AUTONOMOUS PROBABILISTIC HARDWARE FOR UNCONVENTIONAL COMPUTING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Rafatul Faria

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Supriyo Datta, Chair

School of Electrical and Computer Engineering

Dr. Joerg Appenzeller

School of Electrical and Computer Engineering

Dr. Zhihong Chen

School of Electrical and Computer Engineering

Dr. Ernesto E. Marinero

School of Materials Engineering

Approved by:

Dr. Dimitrios Peroulis

Head of the Electrical and Computer Engineering

This thesis is dedicated to my son, Izaan.

ACKNOWLEDGMENTS

First of all, I am grateful to the Almighty for all the opportunities and experiences in my life and my PhD journey will always be one of the most valuable experiences in my life.

The person who guided me with immense patience to grow as a researcher and contribute solid scientific knowledge is my advisor Professor Supriyo Datta. I consider myself fortunate to work with Professor Datta who has an amazing ability to explain difficult concepts in a simple way. I would always remember his advice to express knowledge in a simple logical way without any loose statements. After becoming a mother for the first time at the beginning of my PhD life, when I was struggling a lot to find a balance between work and family, Professor Datta's support and patience towards me helped me pick up the pace and keep going.

I would like to thank my colleagues for teaching and helping me a lot. The list includes Kerem Camsari, Vinh Diep, Seokmin Hong, Samiran Ganguly, Brian Sutton, Kuntal Roy, Ahmed Zeeshan Pervaiz, Shehrin Sayed, Orchi Hassan, Shuvro Chowdhury, Jan Kaiser, Anirudh Ghantasala and Risi Jaiswal. Vinh was my first mentor who introduced me to the basics of nanomagnet physics and simulation. It was also an absolute pleasure to work with Kerem and learn a lot from him. Kerem was always there for the rescue, be it answering difficult reviewer questions which is his favorite task to do or troubleshooting SPICE simulation errors. He was also the one to constantly challenge us with difficult queries to make the paper that we are trying to write better. Group meetings were my favorite; specially some apparently short meetings extending to nearly five hours.

I would like to thank Professor Joerg Appenzeller and Professor Zhihong Chen for their constant encouragement towards research. I enjoyed attending Thursday meetings every week with their experimental groups including Punyshloka Debashis,

Vaibhav Ostwal, Tingting Shen and others. I would like to thank Professor Ernesto Marinero for educating me on magnetism. I am also grateful to all my teachers for their blessings and valuable lessons in my life.

My friends at Purdue made life easier for an extremely homesick person like me. I was truly blessed to live within a very supportive environment surrounded by kind helpful friends and I am always thankful to them.

Last but not the least, I am thankful for my precious family: my mother (Shahina Yasmin), father (Mahmudul Islam), brother (Fahim), sister-in-law (Rimpa), husband (Ovi) and son (Izaan). It is only because of their selfless sacrifices for me that I could gather my strength over and agian and move forward.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

ABSTRACT

Faria, Rafatul Ph.D., Purdue University, May 2020. Autonomous Probabilistic Hardware for Unconventional Computing. Major Professor: Supriyo Datta.

In this thesis, we have proposed a new computing platform called probabilistic spin logic (PSL) based on probabilistic bits (p-bit) using low barrier nanomagnets (LBM) whose thermal barrier is of the order of a kT unlike conventional memory and spin logic devices that rely on high thermal barrier magnets (∼40-60 kT) to retain stability. p-bits are tunable random number generators (TRNG) analogous to the concept of binary stochastic neurons (BSN) in artificial neural network (ANN) whose output fluctuates between a +1 and -1 states with 50-50 probability at zero input bias and the stochastic output can be tuned by an applied input producing a sigmoidal characteristic response. p-bits can be interconnected by a synapse or weight matrix $[J]$ to build p-circuits for solving a wide variety of complex unconventional problems such as inference, invertible Boolean logic, sampling and optimization. It is important to update the p-bits sequentially for proper operation where each p-bit update is informed of the states of other p-bits that it is connected to and this requires the use of sequencers in digital clocked hardware. But the unique feature of our probabilistic hardware is that they are autonomous that runs without any clocks or sequencers. To ensure the necessary sequential informed update in our autonomous hardware it is important that the synapse delay is much smaller than the neuron fluctuation time. We have demonstrated the notion of this autonomous hardware by SPICE simulation of different designs of low barrier nanomagnet based p-circuits for both symmetrically connected Boltzmann networks and directed acyclic Bayesian networks. It is interesting to note that for Bayesian networks a specific parent to child update order is important and requires specific design rule in the autonomous probabilistic hardware

to naturally ensure the specific update order without any clocks. To address the issue of scalability of these autonomous hardware we have also proposed and benchmarked compact models for two different hardware designs against SPICE simulation and have shown that the compact models faithfully mimic the dynamics of the real hardware.

1. INTRODUCTION

Conventional memory and logic devices are made of stable deterministic units: Metal Oxide Semiconductor (MOS) or stable magnets with thermal barrier of the order of ∼40-60 kT [1–3]. **In this thesis, we present a unique computing platform called "Probabilistic Spin Logic" (PSL) based on stochastic unstable units** [4]. We call these stochastic units "p-bits" that are tunable random number generators (TRNG) and analogous to Binary Stochastic Neuron (BSN) [5] from artificial neural networks (ANN) literature. p-bits are three terminal transistor like entities with input-output isolation and gain and many of them can be electrically interconnected according to a synapse or weight matrix $[J]$ to build p-circuits for solving a wide variety of complex problems such as inference, invertible logic, sampling and optimization. Conventionally BSNs are implemented on digital clocked hardware with pseudo random number generators where it is important to update them sequentially so that each update is informed of the states of other connected units. But the feature that distinguishes our proposed p-circuits from binary stochastic neural networks is their completely autonomous clockless operation that allows very fast sampling due to massive parallelism (potentially peta flips per second sampling speed [6]). In an autonomous p-circuit, p-bits run autonomously in parallel without any clocks and it is very unlikely that two p-bits will update at the exact same time and continue doing that. So the updates are effectively sequential in the autonomous p-circuit. But just sequential updates are not enough, updates have to be *informed*. To make the updates *informed* it is necessary to ensure that the synapse delay (τ_S) is much smaller than neuron fluctuation time (τ_N) which is the design criteria for an autonomous probabilistic hardware for proper operation.

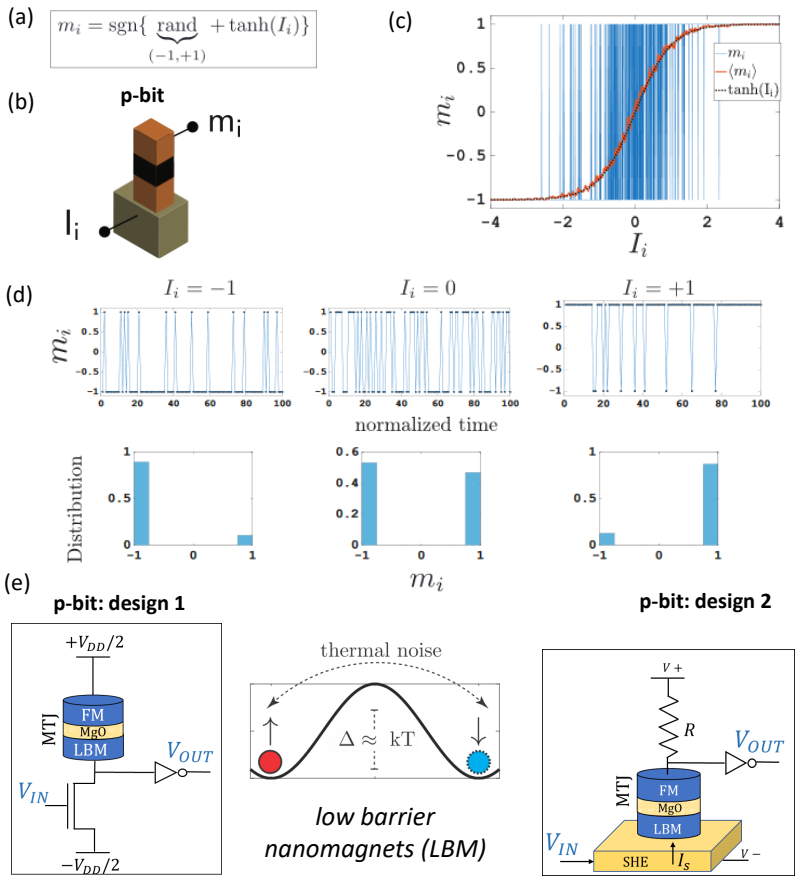


Fig. 1.1.: **Concept of a p-bit:** (a) A generic behavioral model for p-bit described by Eq. (1.1) with the icon shown in (b). (c) The blue trace shows the "magnetization" (m_i) obtained from Eq. (1.1) as the current (I_i) is ramped. The red trace shows the sigmoid response obtained from an RC circuit which provides a moving average of the time-dependent "magnetization" that agrees very well with the black curve showing $\tanh(I_i)$. The bias terminal could involve a voltage (V) instead of a current (I), just as the output could involve quantities other than magnetization. (d) The idealized telegraphic behavior of the model is shown at various bias points [4]. (e) Two hardware implementations of the p-bit unit based on stochastic low barrier nanomagnets (LBM) are shown: design 1 ([7]) and design 2 ([4]).

1.1 What is a p-bit?

A suitable building block for p-bit can be any random signal generator whose randomness can be tuned with an applied bias at a third terminal for input-output isolation. A generic building block is shown in fig. 1.1 whose output state m_i is controlled by the input bias I_i according to the behavioral equation [4]

$$m_i(t + \tau_N) = \text{sgn}\{\text{rand}(-1, 1) + \tanh(I_i(t))\} \tag{1.1}$$

Here τ_N is the average neuron flip time. From eqn. 1.1 we see that in the absence of any bias ($I_i = 0$), m_i randomly fluctuates between +1 and -1 with equal probability giving an average of $\langle m_i \rangle = 0$. When a positive bias ($I_i > 0$) is applied, m_i takes on +1 more likely than -1 resulting in $\langle m_i \rangle > 0$. Similarly for negative applied bias ($I_i < 0$), $\langle m_i \rangle < 0$ is obtained. Strong enough positive or negative bias will pin the states to either +1 or -1 respectively. This tunability of the output state of the p-bit by the applied bias is represented by a sigmoidal $\langle m_i \rangle$ vs I_i response..

There might be various ways to implement a p-bit. For example: (1) CMOS based [8–10] and (2) nanomagnet based [4, 7, 11, 12]. Any mechanism that provides a tunable random signal will qualify as a p-bit that can be interconnected to build interesting probabilistic circuits for various Boolean and non-Boolean operations.

p-bits can be interconnected according to a synapse or weight logic described by:

$$I_i(t + \tau_S) = I_0 \left(\sum_j J_{ij} m_j(t) + h_i \right) \tag{1.2}$$

where τ_S is the evalutation time for the input I_i, $[J]$ is the coupling matrix and $\{h\}$ is the local bias.

In chapter 2, we have shown that p-bits can be connected reciprocally to form Boltzmann Machines. The interesting property about Boltzmann Machines is that inputs and outputs can be treated on equal footing: if inputs are fixed, it gives the relevant output; if output is fixed, it gives all relevant inputs. Thus invertibility of logic functions can be obtained by Boltzmann design of logic gates. We have also

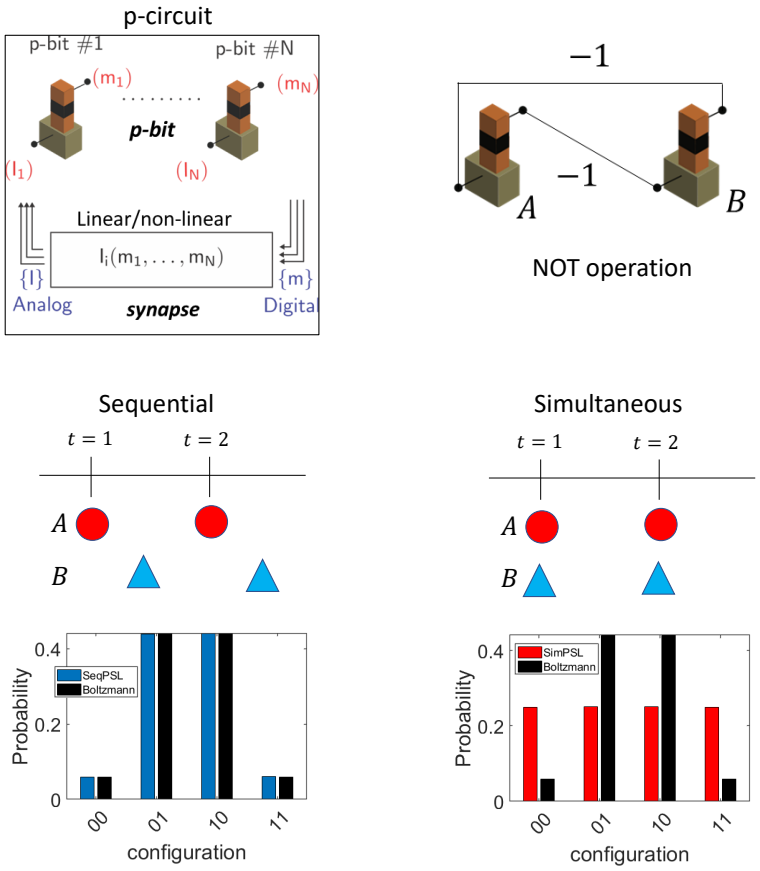


Fig. 1.2.: **p-circuit with sequencers:** A p-circuit is constructed by interconnecting p-bits according to a weight logic or synapse function. As a simple example a p-circuit with two p-bits (A and B) is shown where A and B are interconnected anti-ferromagnetically performing a NOT operation. It is shown that when A and B are updated sequentially one after another by a sequencer in the p-circuit, the network converges to the correct probability distribution from applying Boltzmann law for symmetrically connected networks. But if the sequencer is removed and p-bits A and B are updated simultaneously all at a time, wrong probability distribution is obtained with no preference for 01 or 10 states. Thus the use of sequencers is very important in the ANN literature.

shown that small Boltzmann units can be connected in a directed fashion that still retains the invertibility feature. A 32 bit adder is shown to perform as a subtractor

also. The p-circuit was simulated using the behavioral model defined by equations 1.1 and 1.2 and it is important to update the p-bits sequentially in this model.

1.2 Sequential versus autonomous p-circuit:

In traditional software implementation on a digital hardware, each p-bit is updated sequentially that means after each $\tau_S + \tau_N$ time interval only one p-bit is updated necessiating the use of sequencers. The importance of this sequencer is shown in fig. 1.2. But the interesting fact is that it is possible to design an autonomous p-circuit that does not require any kind of clocks of sequencers and yet can perform properly if the synapse dely τ_S is much smaller than τ_N as shown in fig. 1.3. How much synapse delay can be tolerated by an autonomous p-circuit will depend on the number of fan-in to the p-bits: in general larger the fan-in, lower will be the tolerance to synapse delay. The general design rule for an autonomous p-circuit is $\tau_S \ll (\tau_N/f_{in})$ where f_{in} is the number of fan-in.

Our physics inspired "autonomous" p-circuit as a hardware for ANN is very different from the commonly known "asynchronous" operation of biologically inspired spiking neural networks that require synchronizer and handshaking operation [13,14]. Since all the p-bits are running in parallel in our autonomous hardware, it is potentially a very fast and efficient sampler that can reduce the prefactor t_0 in the convergence time of probabilistic networks defined by $t = t_0 N_{flip}$ where the number of required flips N_{flip} is very much algorithm, network size, interconnection strength and network topology dependent.

We have demonstrated the operation of an autonomous p-circuit by SPICE simulaiton of two types of networks: Symmetrically connected Boltzmann and directed acyclic Bayesian networks.

In this thesis we have presented examples of invertible Boolean logic implemented on a Boltzmann network and Bayesian inference implemented on a Bayesian network by mapping these problems into two low barrier nanomagnet (LBM) based

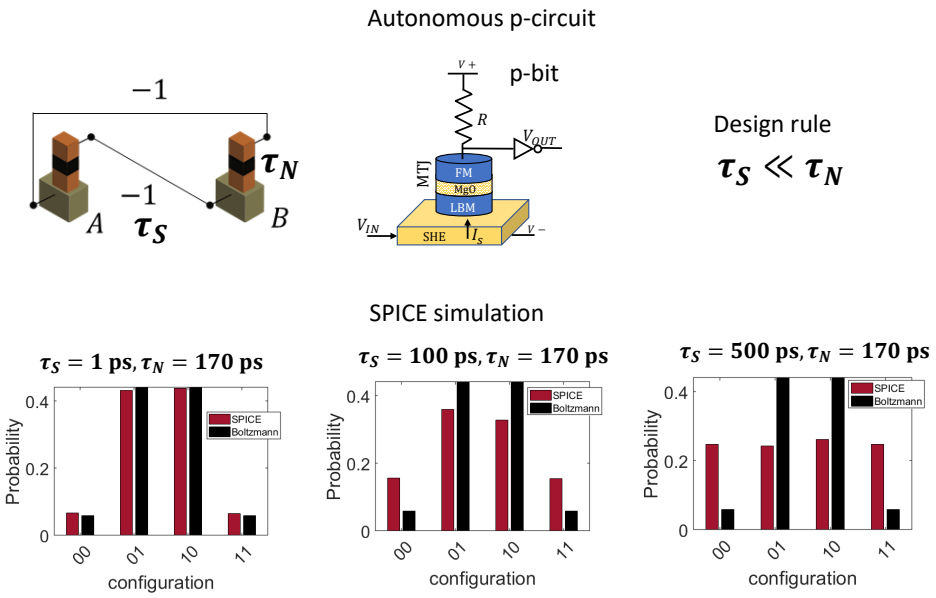


Fig. 1.3.: **Autonomous p-circuit:** As opposed to the sequential p-circuit shown in fig. 1.2, it is possible to design an autonomous p-circuit that does not require any kind of clocks or sequencers and still can operate properly if certain design criterion is met which is synapse delay τ_S has to be much smaller than neuron fluctuation time τ_N. This design rule is varified by SPICE simulation of the same two p-bit network as in fig. 1.2 composed of an LBM based p-bit design (design 2 in fig. 1.1). It is shown that when $\tau_S \ll \tau_N$, the system converges to the correct probability distribution consistent with equilibrium Boltzmann law, but as τ_S gets comparable to τ_N the system starts to fail.

autonomous p-circuit designs (design 1 and design 2 shown in fig. 1.1). Use of LBMs provides low power compact implementation of p-bits where the desired stochasticity comes naturally from the thermal fluctuations of nearly $\sim$ 0kT magnets. One design involves using a stochastic magnetic tunnel junction (MTJ) with the free layer replaced by an LBM and tuning the magnetization fluctuation by a spin current generated from a spin Hall material underneath. Another design is very similar to the commercially available 1T/1MTJ MRAM memory cell with the free layer of the

MTJ replaced by an LBM and tuning the fluctuating output by tuning the resistance of the NMOS transistor connected in series with the stochastic MTJ. We have also proposed and benchmarked two compact models for the two autonomous p-circuits against SPICE simulation of the actual hardware using experimentally benchmarked modules for LBMs and CMOS components. The compact models are useful for exploring very large scale autonomous probabilistic hardware.

