Magneto-electric Approximate Computational Circuits for Bayesian Inference

Sourabh Kulkarni, Sachin Bhat, Santosh Khasanvis*, Csaba Andras Moritz
Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA
*BlueRISC, Amherst, MA, USA
skulkarni@umass.edu, andras@ecs.umass.edu

***Abstract*— Probabilistic graphical models like Bayesian Networks (BNs) are powerful cognitive-computing formalisms, with many similarities to human cognition. These models have a multitude of real-world applications. New emerging-technology based circuit paradigms leveraging physical equivalence e.g., operating directly on probabilities vs. introducing layers of abstraction, have shown promise in raising the performance and overall efficiency of BNs, enabling networks with millions of random variables. While previous BNs of up to 100s of nodes have been shown to require single-digit precision without affecting application outcomes, the significantly larger number of variables requires the computational precision to be scaled to correctly support BN operations. We introduce a new computational circuit fabric based on mixed-signal magneto-electric computations operating with physical equivalence and supporting probabilistic computations with a new approximate circuit style. Precision scaling impacts area at a logarithmic vs. linear scale offering a much lower power and performance cost than in prior directions. Results show 30x area reduction for a 0.001 precision vs. prior direction, while maintaining three orders of magnitude benefits vs. 100-core processor implementations.**

Keywords*— *Bayesian Networks, Nanoscale Fabrics, Approximate Computing, Probabilistic Computing, Magneto-Electric Circuits.

I. INTRODUCTION

BAYESIAN Networks (BNs) [1][2] are a class of probabilistic graphical models enabling reasoning under uncertainty. Several studies in neuroscience[3]-[6] suggest that cognition and higher order reasoning in the human brain may closely resemble Bayesian inference. The Bayesian Model has been successful in explaining several of the brain's abilities as well as shortcomings[3][5]. The use of BNs has also proven effective in many important real-world applications [7]-[11] e.g., gene expression, medical diagnosis, text-classification, troubleshooting, macro-finance, etc. BN implementations in conventional processor architectures are limited by several major issues: i) software solutions involve multiple layers of abstraction to support a non-deterministic framework like BNs; ii) the von Neumann processor architecture inherently separates memory and computation introducing bottlenecks in accessing data; and (iii) non-volatility requirements of cognitive applications are challenging to fulfill efficiently.

Emerging technology[12] based implementations[13][14] show great promise to achieve unique benefits that enable large BNs with potentially thousands to millions of nodes, with orders of magnitude efficiency improvements vs. state-of-the-art. For example, by utilizing a new style of mixed-signal magneto-electric computation based on physical equivalence at the level of physical signals, three orders of magnitude efficiency improvement is projected [14]. Although these new fabric architectures could enable large BNs and therefore many new applications, they do not scale well to higher resolutions. This is because, unlike the radix representation used in conventional digital designs, their flat linear representation (where a single probability value requires multiple physical signals) increases area linearly. For example, adding a digit of precision, i.e., increasing precision by tenfold, would increase area similarly by tenfold. This scaling is prohibitive for very large-scale BNs where precision would need to increase by several digits to support BN inference.

In this paper, we propose a new magneto-electric circuit paradigm for Bayesian applications, which keeps physical equivalence while providing a new direction on scaling computational resolution. We introduce a new hybrid way of representing probabilities and an associated approximate magneto-electric circuit style inspired by approximate computations[15]. The error resilience of BNs toward the arithmetic computations performed in the Bayesian nodes [16] enable the use of approximate computation circuits, which significantly reduce the area and power requirements. Results show that the benefits of achieving higher resolution computation far exceed the loss in accuracy incurred due to the use of approximate techniques. The loss in accuracy is also less of a factor in the quality of the Bayesian inference [16][17], since the increased precision achieved by scaling enables the implementation of much larger BNs incorporating a larger number of random variables which are known to be the primary factors of determining accuracy at the application level.

The paper is structured as follows – In section II, we will briefly discuss the Bayesian formalism and previous magneto-electric frameworks implementing BNs. In Section III we discuss the core ideas of the new fabric, and detail the implementation, which supports scalable precision. In section IV we evaluate the benefits of the new circuit framework over the previous magneto-electric fabric designs. Section V concludes this paper.

II. BACKGROUND

BNs are probabilistic graphical models[1][2], which capture the domain knowledge in a graphical structure (Figure 1). The knowledge of the qualitative relations is encoded as probabilities, which enables these models to provide reasoning under uncertainty. BNs are typically structured as Directed Acyclic Graphs (DAGs), with the nodes representing knowledge about variables in the system. The edges of the graph are directed links, which represent the dependencies. A directed link from one node to another makes the former node the parent of the latter. Each node has several variables and each variable can have multiple states. For every node, the strength of the dependency of the child node on its parents is encoded in the conditional probability table (CPT). Performing inference operations on a BN involves several arithmetic operations in each node of the BN operating on probabilities. It also involves frequent lookups of the CPT probability values for each node. These factors along with the distributed nature of the BNs make it inefficient to perform these computations on traditional computing platforms.

In [13][14], a magneto-electric circuit based fabric was proposed, which performed Bayesian inference with area, performance and power-efficiency many orders of magnitude greater than inference on state of the art 100 core processors. However, the information representation and circuit style adopted by the fabric had limitations in scaling precision that is essential on large BNs, such as in life-science applications benefitting from a representation involving thousands to millions of random variables. This paper introduces a new magneto-electric circuit framework, which maintains the fabric efficiency at lower resolutions while providing an efficient magneto-electric probabilistic framework for applications with potentially millions of nodes requiring higher computational resolution. This direction is partially based on approximate computations with probabilities and achieves exponential scaling of computational resolutions with only a linear increase in area.

