF}/I_{READ} falls in between the resistance distributions of SET and RESET cells (Fig.1-(B)). As a result, for a PCM row being read, V_{SET} for all SET cells is less than V_{REF}, whereas V_{RESET} for all RESET cells is greater than V_{REF}, enabling the distinction of the SET cells (logic '1's) from the RESET cells (logic '0's) with 100% probability.

Now consider that a SET PCM row with all its cells pre-programmed in the SET state (storing '1's) is being read. For this read operation, if we can control V_{REF} such that V_{REF}/I_{READ} falls somewhere on the resistance distribution of the SET cells, we can control the number of cells that would be read as '1's. For example, if we can control V_{REF} to be V_{REF}', as shown in Fig. 1-(B), V_{REF}'/I_{READ} would fall at the center of the resistance distribution for SET cells. As a result, only 50% of the cells in the SET PCM row would be read as '1's (as these cells fall on the left of the V_{REF}'/I_{READ} reference), the remaining 50% cells would be read as '0's. *Learning from this observation, we leverage a judicious control of the PCM read operation to design an efficient SNG, as described next.*

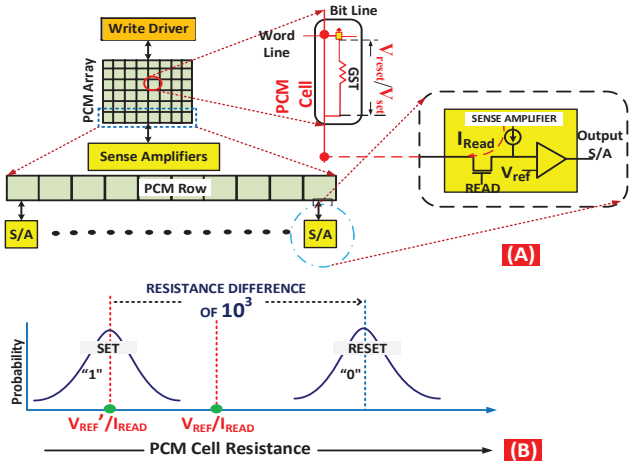

Fig. 1: Illustration of (A) PCM cell array, and (B) PCM resistance distributions for SET and RESET cells.

2 STOCHASTIC NUMBER GENERATOR

Fig. 2 presents a functional block diagram of our proposed stochastic number generator (SNG). To convert an N-bit input number (Fig. 2-1) into a 2^N-bit stochastic bit-vector (Fig. 2-4), our SNG employs a PCM-based Gaussian digital-to-analog converter (GDAC) (Fig. 2-2) that provides V_{REF} for reading a row of 2^N PCM cells (Fig. 2-3) pre-programmed in the SET state. Fig. 2-2 illustrates the GDAC operation for a 3-bit binary input ($B_2B_1B_0$), which can be generalized to any N-bit binary input ($B_{N-1}..B_1B_0$). In

CODES/ISSS '19 Companion , October 13–18, 2019, New York, NY, USA

ACM ISBN 978-1-4503-6923-7/19/10…$15.00
https://doi.org/10.1145/3349567.3351717

general, an N-bit GDAC contains N switches (S_0 to S_{N-1}), each of which corresponds to a bit (B_0 to B_{N-1}) in the N-bit input number, and depending on the bit's value ('0' or '1'), connects its respective voltage source (V_0 to V_{N-1} – implemented using charge pumps [5]) to the voltage summer circuit. We assume that the cumulative distribution function (CDF) for the resistance of PCM SET cells (Fig. 2-3) is available at the design time. Therefore, the voltage level V_X of the voltage source, corresponding to a bit B_X from the N-bit input, can be determined at the design time as: $V_X = I_{READ} \times CDF^{-1}\{(2^X/2^N)\}$, where $CDF^{-1}\{\}$ is the inverse CDF that gives the resistance value R_X (Fig. 2-3 and 2-4) for the given probability number. Accordingly, our proposed N-bit GDAC produces V_{REF} as: $V_{REF} = \sum_{X=0}^{N-1} B_X V_X$, using the embedded voltage summer circuit (Fig. 2-2). When this V_{REF} is used to read the PCM row with 2^N SET cells (Fig. 2-3), only $(\sum_{X=0}^{N-1} B_X 2^X)$ number of cells out of total 2^N cells are read as '1's, and the remaining cells are read as '0's, thereby converting an N-bit input into a 2^N-bit stochastic output.

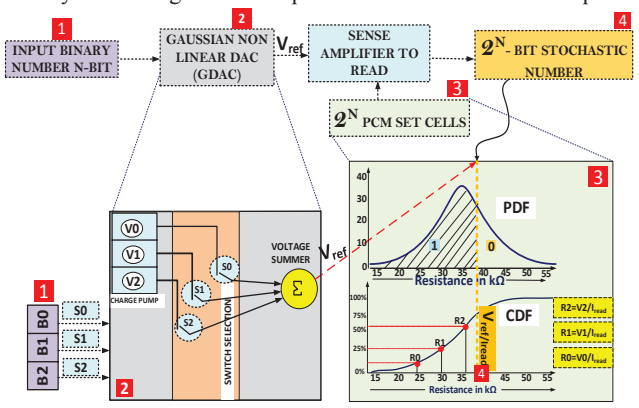

Fig. 2: Functional block-diagram of our proposed Phase Change Memory (PCM) based Stochastic Number Generator (SNG).

