

Spintronics based Stochastic Computing for Efficient Bayesian Inference System

Xiaotao Jia¹, Jianlei Yang², Zhaohao Wang¹, Yiran Chen³, Hai (Helen) Li³, and Weisheng Zhao¹

¹Fert Beijing Research Institute, BDBC, School of Electronic and Information Engineering,

²Fert Beijing Research Institute, BDBC, School of Computer Science and Engineering,
Beihang University, Beijing, 100191, China

³Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, U.S.A.
Email: jianlei@buaa.edu.cn

Abstract— Bayesian inference is an effective approach for solving statistical learning problems especially with uncertainty and incompleteness. However, inference efficiencies are physically limited by the bottlenecks of conventional computing platforms. In this paper, an emerging Bayesian inference system is proposed by exploiting spintronics based stochastic computing. A stochastic bitstream generator is realized as the kernel components by leveraging the inherent randomness of spintronics devices. The proposed system is evaluated by typical applications of data fusion and Bayesian belief networks. Simulation results indicate that the proposed approach could achieve significant improvement on inference efficiencies in terms of power consumption and inference speed.

Keywords— Bayesian Inference, Stochastic Computing, Spintronics, Magnetic Tunnel Junction

I. INTRODUCTION

The rise of deep learning has greatly promoted the development of artificial intelligence, however, most modern deep learning models face several difficulties such as the requirement of large scale training data and overfitting problem during learning. Furthermore, they can neither represent the uncertainty and incompleteness of the world nor take advantages of well-studied experience and theories. In order to overcome these limitations, some researches trend to utilize Bayesian inference or combine Bayesian approaches with deep learning. Bayesian inference provides a powerful approach for information fusion, reasoning and decision making that has established it as the key tool for data-efficient learning, uncertainty quantification and robust model composition. It is widely used in applications of artificial intelligence and expert systems, such as multisensor fusion [1] and Bayesian belief network [2]. Recent years, Bayesian approaches attract the attention of neural network researches. Several studies (such as [3]) have been proposed to combine advances in Bayesian approaches into neural network learning.

Bayes' theorem is the theoretical foundation of Bayesian inference and the key operation is probabilistic computing. The implementation of probabilistic algorithms on floating-point architecture has some disadvantages such as inefficiency in terms of power consumption, computing speed and memory usage and the inability to exploit parallelism of the Bayesian inference [4]. Further, as the scaling of feature size

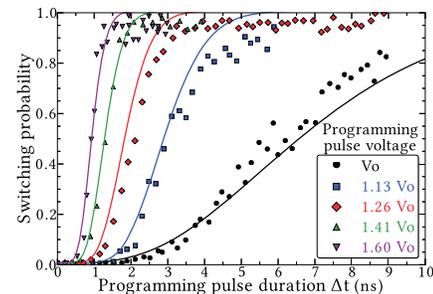


Fig. 1: Experimental measurements of the switching probability with respect to the duration of the applied programming pulse, for different programming voltages. [6]

of transistor, physical phenomena, such as low noise margin, low supply voltage, manufacturing process variations and soft errors, makes traditional integrated circuits much error-prone [5]. Consequently, unconventional computing method - stochastic computing (SC), that directly addresses these issues has attracted much attention. Enable very low-cost implementations of arithmetic operations using standard logic elements and high degree of error tolerance are two main attractive features of stochastic computing [5].

The separation of processing units and memories remains a fundamental principle of von Neumann architecture computers even though there are many efforts towards increasing parallelism [7]. In order to improve Bayesian inference efficiency, several different specific hardware or circuits have been proposed such as FPGA [8] and analog circuits [9]. Even though these works make an improvement on inference efficiency, there are still some shortcomings with the consideration of stochastic computing. Stochastic computing is executed using stochastic bitstreams. In most previous works, stochastic bitstreams (SB) are generated utilizing pseudo-random number generators (RNG) and comparators as shown in Fig. 2(a). Unfortunately, generating (pseudo-)random bits is fairly costly. Therefore, the gate-level advantage of stochastic computing is typically lost. Towards to resolve these shortcomings, emerging nanometer-scale devices such as spintronics are considered as the major breakthroughs. In particular, magnetic tunnel junctions (MTJ) are well suited for bitstream generation because of its attractive feature such as non-volatile, low power and stochastic (Fig. 1). Several strategies have been proposed to generate stochastic bitstreams with spintronic devices [10]–[12]. However, shortcomings still exist in terms of power, area or speed. And none of them

This work was supported in part by the National Natural Science Foundation of China (Grand No. 61602022) and the 111 Talent Program B16001.