1.3 Low barrier nanomagnet based p-circuit for invertible Boolean logic:

One possible implementation of p-bits is using low barrier stochastic nanomagnets with thermal barrier of the order of $\sim$ 1 kT where the tunability comes naturally from nanomagnets driven by spin currents. From fig. 1.4, we see that nanomagnet based p-bits do not have definite +1 and -1 states as described by the behavioral model (eq. 1.1). Rather the magnetization distribution is continuous.

Now the question is if all the interesting functionalities obtained from interconnecting binary p-bits can still be obtained when p-bits are not binary, rather continuous teleghaphic in nature. The question is answered in chapter 3. The answer is a resounding "yes". Even if nanomagnet based p-bits don't have specific +1 and -1 states, they can become strongly correlated to perform precise Boolean operations (e.g. 32 bit addition). No thresholding is applied during the operation of the circuit except at the READ out when all positive values are transformed to +1 and all negative values to -1.

To demonstrate the example of invertible Boolean logic, we have implemented a 32-bit adder interconnecting nearly five hundreds p-bits into a p-circuit that can do precise Boolean operation. We can build Boltzmann Machines (BM) by interconnecting p-bits symmetrically according to a properly designed coupling matrix $[J]$ and bias $\{h\}$ for doing specific Boolean operation. The BM design of Boolean gates provides the unique feature of invertibility. if the input p-bits are clamped, it gives the correct output with maximum probability. But if the output is clamped,

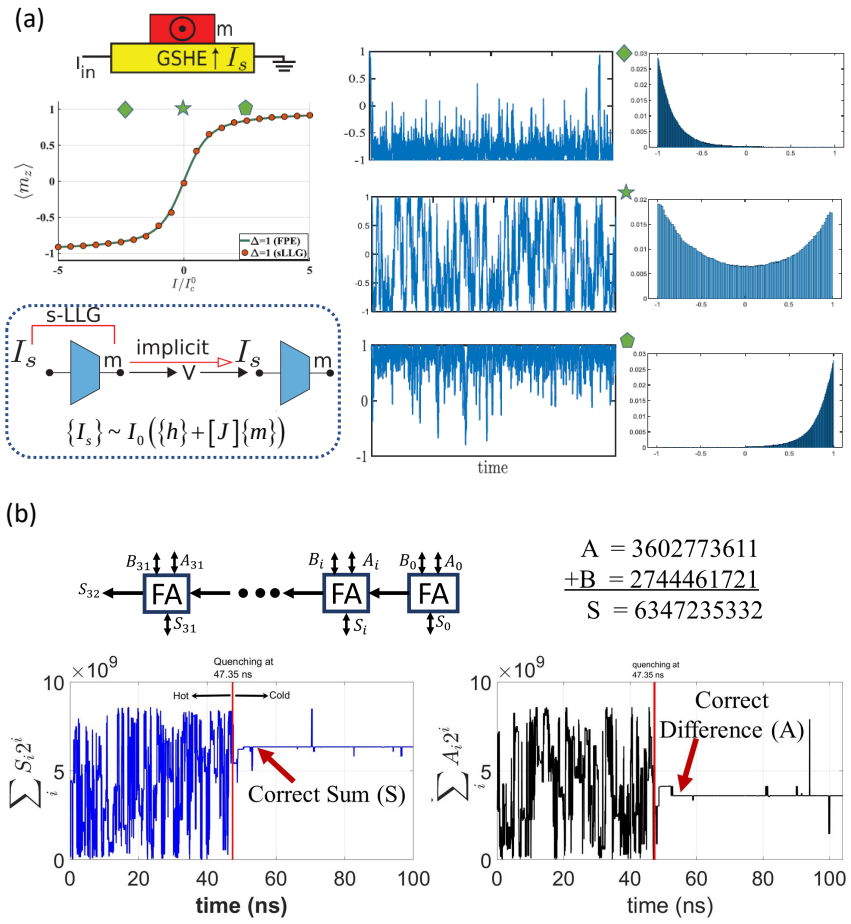


Fig. 1.4.: **Low barrier nanomagnet with continuous magnetization as p-bit for invertible logic:** (a) Implementation of a p-bit using a 1kT nanomagnet as the free layer on a GSHE material that converts the applied charge current to spin current to tune the average magnetization. At zero applied current, the magnetization fluctates among all values between +1 and -1 and the distribution is quite broad. When a positive current is applied, magnetization is biased towards +1 and for a negative current the magnetization distribution is concentrated around -1. (b) Implementation of an invertible 32-bit adder connecting 448 nanomagnets in an autonomous p-circuit.

it gives all the inputs consistent with the given output. We have proposed a unique hybrid architecture where small BM units are connected in a directed fashion. A 32-bit adder is shown to be composed of 32 full adders which are individual Boltz-

mann Machines. This unique architecture not only gives CMOS-like determinism, but also shows invertibility. When inputs A and B are clamped, this huge network of several hundred p-bits gets precisely correlated to to give the correct sum S out of $2^{33} \approx 8$ billion possibilities. But when the output bits S and one set of input bits A are clamped, the network gives the correct difference $B = S - A$. This invertibility is a unique feature of our proposed PSL. Solving stochastic Landau-Lifshitz-Gilbert (sLLG) equation, we show that even if these nanomagnets do not have specific 0 and 1 states, rather they have continuous distribution of magnetization, they can become properly correlated to do precise boolean operation. Here we want to clarify that the use of stochastic nanomagnets has been discussed both theoretically and experimentally for unconventional applications such as random number generation, autonomous learning, stochastic oscillator etc. [15–23]. The novelty of our PSL approach is that we are proposing to use these magnets in a clockless p-circuit to perform precise boolean logic that not only provides CMOS like determinism, but also is invertible. Such low barrier nanomagnets are usually of no interest because they cannot represent a specific 0 or 1. Use of these low kT magnets in PSL not only provides the exciting feature of invertibility in logic, they are also promising for low power operation since the pinning current for such low barrier magnets is much (at least an order of magnitude) less than what is needed for switching a 40 kT magnet.

Figure. 1.4 shows the sLLG simulation result of a network of 448 p-bits performing 32 bit addition. At zero bias, the network is like a molten liquid fluctuating randomly. As the interaction is turned on, the network quickly converges to the correct answer out of many billion possibilities. This probabilistic network is completely autnomous without any clocks and the synapse was assumed to be instantaneous thus satisfying the necessary criterion for an autonomous probabilistic hardware.

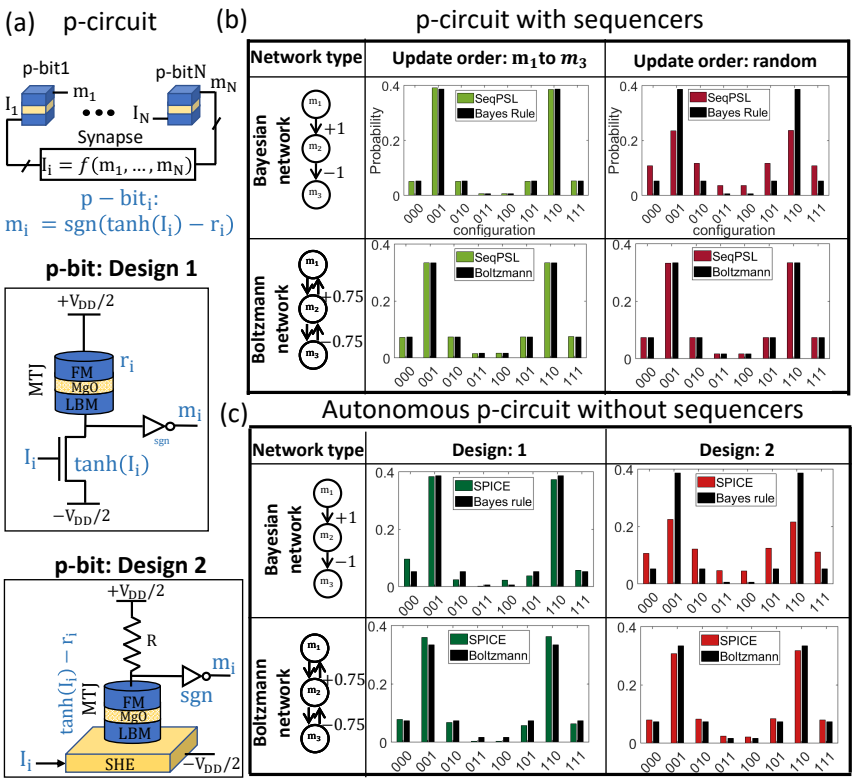


Fig. 1.5.: In a Bayesian network, p-bits representing each random variable of the network need to be updated sequentially from the parent to child nodes. We have proposed the design criteria for an autonomous hardware that would naturally ensure this specific update order without any clock circuitry by comparing two p-bit designs. It is seen that design 1 works well as a BN, but design 2 does not.

1.4 Autonomous p-circuit design for Bayesian network:

The autonomous p-circuit is particularly an interesting idea in terms of implementing directed acyclic networks also known as Bayesian networks (BN) or belief networks or causal networks that are popular in many AI related sectors for probabilistic reasoning and inference, because in this case a specific parent to child informed update order is very important in terms of matching standard statistical results that necessiates the use of sequencers as in digital circuits. We have shown that if certain design criteria are met, an autonomous p-circuit can implement a Bayesian net-

work without any sequencers and match analytical results reasonably well (fig. 1.5). In chapter 4 we have demonstrated a popular example of a BN named "Genetic relatedness" . We have shown how a BN described by conditional probability table (CPT) relating how each child node is dependent on its parent nodes can be translated to a p-circuit coupled by coupling matrix $[J]$ and bias h that are converted to coupling resistances R_{weight} and bias voltage V_{bias} respectively. We have shown that different correlation values coming directly out of the hardware nodes implemented in SPICE matches nicely with standard analytical values. In chapter 5, we have compared two autonomous p-circuit designs (design 1 and 2) as in fig. 1.1 in terms of implementing BNs and elucidated why design 1 works for a BN and design 2 does not in general.

In short, we have shown how to design autonomous probabilistic hardware that can implement different probabilistic networks (both directed and bidiectional) and solve a large class of AI and quantum computing related complex problems with very fast sampling speed. We have also proposed and benchmarked compact models for two different autonomous p-circuits to address the important issue of scalability of these networks.

2. STOCHASTIC P-BITS FOR INVERTIBLE LOGIC

Materials in this chapter have been extracted verbatim from the paper: " Stochastic p-bits for Invertible Logic", K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, published in Physical Review X, 2017. Reprinted with permision from [4].

Conventional semiconductor-based logic and nanomagnet-based memory devices are built out of stable, deterministic units such as standard MOS (metal oxide semiconductor) transistors, or nanomagnets with energy barriers in excess of $\approx$ 40-60 kT. In this paper we show that unstable, stochastic units which we call "p-bits" can be interconnected to create robust correlations that implement *precise Boolean functions* with impressive accuracy, comparable to standard digital circuits. At the same time they are *invertible*, a unique property that is absent in standard digital circuits. When operated in the direct mode, the input is clamped, and the network provides the correct output. In the inverted mode, the output is clamped, and the network fluctuates among all possible inputs that are consistent with that output. First, we present a detailed implementation of an invertible gate to bring out the key role of a single three-terminal transistor-like building block to enable the construction of correlated p-bit networks. The results for this specific, CMOS-assisted nanomagnet-based hardware implementation agree well with those from a universal model for p-bits, showing that p-bits need not be magnet-based: any three-terminal tunable random bit generator should be suitable. We present a general algorithm for designing a Boltzmann machine (BM) with a symmetric connection matrix [J] ($J_{ij} = J_{ji}$), that implements a given truth table with p-bits. The [J] matrices are relatively sparse with a few unique weights for convenient hardware implementation. We then show how BM Full Adders can be interconnected in a *partially directed* manner ($J_{ij} \neq J_{ji}$) to implement large logic operations such as 32-bit binary addition. Hundreds of stochastic p-bits get precisely correlated such that the correct answer out of 2^{33} ($\approx$ 8 billion) possibilities

can be extracted by looking at the statistical mode or majority vote of a number of time samples. With perfect directivity (J_{ji}=0) a small number of samples is enough, while for less directed connections more samples are needed, but even in the former case logical invertibility is largely preserved. This combination of digital accuracy and logical invertibility is enabled by the hybrid design that uses bidirectional BM units to construct circuits with partially directed inter-unit connections. We establish this key result with extensive examples including a 4-bit multiplier which in inverted mode functions as a factorizer.

2.1 Introduction

Conventional semiconductor-based logic and nanomagnet-based memory devices are built out of stable, deterministic units such as standard MOS (metal oxide semiconductor) transistors, or nanomagnets with energy barriers in excess of $\approx$ 40-60 kT. The objective of this paper is to introduce the concept of what we call "p-bits" representing unstable, stochastic units which can be interconnected to create robust correlations that implement precise Boolean functions with impressive accuracy comparable to standard digital circuits. At the same time this "probabilistic spin logic" (PSL) is *invertible*, a unique property that is absent in standard digital circuits. When operated in the direct mode, the input is clamped, and the network provides the correct output. In the inverted mode, the output is clamped, and the network fluctuates among all possible inputs that are consistent with that output.

Any random signal generator whose randomness can be tuned with a third terminal should be a suitable building block for PSL. The icon in Fig. 3.1b represents our generic building block whose input I_i controls the output m_i according to the equation (Fig. 3.1a),

$$m_i(t) = \text{sgn}\{\text{rand}(-1,1) + \tanh(I_i(t))\} \tag{2.1}$$

where rand(−1,+1) represents a random number uniformly distributed between −1 and +1. It is assumed to change every τ seconds which represents the retention time

of individual p-bits. We normalize the time axis to τ so that t is dimensionless and progresses in steps (0, 1, 2, ...). At each time step, if the input is zero, the output takes on a value of -1 or $+1$ with equal probability, as shown in the middle panel of Fig. 3.1d. A negative input I_i makes negative values more likely (left panel) while a positive input makes positive values more likely (right panel). Fig. 1c shows $m_i(t)$ as the input is ramped from negative to positive values. Also shown is the time-averaged value of m_i which equals $\tanh(I_i)$.

A possible physical implementation of p-bits could use stochastic nanomagnets with low energy barriers Δ whose retention time [24]:

$$\tau = \tau_0 \exp(\Delta/kT)$$

is very small, on the order of τ_0 which is a material dependent quantity called the attempt time and is experimentally found to be ≈ 10 ps $-$ 1 ns [24] among different magnetic materials. Such stochastic nanomagnets can be pinned to a given direction with spin currents that are at least an order of magnitude less than those needed to switch 40 kT magnets. The sigmoidal tuning curve in Fig. 3.1c describing the time average of a fluctuating signal represents the essence of a p-bit. Purely CMOS implementations of a p-bit are possible [8, 9], but the sigmoid seems like a natural feature of nanomagnets driven by spin currents. Indeed, the use of stochastic nanomagnets in the context of random number generators, stochastic oscillators and autonomous learning [18, 25, 26] has been discussed in the literature. But performing "invertible" Boolean logic utilizing large scale correlations has not been discussed before to our knowledge.

Note that we are using the term *invertibility* in the broader sense of relation inverses and not in the narrower sense of function inverses. For example, AND, when interpreted as a relation, consists of the set $\{\{1,1 \rightarrow 1\}, \{0,0 \rightarrow 0\}, \{1,0 \rightarrow 0\}, \{0,1 \rightarrow 0\}\}$ where each term is of the form $\{A, B \rightarrow$ AND(A, B)$\}$. The relation inverse of 0 is the set $\{\{0,0\}, \{0,1\}, \{1,0\}\}$ even though the corresponding functional inverse is not defined. What our scheme provides, probabilistically, is the relation inverse [27, 28].

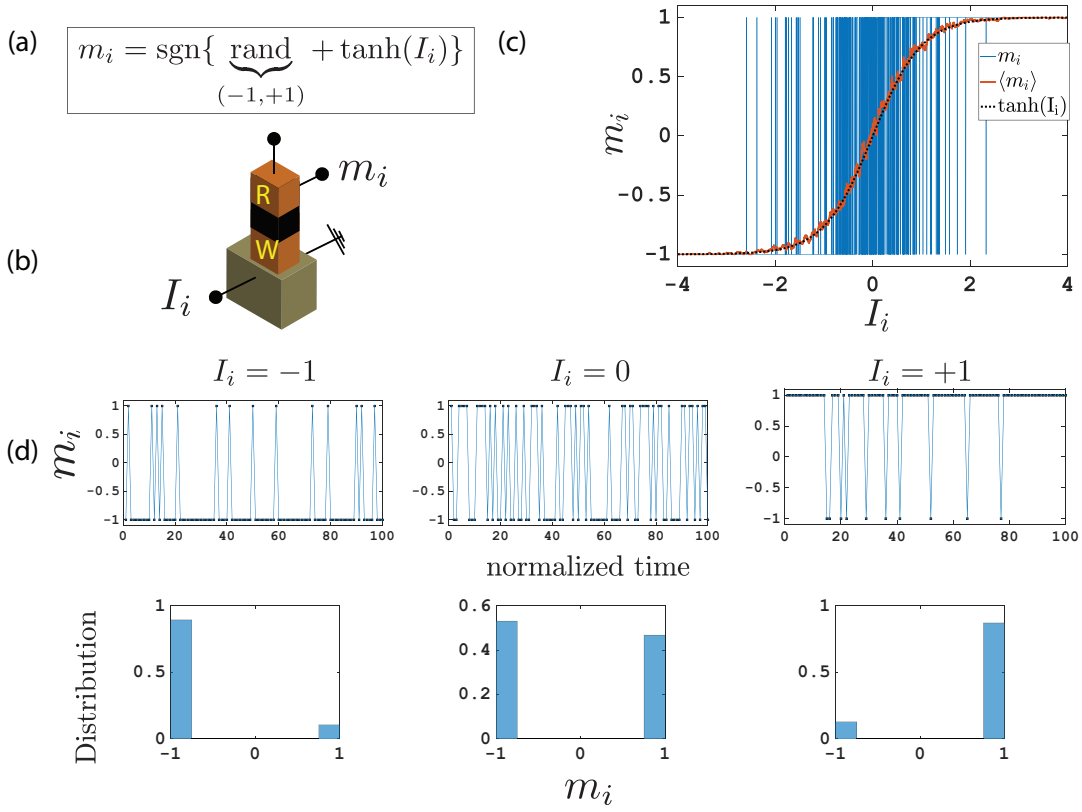


Fig. 2.1.: **Generic building block for PSL:** (a) A generic model for PSL described by Eq. (2.1) with distinct READ and WRITE units represented by the R/W icon shown in (b). Useful functionalities are obtained by interconnecting R/W units according to Eq. (2.2), $I_i = I_0 \times (h_i + \sum J_{ij} m_j)$, with appropriately designed $\{h\}$ and $[J]$. (c) The blue trace shows the "magnetization" (m_i) obtained from Eq. (2.1) as the current (I_i) is ramped. The red trace shows the sigmoid response obtained from an RC circuit which provides a moving average of the time-dependent "magnetization" which agrees very well with the black curve showing $\tanh(I_i)$. The bias terminal could involve a voltage (V) instead of a current (I), just as the output could involve quantities other than magnetization. (d) The idealized telegraphic behavior of the model is shown at various bias points along with corresponding distributions.