III. CORE FRAMEWORK CONCEPTS

The framework builds on the concept of physical equivalence – the probabilities in the BN are encoded in an information representation scheme, which is implemented in a mixed signal magneto-electric circuit style. Efficient scalability of computational resolution is achieved by selective use of approximate computation. The computation circuits are designed such that the inaccuracies incurred by using approximate computing have a much lower impact than the increase in accuracy due to performing computation at a higher resolution.

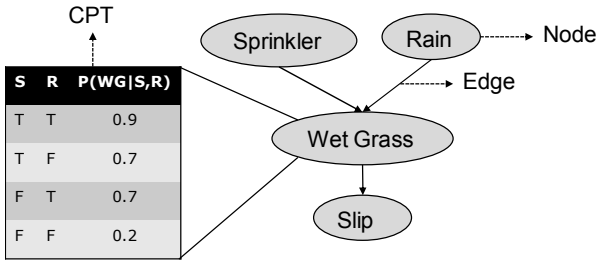

Figure 1. An intuitive example of a BN. The network encodes the causal dependence of someone slipping on the grass, the grass being wet, the sprinkler being on, and on whether it rains.

A. Information representation

Conventionally, information in digital circuits has been represented in a radix format. This style performs well in all digital circuits, and provides for an efficient way to scale precision. However, when we consider radix representation for mixed-signal architectures, which employ analog domain to perform computations, and rely on devices that are stochastic switches with much lower reliability than MOS counterparts, the lack of error resilience is prohibitive. This is since any single bit error would make an entire computation potentially useless.

An alternative approach, considered in [13][14], involves a flat information representation scheme, which is more suitable for mixed signal architectures. This approach assigns the same 'weight' to each digit in the representation and the value represented is simply the sum of the values in the individual digits. Although this provides error resilience and can perform efficient analog computations, scaling to higher computational resolutions is very inefficient.

We propose a more generalized information representation scheme, which combines the scalability of the radix representation and the error resilience of the flat representation. This new 'Flat-Radix' representation consists of M segments in a radix arrangement with n elements each in a flat arrangement as shown in Figure 2(a). Within a single segment, all the flat elements f_i contribute equally toward the value of the segment, making it error resilient. A segment S_J represents a value equal to the sum of all the flat elements f_i in the segment. Because of the radix nature of the segments, a segment S_J has a value n times the value of segment S_{J+1}, providing efficient scalability. The probability is represented by the segments arranged in a radix format with base n. This new representation opens a spectrum of possible representation states with flat representation at one end (M=1) and radix representation at the other (n=1). This vast range of possible configurations is attractive as the representation scheme could be potentially tailored to match the application precision requirement.