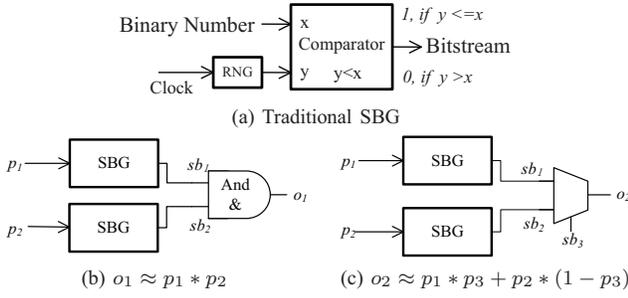


Fig. 2: (a) SBG circuit; (b) Multiplication operation using AND gate; (c) scaled addition operation using multiplexer (MUX).

explain how to incorporate the stochastic bitstream generator with real world applications.

In this paper, a Bayesian inference system with less power consumption and high inference speed is built by stochastic computing based on spintronic devices and applied to traditional Bayesian inference applications. The main contributions of this work are listed as follows:

- A complete scheme of MTJ based stochastic bitstream generator (SBG) is proposed. Simulation results indicate that the stochastic bitstreams generated by the proposed SBG are with high accuracy and low correlation.
- Two efficient Bayesian inference systems are proposed utilizing the SBG and applied to data fusion and Bayesian belief network. Simulation results show that both two applications could achieve reasonable results with less energy, higher speed.

The remainder of this paper is organized as follows. Section II states some preliminaries and related works. The diagram of Bayesian inference system is illustrated in Section III. Section IV describes details of SBG. Bayesian inference systems for two real world applications are proposed in Section V. Finally, conclusion is given in Section VI.

II. BACKGROUND

A. Stochastic Computing

SC was first introduced in the 1960's by von Neumann [13]. The basic idea of SC is that a number is presented by the ratio of '1' in a SB and arithmetic operations are implemented using simple logic gate(s) as shown in Fig. 2(b)(c). It is worth to note that SBs which are highly correlated are not as expected, because higher correlation would lead to lower computing precision. In order to meet the requirement of sufficient random and uncorrelated, pioneer researcher proposed several SBG models such as linear feedback shift registers (LFSRs) [14], weighted binary SNG [15]. However, these CMOS based SBGs consume too much energy and area.

B. MTJ Basics

The core part of the MTJ is a sandwich structure consisting of two ferromagnetic (FM) layers sandwiched with a tunneling barrier. One FM layer is called as reference layer with fixed magnetization direction. The other FM layer is called as free layer whose magnetization direction could be parallel (P) or antiparallel (AP) with that of reference layer. Because of the tunnel magnetoresistance effect, the

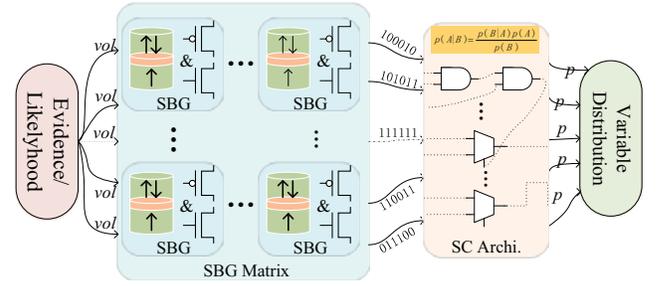


Fig. 3: Diagram of the proposed Bayesian inference system.

nanopillar resistance depends on the relative orientation (P or AP) of the magnetization directions of the two FM layers. An applied field can switch the free layer between the two directions. The stochastic behavior of MTJ switching has been revealed by [16], which results from the unavoidable thermal fluctuations of magnetization [17]. The stochastic switching is very suitable for generating stochastic bitstreams.

Recently, the work in [10] proposes an SBG based on MTJ. But the circuit is too simple, and its implementation may be incomplete. Furthermore, it does not consider the correlation of different SBs which may result in inaccuracy of SC. A novel computing system using stochastic logic built by voltage-controlled MTJs (VC-MTJs) is proposed in [12]. This system consumes less energy and circuit area compared with LFSR circuits. But in this system, the bit generation still involves too many MTJs and transistors. Bitstream correlation is considered in this paper, but the proposed shuffle operation could not remove the relevance essentially and arithmetic operations between them maybe result in an unexpected number. For example, a bitstream sb_1 ('10101010') presenting 0.5 will be turned into sb_2 ('01010101') with the proposed shuffle operation in [12]. However, the result of $sb_1 \& sb_2$ will be 0 rather than 0.25.