Ensemble-average versus time-average: A sigmoidal response was presented in [29] for the ensemble-averaged magnetization of large barrier magnets biased along

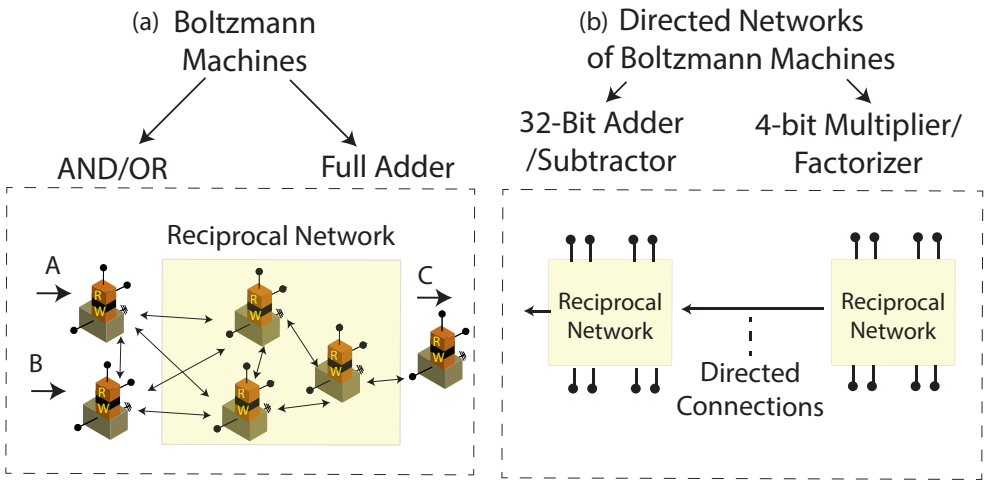


Fig. 2.2.: **PSL designs discussed in this paper**: (a) Basic Boolean elements (AND/OR, Full Adder) are implemented as Boltzmann Machines based on symmetrically coupled networks with $J_{ij} = J_{ji}$. (b) Complex Boolean functions like a 32-bit Ripple Carry Adder/Subtractor and 4-bit Multiplier/Factorizer are implemented by combining the reciprocal Boltzmann machines in a directed fashion.

a neutral state. This was proposed as a building block for both Ising computers as well as directed belief networks and a recent paper [30] describes a similar approach applied to a graph coloring problem. By contrast low barrier nanomagnets provide a sigmoidal response for the time-averaged magnetization and a suitably engineered network of such nanomagnets could cycle through the 2^N collective states at GHz rates, with an emphasis on the "low energy states" which can encode the solution to the combinatorial optimization problems, like the traveling salesman problem (TSP) as shown in [11]. Once the time-varying magnetization has been converted into a time-varying voltage through a READ circuit, a simple RC circuit can be used to extract the answer through a moving time average. For example, in Fig. 3.1c the red trace was obtained from the rapidly varying blue trace using an RC circuit in a SPICE simulation.

The central feature underlying both implementations is the *p-bit* that acts like a tunable random number generator, providing an intrinsic sigmoidal response for

the ensemble-averaged or the time-averaged magnetization as a function of the spin current. It is this response that allows us to *correlate* the fluctuations of different p-bits in a useful manner by interconnecting them according to

$$I_i(t) = I_0 \times (h_i(t) + \sum_j J_{ij} m_j(t)) \tag{2.2}$$

where h_i provides a local bias to magnet i and J_{ij} defines the effect of bit j to bit i, and I_0 sets a global scale for the strength of the interactions like an inverse "pseudo-temperature" giving a dimensionless current I_i to each p-bit. The computation of $I_i(t)$ in terms of $m_j(t)$ in Eq. (2.2) is assumed instantaneous, in hardware implementations there can be interconnect delays that relate $m_j(t)$ to currents at a later time, $I_i(t')$.

Equation (2.1) arises naturally from the physics of low barrier nanomagnets as we have discussed above. Equation (2.2) represents the "weight logic" for which there are many candidates such as memristors [31], floating-gate based devices [32], domain-wall based devices [33], standard CMOS [34]. The suitability of these options will depend on the range of J values and the sparsity of the J-matrix.

Equations (2.1-2.2) are essentially the same as the defining equations for Boltzmann machines introduced by Hinton and his collaborators [35] which have had enormous impact in the field of machine learning, but they are usually implemented in software that is run on standard CMOS hardware. The primary contributions of this paper are threefold:

- *Hardware implementation:* It may seem "obvious" that an unstable magnet could provide a natural hardware for representing a p-bit, but we would like to stress a less obvious point. To the best of our knowledge, simple two-terminal devices are not suitable for constructing large scale correlated networks of the type envisioned here. Instead, we need three-terminal building blocks with transistor-like gain and input-output isolation as shown in Fig. 3.1b [29]. To stress this point, we describe a concrete implementation of a Boolean function using detailed nanomagnet and transport simulations that are in good agreement with those obtained by the generic model based on Eq. (2.1). All other

results in this paper are based on Eq. (2.1) in order to emphasize the generality of the concept of p-bits which need not necessarily be nanomagnet-based [36,37].

- *Boltzmann machines (BM) for invertible Boolean logic (Fig.2a):* Much of the current emphasis on BMs is on "learning" giving rise to the concept of restricted Boltzmann machines [38]. By contrast this paper is about Boolean logic, extending an established method for Hopfield networks [39] to provide a mathematical prescription to turn *any* Boolean truth table into a symmetric J-matrix (Eq. (2.2), with $J_{ij} = J_{ji}$), in one shot with no "learning" being involved. This design principle seems quite robust, functioning satisfactorily even when the J-matrix elements are rounded off, so that the required interconnections are relatively sparse and quantized which simplifies the hardware implementation. The numerical probabilities agree well with those predicted from the energy functional.

$$E(\{m\}) = -I_0 \times \left(\sum_{i,j} \frac{1}{2} \left(J_{ij} m_i m_j \right) + \sum_i h_i m_i \right) \tag{2.3}$$

using the Boltzmann law:

$$P(\{m\}) = \frac{\exp(-E)}{\sum_{i,j} \exp(-E)} \tag{2.4}$$

Most importantly we show that *the resulting Boolean gates are invertible:* not only do they provide the correct output for a given input, for a given output they provide the correct input(s). If the given output is consistent with multiple inputs, the system fluctuates among all possible answers. This remarkable property of invertibility is absent in standard digital circuits and could help provide solutions to the Boolean satisfiability problem (Fig. 2.8) [40].

- *Directed networks of BM (Fig.2b):* Finally we show that individual BM's can be connected to perform *precise* arithmetic operations which are the norm in standard digital logic, but quite surprising for BM which are more like a collection

of interacting particles than like a digital circuit. We show that a 32-bit adder converges to the one correct sum out of $2^{33} \approx 8$ billion possibilities when the interaction parameter is suddenly turned up from say $I_0 = 0.25$ to $I_0 = 5$. This can be likened to quenching a molten liquid and *getting a perfect crystal.* What we expect is plenty of defects, distributed differently everytime we do the experiment. That is exactly what we get if the individual BM Full adders comprising the 32-bit adder are connected bidirectionally ($J_{ij} = J_{ji}$). But by making the connection between Adders directed ($J_{ij} \neq J_{ji}$), we obtain the striking accuracy of digital circuits while largely retaining the invertibility of BM. This is a key result that we establish with extensive examples including a 4-multiplier which in inverted mode functions as a factorizer.

Each of these three contributions is described in detail in the three sections that follow.

2.2 An example hardware Implementation of PSL

To ensure that individual p-bits can be interconnected to produce robust correlations, it is important to have separate terminals for writing (more correctly biasing) and reading, marked W and R respectively in Fig. 2.3a. With IMA nanomagnets (e.g circular nanomagnets) this could be accomplished following existing experiments [15, 42] using the giant spin Hall effect (GSHE). Recent experiments using a built-in exchange bias [43–46] could make this approach applicable to PMA as well. Note however, that these experiments have all been performed with stable free layers, and would have to be carried out with low barrier magnets in order to establish their suitability for the implementation of p-bits. As the field progresses, one can expect the bias terminal to involve voltage control [47, 48] instead of current control, just as the output could involve quantities other than magnetization. We will now show a concrete implementation of a Boolean function using minimal CMOS circuitry in conjunction with stochastic nanomagnets through detailed nanomagnet and trans-

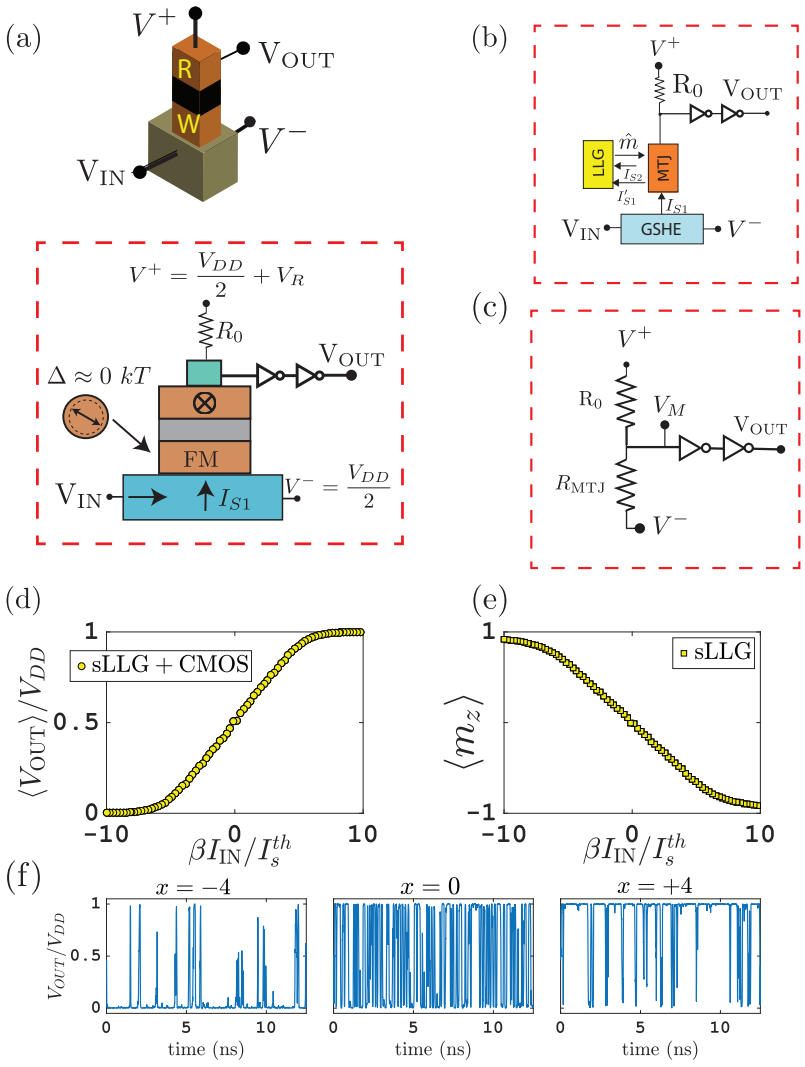


Fig. 2.3.: **CMOS-assisted implementation of p-bits:** (a) A possible CMOS-assisted implementation of p-bits that have a separate READ/WRITE paths. A GSHE layer provides a spin current that pins the magnetization of circular magnets ($\Delta \approx 0\ kT$). The change in magnetization is sensed by an MTJ and amplified by two CMOS inverters that act as a buffer, providing the necessary isolation and gain. (b) Self-consistent, modular modeling of transport and magnetization dynamics. See "Assumptions of the model" in the text. (c) Equivalent READ circuit. (d) SPICE-based average output voltage normalized to the $V_{DD} = 0.8$ V of 14 nm FinFET HP-inverters [41]. (e) sLLG-based average magnetization of the circular magnet as a function of the spin current (averaged over 500 ns for each bias point with a time step of $\Delta t = 0.05$ ps, 10 million points per marker), normalized to the GSHE gain and the thermal noise strength, I_s^{th}. (f) The time-dependent output voltage at various bias points.

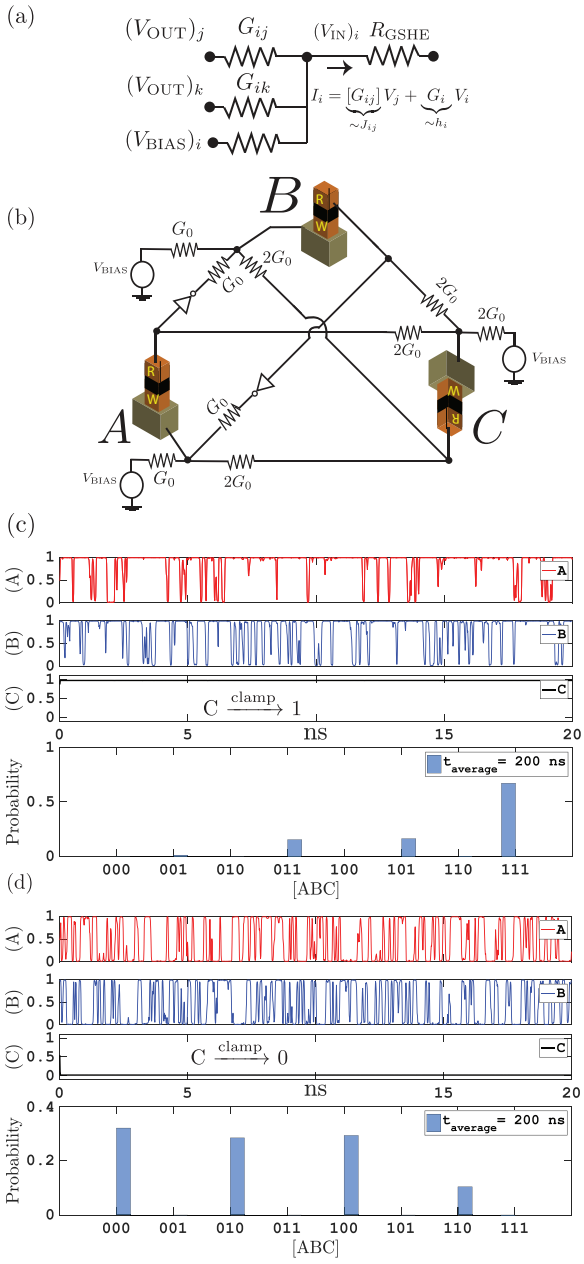


Fig. 2.4.: **An invertible AND gate:** (a) Passive resistor network that is used to obtain the connection terms J_{ij} to correlate p-bits. The output impedance $R_{ij} = 1/G_{ij}$ is much smaller than the input impedance R_{GSHE}, allowing separate voltages to add at the input of the ith p-bit. (b) Explicit implementation of an AND gate based on Eq. (2.10). (c) When C is clamped to 1, A and B spend most of their time in the (11) state, the only combination consistent with C=1. (d) The invertible operation of the AND gate when the C gate is clamped to a zero, while A and B are left floating. A and B bits fluctuate between 3 possible combinations consistent with C=0, (A,B)=(00),(01),(10). The time response of A,B,C voltages are normalized by V_{DD}. Histogram is obtained by averaging over 200 ns of thresholded voltages, only the first 20 ns of A,B,C voltages are shown for clarity.

port simulations that are in good agreement with those obtained from the generic model based on Eq. (2.1).

Fig. 2.3a shows a possible, CMOS-assisted p-bit that has a separate READ and WRITE path. The device consists of a heavy metal exhibiting Giant Spin Hall Effect (GSHE) that drives a circular magnet which replaces the usual elliptical magnets in order to provide the stochasticity needed for the magnetization. A small read current, which is assumed to not disturb the magnetization of the free layer in our design, that flows through the fixed layer is used to sense the instantaneous magnetization, which is amplified and isolated by two inverters that act as a buffer. This structure is very similar to the experimentally demonstrated GSHE switching of elliptical magnets that were similarly read-out by an MTJ [42], with the only exception that the elliptical magnets are replaced by circular magnets with an aspect ratio of one. This device could be viewed as replacing the free layers of the GSHE-driven MTJs demonstrated in [42] with those in the telegraphic regime [15, 49–51] .

In the presence of thermal noise the magnetization of such a circular magnet rotates in the plane of the circle without a preferred easy-axis that that would have arisen due to the shape anisotropy, effectively making its thermal stability $\Delta \approx 0$ kT [52]. This magnetization can be pinned by a spin current that is generated by flowing a charge current through the GSHE layer. The magnetic field driven sigmoidal responses of magnetization for such circular magnets have experimentally been demonstrated [53, 54], while the spin current driven pinning has not been demonstrated to our knowledge. Using validated modules for transport and magnetization dynamics [55] (Fig. 2.3b), we solve the stochastic Landau-Lifshitz-Gilbert (sLLG) equation in the presence of thermal noise and a GSHE current. The following subsection shows detailed simulation parameters.

Sigmoidal response: A long-time average ($t = 500$ ns) of the magnetization $\langle m_z \rangle$ as a function of a GSHE-generated spin current is plotted in Fig. 2.3e that displays the desired sigmoidal characteristic for p-bits dictated by Eq. (2.1). The x-axis of

Fig. 2.3e is normalized to the geometric gain factor that relates the charge current to the spin current exerted [56, 57]:

$$\beta \equiv \frac{I_s}{I_c} = \theta_{\mathrm{SH}} \frac{L_{FM}}{t} \left(1 - \mathrm{sech}\left(\frac{t}{\lambda} \right) \right) \tag{2.5}$$

where θ_{SH} is the Hall angle, t is the thickness and λ is the spin-relaxation length of the heavy metal. The quantity β can be made to be much greater than 1 providing an intrinsic gain [2], however for the parameters used in the present examples, β is ≈ 1.5.