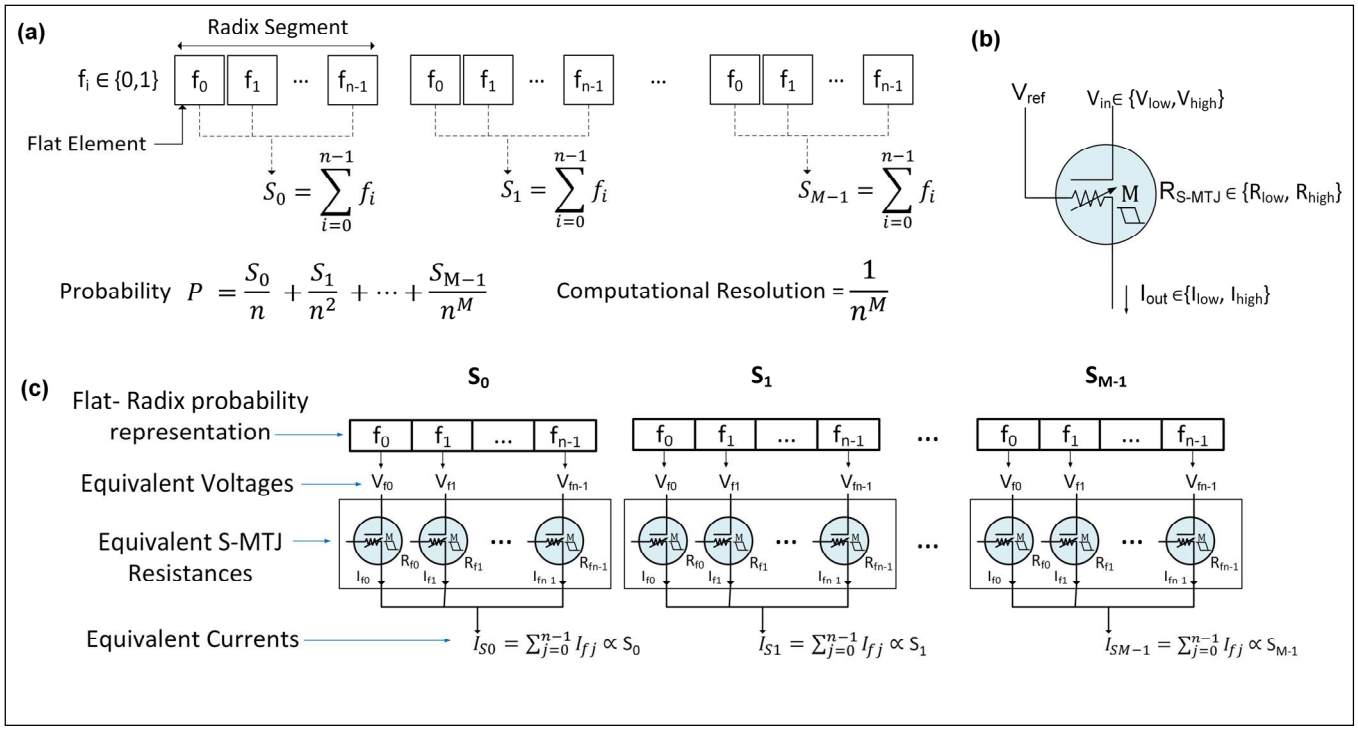

Figure 2. (a) Flat-Radix information representation scheme. Probability is encoded in segments which are in a radix arrangement, with each segment containing flat elements; (b) Schematic view of the 2-state S-MTJ device [12][13]; (c) Physically equivalent encoding of probability using the proposed information representation scheme into S-MTJ circuits.

Consider an example with a 10-2 configuration of the scheme; i.e. with n=10 and M=2. This corresponds to a representation with two segments in a radix representation with each containing 10 flat elements. The effective computational resolution of this configuration is 0.01. Now if we change M to 3, the configuration now has an effective resolution of 0.001. This corresponds to an exponential increase in computational resolution with linear increase in number of devices. It is important to note however, that some of the computations performed are approximate in nature and the increase in computational resolution doesn't directly translate to increased accuracy in computation.

The probabilities in BNs encoded using the proposed scheme are represented physically in a persistent manner using non-volatile 2-state S-MTJ devices detailed in [12][13]. Briefly, S-MTJs are magneto-electric devices, which have variable resistance states based on the relative spin-polarization of two magnetic layers (free and fixed). The magnetization of the free layer can be changed by a strain induced from an external voltage. Although this device is known to have switching error greater than CMOS transistors, its stochastic behavior makes it ideal to encode probabilities, as computations on probabilities are usually error resilient. This device is schematically shown in Figure 2(b). A constant reference voltage V_{ref} is applied at one terminal and current I_{out} is read-out from the other terminal. The voltage V_{in} changes the resistance value of the device $R_{S\text{-}MTJ}$, which is reflected in the change in output current I_{out}.

Figure 2(c) shows the physically equivalent circuit representation of probabilities encoded in the proposed information representation scheme. This paper discusses a spatial representation scheme, but a temporal scheme is also possible. In each segment, the flat elements f_i are represented by S-MTJ device resistance states R_{fi} and can be changed by input voltages V_{fi}. The outputs of all devices in a segment are summed together. The summed currents I_{SJ} now represent segments S_J. The total probability P is interpreted by considering the segments in a radix arrangement. The base of this radix structure is based on the number of flat elements in each segment.

Although the representation has radix segments, the computations remain atomic to the flat portions, as there is no notion of radices when computing in analog domain. Using approximate computing when required, we design circuits that bring together the flat computations into a radix representation. The approximate computing techniques (discussed in further subsections) have a lower error bound; the inaccuracies in computation can never be greater than the computations in a representation with a single radix segment (i.e., M=1). This case occurs when all the radix segments except the first one have value of 0 and essentially get excluded from the computation.

The scheme shown in Figure 2 assumes devices with two stable states. However, several emerging devices like the magnetic domain-wall devices [18][19], show potential to support multiple stable states. For such devices, the flat elements in a radix segment could collapse, yielding in a radix