III. DIAGRAM OF BAYESIAN INFERENCE SYSTEM

Fig. 3 describes the diagram of the proposed Bayesian inference system (BIS). The input of BIS is a series of bias voltages corresponding to evidence or likelihood. These evidences or likelihood may come from sensors in robot, autonomous, etc., also may come from clearly fact such as the X-ray results in Bayesian belief net for cancer diagnosis. SBG matrix within light blue rectangle and SC architecture within light yellow rectangle are two key components of BIS. SBG matrix is utilized to generate SBs based on input voltages. Its scale is related with evidence count and variable relations. Each SBG is a hybrid MTJ/CMOS circuit yielding SB with fast speed, low power and high accuracy. Details of SBG are described in Section IV. SC architecture is constructed by simple logic gates such as AND gate or multiplexer (MUX) and takes SBs as inputs. The goal of SC architecture is to implement Bayesian inference utilizing SBs and SC theory on the basis of Bayes' Rule. In this architecture, stochastic computing is achieved by a novel arrangement of AND gates and MUXs and the interconnections between them. Usually, different applications are solved by different inference algorithms, thus, require different computing architectures which could be found in

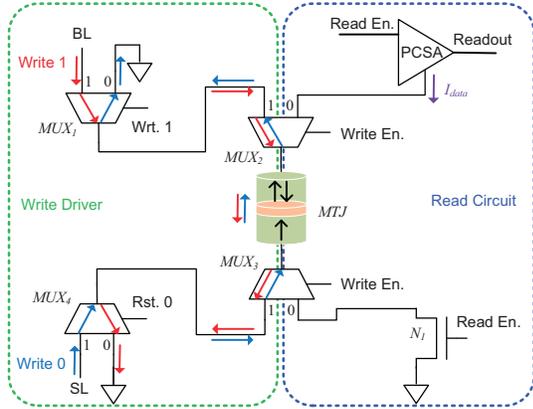


Fig. 4: Schematic of SBG circuit.

Section V. Finally, inference results are presented by the format of random variable distribution which could provide guidance for decision making.

IV. MTJ BASED STOCHASTIC BITSTREAM GENERATOR

Accuracy of Bayesian inference is mainly determined by the quality of bitstreams. A “Good” bitstream should accurately represent a given probability number and also have low correlation with other bitstreams. In this section, we introduce an SBG utilizing stochastic switching behavior of MTJ and then exhibit the simulation results.

A. Schematic of SBG

In the proposed system, every bitstream is constructed based on the state of MTJ. If MTJ is with high resistance *i.e.* ‘AP’ state, ‘0’ will be added to the bitstream; otherwise, ‘1’ will be added. Generally, the state of MTJs could be easily detected by CMOS sense amplifiers.

The circuit diagram of proposed SBG is illustrated as Fig. 4 which is composed by CMOS transistors and MTJs. Both write and read operations could be achieved with this circuit. Bit-line (BL) and source-line (SL) are driven by two different voltage sources. MUX₂ and MUX₃ are used to control either read current or write current would go through the MTJ. During the write operation, signal ‘Write En.’ is at high level, thus terminal ‘1’ of MUX₂ and MUX₃ are ON. The write operation consists of two phases: resetting MTJ state to ‘AP’ state and switching the MTJ state from ‘AP’ to ‘P’. In the first phase, terminal ‘0’ of MUX₁ and terminal ‘1’ of MUX₄ are ON because signal ‘Wrt. 1’ is at low level and signal ‘Rst. 0’ is at high level. Current flows through the MTJ from bottom to top as the blue arrow shows. In this phase, bias voltage and duration time are set to guarantee that the state of MTJ switches to ‘AP’ state at 100% probability. In the second phase of write operation, terminal ‘1’ of MUX₁ and terminal ‘0’ of MUX₁ are ON because signal ‘Wrt. 1’ is at high level and signal ‘Rst. 0’ is at low level. Current flows through the MTJ from top to bottom as the red arrow shows. In this phase, bias voltage and duration time are set based on the wanted probability value. During the read operation, terminal ‘0’ of MUX₂ and MUX₃ and transistor N₁ are ON. A

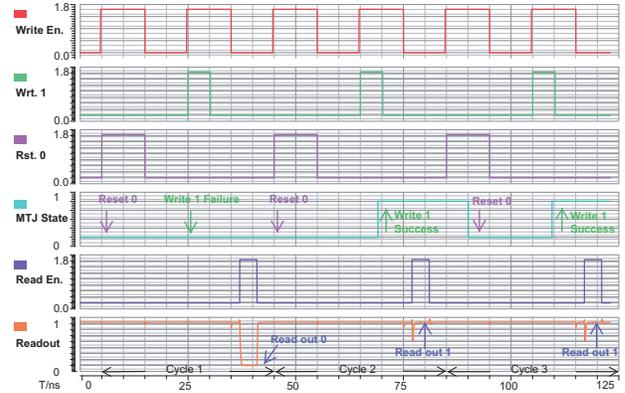


Fig. 5: Simulation results of SBG circuit.