Another quantity that is used to normalize the x-axis of Fig. 2.3e is the "thermal spin current" that corresponds to the strength of the thermal noise that needs to be overcome for a circular magnet to be pinned in a given direction:

$$I_s^{th} = \left(\frac{4q}{\hbar} \right) \alpha (kT) \tag{2.6}$$

where q is electron charge, α is the damping coefficient of the magnet. I_s^{th}, I_s and I_c all have units of charge current, therefore we can define the dimensionless interaction parameter, I_0 of Eq. 2.2 as $I_0 \equiv \beta I_c / I_s^{th} = I_s / I_s^{th}$.

It can be seen from Fig. 2.3e that when the applied spin current $\beta I_c / I_s^{th} = I_s / I_s^{th} \approx 10$, the magnetization of the circular magnet is pinned in the $\pm z$ directions for these particular parameters. For PMA magnets with low barriers ($\Delta \ll kT$), the pinning current is independent of the volume as long as increasing the volume does not invalidate the $\Delta \ll kT$ assumption. This can be analytically shown from a 1D Fokker-Planck equation [58], and we have reproduced this behavior directly from sLLG simulations. For the in-plane (circular) magnets considered here, the pinning current in general has a M_s and Vol. dependence and the dimensionless pinning current can be larger.

Nevertheless, it is possible to estimate the thermal spin current for typical damping coefficients of $\alpha = 0.01 - 0.1$, I_s^{th} is $\approx 0.25\ \mu A - 2.5\ \mu A$. Pinning currents for superparamagnets are at least an order of magnitude smaller than the critical switching currents of stable magnets [59]. I_s^{th}, defined by Eq. (2.6) also sets the scale for I_0

defined in Eq. (2.2) suggesting that a stochastic nanomagnet based implementation of PSL could be more energy efficient than the standard spin-torque switching of stable magnets that suffer from high current densities.

Need for three-terminal devices with READ-WRITE separation: Note that a crucial function of the READ circuit and the CMOS transistors in this design is the ability to turn the magnetization into an output voltage that is proportional to m_z, providing gain for fan-out and isolation to avoid any read disturb. Indeed, a critical requirement for any other alternative implementations of p-bits is the need for three terminal devices with separate READ and WRITE paths to provide gain and isolation. In this particular design these features come in by directly integrating CMOS transistors, but CMOS-free, all-magnetic designs with these characteristics have been proposed [2, 60]. Our purpose is to simply show how a p-bit can be realized by using experimentally demonstrated technology. Alternative designs are beyond the scope of this paper.

READ Circuit: For the output to provide symmetric voltage swings on the GSHE layer, the minus supply V^- needs to be set to $V_{DD}/2$ since $V_{\rm OUT}$ ranges between 0 and V_{DD}. V^+ is set to $V_{DD}/2 + V_R$ where V_R is a small READ voltage that is amplified by the inverters. We assume a simple, bias-independent MTJ model [61]:

$$G_{\rm MTJ} = G_0(1 + P^2 m_z), \tag{2.7}$$

where P is the interface polarization and G_0 is the average MTJ conductance. Setting the reference resistance (Fig. (2.3c)) R_0 equal to G_0^{-1}, the input voltage to the inverters, V_M in FIG. (3.2d) becomes:

$$V_M = \frac{V_{DD}}{2} + \frac{V_R}{2 + m_z P^2} \tag{2.8}$$

In the absence of a bias $\langle m_z \rangle$ becomes 0 and the middle voltage fluctuates around the mean $\langle V_M \rangle = V_{DD}/2 + V_R/2$. This requires the inverter characteristic to be shifted to this value to produce a telegraphic output that fluctuates between 0 and V_{DD} with equal probability (Fig. 2.3f). This shift is easily engineered by sizing the pFET and

nFET transistors differently, a wider pFET shifts the inverter characteristic towards V_{DD}, as we will show in the next subsection.

Interconnection matrix: A passive resistor network can be used as a possible interconnection scheme to correlate the p-bits as shown in Fig. 2.4. A proper design of the interconnection matrix J that has only a few discrete values ensures a minimal number of different conductances (G_{ij}). In this demonstrated example the AND gate requires only 2 unique, discrete conductance values.

The spin currents that need to be delivered to each p-bit are on the order of a few μA and can be generated with charge currents that are even smaller, due to the GSHE gain. This means the interconnection resistances R_{ij} could be on the order of 100 kΩ's since the voltage drops across these resistances are around $V_{\rm OUT} - V^- \approx \pm 0.5$ V. Since the GSHE ground $V^- = V_{DD}/2$ simply shifts all the voltages to get symmetric $\pm$ swings, we define the voltages $(V'_{\rm OUT})_i = (V_{\rm OUT})_i - V^-$. Then input currents to each p-bit can be expressed (Fig. 2.4a):

$$\left(I_{\rm IN}\right)_i = \sum_j G_{ij}(V'_{\rm OUT}) + G_i(V'_{\rm BIAS}) \tag{2.9}$$

assuming $\sum_j G_{ij} \ll G_{GSHE}$ since the heavy metal resistances are typically much less than hundreds of kΩ. We have verified the validity of Eq. (2.9) by SPICE simulations, for the parameters chosen for these examples.

As a result, we observe that Eq. (2.9) constitutes a hardware mapping for the interconnections of Eq. (2.2). In this scheme G_{ij} conductances are initially adjusted to obtain a global interaction strength I_0 for a given problem. Alternatively, the interaction strength can be adjusted electrically by varying the supply voltages.

Invertible AND Gate: Fig. 2.4b shows an explicit implementation of an invertible AND gate ($A \cap B = C$) corresponding to [J] and $\{h\}$ matrices [62] that have 3 unique, integer entries:

$$J = \begin{array}{c} \\ \text{A} \\ \text{B} \\ \text{C} \end{array} \begin{array}{c} \begin{array}{ccc} \text{A} & \text{B} & \text{C} \end{array} \\ \begin{pmatrix} 0 & -1 & +2 \\ -1 & 0 & +2 \\ +2 & +2 & 0 \end{pmatrix} \end{array} \quad h^T = \begin{bmatrix} +1 & +1 & -2 \end{bmatrix} \tag{2.10}$$

In Fig. 2.4d, we show the *inverse* operation of the AND gate where we clamp the output bit C to a 0 or 1 by the bias voltage attached to its input terminal. The interconnection resistance is chosen to be $R_0 = 125$ kΩ that roughly provides $\approx \pm 6$ μA of charge current to each p-bit, corresponding to an $I_0 \approx 3.5$ for the chosen parameters.

Generating the histogram: At the end of the simulation (t=200 ns), we threshold the voltage output of A,B and C by legislating all voltages above $V_{DD}/2 = 0.4$ V to be 1, and below $V_{DD}/2$ to be 0. Then a histogram output for the thresholded word [ABC] is obtained and normalized to unit probability. Clamping the output to 0 and letting A and B float, make A and B fluctuate in a correlated manner and they visit the three possible states (00, 01, 10) with approximately equal probability. Resolving the output 0 to the three possible input combinations is, in a way "factorizing" the output. Conversely, clamping the output to 1 produces a strong (11) peak in the histogram of [ABC], which is the only consistent input combination for C=1 (Fig. 2.4c-d).

Assumptions of the model: We have made several simplifying assumptions while modeling the hardware implementation of a p-bit. (1) The READ voltage that is amplified by the inverters produces a small current that passes through the circular magnet and might potentially disturb its current state. We assumed that this current (labeled as I_{S2} in Fig. 2.3b) is negligible and do not affect the magnetization of the stochastic magnet. (2) We assumed that the spin current generated by the heavy metal is deposited to the free layer with perfect efficiency ($I'_{S1} = I_{S1}$ in Fig. 2.3b), however, depending on the interface properties this conversion factor can be less than

100%. (3) We have also assumed that the fixed layer does not produce a notable stray field on the circular magnet. Note that the presence of such a constant field would simply shift the sigmoidal behavior presented in Fig. 2.3d-e to the right (or left) and could have been offset by a constant bias current. (4) Finally, we have neglected the resistance of the GSHE portion in the READ circuit (Fig. 2.3c), assuming the MTJ resistance would be dominant in this path.

2.2.1 Detailed Simulation Parameters

This section shows the details of simulation parameters for the hardware implementation of p-bits that are used for Fig. 2.3–2.4.

sLLG for stochastic circular magnets: The magnetization of a circular nanomagnet described as $\hat{m}_i$ is obtained from the stochastic Landau-Lifshitz-Gilbert (sLLG) equation:

$$(1+\alpha^2)\frac{d\hat{m}_i}{dt} = -|\gamma|\hat{m}_i \times \vec{H}_i - \alpha|\gamma|(\hat{m}_i \times \hat{m}_i \times \vec{H}_i) + \frac{1}{qN_i}(\hat{m}_i \times \vec{I}_{Si} \times \hat{m}_i) + \left(\frac{\alpha}{qN_i}(\hat{m}_i \times \vec{I}_{Si})\right) \tag{2.11a}$$

where α is the damping coefficient, q is the electron charge, γ is the electron gyromagnetic ratio, I_s is the spin current that is assumed to be uniformly distributed over the total number of spins in the macrospin, $N_i = M_s\text{Vol.}/\mu_B$, μ_B being the Bohr magneton. It is assumed that the spin current generated from the GSHE layer is polarized in the z-direction, such that $\vec{I}_{Si} = I_S\hat{z}$. $\vec{H}_i$ is the effective field of the circular magnet, where the uniaxial anisotropy is assumed to be negligible, but there is still a strong demagnetizing field. The thermal fluctuations also enter through the effective magnetic field: $\vec{H}_i = -4\pi M_s m_x \hat{x} + \vec{H}_{th}$, x-axis being the out-of-plane direction of the magnet, and $\langle|\vec{H}_{th}|^2\rangle = 2\alpha kT/(|\gamma|M_s\text{Vol.})$ in units [Oe^2/Hz] with zero mean, and equal in all three directions. Table 2.1 shows the parameters used in Figs. 2.3–2.4. We note that this parameter selection is simply one possibility, many other parameters could have been used with no change in the basic conclusions.

Obtaining the sigmoidal response of CMOS+sLLG: Each data point in the sigmoids shown in Figs. 2.3−2.4 is obtained by averaging the z-component of the magnetization after 500 ns, with a time-step of $\Delta t = 0.05$ ps. The CMOS inverter char-

Table 2.1.: Parameters used for simulations in Figs. 2.3−2.4.

Parameters	**Value**
Saturation magnetization (M_s)	300 emu/cc
Magnet diameter (Φ), thickness (t)	15 nm, 0.5 nm
MTJ Polarization (P) (Eq. (2.7))	0.5
MTJ Conductance (G_0) (Eq. (2.7))	176 μS
Damping coefficient (α)	0.1
Spin Hall Length, Width (Eq. (2.5))	$L = W = 15$ nm
Hall Angle, Spin relax. length	θ=0.5 [63], λ_{sf} =2.1 nm [64]
Spin Hall res. (ρ), thickness (t)	200 $\mu\Omega$-cm [65], 3.15 nm
Temperature (T)	300 K
CMOS Models	14nm HP-FinFET [41]
Supply and READ Voltage	$V_{DD} = 0.8$ V, $V_R = 0.5$ V
Timestep for transient sim. (SPICE)	Δt = 0.05 ps

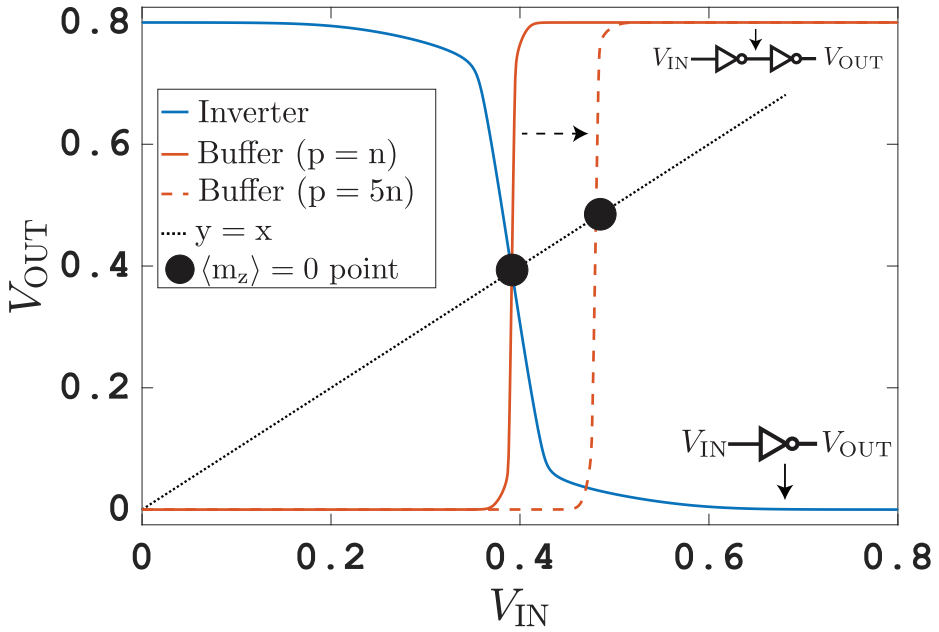


Fig. 2.5.: **14 nm PTM, Inverter/Buffer:** DC response of 14 nm high performance (HP) FinFETs based on [41] for an inverter and buffer. Sizing the transistors differently allows the switching point to be shifted.

acterestics in conjunction with a spherical representation-based sLLG are obtained using the modular framework developed in [55] using HSPICE.

14 nm FinFET Inverter Characteristics: Fig. 2.5 shows the input/output characteristics of the single and double inverters that are used to amplify the stochastic signal that is generated by the MTJ (Fig. 2.3). At zero-bias from the GSHE, the amplified signal V_M (Eq. 2.8) is in the middle of V^+ and V^- which is $V_{DD}/2 + V_R/2$. The buffer response can be shifted to this value by increasing the size of pFETs, as shown in Fig. 2.5.

2.3 Invertible Boolean logic with Boltzmann Machines

We now present a mathematical prescription that shows how any given truth table can be implemented in terms of Boltzmann Machines, in "one shot" with no learning being involved, unlike much of the past work in this area (See for example, [66, 67]).

In Section 2.2, we chose a simple [J] and $\{h\}$ matrix to implement an AND gate based on [62]. In this section, we outline a general approach to show how any truth table can be implemented in terms of such matrices. Our approach, pictorially described in Fig. 2.6, begins by transforming a given truth table from binary $(0, 1)$ to bipolar $(-1, +1)$ variables. The lines of the truth table are then required to be eigenvectors each with eigenvalue $+1$, all other eigenvectors are assumed to have eigenvalues equal to 0. This leads to the following prescription for J as shown in Fig. 2.6:

$$[J] = \sum_{i,j} [\mathrm{S}^{-1}]_{\mathrm{ij}} \mathrm{u_i} \mathrm{u_j^\dagger} \tag{2.12a}$$

$$\mathrm{S_{ij}} = \mathrm{u_i^\dagger u_j} \tag{2.12b}$$

where u_i are the eigenvectors corresponding to lines in the truth table of a Boolean operation and S is a projection matrix that accounts for the non-orthogonality of the vectors defined by different lines of the truth table. Note that the resultant J-matrix is always symmetric ($J_{ij} = J_{ji}$) with diagonal terms that are subtracted in our models such that $J_{ii} = 0$. The number of p-bits in the system is made greater than the number of lines in a truth table through the addition of hidden units (Fig. 2.6) to ensure that the number of conditions we impose is less than the dimension of the space defined by the number of p-bits.

Another important aspect in the construction of [J] is that an eigenvector $\mathrm{u_i}$ implies that its complement $-\mathrm{u_i}$ is also a valid eigenvector. However only one of these might belong to a truth table. We introduce a "handle" bit to each $\mathrm{u_i}$ that is biased (h_i) to distinguish complementary eigenvectors. These handle bits provide the added benefit of reconfigurability. For example, AND and OR gates have complementary truth tables, and a given gate can be electrically reconfigured as an AND or an OR gate using the handle bit.

J-Matrices for AND/FA: We now provide the details of the J-matrix for the AND gate, obtained using the prescription shown in Fig. 2.6 based on Eq. (2.12a). The eigenvectors of the truth table for the AND in Fig. 2.6 are placed into a matrix U,

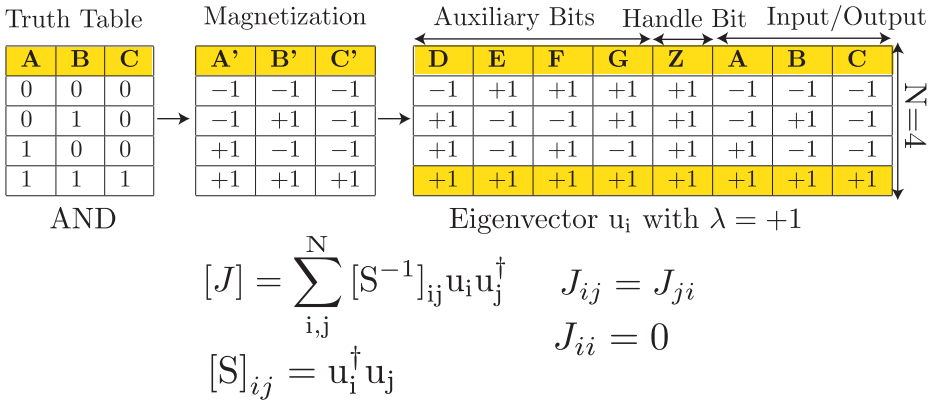


Fig. 2.6.: **Truth Table to J-Matrix:** A given truth table is first transformed from binary to bipolar variables by using the transformation $m = 2t - 1$, where m and t represent the magnetization and binary values of the truth table. Additional bits are introduced to each line of the truth table to ensure that the resultant S-matrix is invertible. The indices i, j correspond to the number of lines in the truth table. u_i, u_j are column vectors. As an example, we have shown auxiliary bits that result in an S-matrix equal to the identity matrix, since the eigenvectors are orthogonal. The J-matrix is then obtained by Eq. (2.12a) which ensures that the truth table corresponds to the low energy states of the Boltzmann machines according to Eq. (2.4). A handle bit of +1 is introduced to each line of the truth table which can be biased to ensure that the complementary truth table does not appear along with the desired one. This bit also allows a truth table to be electrically reconfigured into its complement.

such that $U = [u_1\ u_2\ u_3\ u_4]$, where u_1 is the first row of the matrix shown in Fig. 2.6, $u_1 = [-1+1+1+1+1-1-1-1]^T$ and so on. In matrix notation, the S-matrix can be written as:

$$S = U^T U = 8\ I_{4\times 4} \tag{2.13}$$

Then the J-matrix becomes:

$$J = \sum_{ij} \underbrace{[S^{-1}]_{ij}}_{1/8\ \delta_{ij}} u_i u_j^\dagger = 1/8 \sum_i u_i u_i^\dagger \tag{2.14}$$

Removing the diagonal entries by making $J_{ii} = 0$ and multiplying the matrix entries