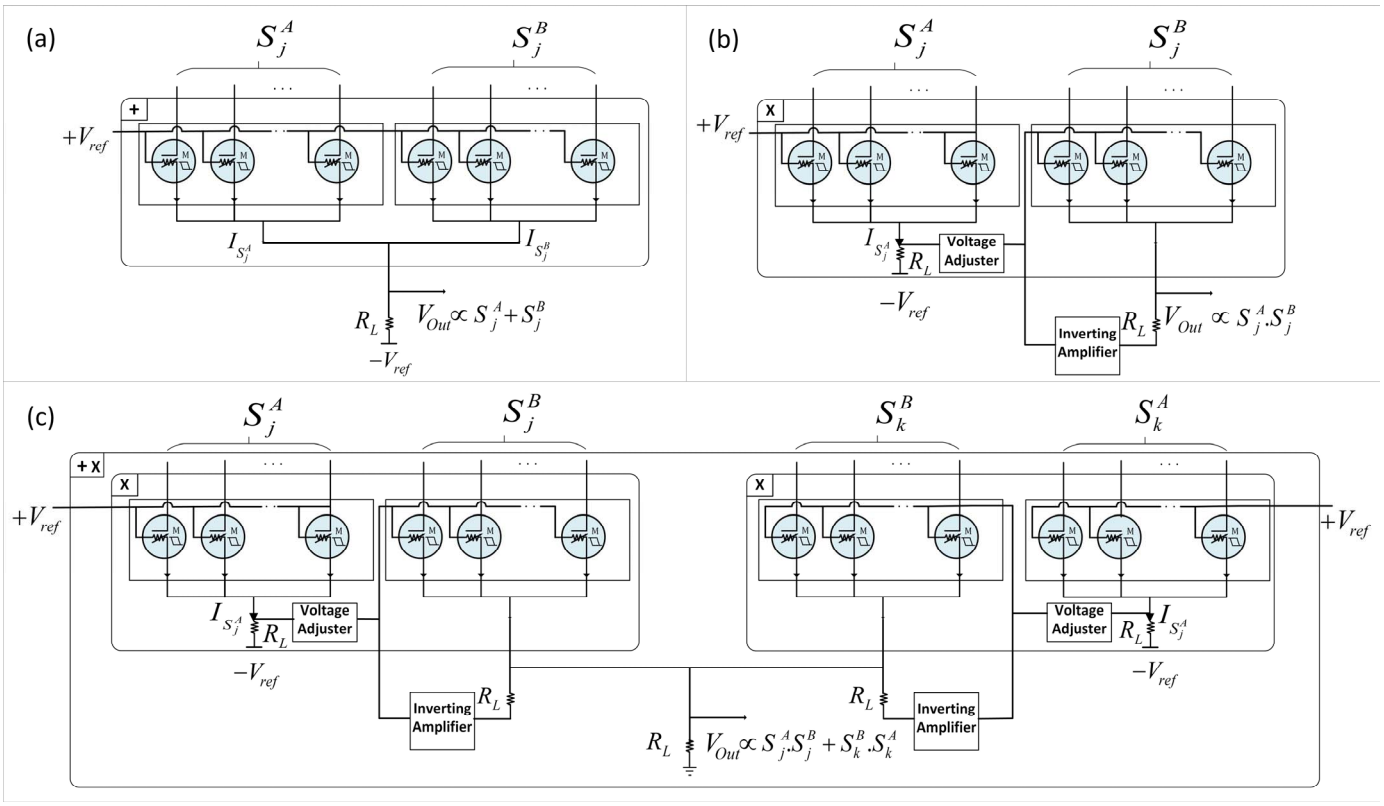

Figure 3. Segment-level Composers; (a) Addition SLC, current summation is used to perform addition; (b) Multiplication SLC, the current of one multiplicand is used to modulate the reference voltage of the other multiplicand, resulting in a multiplication operation; and (c) Add-Multiply SLC, a hierarchical combination of the addition and multiplication SLCs which implements a sum-of-products operation.

only representation. Furthermore, memristive devices shown to have continuous persistence states, could potentially collapse the proposed scheme into a single device.

B. Segment-level circuit framework

The representation scheme discussed above provides a physically equivalent way to encode probabilities and in the process, enables unique ways to perform computation. In this architecture, the computation operations like addition and multiplication are consequences of circuit design, in contrast to traditional computation schemes in which there are complex arithmetic units dedicated to perform these computations. The computation circuits start from simple adder and multiplier circuits operating on individual flat segments and a hierarchical encapsulation of these circuits performs higher order functionality, i.e., the flat-radix adders and multipliers.

The computational elements operating on flat segments are primarily composed of two components – i) a composer which converts input data stored in form of resistances of the S-MTJs in current domain and performs the computation on the current; and ii) a decomposer which converts the computed result back into the format which can be stored into the S-MTJs. The composer circuits are discussed first and the decomposer will be described later.

Figure 3.(a), (b) shows circuit diagram and working of the addition and multiplication composers operating on a segment-level computation. The design of these circuits is based on the arithmetic composer circuits discussed in [13][14]. The addition composer works by summing together the currents from both the flat segments, while the multiplication composer works by the current obtained from one of the segments moderating the V_{ref} of the other segment.

Building on these two composers, we hierarchically construct an add-multiply composer, which combines the operation of both addition and multiplication composer. Figure 3.(c) shows the circuit diagram and working of the add-multiply composer.

C. Flat-Radix Addition Circuit

After describing all the composers operating at the segment level, we shall now discuss the design of the flat-radix composers (FRCs), which utilize the unique structure of the flat-radix scheme to perform the computations. For purpose of clarity we shall from now on refer to the segment-level composers as 'SLCs'. In implementing FRCs, we perform few changes in the design of SLCs; we combine multiple SLCs which perform computation on individual segments and design special circuits to propagate 'carry' signals as shown in Figure 4 from lower order segment to higher order one. The