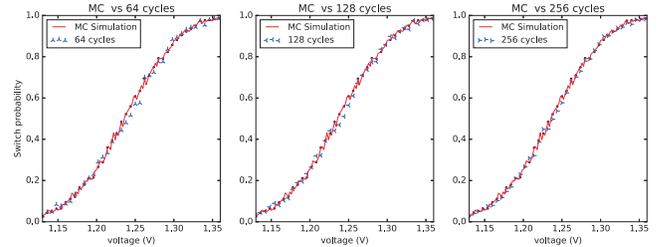


Fig. 6: Applied voltage v.s. switching probability.

Pre-Charge Sense Amplifier (PCSA) [18] is used to read the state of MTJ.

Three-cycle Cadence simulated waveform is illustrated in Fig. 5. Each cycle consists of three operations of resetting 0, writing 1 and reading MTJ state. In each cycle, the MTJ state is first reset to be ‘AP’ state during which ‘Write En.’ and ‘Rst. 1’ are high and ‘Wrt. 1’ is low. Current goes through the MTJ from bottom to up as the blue arrow shows in Fig. 4. Then comes writing 1 stage during which ‘Rst. 0’ is low and ‘Wrt. 1’ is high. Current goes through the MTJ from up to bottom as the red arrow shows in Fig. 4. In this stage, the state of MTJ may or may not switch from ‘AP’ to ‘P’. Then comes the reading stage during which ‘Read En.’ becomes high and ‘Write En.’ becomes low. In this stage, the state of MTJ is read out by PCSA circuit as the last wave shows. In the given example, writing 1 operation fails in the first cycle and successes in the following two cycles. Thus, bitstream is generated as ‘011’.

B. Probability-Voltage relationship based on MC simulation

SBG is used to generate SBs to represent probability number. Different bias voltages correspond to different probability values. In this section, the Probability-Voltage relationship of proposed SBG is analyzed using Monte-Carlo simulation strategy. The simulation is processed by Cadence Virtuoso with 45 nm CMOS and 40 nm MTJ technologies. In the simulation, a behavioral model of MTJ considering the stochastic switch feature is described by Verilog-A language [19]. The write duration time is set to be 5 ns because the relationship of voltage and probability is closed to linear under this setting. The reset duration time is set to be 10 ns in order to guarantee a 100% reset switching. For each bias voltage

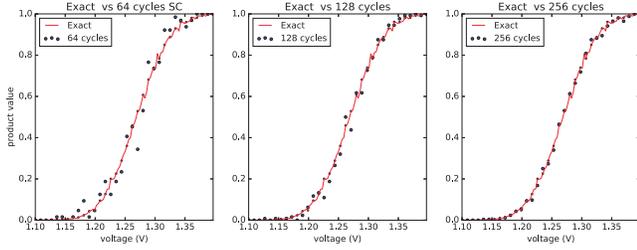


Fig. 7: Comparison between arithmetic computing and stochastic computing.

ranging from 1.13 V to 1.36 V, 1000 Monte-Carlo simulations are performed. The simulated P-V relationship is illustrated in Fig. 6 by the red line. From the figure we can find that as the increasing of voltage, the switching probability also increases monotonously. It means that voltages and probability values are almost corresponding one by one.

C. Evaluation

Two evaluation experiment results are presented in this section which prove that the stochastic bitstreams generated by the proposed SBG are high accuracy and low correlation.

Firstly, bitstreams are generated with length of 64, 128 and 256. As shown in Fig. 6, results of all the three classes bitstreams are well coincident with Monte-Carlo simulation results. Compared with Monte-Carlo simulation results, the average errors are only 1.6%, 1.3% and 1.1% for length of 64, 128 and 256, respectively. It is obvious that the longer the bitstream, the smaller the error. As described above, “good” bitstream requires low correlation with other bitstreams. In the Verilog-A model, an effective seed generation strategy is integrated into MTJ model. The strategy could guarantee that different MTJs use different seeds. Because the seeds are independent of each other, there is no correlation between any two bitstreams. To verify the random strategy, in the second experiment, a multiplication of two bitstreams driven by the same voltage is executed using AND gate. Both the results of exact computing and stochastic computing with different bitstream lengths are shown in Fig. 7. Statistical results show that the average errors are only about 2.8%, 2.0% and 1.2%, respectively.

So far, an SBG circuit is constructed based on MTJ and its efficiency has been proven by simulation results. It is served as the most important component of the Bayesian inference system proposed in Section V.