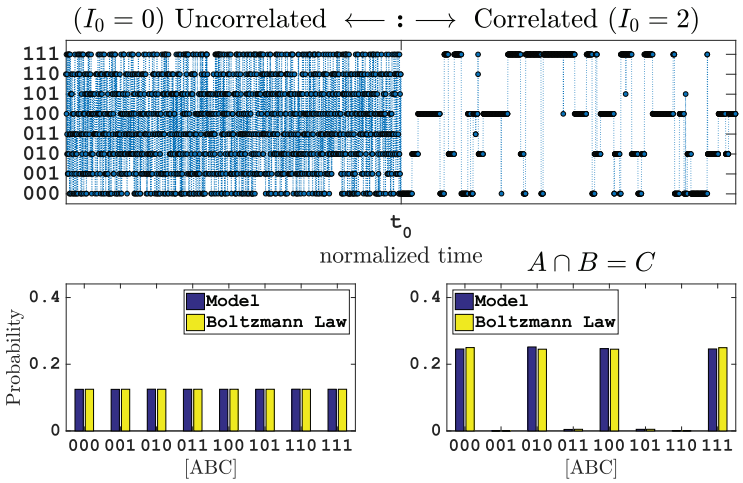


Fig. 2.7.: **Correlated p-bits, AND Gate**: When the interaction strength (I_0) is zero, p-bits produce uncorrelated noise, visiting all possible states with equal probability. In this example, the interaction strength (pseudo inverse-temperature) is suddenly increased from 0 to 2 as a step function at $t = t_0$, to effectively "quench" the network. This correlates the p-bits to produce the truth table of an AND gate (AND: $A \cap B = C$). Note that after this quenching, the p-bits only visit the low energy states corresponding to the truth table of the AND gate and once the system is in one of the low energy states, it tends to stay there for a while, until being kicked out by the thermal noise. The time averages of the uncorrelated and the correlated system are well-explained by the Boltzmann law stated in Eq. (2.4). The total simulation used a $T = 4e6$ steps to compare the results with the Boltzmann distribution, though only a fraction is shown in the upper panel for clarity.

by 2, to obtain simple integers, J_{AND} evaluates to:

$$J_{\text{AND}} = \begin{pmatrix} 0 & -1 & 0 & 0 & 1 & 1 & 1 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 & -1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & -1 & 1 & 1 & 0 \end{pmatrix} \tag{2.15}$$

with the notation, [1-5: auxiliary bit and handle bit, 6:"A", 7:"B", 8:"C"]. Following a similar procedure, we use the following 14×14 Full Adder matrix, $J_{\rm FA}$:

$$J_{\rm FA} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & -1 & -1 & -1 & -1 & -2 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & -1 & -1 & 2 & -1 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & -1 & -1 & -1 & 2 & 1 & -1 \\
0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & -1 & -2 & 1 & 1 & -1 & 1 \\
0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & -1 & 1 & -1 \\
0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & -2 & -1 & 1 \\
0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -2 & 1 & -1 & 1 \\
4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 2 & 1 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & -1 & -2 & 2 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & 1 & 2 \\
-1 & 2 & -1 & 1 & -1 & 1 & -2 & 1 & 0 & -1 & 0 & -1 & 1 & 2 \\
-1 & -1 & 2 & 1 & -1 & -2 & 1 & 1 & 0 & -1 & -1 & 0 & 1 & 2 \\
-2 & 1 & 1 & -1 & 1 & -1 & -1 & 2 & 0 & 1 & 1 & 1 & 0 & -2 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & 0 & 2 & 2 & 2 & -2 & 0
\end{pmatrix} \tag{2.16}$$

with the notation, [1−9: auxiliary bits and handle bit, 10: "$C_{\rm in}$", 11: "B", 12: "A", 13: "S" 14: "$C_{\rm out}$"].

These are the J-matrices (AND and FA) that are used for all examples in the paper, except for the AND gate described in Section 2.2. Fig. 2.10 shows the "truth table" operation of the Full Adder where all input/output terminals are "floating" using the J-matrix of Eq. (2.16), showing excellent quantitative agreement with the Boltzmann distribution of Eq. (2.4) at steady state even for the undesired peaks of the truth table.

Note that this prescription for [J] is similar to the principles developed originally for Hopfield networks ([68], and Eq. (4.20) in [39]). However, other approaches are possible along the lines described in the context of Ising Hamiltonians for quantum computers [62]. We have tried some of these other designs for [J] and many of them lead to results similar to those presented here. For practical implementations, it will be important to evaluate different approaches in terms of their demands on the dynamic range and accuracy of the weight logic.

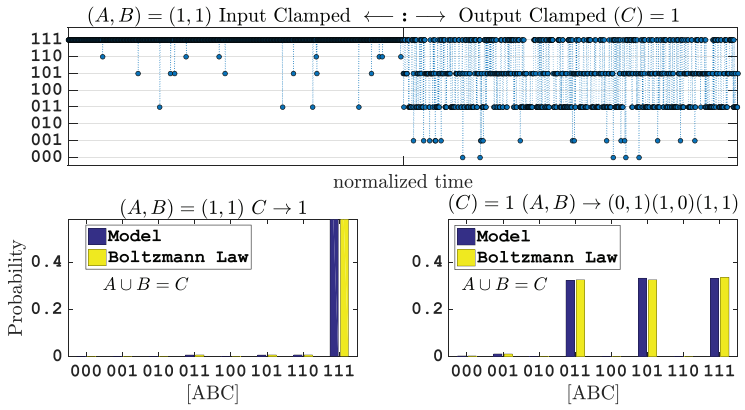


Fig. 2.8.: **Implementing a Boolean function and its inverse**: The input or output terminals of an appropriately interconnected network of p-bits can be "clamped" to perform a specific logic operation or its *inverse*. In this example, the input bits (A,B) of an OR Gate are clamped to be +1, forcing the output bit C to be 1, during the first phase of operation ($t < t_0$). In the second phase of operation ($t > t_0$), the output of the OR gate C is clamped to the value +1, which is consistent with three different combinations of (A,B). As shown in the time response and the long-time histogram plots, all three possibilities emerge with equal probability, demonstrating the "inverse" OR operation. In each case, the expected probabilities from the Boltzmann Law (Eq. (2.4)) closely match those produced by the generic model, Eq. (2.1-2.2) after running the system for one million steps, only a fraction is shown in the upper panel for clarity.

Description of universal model: Once a J-matrix and the h-vector are obtained for a given problem, the system is initialized by randomizing all m_i at time, $t = t_0$. First, the current (voltage) that a given p-bit (m_i) feels due to the other coupled m_j is obtained from Eq. (2.2), and the m_i value is updated according to Eq. (2.1). Next the procedure is repeated for the remaining p-bits by finding the current they receive due to all other m_i using the *updated* values of m_i. For this reason, the order of updating was chosen randomly in our models and we found that the order of updating has no effect in our results. However, updating the p-bits in *parallel*

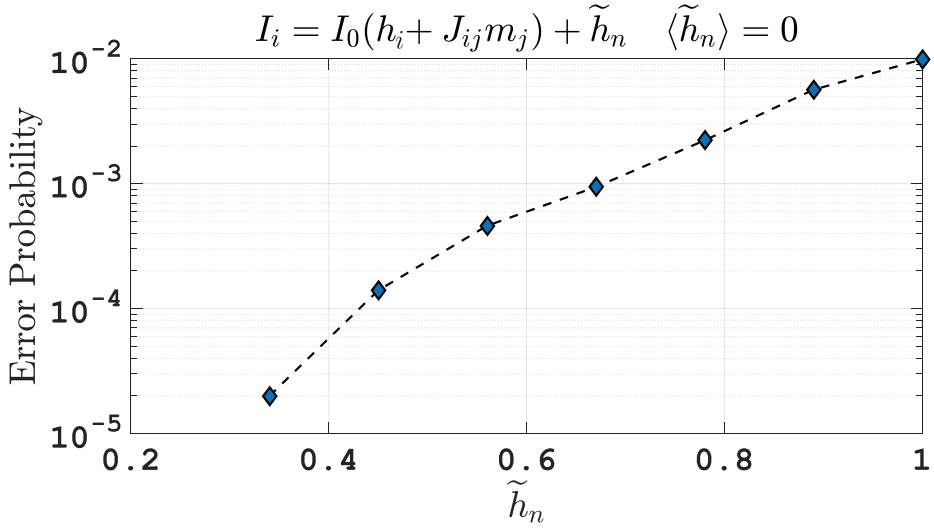


Fig. 2.9.: **Noise Tolerance of AND**: The probability of a wrong output for an (AND) gate (Eq. 2.15) operated with clamped inputs is investigated in the presence of a random noise field which enters Eq. (2.2) as indicated in the figure. The noise is assumed to be uniformly distributed over all p-bits in a given network, and centered around zero with magnitude $\pm\tilde{h}_n$, where $(I_0 = 2, h_i = \pm 1)$. Each gate is simulated 50000 times for T=100 time steps to produce an error probability for a given noise value, and the maximum peak produced by the system is assumed to be an output that can be read with certainty. The system shows robust behavior even in the presence of large levels of noise.

leads to incorrect results. These two observations are well-known in the context of Hopfield networks and Boltzmann Machines [69–71]. This type of serial updating corresponds to the "asynchronous dynamics" [39, 72]. We note that the hardware implementation discussed in this paper naturally leads to an asynchronous updating of p-bits in the absence of a global clock signal. We have set up an online simulator based on this model in Ref. [73] so that interested readers can simulate some of the examples discussed in this paper.

Fig. 2.7 shows the time evolution of an AND based on Eq. (2.15). Initially for $t < t_0$ the interaction strength is zero ($I_0 = 0$), making the pseudo-temperature of the

system infinite and the network produces uncorrelated noise visiting each state with equal probability. In the second phase ($t > t_0$), the interaction strength is suddenly increased to $I_0 = 2$, effectively "quenching" the network by reducing the temperature. This correlates the system such that only the states corresponding to the truth table of the AND gate are visited, each with equal probability when a long time average is taken. The average probabilities in each phase quantitatively match the Boltzmann Law defined by Eq. (2.4).

In Fig. 2.8, we show how a correlated network producing a given truth table can be used to do directed computation analogous to standard CMOS logic. An OR gate is constructed by using the same [J] matrix for an AND gate, but with a negated handle bit. By "clamping" the input bits of an OR gate ($t < t_0$) through their bias terminals, h_i, to (A,B)=(+1,+1), the system is forced to only one of the peaks of the truth table, effectively making C=1.

The PSL gates however exhibit a remarkable difference with standard logic gates, in that inputs and outputs are on an equal footing. Not only do clamped inputs give the corresponding output, *a clamped output gives the corresponding input(s).* In the second phase ($t > t_0$) the output of the OR gate is clamped to +1, that produces three possible peaks for the input terminals, corresponding to various possible input combinations that are consistent with the clamped output (A,B)=(0,1),(1,0) and (1,1). The probabilistic nature of PSL allows it to obtain multiple solutions (Fig. 2.8c). It also seems to make the results more resilient to *unwanted* noise due to stray fields that are inevitable in physical implementations as shown in Fig. 2.9. Here, we simulate an AND gate in the presence of a normally distributed random noise that enters the bias fields of each p-bit and define the computation to be faulty, if the mode (most frequent value) of the output bit is not consistent with the programmed input combinations after $T = 100$ time steps. We observe that even large levels of uncontrolled noise produces correct results with high probabilities.

Fig. 2.10 shows the design of a Full Adder (FA) with the 8-line truth table shown. There are three inputs in all, two from the numbers to be added, and one carry

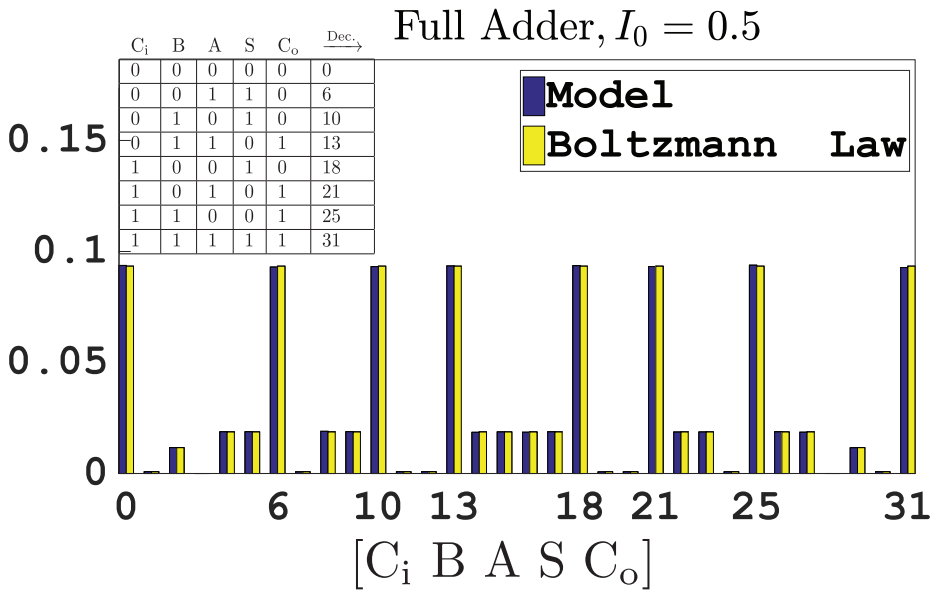


Fig. 2.10.: **Full Adder**: Full Adder in the truth table mode, where all inputs and outputs are floating, calculated using J_{FA} from Eq. (2.16), with $I_0 = 0.5$. The statistics are collected for $T = 10^6$ steps, and each terminal output is then placed in the histogram. The states are numbered using the decimal number corresponding to the binary number [C_i A B S C_o]. The decimal numbers corresponding to the truth table are shown in the inset, and these match the location of the taller peaks in the histogram. Note that the Boltzmann distribution (Eq. (2.4)) quantitatively matches the model even for the suppressed peaks.

bit from previous FA. It produces two outputs, one the sum bit and the other a carry bit to be passed on to the next FA. The probabilities of different states are calculated using J_{FA} from Eq. (2.16), with $I_0 = 0.5$ in the truth table mode, where all inputs and outputs are floating and the states are numbered using the decimal number corresponding to the binary word [C_i A B S C_o]. The decimal numbers corresponding to the truth table are shown in the inset, and these match the location of the taller peaks in the histogram. Note that the Boltzmann distribution (Eq. (2.4)) quantitatively matches the model even for the suppressed peaks. A higher I_0 reduces

these suppressed peaks further. The statistics are collected for $T = 10^6$ steps, and each terminal output is then placed in the histogram.

2.4 Directed Networks of Boltzmann Machines

When constructing larger circuits composed of individual Boltzmann machines, the reciprocal nature of the Boltzmann machine often interferes with the directed nature of computation that is desired. It seems advisable to use a hybrid approach. For example in constructing a 32-bit adder we use Full-Adders (FA) that are individually BMs with symmetric connections, $J_{ij} = J_{ji}$. But when connecting the carry bit from one FA to the next, the coupling element J_{ij} is non-zero in only one direction from the least significant to the most significant bit. This directed coupling between the components distinguishes PSL from purely reciprocal Boltzmann machines. Indeed, even the Full Adder could be implemented not as a Boltzmann machine but as a directed network of more basic gates. But then it would lose its invertibility. On the other hand, the directed connection of BM Full Adders largely preserves the invertibility of the overall system as we will show.

2.4.1 32-bit Adder/Subtractor

Fig. 2.11 shows the operation of a 32-bit adder that sums two 32-bit numbers A and B to calculate the 33-bit sum S. In the initial phase ($t < t_0$) we have $I_0 = 0$ corresponding to infinite temperature so that the sum bits (S) fluctuate among $2^{33} \approx$ 8 billion possibilities. With $I_0 = 1$, Fig. 2.11 shows that the correct answer has a probability of $\approx 12\%$ which is much lower than the $\approx 100\%$ that can be achieved with larger I_0 values (as in Fig.2.13 a-c with I_0=5). Nevertheless the peak is unmistakable as evident from the expanded scale histogram and the correct answer is extracted from the majority vote of T=100 samples as shown in Fig. 2.13. This ability to extract the correct answer despite large fluctuations is a general property of probabilistic algorithms.

Interestingly, although the overall system includes several unidirectional connections, it seems to be able to perform the inverse function as well. With A and B clamped it calculates S=A+B as noted above. Conversely with S clamped, the input bits A and B fluctuate in a correlated manner so as to make their sum sharply peaked around S. Fig. 2.11 shows the time evolution of the input bits that have broad distributions spanning a wide range. Initially, when I_0 is small, the sum of A and B also shows a broad distribution, but once I_0 is turned up to 1, the distributions of A and B get strongly correlated making the distribution of A+B sharply peaked around the fixed value of S. It must be noted that the 32-bit adder shown in Fig. 2.11 is not like standard digital circuits which are not invertible. The demonstration of such an invertible 32-bit adder could be practically significant, since binary addition is noted to be the most fundamental and frequently used operation in digital computing [74].

Delay of Ripple Carry Adder: Just as in CMOS-based Ripple Carry Adders, the delay of the p-bit based RCA is a function of the inputs A and B. In Fig. 2.12 we have systematically studied the worst-case delay of the p-bit based Ripple Carry Adder (RCA) as a function of increasing bit size. We selected a "worst-case" combination that results in a carry that needs to be propagated from bit 1 to bit N which results in a linear increase in the delay, exhibiting O(n) complexity with input size similar to CMOS implementations [75]. When the inputs are random, the delay seems to increase sub-linearly. The system is quenched at t=0 for different interaction parameters I_0 and the delay is defined to be the time it takes for the system to settle to the mode of the array for T=200. An error check has been carried out separately to ensure the calculated sum (mode) is always exactly equal to the expected sum. For random inputs the 32-bit adder is close to 20 time steps, in accordance with the example shown in Fig. 2.11.

Digital accuracy AND logical invertibility: The striking combination of accuracy and invertibility is made possible by our hybrid design, whereby the individual Full Adders are Boltzmann Machines, even though their connection is directed. Our 32-bit adder is more like a *collection of interacting particles* than like a digital circuit as

evident from Fig. 2.13a which shows a colormap of the binary state of each of the 448 p-bits as a function of time with the interaction parameter I_0 suddenly increased from 0.25 to 5 at $t_0 = 50$, thereby quenching a "molten liquid" into a "solid". Nevertheless it shows the striking accuracy of a digital circuit, with S−A−B exactly equal to zero in each of the 1000 trials as shown in Fig. 2.13b. We do not expect a "molten liquid" to be quenched into a "perfect crystal" every time. Instead, we would expect a "solid full of defects" with different non-zero values for S−A−B in each trial. That is exactly what we get if the carry bits are bidirectional as in a fully BM implementation (Fig. 2.13d).