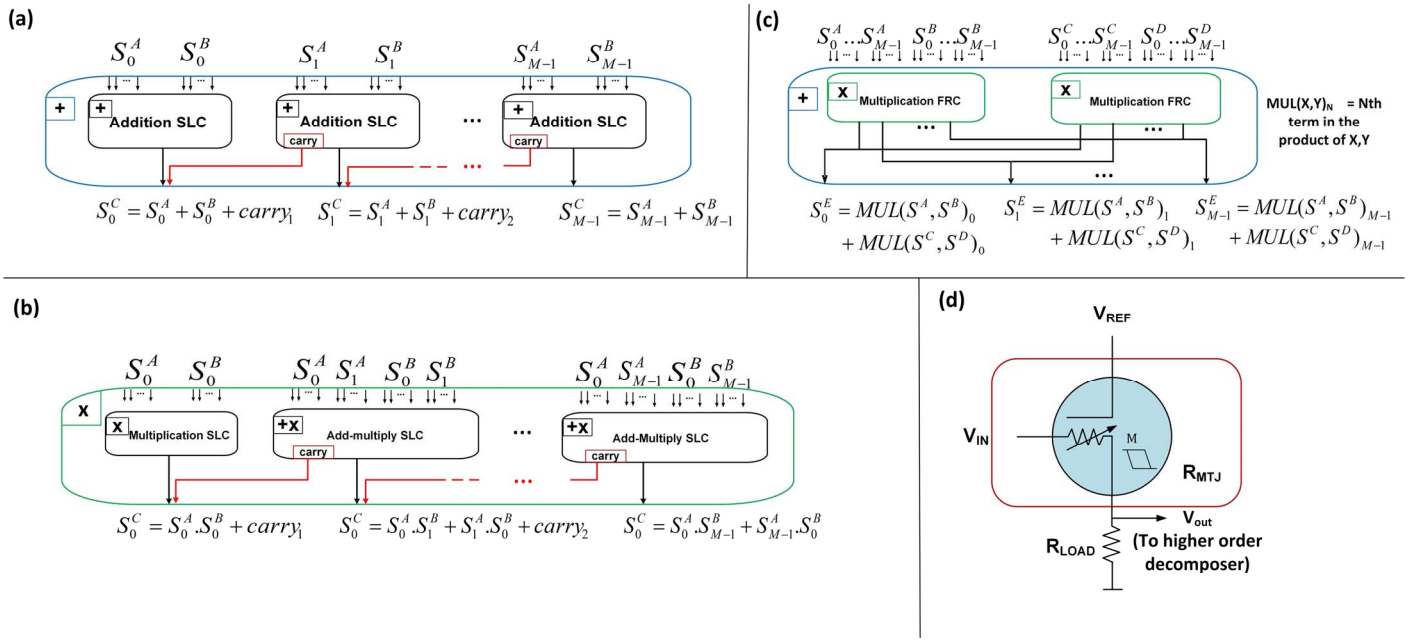

Figure 4. The Flat-Radix Composers; (a) Addition FRC made up of several addition SLCs and carry circuits; (b) Multiplication FRC made up of one multiplication SLC and several Add-Multiply SLCs and carry circuits; (c) Add-Multiply FRC made from a hierarchical combination of Multiplication FRC and Addition FRC; and (d) Carry propagation circuit providing carry support from lower order segment to higher order segment.

carry circuit consists of a S-MTJ based voltage comparator, which detects carry generation in lower order circuit and propagates it to higher order decomposer. Consider two probabilities A and B, which are represented based on the new scheme as follows:

$$A = \left(\frac{1}{n}\right) S_0^A + \left(\frac{1}{n^2}\right) S_1^A + \cdots + \left(\frac{1}{n^M}\right) S_{M-1}^A \quad (1)$$

$$B = \left(\frac{1}{n}\right) S_0^B + \left(\frac{1}{n^2}\right) S_1^B + \cdots + \left(\frac{1}{n^M}\right) S_{M-1}^B \quad (2)$$

The addition operation in this new representation is done using the following formula:

$$A + B = \left(\frac{1}{n}\right)(S_0^A + S_0^B + carry_1) + \left(\frac{1}{n^2}\right)(S_1^A + S_1^B + carry_2) + \cdots + \left(\frac{1}{n^M}\right)(S_{M-1}^A + S_{M-1}^B) \quad (3)$$

In the above formula, each individual addition term is computed in its own segment, while the carry is propagated to the segment one order higher. Figure 4(a) shows the circuit diagram which implements equation (3). This design makes it very efficient to scale to higher resolution just by adding more segments to the hybrid representation. This implementation is an exact expression of addition formula hence the calculations have no approximation involved. This accuracy is achieved because in an addition operation, the causality flows strictly in one direction (lower-order to higher-order) in the form of carry; i.e., a lower order term may affect the result of a higher order calculation (through carry), but a higher order term cannot affect the result of a lower order calculation.

D. *Flat-Radix Approximate Multiplication Circuit*

The Flat-radix computation scheme utilizes approximate computation techniques in the FRC multiplier. Consider the sum-of-segments form of probabilities A and B from equations (1) and (2). A multiplication operation between these would be:

$$A.B = \left(\left(\frac{1}{n}\right) S_0^A + \left(\frac{1}{n^2}\right) S_1^A + \cdots + \left(\frac{1}{n^M}\right) S_{M-1}^A\right).\left(\left(\frac{1}{n}\right) S_0^B + \left(\frac{1}{n^2}\right) S_1^B + \cdots + \left(\frac{1}{n^M}\right) S_{M-1}^B\right) \quad (4)$$

Upon expanding equation (4), we end up with a much higher number of terms; in this case, a multiplication of two probabilities with *m* terms each would result in m^2 terms. The error resilient nature of BNs [17], whose quality of inference is dependent more on the graph structure and random variable selection than the accuracy of the arithmetic operations, enable us to optimize the design of the multiplier circuit. To optimize the circuit, we include only the top *m* contributing terms of the expansion. The effect of performing this optimization on the accuracy of the multiplier circuit is discussed in later sections. Using this optimization process, we design an approximate multiplication formula which maximizes efficiency while attempting to minimize approximation error:

$$A.B = \left(\frac{1}{n}\right)(S_0^A.S_0^B + carry_1) + \left(\frac{1}{n^2}\right)(S_0^A.S_1^B + S_1^A.S_0^B + carry_2) + \cdots + \left(\frac{1}{n^2}\right)(S_0^A.S_{M-1}^B + S_{M-1}^A.S_0^B) \quad (5)$$