V. APPLICATIONS

Different applications may be solved by different Bayesian inference mechanisms. Thus, structures of BIS are also different. In this section, two different types of applications with different inference mechanisms are considered. Using the MTJ based SBG and stochastic computing theory, we build two Bayesian inference systems for the two applications.

A. Data fusion for target location

Data fusion is the process of integrating multiple data sources to produce more consistent, accurate, and useful information than that provided by any individual data source. In this section, a simple data fusion example and corresponding Bayesian inference system are studied.

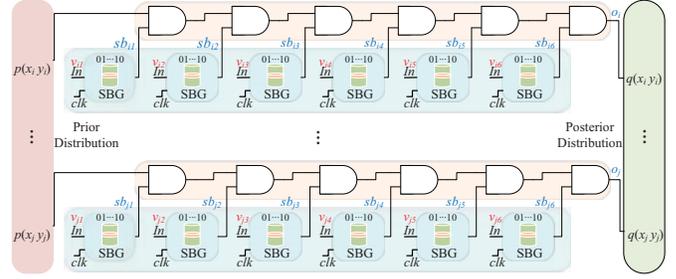


Fig. 8: Bayesian inference system for object location problem.

1) Problem definition and Bayesian inference algorithm:

There are three noisy sensors on the 2D plane and each of them could provide two sensor data independently: Distance (D) and Bearing (B). The problem is to calculate the object location (x^*, y^*) on the plane under the estimated data $(D_1, B_1, D_2, B_2, D_3, B_3)$. The values of the problem parameters are similar to that in [20] as following. Three sensors locate at $(0,0)$, $(0, 32)$ and $(32, 0)$ and the object actual position is $(28,29)$. For each sensor i , Given a position (x, y) , the distance model $p(D_i|x y)$ and bearing model $p(B_i|x y)$ satisfy the following Gaussian distributions:

$$p(D_i|x y) = N(\mu_{di}, \theta_{di}), \quad p(B_i|x y) = N(\mu_{bi}, \theta_{bi}).$$

where, μ_{di} means the Euclidian distance between the sensor i and position (x, y) , $\theta_{di} = 5 + \mu_{di}/10$. And μ_{bi} is the viewing angle of the sensor i and position (x, y) , θ_{bi} is set as 14.0626 degree.

The inference algorithm using sensor data can be expressed as $p(x y|D_1 B_1 D_2 B_2 D_3 B_3)$. Based on Bayes' theory,

$$p(x y|D_1 B_1 D_2 B_2 D_3 B_3) \propto p(x y) * \prod_i p(B_i|x y)p(D_i|x y) \quad (1)$$

In Eqn. (1), $p(x y)$ is known as the prior probability and the following six conditional probabilities are known as evidence or likelihood information. In this problem, the object may locate at any position. The prior probability $p(x y)$ has the same value for every position. So $p(x y)$ is ignored in the following Bayesian inference system.

2) *Bayesian inference system:* It can be seen from Bayesian inference mechanism (Eqn. (1)) that the distribution of object location is calculated by the product of a series of conditional probabilities. In stochastic computing, this is processed using AND gates. In addition, we could find that the calculation of probability value that the object locates at one position is independent for each other. Based on the analysis, the Bayesian inference architecture is illustrated in Fig. 8 as a matrix structure for this application. For each position, 6 SBGs are deployed to yield stochastic bitstreams and 5 AND gates are deployed to achieve multiplication. Thus, for a 64×64 grid, 24576 SBGs and 20480 AND gates are needed. In Fig. 8, the output of each row is the posterior probability value that the object locates at this position. In our simulation, 64×64 counters are used to decode the outputs from stochastic bitstreams to float-point numbers by calculating the proportion of ‘1’. The proposed system makes the best use of high parallel attribute of Bayesian inference and stochastic computing. Utilizing the independent

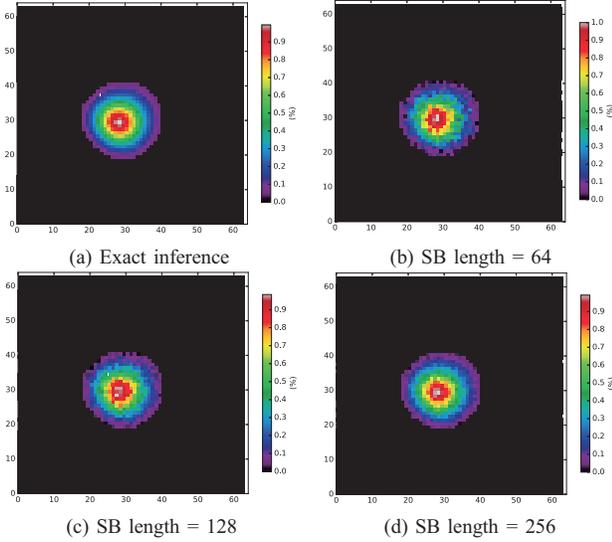


Fig. 9: Data fusion result of target location problem on 64×64 grid. (a) Exact inference results. (b)-(d) Stochastic computing results with length of 64, 128, 256.