Note however, that this digital accuracy is achieved while maintaining the property of invertibility that is absent in digital circuits. Fig. 2.13 is not for direct mode operation, but for the adder operating in reverse mode as a subtractor. It might be expected that the directed connection of carry bits from the less significant to the more significant bit could lead to a loss of invertibility. To investigate this point, we show the error S−A−B as a function of trial number (Fig. 2.14) for four different modes of operation with (i) A and B clamped (Addition), (ii) S and A clamped (Subtraction), (iii) A, B and S for the 16 most significant bits (msb) clamped, and (iv) A, B and S for the 16 least significant bits (lsb) clamped. The fully bidirectional implementation shows very large errors for all modes of operation. The directed implementation, on the other hand, works perfectly for both the adder and the subtractor modes. It also works if we clamp the least significant bits, but not if we clamp the most significant bits. This seems reasonable since we expect to be able to control a flow by making changes upstream (lsb), but not downstream (msb).

Partial directivity: So far in our examples we have only considered fully directed ($J_{ij} = 2\ J_0, J_{ji} = 0$) or fully bidirectional ($J_{ij} = J_0, J_{ji} = J_0$) carry bits when connecting the individual Full Adders. In Fig. 2.15 we systematically analyze the effects of partial directivity in the operation of a 32-bit adder. We observe that the 32-bit adder operates correctly even when there is large degree of bidirectionality ($J_{ji} = J_{ij} \times 0.75$) provided that the system is allowed to run for a long time, $T = 50000$,

in stark contrast with the fully directed case that could resolve the right answer within $T = 100$, shown in Fig. 2.14b. Decreasing the time steps systematically increases the error. Increasing the correlation parameter while keeping T constant also seems to adversely affect the bidirectional designs, that might be getting the system stuck in local minima.

Directionality and computation time, 2 p-bit model: The qualitative relation between I_0, T and bidirectionality J_{12}/J_{21} described above is derived from extensive numerical simulations based on Eq. 2.1-2.2. However, the broad features can be understood from a model involving just two p-bits, 1 and 2, with

$$h = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad J = \begin{bmatrix} 0 & J_{12} \\ J_{21} & 0 \end{bmatrix}$$

It is straightforward to write a master equation describing the time evolution of the probabilities of different configurations:

$$\frac{d}{dt} \begin{bmatrix} P_{11} \\ P_{10} \\ P_{01} \\ P_{00} \end{bmatrix} = [W] \begin{bmatrix} P_{11} \\ P_{10} \\ P_{01} \\ P_{00} \end{bmatrix}$$

W being the transition matrix [39], P_{00} representing the probability of both p-bits being -1, P_{11} both being $+1$, and so on. We can write two matrices W_1 and W_2 describing the updating of p-bits 1 and 2 respectively:

$$W_1 = \begin{array}{c} (1,2) \\ (11) \\ (10) \\ (01) \\ (00) \end{array} \begin{array}{c} \begin{array}{cccc} (11) & (10) & (01) & (00) \end{array} \\ \begin{pmatrix} p & 0 & p & 0 \\ 0 & \overline{p} & 0 & \overline{p} \\ \overline{p} & 0 & \overline{p} & 0 \\ 0 & p & 0 & p \end{pmatrix} \end{array}$$

$$W_2 = \begin{array}{c} \\ (11) \\ (10) \\ (01) \\ (00) \end{array} \begin{array}{c} \begin{array}{cccc} (1,2) & (11) & (10) & (01) & (00) \end{array} \\ \begin{pmatrix} q & q & 0 & 0 \\ \overline{q} & \overline{q} & 0 & 0 \\ 0 & 0 & \overline{q} & \overline{q} \\ 0 & 0 & q & q \end{pmatrix} \end{array}$$

where $W(i,j)$ represents the probability that state (j) makes a transition to state (i), and $\bar{p} = 1-p$, $\bar{q} = 1-q$. p and q are obtained from Eq. 2.1-2.2:

$$p = \frac{1}{2}(1 + \tanh(I_0(J_{12} + h_1))) = \frac{1}{2}(1 + \tanh(I_0 J_{12}))$$

$$q = \frac{1}{2}(1 + \tanh(I_0(J_{21} + h_2))) = \frac{1}{2}(1 + \tanh(I_0 J_{21}))$$

The overall transition matrix W is given by $W_2 \times W_1$ or $W_1 \times W_2$ depending on which bit is updated first. Either way the matrix W has four eigenvalues $\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0$ and $\lambda_4 = (2p-1)(2q-1) = \tanh(I_0 J_{12}) \times \tanh(I_0 J_{21})$ and the corresponding eigenvectors evolve with time $\sim \lambda^T$.

The components corresponding to λ=0 decay instantaneously while the eigenvector corresponding to λ=1 is the stationary result representing the correct solution. But for the system to reach this state, we have to wait for the fourth eigenvector corresponding to λ_4 to decay sufficiently. A fully directed network has J_{21} =0, so that $\lambda_4 = 0$ and the system quickly reaches the correct solution. But in a bidirectional network with $J_{12} = J_{21}$, the fourth eigenvalue can be quite close to one, especially for large I_0 and take an exponentially long time to decay, as $\lambda^T = \exp(T \ln \lambda) \approx \exp(-T(1-\lambda))$ when λ is close to 1.

This 2 p-bit model provides some insight into our general observation that directivity can be used to obtain accurate answers quickly. However, depending on the problem at hand it may be desirable to retain some degree of bidirectionality, since full directivity does lead to some loss of invertibility as seen for one set of inputs in Fig. 2.14. An example of a partially directed p-bit network is discussed in the next section.

2.4.2 4-Bit Multiplier / Factorizer

Fig. 2.16 shows how the invertibility of PSL logic blocks can be used to perform integer factorization using a multiplier in reverse. Normally, the factorization problem requires specific algorithms [76] to be performed in CMOS-like hardware, here we simply use a digital 4-bit multiplier working in *reverse* to achieve this operation.

Specifically with the output of the multiplier clamped to a given integer from 0 to 15, the input bits float to the correct factors. The interconnection strength I_0 is increased suddenly from 0 to 2 at $t = t_0$ (Fig. 2.16) and the input bits get locked to one of the possible solutions. For example, when the output is set to 9, both inputs float to 3. With the output set to 6, both inputs fluctuate between two values, 2 and 3. Note that factors like $9 = 9 \times 1$ do not show up, since encoding 9 in binary requires 4-bits (1001) and the input terminals only have 2-bits. We have checked other cases where factorizing 3 shows both 3×1 and 1×3, and factorizing zero shows all possible peaks since there are many solutions such that $0 = 0 \times 1, 2, 3$ and so on.

We also kept the same directed connections between the Full Adders for the carry bits, making them a directed network of Boltzmann Machines, similar to the 32-bit Adder. Moreover, we kept a directed connection *from* the Full Adders *to* the AND gates as shown in Fig. 2.16a since the information needs to flow from the output to the input in the case of factorization. The input bits that go to multiple AND gates are "tied" to each other with a positive exchange ($J > 0$) value much like 2-spins interacting ferromagnetically, however in PSL we envision these interactions to be controlled purely electrically. In this example, we have observed that the system is sensitive to the relative strengths of couplings within the AND gates and between the AND gates and the Full Adders which can also depend on a chosen annealing profile.

The design of factorizers of practical relevance is beyond the scope of this paper. Our main purpose has been to establish how the key feature of invertibility of p-bits can be creatively used for different circuits with unique functionalities. The demonstration of 4-bit factorization through reverse multiplication is similar to mem-

computing [77] based on deterministic memristors. Note, however, that the building blocks and operating principles of stochastic p-bits and memcomputing [78] are very different and the only similarity noted here is the fact that both approaches treat the input and output terminals on an equal footing.

2.5 Summary

It is generally believed that (1) probabilistic algorithms can tackle specific problems much more efficiently than classical algorithms [79], and that (2) probabilistic algorithms can run far more efficiently on a probabilistic computer than on a deterministic computer [79, 80]. As such, it seems reasonable to expect that probabilistic computers based on robust room temperature p-bits could provide a practically useful solution to many challenging problems by rapidly sampling the phase space in hardware.

In this paper we have presented a framework for using probabilistic units or "p-bits" as a building block for a probabilistic spin logic (PSL) which is used to implement precise Boolean logic with an accuracy comparable to standard digital circuits, while exhibiting the unique property of invertibility that is unknown in deterministic circuits. Specifically we have:

- presented an implementation based on stochastic nanomagnets to illustrate the importance of three-terminal building blocks in the construction of large scale correlated networks of p-bits. We emphasize that this is just one possible implementation that is by no means the only one **(Section 2.2)**.

- presented an algorithm for implementing Boolean gates as BM with relatively sparse and quantized J-matrix elements, benchmarked their operation against the Boltzmann law, and established their capability to perform not just direct functions but also their inverse **(Section 2.3)**, and

- presented a 32-bit adder implemented as a hybrid BM that achieves digital accuracy over a broad combination of the interaction parameter I_0, directionality and the number of samples T. This striking accuracy is reminiscent of digital circuits, but it is achieved while preserving a certain degree of invertibility which is absent in digital circuits. The accuracy is particularly surprising with high degrees of bidirectionality ($J_{12} = 0.75 \times J_{21}$) where the system is picking out the one correct answer out of nearly $2^{33} \approx 8$ billion possibilities. This may require a larger number of time samples, but these could be collected rapidly at GHz rates. **(Section 2.4)**.

We hope these findings will help emphasize a new direction for the field of spintronic and nanomagnetic logic by shifting the focus from stable high barrier magnets to stochastic, low barrier magnets, while inspiring a search for other possible physical implementations of p-bits.

Acknowledgment

It is a pleasure to acknowledge many helpful discussions with Behtash Behin-Aein (Globalfoundries) and Ernesto E. Marinero (Purdue University). We thank Jaijeet Roychowdhury (UC Berkeley) for suggesting the phrase "invertible". This work was supported in part by C-SPIN, one of six centers of STARnet, a Semiconductor Research Corporation program, sponsored by MARCO and DARPA, in part by the Nanoelectronics Research Initiative through the Institute for Nanoelectronics Discovery and Exploration (INDEX) Center, and in part by the National Science Foundation through the NCN-NEEDS program, contract 1227020-EEC.

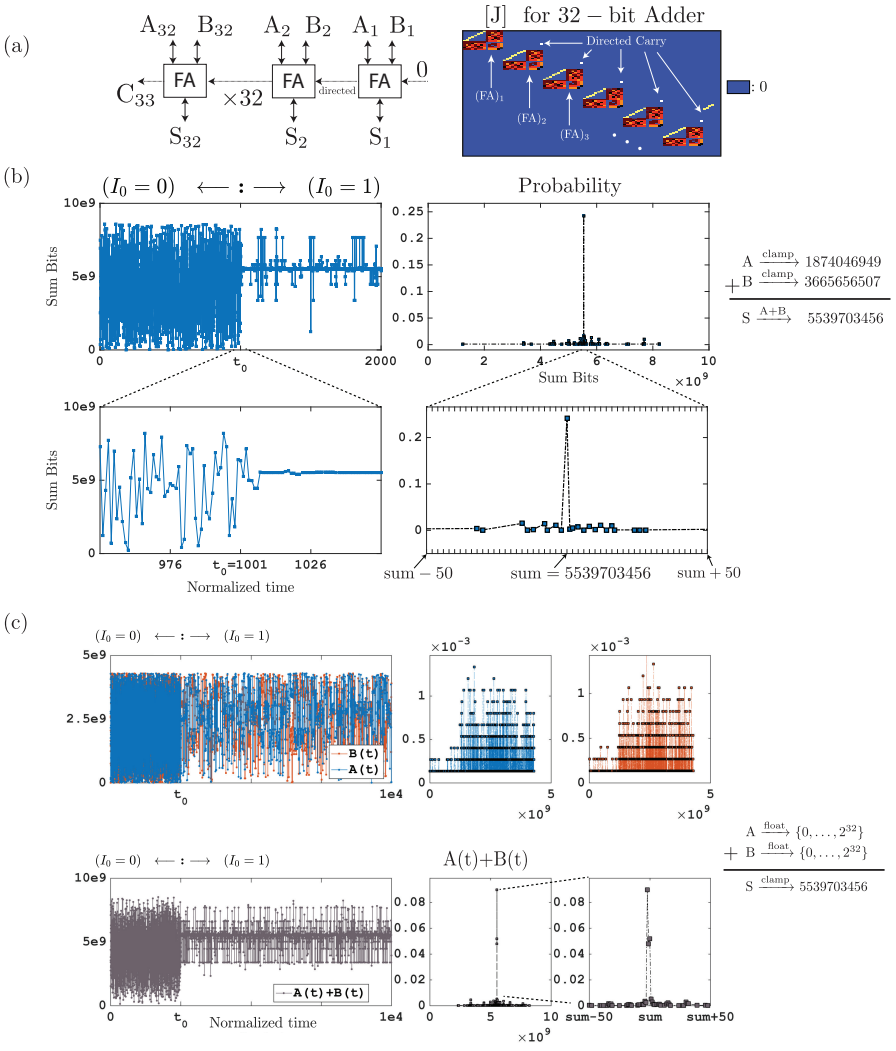


Fig. 2.11.: **32-bit Ripple Carry Adder (RCA)**: (a) A 32-bit Ripple Carry Adder (RCA) is designed using individual Full Adder (FA) units with the carry bit designed as a *directed* connection from the least significant bit to the most significant bit. The overall J-matrix for a 32-bit adder J-matrix is shown, and it is quite sparse and quantized. (b) For $t < t_0, I_0 = 0$ and the sum fluctuates randomly. At $t = t_0, I_0$ is suddenly increased, and the adder converges on the correct result for two random inputs A and B. The distribution of 1000 data points ($t > t_0$) show a single peak with 24% probability of time spent in the correct state (not including the uncorrelated time points for $t < t_0$). (c) Even though the connections between the Full Adder units are directed, the system performs the inverse function as well. When the output (S) is clamped to a fixed number, the inputs (A) and (B) fluctuate in a correlated manner to make A+B=S when $I_0 = 1$. Note the broad distributions of A and B (collected for $t > t_0$) as compared to the extremely sharp distribution of A+B.

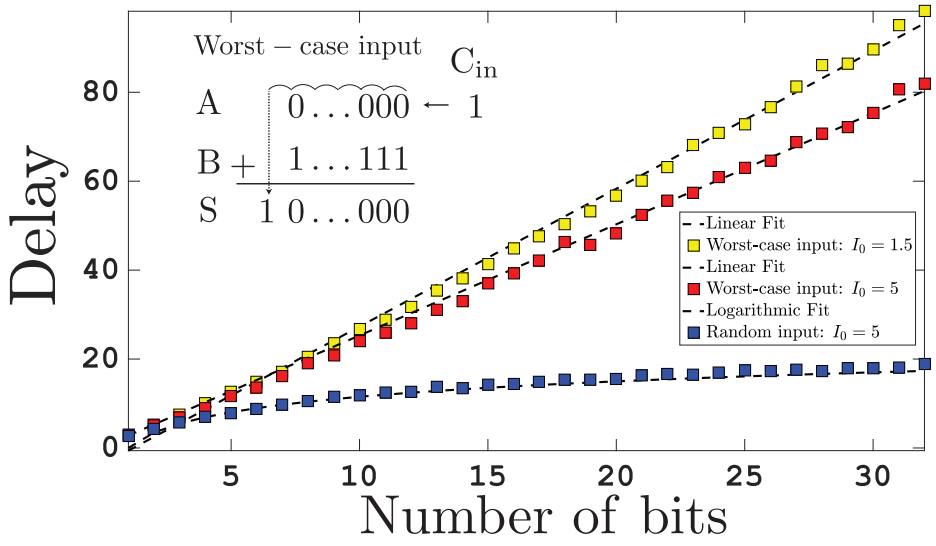


Fig. 2.12.: **Ripple Carry Adder delay**: The delay of the RCA as a function of number of bits in the Ripple Carry Adder (RCA) is shown. The worst case input combination generates a carry that propagates all the way through bit-1 to bit-N, and has a linear dependence on the number of bits, exhibiting O(n) complexity. When the inputs are random, the delay increases logarithmically. The delay is defined to be the time it takes for the network to reach the mode of the array for T=200 after getting quenched at t=0. Each point is an average of 500 trials with random initial conditions for an $I_0 = 1.5$, and the mode of the array was exactly equal to the arithmetic sum of the inputs in each case. The worst-case inputs are A=0...000 and B=1...111 with an input carry (C_{in}) of 1. Results show a weak I_0 dependence.

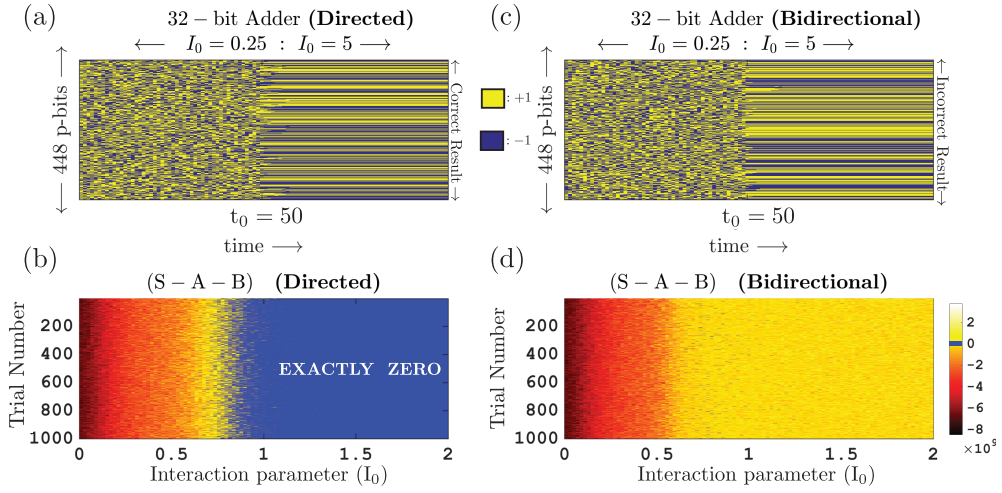


Fig. 2.13.: **Accuracy of 32-bit adder, directed versus bidirectional:** The results are shown for the adder operating in a subtractor mode, clamping one (random) 32-bit input (A) and a (random) 33-bit output (C_{out}+ S), and observing the other 32-bit input B which should provide the difference S−A. (a): Colormap of the binary state of each of the 448 p-bits comprising the directed adder as a function of time with the interaction parameter I_0 suddenly increased from 0.25 to 5 at t_0=50. For low values of I_0 at t<50, the collection of p-bits is like a molten liquid which is quenched at $t_0 = 50$ into a solid. (b) Surprisingly this solid corresponds to a "perfect crystal" in each of the 1000 trial experiments, with S−A−B exactly equal to zero (Dark blue). (c) Same as (a) but for a bidirectional adder. Here too the "liquid" quenches to a solid at $t_0 = 50$, but in this case the resulting "solid" is full of defects (with hardly any zeros), with S−A−B $\neq 0$, yielding a different wrong result for each trial as evident from (d). For (c) and (d) The colorbar is modified to have a dark blue color corresponding to exactly zero. S,A,B are taken to be the statistical mode of the 100×1 array obtained at the end of each trial.