For example, let us understand how this works for a 3-bit input number 101 (i.e., B_2=1, B_1=0, B_0=1). To convert this number into a stochastic number, our SNG should be able to produce a vector of total 8-bits with five '1's and three '0's. In our SNG (Fig. 2), the voltage level V_2 of the voltage source, corresponding to B_2=1, can be evaluated as $V_2 = I_{READ} \times CDF^{-1}\{2^2/2^3\} = I_{READ} \times CDF^{-1}\{4/8\}$. Similarly, V_1 and V_0 can also be evaluated. Consequently, our proposed GDAC produces $V_{REF} = B_0V_0+B_1V_1+B_2V_2 = V_0+V_1$, as B_1 is zero for our example number 101. This V_{REF} is used to read total 2^3=8 PCM SET cells, and doing so results in only $B_02^0+B_12^1+B_22^2 = 2^0+2^2 = 5$ cells to be read as '1's, and the remaining 3 cells to be read as '0's, hence, correctly converting the binary input 101 into an 8-bit stochastic number.

Note that the proposed SNG does not generate true random numbers with uniform distribution. Nevertheless, it can efficiently control the number of '1's in its output bit-stream, which fulfils the sufficient requirement for our proposed SNG to be used in stochastic PIM architectures [1].

3 RESULTS

We evaluated the area and energy consumption of our proposed SNG and compared the results with conventional SNG from [2], for different bit-sizes of the input binary number from 4-bits to 14-bits. We used Cadence's Spectre for SPICE-level simulations of the LFSR-based conventional SNG and our GDAC-based SNG with the op-amp based voltage summer. We take the area, energy, and delay values for PCM from [3] and for charge-pump based voltage-sources from [5]. The μ and σ for the SET resistance distribution for PCM cells are evaluated from [4] to be 34.15kΩ and 6.54kΩ, respectively. (Fig.(3-A)) and (Fig.(3-B)), respectively, show the area consumption and energy values, for the two SNG designs, evaluated for 14nm technology by scaling the SPICE-based results for 45nm technology. From the figures, as the bit-size of the input binary increases from 4-bits to 14-bits, the energy and area values for our GDAC-based SNG hardly increases, compared to the exponential increase in the energy and area values for the conventional LFSR-based SNG. As a result, for a 14-bit input number, our GDAC-based SNG consumes ~300× less area and ~250× less energy, compared to the LFSR-based SNG.

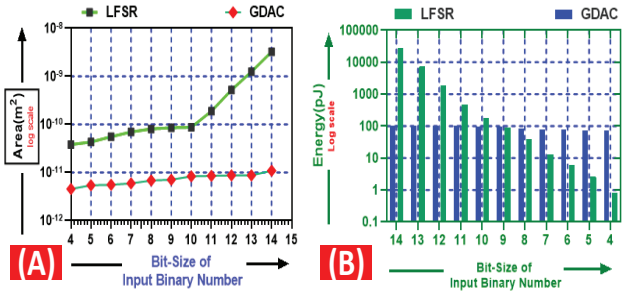

Fig. 3: (A) Area and (B) energy consumption values of our GDAC-based SNG and conventional LFSR-based SNG for various bit-sizes of input binary number.

4 SUMMARY AND FUTURE WORK

In summary, the use of our PCM-based SNG for stochastic PIM architectures provides the following benefits: (i) Our SNG can convert any N-bit binary number into a 2^Y–bit stochastic number, by simply using the V_{REF} generated by our GDAC to read total 2^Y PCM SET cells. Here, Y can be any number ≥ N, as abundant number of PCM cells would already be available in PCM based PIM architectures, which provides an unprecedented opportunity to improve the precision of the generated stochastic numbers without adding any significant conversion logic/circuit and related overheads in our designed SNG. *(ii)* As PCM SET cells are already be available in a PCM-based PIM architecture, GDAC is the only block of our SNG that incurs area overhead. Moreover, unlike the LFSR-based SNGs from [2], our GDAC-based SNG can generate all 2^Y bits of the output stochastic number in parallel, which results in huge latency and energy savings for our GDAC-based SNG.

In the future, we plan to evaluate the conversion-based and precision-based error-efficiencies for our proposed SNG. Moreover, we intend to do a comprehensive comparative analysis of our proposed SNG with other SNG designs from prior work. Further, we will do detailed case studies to explore the utilization of our designed SNG for various stochastic Processing-In-Memory (PIM) applications, including the deep neural network inference.

REFERENCES

[1] S. Li et al., "SCOPE: A Stochastic Computing Engine for DRAM-based In-Situ Accelerator," in Proc. MICRO, 2018.

[2] K. Kim et al., "An Energy-Efficient Random Number Generator for Stochastic Circuits," in Proc. ASPDAC, 2016.

[3] I.G. Thakkar et al., "DyPhase: A Dynamic Phase Change Memory Architecture with Symmetric Write Latency and Restorable Endurance," IEEE TCAD, 2017.

[4] G. W. Burr et al., "The inner workings of phase change memory: Lessons from prototype PCM devices," IEEE Globecom 2010.

[5] L. Jiang et al., "A low power and reliable charge pump design for phase change memories," in Proc. ISCA, 2014.