TABLE I: KL divergence analysis of target location problem

Grid Size	Bitstream Length		
	64	128	256
64×64	0.0090	0.0043	0.0018
32×32	0.0086	0.0041	0.0019
16×16	0.0080	0.0035	0.0011

of inference algorithm (*i.e.* Eqn. (1)), all rows of the system could perform stochastic computing at the same time. In each row, all the SBGs could yield bitstreams in parallel and the “And” operations are also implemented concurrently during reading the MTJ state.

3) *Simulation Results:* Cadence Virtuoso is used to analyze the accuracy and efficiency of the proposed BIS. In the simulation, 64×64 , 32×32 and 16×16 grids are utilized to test our Bayesian inference system. The finer the grid, the more accurate the target position. For every grid scale, stochastic bitstreams with length of 64, 128 and 256 are generated to perform stochastic computing. The longer the stochastic bitstream, the higher the stochastic calculation accuracy. In Fig. 9, four object location inferred results are shown by heat map on 64×64 grid. Fig. 9(a) is the exact inference result using arithmetic computing in float-point arithmetic computer. Fig. 9(b), 9(c) and 9(d) are the inference results by the proposed Bayesian inference system with stochastic bitstreams length of 64, 128 and 256, respectively. The simulation results indicate that the proposed system could achieve the Bayesian inference results correctly. Compared with exact inference results, the longer the stochastic bitstream, the smaller the error. To quantify the precision of the inference system, the Kullback-Leibler divergence (KL divergence) between stochastic inference distribution and the exact reference distribution is calculated. As shown in Table I, the first column shows the grid scale. The following 3 columns are the KL divergence value for different bitstream lengths. Taking 32×32 grid for example, 10^{-3} KL divergence requires

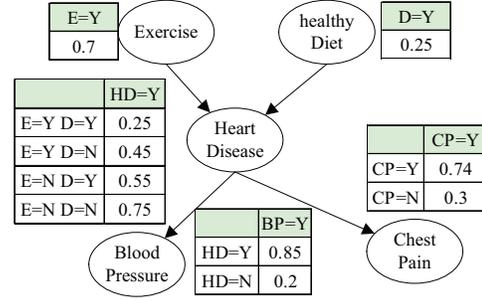


Fig. 10: Bayesian belief network for heart disaster.

length of 256. But for the same precision, the work in [20] requires length of 10^5 . The outstanding results benefit from the high accuracy and low correlation bitstreams generated by the MTJ based SBG. As reported in [20], for a problem with 32×32 grid, the software version on a typical laptop takes 919 mJ, and the FPGA based Bayesian machine only takes 0.23 mJ with stochastic bitstream length of 1000. Benefiting from the low power consumption of MTJs and high quality of SBG, the proposed Bayesian inference system only spends less than 0.01 mJ to achieve the same accuracy with the 32×32 grid. Speed of the proposed Bayesian inference system depends on the bitstream length. Because of the high parallel, the whole inference process only takes $40T$ ns, where ‘T’ means the bitstream length.

B. Bayesian Belief Network

Bayesian belief network is a probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph. In this section, a Bayesian belief network for heart disaster is studied.

1) *Problem definition and Bayesian inference algorithm:* Fig. 10 is a Bayesian belief network (BBN) example for heart disaster prediction. In this network, the parent nodes of heart disaster (HD) are factors that cause heart disaster, including exercise (E) and diet (D). The child nodes are clinical manifestations of HD, including blood pressure (BP) and chest pain (CP). In addition to the graph structure, Conditional probability tables (CPT) are also given. For example, the second value 0.45 in the CPT of node HD means that if a person takes regular exercise but unhealthy diet, the risk of HD is 0.45. In this problem, we pay more attention to inference based on given evidences. The inference mechanism could be classed as two groups based on the junction tree algorithm. The first case is considering E, D and HD as a group and calculating $p(HD)$ as Eqn. (2):

$$\begin{aligned}
 p(HD = Y) = & [p(HD|E = YD = Y)p(D = Y) + \\
 & p(HD|E = YD = N)p(D = N)]p(E = Y) + \\
 & [p(HD|E = ND = Y)p(D = Y) + \\
 & p(HD|E = ND = N)p(D = N)]p(E = N)
 \end{aligned} \quad (2)$$