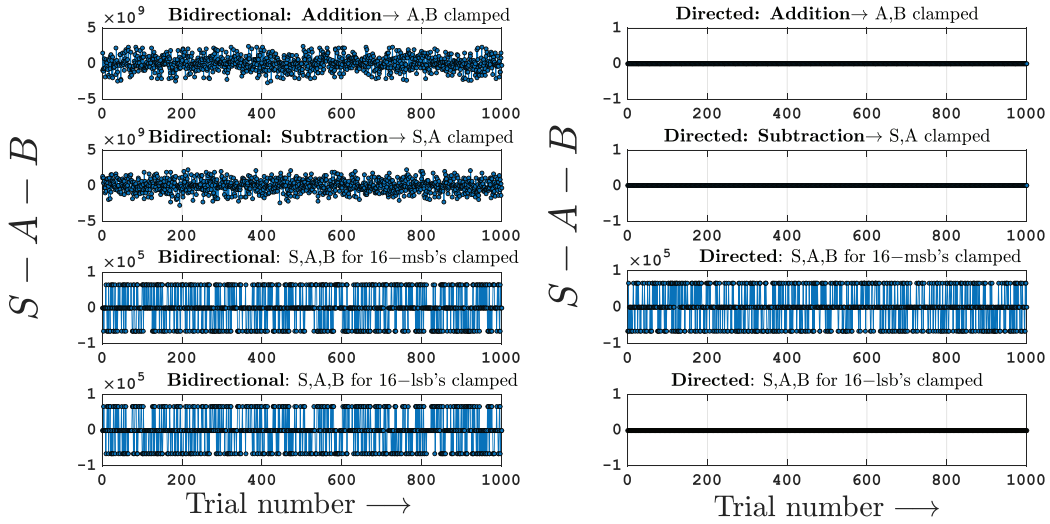


Fig. 2.14.: **Invertibility of 32-bit adder, directed vs bidirectional:** An adder that provides the sum S of two 32-bit numbers A and B: $S = A + B$. The left panel shows the adder implemented with bidirectional carry bits, while the right panel shows one with carry bits directed from the least significant to the most significant bit. Four different modes are shown with (i) A and B clamped (Addition), (ii) S and A clamped (Subtraction), (iii) A, B and S for the 16 most significant bits (msb) clamped, and (iv) A, B and S for the 16 least significant bits (lsb) clamped. Note that that bidirectional implementation shows very large errors for all modes of operation. The directed implementation works perfectly for both the adder and the subtractor modes. It also works if we clamp the least significant bits, but not if we clamp the most significant bits. Correlation parameter $I_0 = 1$, $T = 100$ steps for all trials. S,A,B are taken to be the mode (most frequent value) of the 100×1 array obtained at the end of each trial. Clamped inputs are random 32-bit words for each trial, for a total of 1000 trials.

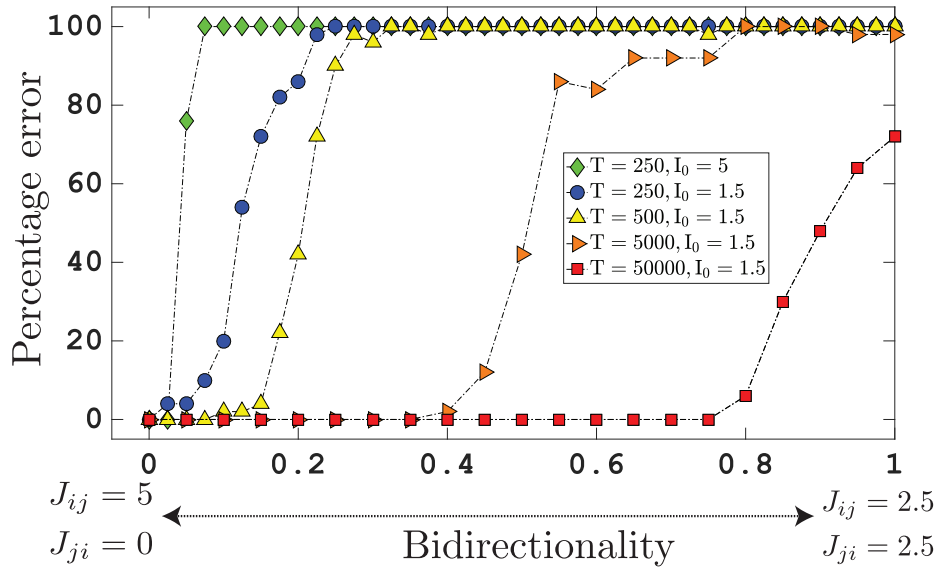


Fig. 2.15.: **Error versus bidirectionality:** The degree of bidirectionality J_{ji}/J_{ij} of the carry-out (j) to carry-in (i) link between the Full Adders is systematically varied while keeping the sum $J_{ij} + J_{ji}$ constant. In each case the sum is obtained from the statistical mode (or majority vote) of T time samples over 50 trials. The y-axis shows the fraction of trials that yield the wrong result. Note that for large I_0 and small T, error-free operation is obtained only if bidirectionality is close to zero similar to standard digital circuits. But with $I_0 = 1.5$ and T=50,000, error-free operation (at least for 50 trials) is obtained even with $\approx 75\%$ bidirectionality.

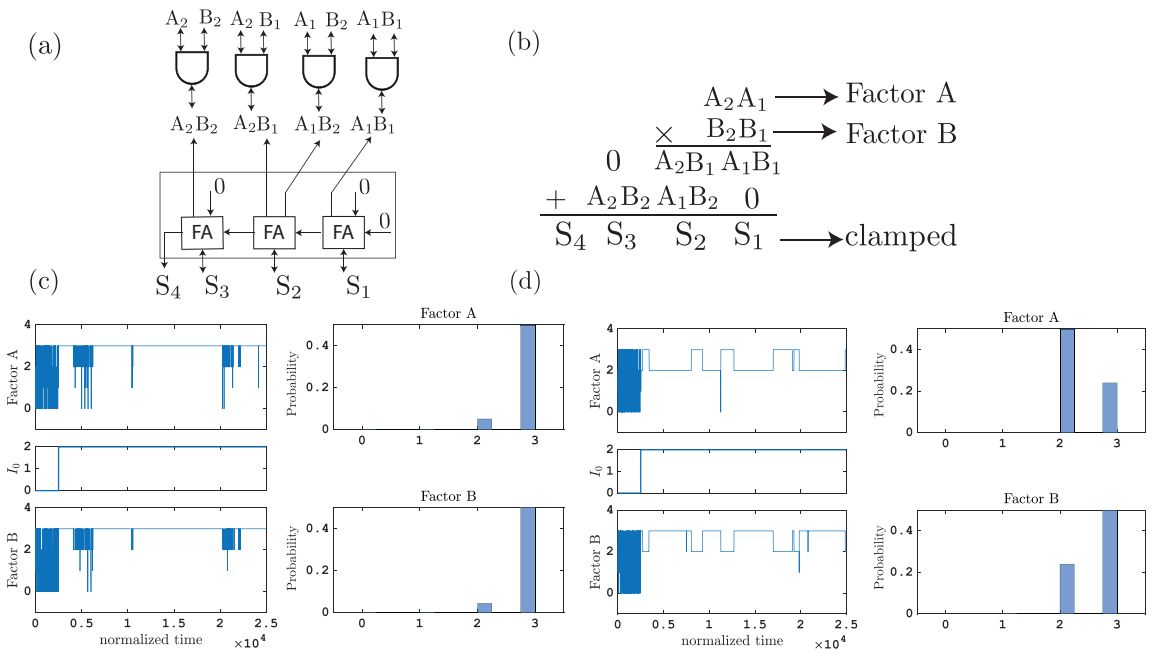


Fig. 2.16.: **Factorization through inverse multiplication**: The reversibility of PSL allows the operation of integer factorization using a binary multiplication circuit implemented using the principles of digital logic using AND gates and Full Adders (FA) as shown in (a). The output nodes of a 4-bit multiplier are clamped to a given integer, and the system produces the only consistent factors of the product at the input terminals, probabilistically. The interaction parameter I_0 is suddenly increased to a saturation value of 2, and held constant as shown. (b) The output terminal is clamped to 9 and is factored into 3×3, note that 9×1 is not an achievable solution in this setup since encoding 9 requires 4-bit inputs in binary, whereas inputs are limited to 2-bits. (c) The output terminal is clamped to 6 and after being correlated, the factors cross-oscillate between 2 and 3. In both cases the histogram is obtained by counting outputs after $t > t_{\text{total}}/2 = 1.25 \times 10^4$ time steps to collect statistics after the system is thermalized.

3. LOW BARRIER NANOMAGNETS AS P-BITS FOR SPIN LOGIC

Materials in this chapter have been extracted verbatim from the paper: " Low Barrier Nanomagnets as p-bits for Spin Logic", R. Faria, K. Y. Camsari, and S. Datta, published in *IEEE Magnetics Letters*, 2017. Reprinted with permision from [12].

It has recently been shown that a suitably interconnected network of tunable telegraphic noise generators or "p-bits" can be used to perform even precise arithmetic functions like a 32-bit adder. In this paper we use simulations based on the stochastic Landau-Lifshitz-Gilbert (sLLG) equation to demonstrate that similar impressive functions can be performed using unstable nanomagnets with energy barriers as low as a fraction of a kT. This is surprising since the magnetization of low barrier nanomagnets is not telegraphic with discrete values of ± 1. Rather it fluctuates randomly among all values between -1 and $+1$, and the output magnets are read with a thresholding device that translates all positive values to 1 and all negative values to zero. We present sLLG-based simulations demonstrating the operation of a 32-bit adder with a network of several hundred nanomagnets, exhibiting a remarkably precise correlation: The input magnets {A} and {B} as well as the output magnets {S} all fluctuate randomly and yet the quantity A+B−S is sharply peaked around zero! If we fix {A} and {B}, the sum magnets {S} rapidly converge to a unique state with S=A+B so that the system acts as an adder. But unlike standard adders, the operation is invertible. If we fix {S} and {B}, the remaining magnets {A} converge to the difference A=S−B. These examples emphasize a new direction for the field of nanomagnetics away from stable high barrier magnets towards stochastic low barrier magnets which not only operate with lower currents, but are also more promising for continued downscaling. **Index Terms:** Spintronic memory and logic, nanomagnetics, Landau-Lifshitz-Gilbert equation, arithmetic functions.

3.1 Introduction

The developments in spintronics and nanomagnetics are having enormous influence on the field of storage and memory devices and it has been shown that the WRITE (W) and READ (R) elements can also be integrated into units that implement Boolean as well as non-Boolean logic [1–3, 32, 33, 48, 81, 82]. These applications, however, usually make use of stable magnets with energy barriers $\sim$ 40 kT which require relatively large currents for their operation. The critical spin current needed to switch a magnet with a thermal energy barrier of $\Delta = H_K M_s V/2$ is given by [83]

$$I_c = I_{c0}\frac{\Delta}{kT} \qquad I_{c0} = \frac{4q\alpha}{\hbar}kT\left(1 + f_I\,\frac{H_d}{2H_K}\right) \tag{3.1}$$

where q is the electronic charge, M_s is the saturation magnetization, H_K is the anisotropy field, H_d is the demagnetization field, V is the volume, α is the Gilbert damping coefficient and the factor f_I is equal to zero for perpendicular anisotropy magnets (PMA) and one for inplane anisotropy magnets (IMA). With $\Delta \sim 40$ kT and $\alpha \sim 0.01$, the critical switching spin current for a PMA magnet is $4q\alpha\Delta/\hbar \approx 10\ \mu A$. Magnets with lower barriers could operate with lower currents but their application in conventional memory or logic is severely limited due to their stochastic nature. However, their possible use in unconventional applications has been discussed both theoretically and experimentally [15–23]. The implementation of logic operations based on an ensemble average over stable nanomagnets has been explored in [29, 30, 84] while [11] describes an approach to the traveling salesman problem based on a time average over unstable nanomagnets that cycle through millions of collective correlated states potentially at GHz rates. Note that for such nanomagnets ($\Delta \ll 25$ kT [24]), the Arrhenius model that predicts a telegraphic change between two magnetizations is no longer applicable, and the magnetization becomes a continuous variable. The present paper describes the application of the latter approach (time average) to implement precise Boolean logic operations like a 32-bit adder that provides the sum S for given inputs A and B. Remarkably the adder also evaluates the inverse function, cycling through all combinations of A and B that add up to a given sum S. We have

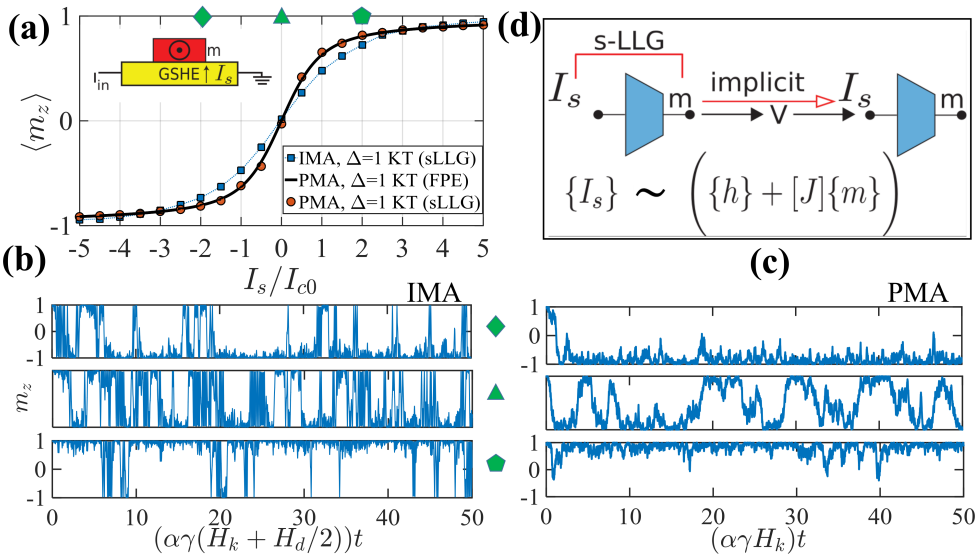


Fig. 3.1.: **Low-barrier stochastic Nanomagnet as a p-bit:** (a) Time-averaged magnetization of low barrier IMA and PMA magnets ($\Delta = 1$ kT, $\mathrm{H_K} = 60$ mT, $\alpha = 0.01$, $\mathrm{H_d} = 1.5$ T for IMA) as a function of the bias spin current which is normalized to I_{c0} (Eq. 3.1). Average magnetization of PMA magnets obtained from sLLG which agrees well with the analytical solution from the FPE, Eq. 3.6. Inset shows a physical structure using a giant spin Hall effect (GSHE) material that could be used to convert a charge current into a spin current with the correct polarization to bias an IMA. (b) The magnetization m(t) for IMA as a function of time for three different bias currents obtained from a numerical solution of sLLG equation. (c) Same plot for PMA with the same barrier height. Note that the fluctuations are much faster and more telegraphic for IMA than for PMA. (d) A connection scheme for two p-bits is shown where the magnetization of a p-bit is implicitly converted into the bias current/voltage for the next p-bit (Eq. 3.2). A possible hardware implementation to turn the magnetization m into a voltage V, could combine a GSHE layer with MTJs as in [2], replacing the stable write magnets by low barrier nanomagnets that are discussed here.

recently shown [85] that a suitably interconnected network of tunable telegraphic noise generators or telegraphic "p-bits" can be used to perform even precise arith-

metic functions like a 32-bit adder. However, it is not clear whether such p-bits can be implemented with real physical systems, especially if the noise in these systems are not telegraphic but continuous. The objective of this paper is to demonstrate that p-bits can be implemented using unstable nanomagnets with energy barriers as low as a fraction of a kT, *even though their magnetization is not telegraphic* and fluctuate among all values from -1 to $+1$. We assume that the magnets can be read with a thresholding device that translates all positive values to $+1$ and all negative values to zero. But this thresholding is applied *only* to the output nodes when we need to read a magnet at the end of an operation and not to the internal nodes or during device operation.

We start in *Section 2* by showing that low barrier magnets, both PMA and IMA, exhibit the key property of p-bits, namely that they act as electrically tunable random number generators (RNG). Their magnetization $m(t)$ fluctuates randomly in time, and the time-averaged $\langle m \rangle$ can be tuned from -1 to $+1$ with a spin current. IMA magnets require a larger current to tune, but this is offset by a more rapid fluctuation rate, allowing a faster evaluation of the time average, and hence faster operation (Fig. 3.1). Note also that the PMA magnetization is relatively continuous compared to IMA magnetization which is more telegraphic in nature.