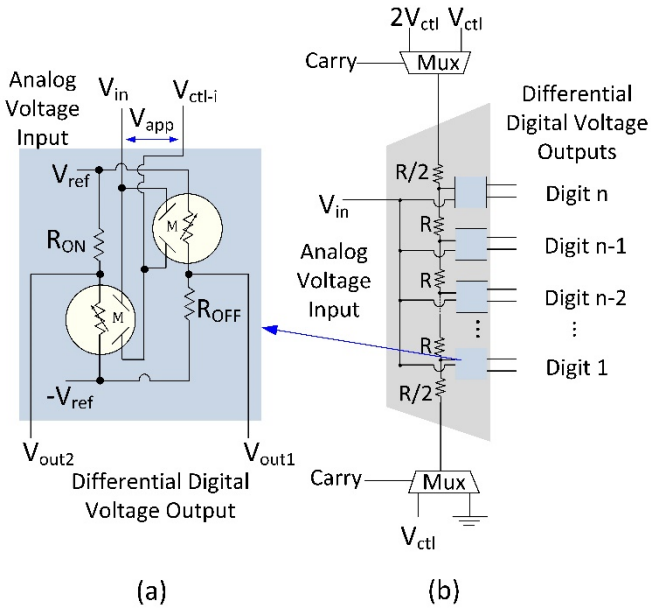

Figure 5. The decomposer circuit. (a) A single comparator element of a segment-level decomposer; and (b) A segment-level decomposer implemented by structuring individual comparators in a R-2R circuit.

Figure 4(b) shows the circuit diagram implementation of equation (5). This formulation is approximate, as it omits the equation terms, which contribute the least to the end-result. These terms are the results of partial multiplications of lower order segments. The omission of these terms is an optimization, resulting from the application of the concept of approximate computing in this case. This approximate multiplier has the worst-case performance when all the segments except the highest order one are zero. In this worst case the FRC performs equally to a SLC. To summarize, the FRCs are cascaded versions of the SLCs, along with extra carry circuitry to propagate the carry through the orders. Figure 4 shows the FRCs as well as the carry circuit. The functionality of the FRCs is verified in HSPICE.

Finally, we discuss the decomposer circuit, which converts the computed results back to the flat-radix format at segment level for further computations. This circuit is designed for devices with two states. The basic design of the segment-level decomposer uses a R-2R ladder circuit commonly used to convert continuous voltages in discrete states. The voltage V_{IN} is compared in several steps to V_{ctl} using MTJ-based comparators shown in Figure 5(a), where each comparator from top to bottom compares V_{IN} with increasingly smaller fractions of V_{ctl}, which is set to be equal to the voltage that corresponds to all the flat elements of a segment being high. Hence, the number of comparators, which output 'high' will be proportional to the ratio of V_{IN} to V_{ctl}. All the segments sans the first can have a maximum value equal to twice the maximum value that can be stored in a single segment. To correctly decompose values greater than V_{ctl} we design a carry-based mechanism to switch the comparator voltage levels as shown in Figure 5(b).

IV. Evaluation

We perform the following evaluations on the proposed architecture. First, we compare the area, power and performance of this architecture with the architecture discussed in [13][14]. Second, we evaluate the error being introduced due to the approximate computing at FRC-level circuits. The benefit of gaining computational resolution over losing accuracy is also discussed.

As discussed in section III (A), the MTJ devices central to this computational framework are known to exhibit certain switching error. The impact of the stochastic nature of the switching of the MTJs on the result of the computations was studied in [13]. It is shown that, due to the error resilient nature of the flat representation scheme, which is utilized by

TABLE I
THE AREA, POWER AND DELAY OF FRCS AND OTHER CIRCUIT COMPONENTS FOR VARIOUS COMPUTATIONAL RESOLUTIONS.

Metrics (Worst case)	Resolution	MUL	ADDMUL	OPAMPs	Decomposers
Area(μm²)	0.1	5	17	95.4	240
	0.01	21	42	190.8	480
	0.001	39	78	286.2	720
	0.0001	56	112	381.6	960
Power(μW)	0.1	1.15	2.81	89.32	11.37
	0.01	3.96	7.92	178.64	22.74
	0.001	6.77	13.54	267.96	34.11
	0.0001	9.58	19.16	357.28	45.48
Delay(ns)	0.1	144	137	100	132.9
	0.01	144	144	100	132.9
	0.001	144	144	100	132.9
	0.0001	144	144	100	132.9

the flat-radix scheme used in this work within each flat segment, the inaccuracies related to errors in the arithmetic operations far exceed the errors due to the switching errors

would have lost most of its performance, power, and area advantages. A further 1,000x improvement (i.e., a 0.000001) in precision would require 6 segments arrangement and would