Here, Y means yes and N means No. If exercise or diet is determined, $p(E)$ or $p(D)$ in Eqn. (2) is 1, otherwise, is the value in CPT. The second case is considering HD, HB and CP as a group and calculating $p(HD|HB\ CP)$ as Eqn. (3):

$$p(HD = Y|HB\ CP) = \frac{p(HB|HD = Y)p(CP|HD = Y)p(HD = Y)}{p(HB\ CP)} \quad (3)$$

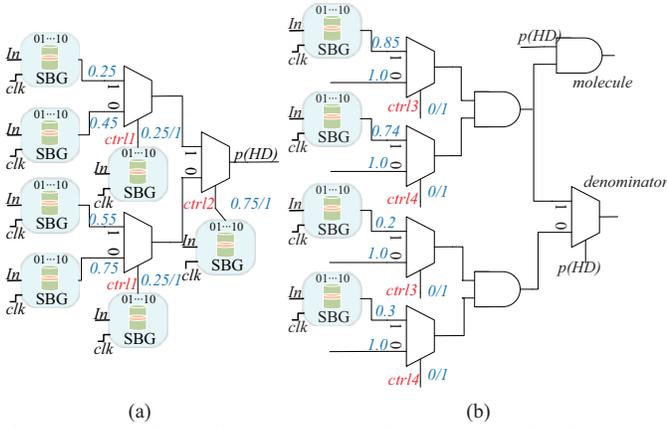


Fig. 11: Bayesian Inference system for BBN. (a) Implementation of Eqn. (2). (b) Implementation of Eqn. (3).

TABLE II: Example of BBN inference setting

Probability	($ctr11$, $ctr12$, $ctr13$, $ctr14$)	[21]	SC
$p(HD BP)$	(0.25, 0.75, 1.00, 0.00)	0.803	0.805
$p(HD D, E, BP)$	(1.00, 1.00, 1.00, 0.00)	0.586	0.592
$p(HD E, BP)$	(0.25, 1.00, 1.00, 0.00)	0.687	0.694
$p(HD D, E, BP, CP)$	(1.00, 1.00, 1.00, 1.00)	0.777	0.742
$p(HD CP)$	(0.25, 0.75, 0.00, 1.00)	0.703	0.700

The denominator of Eqn. (3) can be calculated as Eqn. (4):

$$\frac{p(HB|HD = Y)p(CP|HD = Y)P(HD = Y) + p(HB|HD = N)p(CP|HD = N)P(HD = N)}{p(HB|HD = Y)P(HD = Y) + p(HB|HD = N)P(HD = N)} \quad (4)$$

Here, $p(HD = Y)$ is calculated by Eqn. (2). If HB or CP is diagnosed, the conditional probability value in Eqn. (3) is the value in CPT, otherwise is 1.

2) *Bayesian inference system*: Based on the inference algorithm, the inference system could be easily constructed. Eqn. (2) could be calculated by three MUXes as shown in Fig. 11(a). Eqn. (3) could be calculated by three AND gates and five MUXes as shown in Fig. 11(b). Based on the evidence, the Bayesian inference is performed by different combination of MUX control signal.

3) *Simulation Results*: The simulation of Bayesian inference system for BBN is also used Cadence Virtuoso and the simulation results are shown in Table II. The first column of the table lists some the possible posterior probability. The second columns gives the corresponding settings of control signal for each MUX. Column 3 shows the exact results calculated by [21]. Column 4 is the results calculated by the proposed bayesian inference system using stochastic computing. The comparison between column 6 and column 7 indicates that the proposed Bayesian inference system for BBN could achieve reasonable results.

VI. CONCLUSION

In this paper, a stochastic bitstream generator based on MTJ is proposed firstly. Simulation results shows that the proposed SBG could yield “good” stochastic bitstreams. Not only can the probability values be accurately expressed, but also the correlations between each other are low. Based on MTJ based SBG and stochastic computing theory, two Bayesian inference systems for different applications are proposed. Simulation results indicate that both the two systems could achieve high

inference accuracy with fast running speed and low power consumption. The future work will be carried on from two aspects. The first one is further improving the performance of SBG in terms of accuracy, speed and power in order to build more efficient Bayesian inference system. The second one is improving scalability to larger problems and widening extent of application.