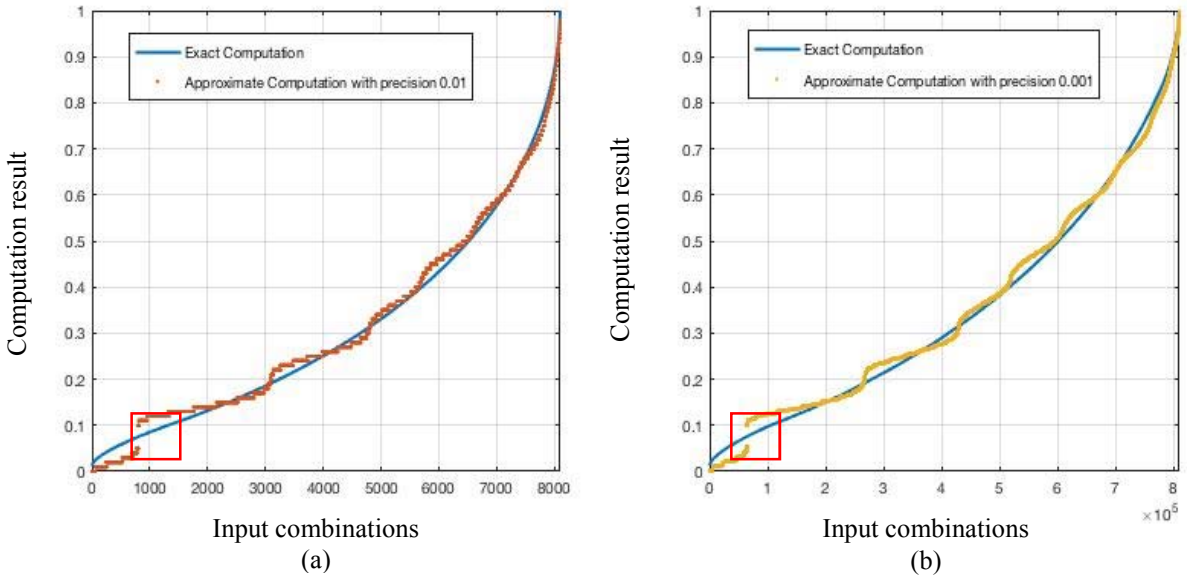

Figure 6. The computational accuracy of Multiplication FRCs with precision 0.01 (10-2 flat-radix with 20 devices per probability) in (a) and 0.001 (10-3 flat-radix with 30 devices per probability) in (b) as compared to the accuracy of 0.01 precision flat-only scheme with 100 devices per probability, and 0.001 flat-only scheme with 1000 devices per probability, respectively. The regions highlighted in red indicate the maximum error of 0.1 due to the approximate nature of the computations. The plot displays the outputs for all possible input combinations sorted in ascending order.

due to the MTJs. In both cases the errors are gracefully tolerated due to the information representation.

A. Area, Power and Performance

Table I shows the area, power and delay values for flat-radix architecture schemes with computational resolutions from 0.1 (1 segment) to 0.0001 (4 segments). The results suggest up to 30x area and power benefits of this architecture over previous approaches[13][14] for a computational resolution of 0.01. The area and power benefits vs. these approaches grow exponentially with resolution.

In [14], the performance of a physically equivalent magneto-electric Bayesian inference system (with a computational resolution of 0.1) was compared with a 100-core processor assuming best-case idealized processor performance. This comparison yielded a 6,000x performance benefit for Bayesian inference over the 100-core processor. For the same resolution, assuming a flat organization with one segment, the delay characteristics of the circuit framework described in this paper matches that of [14], and hence similar performance relative to 100-core processors is expected.

For a precision of 0.001, on the other hand, combining three flat segments into a 10-3 flat-radix magneto-electric framework (10 MTJs for encoding a probability, in each of the three radix segments), this architecture could still maintain a 2,000x performance benefit while the original architecture

maintain a 1,000x improvement. Any single error in a segment would affect that segment by a tenth of a precision. A higher error resiliency may combine 100 MTJs in each segment: e.g., a 100-6 encoding would yield an even more graceful degradation would an error occur in any segment due to say an MTJ not switching correctly, for a 10x additional area impact (and 10x additional precision). The overall computational error would depend on the segment position in a radix, but highest error would occur when highest segment (in radix order) is affected. This calculation does not yet account for any approximate calculations that will be discussed below.

B. Approximation Error in FRCs

We shall now observe the accuracy tradeoffs of the flat-radix approach vs. the flat scaling implemented in [14] by comparing the accuracy performance. These comparisons are done by generating behavioral models of the multiplication composers in Matlab, as exhaustive hardware simulations in HSPICE are infeasible to be done in reasonable time and have convergence issues due to large number of devices and input combinations involved. Figure 6 contains two plots that compare the multiplication operation with scaled flat[13][14] and flat-radix approaches, with 0.01 and 0.001 resolution respectively. The plots were generated by calculating the results for all possible input combinations. From the plots, it is evident that the substantial savings in area and power (5x for 0.01 and 33x for 0.001) are obtained by the flat-radix

TABLE II

PRECISION COMPARISON BETWEEN LOW RESOLUTION ACCURATE COMPUTATION AND HIGHER RESOLUTION APPROXIMATE COMPUTATION.

Computational Resolution	No. of devices	Mean error	Error variance	Max. Error	% Input combinations with max error
0.1(Previous work)	10	0.065	0.006	0.1	10%
0.01	20	0.027	0.000097	0.1	0.01%
0.001	30	0.0037	0.000023	0.1	0.00001%

approach. With a 2x and 3x increase in number of devices, computations at higher resolutions, although approximate, yield lower mean errors (~2.4x and ~17x) and orders of magnitude lower error variance (~61x and ~260x) as shown in Table II. The plot also shows where the new computation system has the highest error of 0.1 (highlighted in red) for precision levels. Although the approximate computation scheme has a worst-case error of 0.1, at higher precision levels, the percentage of input combinations that lead to the maximum error is very low (0.01% and 0.00001%). In the context of Bayesian inference, it has been shown that the quality of inference of a BN depends primarily on the structure of the graph and the number of random variables captured accurately, while the numerical precision required in the arithmetic computations plays a secondary role [17]. Although the computation scheme described in this paper has a maximum error of 0.1, even for higher resolutions, the infrequent occurrence of the high error case is unlikely to affect the outcome of the BN inference; also at the application level accuracy is considered as the likelihood of correct prediction in a belief across a large input measurement set vs. individual inference. On the other hand, the increased precision obtained, along with significantly lower mean error and error variance, will allow for much larger BNs to be implemented in this architecture at a low area cost.

V. CONCLUSION

Probabilistic reasoning frameworks enable many important applications like gene expression, threat detection, text classification and macroeconomics [7]-[11]. As more disciplines of science incorporate probabilistic reasoning into their research process, the list of applications which could benefit from BNs is increasing. The fundamental incompatibility of these probabilistic frameworks with the conventional computing paradigm demands new fabric architecture approaches, which perform probabilistic computations much more efficiently. The magneto-electric circuit framework proposed in this paper performs high resolution probabilistic computations with high efficiency and provides with an easily scalable information representation scheme for analog computations with probabilities. The ability to scale efficiently while maintaining error resiliency will enable accurate representation of very large BNs with sizes up to a million random variables, which could potentially be used in applications like personalized gene-expression networks for cancer treatments [7].

REFERENCES

[1] J. Pearl, Probabilistic reasoning in intelligent systems: Networks of plausible inference, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[2] A. Darwiche, *Modeling and reasoning with Bayesian networks*, Cambridge University Press, 2009.

[3] D. C. Knill and A. Pouget, "The Bayesian brain: The role of uncertainty in neural coding and computation," *Trends Neurosci.*, vol. 27, no. 12, pp. 712–719, 2004.

[4] E. Téglás, E. Vul, V. Girotto, M. Gonzalez, J. B. Tenenbaum, and L. L. Bonatti, "Pure reasoning in 12-month-old infants as probabilistic inference.," *Science (80-.).*, vol. 332, no. 6033, pp. 1054–9, 2011.

[5] D. De Ridder, S. Vanneste, and W. Freeman, "The Bayesian brain: Phantom percepts resolve sensory uncertainty," *Neurosci. Biobehav. Rev.*, vol. 44, pp. 4–15, 2014.

[6] K. Friston, "The free-energy principle: a unified brain theory?," *Nat. Rev. Neurosci.*, vol. 11, no. 2, pp. 127–138, 2010.

[7] C. Su, A. Andrew, M. R. Karagas, and M. E. Borsuk, "Using Bayesian networks to discover relations between genes, environment, and disease," *BioData Mining*, vol.6, no. 6, 2013.

[8] Valdes and K. Skinner, "Adaptive, model-based monitoring for cyber attack detection," in *Recent Advances in Intrusion Detection (RAID)*, pp. 80–92, 2000.

[9] G.J. Brostow, R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.1, pp.594-601, 17-22 June 2006.

[10] P. Lucas, "Bayesian networks in medicine: A model-based approach to medical decision making," *in Proceedings of the EUNITE workshop on intelligent systems in patient care*, 2001.

[11] M. A. Hooker, "Macroeconomic factors and emerging market equity returns: a Bayesian model selection approach" *Emerging Markets Review, Elsevier,* 2004.

[12] A. K. Biswas, S. Bandyopadhyay and J. Atulasimha, "Energy-efficient magnetoelastic non-volatile memory," Appl. Phys. Lett., 104, 232403, 2014

[13] S. Khasanvis, M. Li, M. Rahman, M. Salehi-Fashami, A. K. Biswas, J. Atulasimha, S. Bandyopadhyay, and C. A. Moritz. "Self-similar magneto-electric nanocircuit technology for probabilistic inference engines." *IEEE Transactions on Nanotechnology,* vol. 14, no. 6, pp980-991. 2015.

[14] S. Khasanvis, M. Li, M. Rahman, A. K. Biswas, M. Salehi-Fashami, J. Atulasimha, ... & C. A Moritz, "Architecting for Causal Intelligence at Nanoscale". *IEEE Computer*, vol.48, no.12, pp54-64, 2015.

[15] Han, Jie, and Michael Orshansky. "Approximate computing: An emerging paradigm for energy-efficient design." *2013 18th IEEE European Test Symposium (ETS).* IEEE, 2013.

[16] A. Onisko, and M. J. Druzdzela, "Impact of precision of Bayesian network parameters on accuracy of medical diagnostic systems," *Artificial Intelligence in Medicine, Elsevier*, vol. 57, no. 3, pp. 197 – 206, 2013.

[17] S. Tschiatschek, and F. Pernkopf, "On Bayesian Network classifiers with reduced precision parameters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp.774-785, 2015.

[18] J. A. Currivan, Y. Jang, M. D. Mascaro, M. A. Baldo, and C. A. Ross, "Low energy magnetic domain wall logic in short, narrow, ferromagnetic wires," *IEEE Magn. Lett.*, vol. 3, 2012

[19] M. Sharad, D. Fan, K. Aitken, and K. Roy, "Energy efficient non-Boolean computing with spin neurons and resistive memory," *IEEE Trans. Nanotechnol.*, vol. 13, no. 1, pp. 23–34, Jan. 2014.