REFERENCES

- [1] P. Pinheiro and P. Lima, “Bayesian sensor fusion for cooperative object localization and world modeling,” in *CIAS*. Citeseer, 2004.
- [2] N. Cruz-Ramírez, H. G. Acosta-Mesa, H. Carrillo-Calvet, L. A. Nava-Fernández, and R. E. Barrientos-Martínez, “Diagnosis of breast cancer using bayesian networks: A case study,” *Computers in Biology and Medicine*, vol. 37, no. 11, pp. 1553–1564, 2007.
- [3] Y. Gal, R. Islam, and Z. Ghahramani, “Deep bayesian active learning with image data,” *arXiv preprint arXiv:1703.02910*, 2017.
- [4] C. S. Thakur, S. Afshar, R. M. Wang, T. J. Hamilton, J. Tapson, and A. Van Schaik, “Bayesian estimation and inference using stochastic electronics,” *Frontiers in neuroscience*, vol. 10, 2016.
- [5] A. Alaghi and J. P. Hayes, “Survey of stochastic computing,” *TECS*, vol. 12, no. 2s, p. 92, 2013.
- [6] A. F. Vincent, N. Locatelli, J.-O. Klein, W. S. Zhao, S. Galdin-Retailleau, and D. Querlioz, “Analytical macrospin modeling of the stochastic switching time of spin-transfer torque devices,” *IEEE Transactions on Electron Devices*, vol. 62, no. 1, pp. 164–170, 2015.
- [7] J. Grollier, D. Querlioz, and M. D. Stiles, “Spintronic nanodevices for bioinspired computing,” *Proceedings of the IEEE*, vol. 104, no. 10, pp. 2024–2039, 2016.
- [8] M. Lin, I. Lebedev, and J. Wawrzyniak, “High-throughput bayesian computing machine with reconfigurable hardware,” in *FPGA*. ACM, 2010, pp. 73–82.
- [9] P. Mrosczyk and P. Dudek, “The accuracy and scalability of continuous-time bayesian inference in analogue cmos circuits,” in *ISCAS*. IEEE, 2014, pp. 1576–1579.
- [10] L. A. de Barros Naviner, H. Cai, Y. Wang, W. Zhao, and A. B. Dhia, “Stochastic computation with spin torque transfer magnetic tunnel junction,” in *NEWCAS*. IEEE, 2015, pp. 1–4.
- [11] Y. Wang, H. Cai, L. A. Naviner, J.-O. Klein, J. Yang, and W. Zhao, “A novel circuit design of true random number generator using magnetic tunnel junction,” in *NANOARCH*. IEEE, 2016, pp. 123–128.
- [12] S. Wang, S. Pal, T. Li, A. Pan, C. Grezes, P. Khalili-Amiri, K. L. Wang, and P. Gupta, “Hybrid vc-mtj/cmos non-volatile stochastic logic for efficient computing,” in *DATE*. IEEE, 2017, pp. 1438–1443.
- [13] J. Von Neumann, “Probabilistic logics and the synthesis of reliable organisms from unreliable components,” *Automata studies*, vol. 34, pp. 43–98, 1956.
- [14] P. Jeavons, D. A. Cohen, and J. Shawe-Taylor, “Generating binary sequences for stochastic computing,” *IEEE Transactions on Information Theory*, vol. 40, no. 3, pp. 716–720, 1994.
- [15] P. K. Gupta and R. Kumaresan, “Binary multiplication with pn sequences,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 603–606, 1988.
- [16] T. Devolder, J. Hayakawa, K. Ito, H. Takahashi, S. Ikeda, P. Crozat, N. Zerounian, J.-V. Kim, C. Chappert, and H. Ohno, “Single-shot time-resolved measurements of nanosecond-scale spin-transfer induced switching: Stochastic versus deterministic aspects,” *Physical review letters*, vol. 100, no. 5, p. 057206, 2008.
- [17] M. Marins de Castro, R. Sousa, S. Bandiera *et al.*, “Precessional spin-transfer switching in a magnetic tunnel junction with a synthetic antiferromagnetic perpendicular polarizer,” *Journal of Applied Physics*, vol. 111, no. 7, p. 07C912, 2012.
- [18] W. Zhao, C. Chappert, V. Javerliac, and J.-P. Noziere, “High speed, high stability and low power sensing amplifier for mtj/cmos hybrid logic circuits,” *IEEE Transactions on Magnetics*, vol. 45, no. 10, pp. 3784–3787, 2009.
- [19] Y. Wang, Y. Zhang, E. Deng, J.-O. Klein, L. A. Naviner, and W. Zhao, “Compact model of magnetic tunnel junction with stochastic spin transfer torque switching for reliability analyses,” *Microelectronics Reliability*, vol. 54, no. 9, pp. 1774–1778, 2014.
- [20] A. Coninx, P. Bessière, E. Mazer, J. Droulez, R. Laurent, M. A. Aslam, and J. Lobo, “Bayesian sensor fusion with fast and low power stochastic circuits,” in *ICRC*. IEEE, 2016, pp. 1–8.
- [21] (2017) Pythonic bayesian belief network framework. <https://github.com/eBay/bayesian-belief-networks>.