To harness either for logic applications, they have to be interconnected such that the spin current I_{sk} driving magnet 'k' has to be derived from the magnetization of other magnets.

$$\frac{2I_{sk}}{I_{c0}} = -I_0\left(h_k + \sum_j J_{kj} m_j\right) \tag{3.2}$$

where I_{c0} is normalization constant defined as the critical current (Eq. 3.1) for a magnet with a barrier $\Delta = 1$ kT and I_0 determines the overall strength of the interconnections. The bias $\{h\}$ and interconnection $[J]$ matrices have to be designed appropriately in implementing specific operations. We will not go into the implementation of these matrices since there are many options requiring careful discussion [32], [33], [31], [34]. We will assume that a network of stochastic nanomagnets (PMA and IMA) has been interconnected according to Eq. 3.2 and simulate their be-

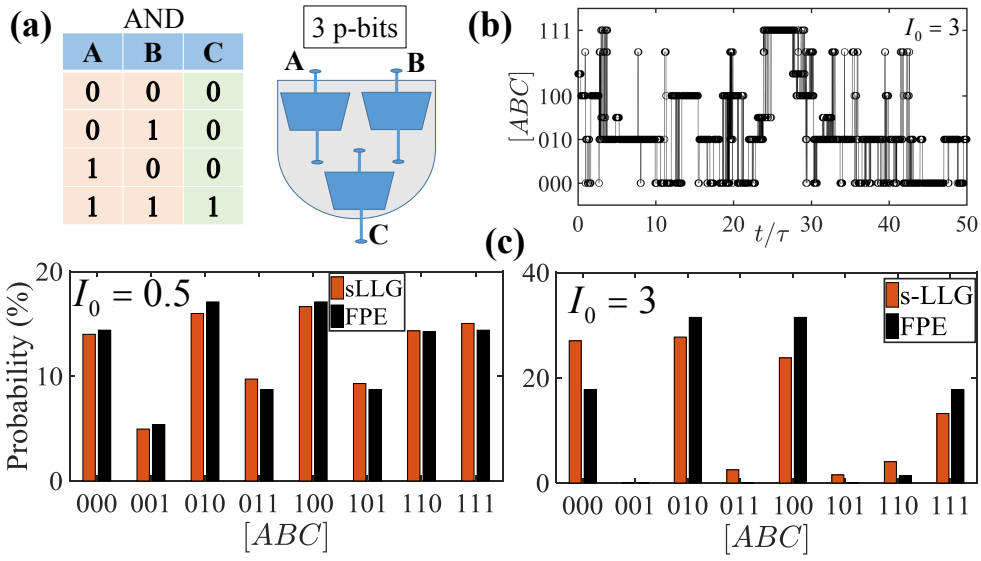


Fig. 3.2.: **Implementation of a basic boolean element (AND) using p-bits:** (a) The truth table for AND is shown along with a schematic for the network of three p-bits used to perform the operation. The p-bits are connected symmetrically with $J_{ij} = J_{ji}$. (b) The decimal value of each configuration of the input-output nodes at each time step (normalized by the factor $\tau = (\alpha\gamma(H_k + H_d/2))^{-1}$) is calculated according to $A \times 2^2 + B \times 2^1 + C \times 2^0$ where A, B and C are thresholded to obtain binary values (0,1) at the read out. (c) Histograms of the different configurations of the p-bits are shown for a weaker ($I_0 = 0.5$) and stronger ($I_0 = 3$) correlation strength. Note the close match between the numerical values obtained from the sLLG equation with the probabilities obtained analytically from the FPE result in Eq. 3.7 which is related to the Boltzmann law, especially for I_0=0.5. For higher values of I_0 the numerical results tend to be stuck in metastable states requiring longer simulation times to converge to the steady-state FPE result.

havior using the stochastic Landau-Lifshitz-Gilbert (sLLG) equation to demonstrate useful functionalities. We assume that the currents specified by Eq. 3.2 are applied to each magnet on a time scale that is much shorter than the magnet dynamics, and new features could arise if delays associated with these interconnections are comparable

to magnet dynamics. These issues are beyond the scope of this paper. All numerical examples are presented for IMA with parameters shown in Fig. 3.1 but similar results are obtained with PMA as well.

In *Section 3* we describe how simple logic gates can be implemented by suitably designing the $\{h\}$ and $[J]$ matrices so that the magnet configurations corresponding to the desired truth table represent 'low energy' states where the network spends most of its time according to the Boltzmann law of equilibrium probabilities: $P(\{m\}) \sim \exp(-E(\{m\})/kT)$. Although the use of spin currents does not in general permit us to write an energy functional [86], for symmetrically interconnected PMA magnets we can use a functional of the form [11, 87]:

$$-\frac{E(\{m\})}{kT} = \sum_i \frac{\Delta_i}{kT}\, m_i^2 + I_0\left(\sum_i h_i m_i + \frac{1}{2}\sum_{i,j} J_{ij} m_i m_j\right) \tag{3.3}$$

to describe the network of interconnected magnets. This can be seen by noting that from the Boltzmann law and Eq. 3.3

$$\frac{\partial\, lnP}{\partial\, m_k} = 2\frac{\Delta_k}{kT} m_k + I_0\left(h_k + \frac{1}{2}\sum_j (J_{kj} + J_{jk}) m_j\right)$$

so that for a symmetric $[J]$ matrix, from Eq. 3.2

$$P(m_k) \sim \exp\left(\frac{\Delta_k}{kT} m_k^2 - \frac{2I_{sk}}{I_{c0}} m_k\right) \tag{3.4}$$

which is exactly the steady-state condition for magnet 'k' that we would obtain from the Fokker-Planck equation (FPE) ([58] Eq. (3.9)) for PMA. Moreover, our "empirical" results show that the energy functional shows good agreement even when magnets have an additional shape anisotropy. Note that even though I_{c0} is size-independent, the distribution of the nanomagnet depends on size through Δ: for higher Δ magnets, more spin current is required to pin the magnetization. We will refer to Eq. 3.4 as the FPE probability.

The probability distributions obtained from the numerical solution of the sLLG equation for both PMA and IMA magnets follow the FPE result quite well (Fig. 2). The highest probabilities correspond to the lowest energy states, which correspond to the desired truth table relating the input magnets A and B to the output magnet C. If

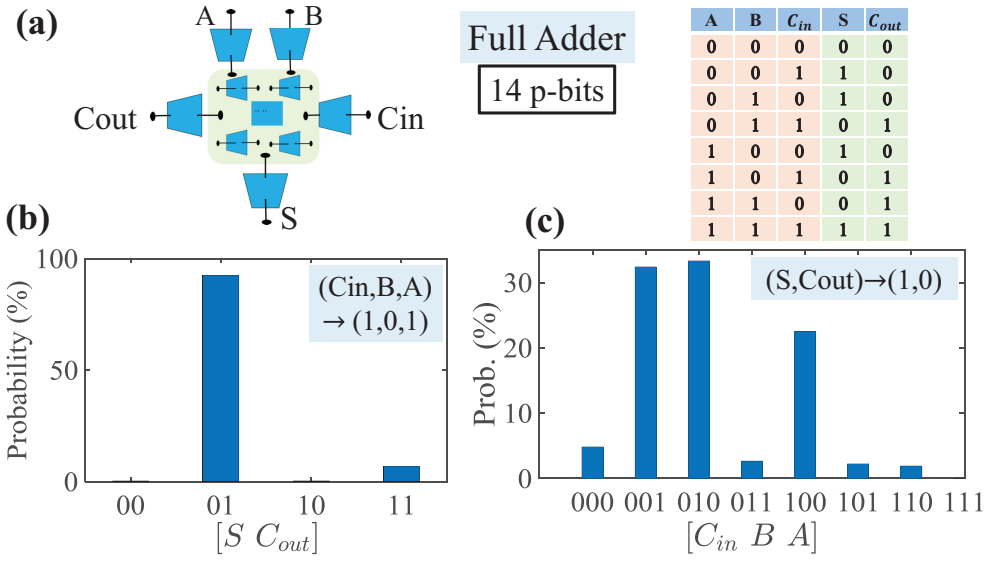


Fig. 3.3.: **Full Adder**: (a) A full adder (truth table shown) implemented by connecting 14 p-bits symmetrically. (b) In forward mode, when the inputs (A, B, C_{in}) are clamped, the adder gives the correct output (S *and* C_{out}). (c) Unlike standard logic, these gates *are invertible*: If the output nodes of the adder are clamped to fixed values, the adder gives all possible input combinations satisfying the output constraint.

we force the inputs A and B to specific values by using appropriate values for h_A and h_B, C would take on the specific value required by the truth table, just like standard digital gates. But unlike standard gates, these gates are invertible, similar to those discussed in the context of memcomputing [78]. They can be operated in reverse: if we clamp the output C to a specific value, the inputs A and B will spend most of its time in those configurations $\{AB\}$ that produce that output. We also illustrate this reversible operation with a more complex logic gate, namely a full adder treating it as a Boltzmann machine (BM) and using the same principle of energy minimization to design the $\{h\}$ and $[J]$ matrices.

Finally in *Section 4* we demonstrate the operation of a 32-bit adder obtained from 31 full adders and one half adder with the output carry from each bit connected to

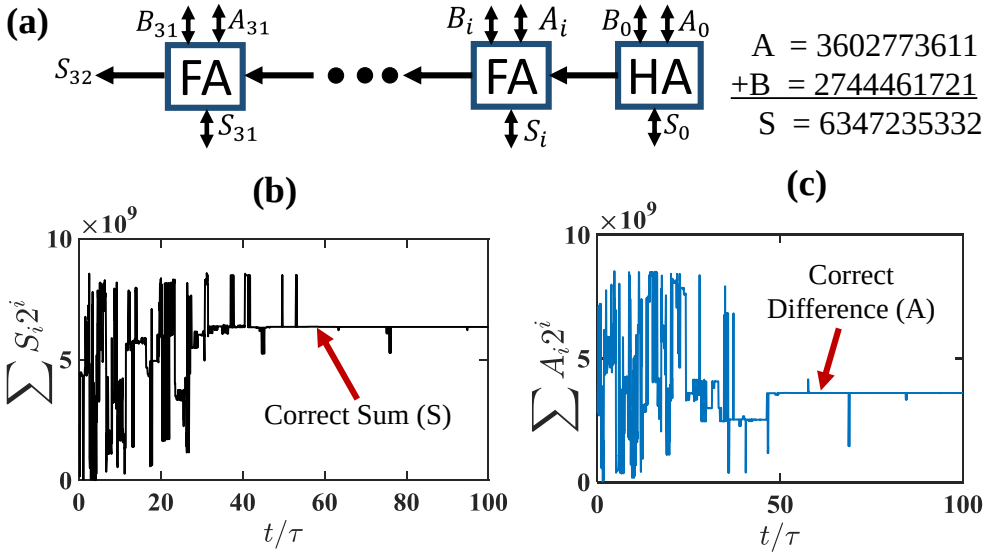


Fig. 3.4.: **32-bit Adder/ Subtractor:** (a) Schematic of an adder constructed from 31 full adders (from Fig. 3.3) and one half adder (composed of 6 p-bits) with the carry out bit C_{out} from each adder communicated in a directed fashion to the carry in bit C_{in} of the next adder. (b) Time evolution of the sum $S = \sum_i S_i 2^i$ obtained from the sum bits {S} as the coupling strength I_0 is ramped up starting from zero. Note that in a time $\sim 60\ \tau$ (τ is defined in Fig. 2), the sum converges (with occasional jumps) to the correct value which represents one out of $2^{33} \sim 8$ billion possibilities. (c) Although the individual adders are connected in a directed fashion through the carry bits, the overall 32-bit adder performs the inverse function as well. If the sum bits {S} are clamped along with one set of input (B), the other input converges rapidly to the correct difference (A).

the carry in of the next higher bit through the appropriate element of the overall $[J]$ matrix. Note that these are unidirectional connections so that the overall $[J]$ matrix is not symmetric, though the $[J]$ matrix for each full adder is symmetric. We show that this network of nearly five hundred nanomagnets exhibits a remarkably precise correlation that provides the exact sum S of any two given inputs, A and B (Fig.4). What is even more remarkable is that if we do not fix either the inputs or

the outputs, the quantities A, B and S all fluctuate randomly and yet the quantity A+B−S is sharply peaked around zero, so that the network can be used to extract either A, B or S, if the other two are fixed, which is similar to the NP-complete "subset sum" problem (Fig.5) [88, 89].

3.2 Stochastic nanomagnet model

Fig.1(b,c) shows the time response of the magnetization m_z along the easy axis calculated using the sLLG equation (integrated by Heun's method within the Stratonovich calculus [90]) with $\Delta t = 0.95\ ps$ for IMA and $\Delta t = 11.8\ ps$ for PMA.

$$(1+\alpha^2)\frac{d\hat{m}_i}{dt} = -|\gamma|\hat{m}_i \times \vec{H}_i - \alpha|\gamma|(\hat{m}_i \times \hat{m}_i \times \vec{H}_i) + \frac{1}{qN_i}(\hat{m}_i \times \vec{I}_{Si} \times \hat{m}_i) + \left(\frac{\alpha}{qN_i}(\hat{m}_i \times \vec{I}_{Si})\right) \tag{3.5a}$$

where H_i is the effective field including the uniaxial and shape anisotropy terms, as well as the thermally fluctuating magnetic field due to three dimensional uncorrelated thermal noise H_n having Gaussian distribution with mean $\langle H_n \rangle = 0$ and standard deviation $\langle H_n^2 \rangle = 2\alpha \mathrm{kT}/|\gamma|\mathrm{M_s V}$ along each direction [90–94], γ is the gyromagnetic ratio and $N_i = M_s V/\mu_B$ is the total number of Bohr magnetons comprising the magnet. Our simulations are based on the macrospin approximation, as is common in the literature [24, 95, 96]. This approximation may not be adequate for larger magnets with multiple domains, but is expected to work better as the magnets are scaled down. The time-averaged magnetization (Fig. 1a) obtained from the sLLG equation for PMA magnets is in good agreement with that obtained analytically by averaging over the FPE result (Eq. 3.4):

$$\langle m \rangle = \int_{-1}^{+1} dm\ m\ P(m) \Big/ \int_{-1}^{+1} dm\ P(m) \tag{3.6}$$

3.3 Basic Boolean Gates

In implementing any given truth table we need the $\{h\}$ and $[J]$ matrices that make the truth table correspond to the lowest energy states of the energy functional given in Eq. 3.3. The choice of these matrices is not unique and [62] provides a suitable

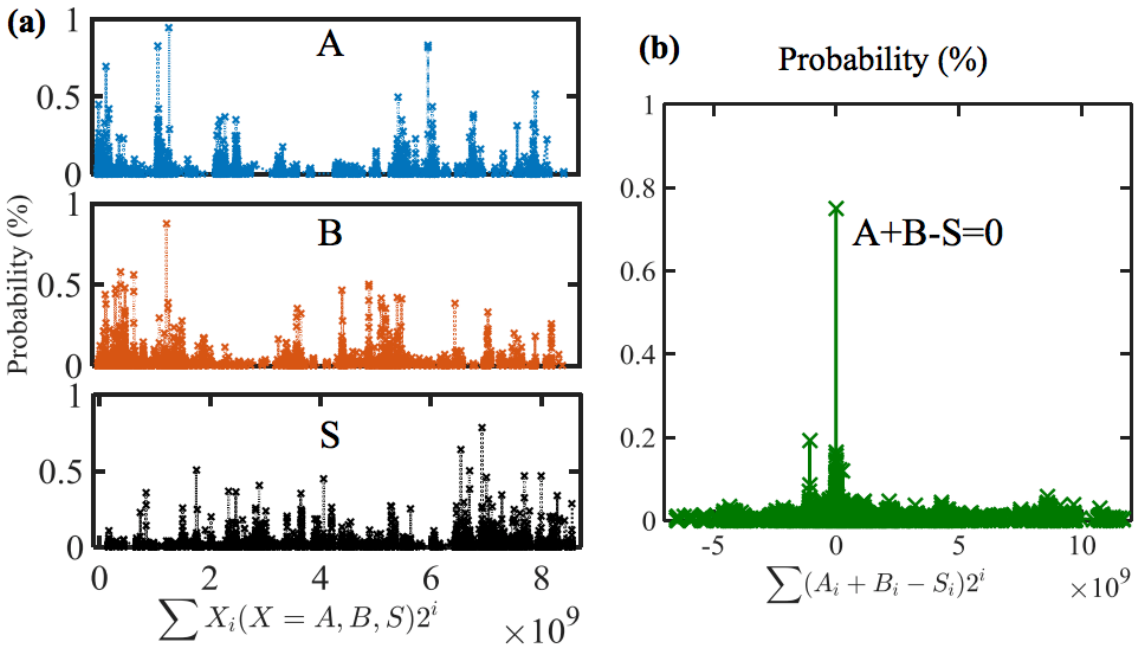


Fig. 3.5.: **Correlated Adder**: A remarkable property of the adder (in Fig. 3.4) is that it works even when the inputs (A,B) and the output (S) are not unique and fluctuate in time amongst many allowed values as shown in (a). Nevertheless, the quantity A+B-S is sharply peaked at zero (b), demonstrating the correlation of hundreds of nanomagnets consistent with the addition function A+B−S=0.

set for AND, OR gates along with many other functions. Fig. 2a shows one possible implementation of an AND gate using a network of three nanomagnets, representing A,B and C.

The magnetization of the magnets A, B and C fluctuates continuously between −1 and +1 and are mapped into the binary values of 0 and 1 by a thresholding operation: all negative values map to zero, while positive values map to +1. The y-axis in Fig. 2b shows the resulting binary number $\{ABC\}$ converted into a single number $A \times 2^2 + B \times 2^1 + C \times 2^0$. Note how the values on the y-axis are clustered around 0, 2, 4 and 7 which correspond to the lines of the truth table shown in Fig. 3.2a. Occasionally the system jumps to other values but it quickly returns to one of these preferred values.

This clustering is reflected in the histogram constructed from 678 normalized time steps (Fig. 3.2c) which shows peaks around the preferred states defined by the truth table. This agrees well with the probability plot constructed from the FPE result in Eq. 3.4 noting that we can label the thresholded states as $m_i = s_i m$ where $s_i = \pm 1$ and $0 < m < 1$ so that from Eq. 3.3:

$$E(\{s\}, m) = \Big(\sum_i \frac{\Delta_i}{kT} + \frac{1}{2} I_0 \sum_{i,j} J_{ij} s_i s_j \Big) m^2 + \Big(I_0 \sum_i h_i s_i \Big) m$$

$$P(\{s\}) \sim \int_0^1 dm \ \exp\big(- E(\{s\}, m) \big) \tag{3.7}$$

The peaks corresponding to the preferred states in Fig. 2c do not have equal probability, even at steady state as predicted by Eq. (3.7). This skew is due to the continuous nature of magnetization with small Δ magnets that affect the thresholded results.

Note that the probabilities are strongly affected by the choice of I_0 as we might expect from the exponential dependence of the Boltzmann function. If we use a much smaller value of I_0 we obtain a uniform probability across all eight states as we would expect for three uncorrelated magnets. If we use a much larger value of I_0 the Boltzmann law predicts all states with equal energy to be equally occupied, but in a numerical simulation, the system tends to get stuck for long periods in one of the preferred states, instead of moving freely among them.

Consider now a full adder having three inputs A, B, C_{in} and two outputs S, C_{out}, S being the sum bit, and C_{in}, C_{out} being the incoming carry and the outgoing carry bits. Fig. 3 shows a full adder constructed out of 14 p-bits treating it as a BM with a symmetric J-matrix [1] which is obtained by a suitable extension of the principles developed in the context of Hopfield networks ([39], Eq. 4.20) and extended in [85]. This design not only gives the correct output for a given input, *but also the correct set of inputs for a given output.*

[1]The design of [J] matrices has been discussed in [39] and [4] and are assumed not to change during operation.

3.4 32-Bit Adder/Subtractor

Finally we demonstrate the operation of a 32-bit adder obtained from 31 full adders and one half adder with a single directed connection from the C_{out} of one bit to the C_{in} of the next bit, in accordance with the standard design of ripple carry adders (RCA). Here, we treat the RCA as a standalone block without any peripheral read-out circuitry to simply demonstrate how the nanomagnet network can operate as a directed combinational logic unit. If we provide two input numbers A and B, and look at the sum S, which includes all the sum bits along with the carry-out from the last bit, $C_{out}(32)$ we find numerically that the system relaxes to the correct sum with occasional jumps from the correct state. It is really quite surprising that a network of $14 \times 31 + 6 = 440$ nanomagnets fluctuating continuously over the range $-1 < m < +1$ get correlated precisely enough to point to the correct answer out of $2^{33} \approx 8$ *billion* possibilities without getting stuck in metastable states [85]. Interestingly it also works as a subtractor: if we fix the sum and one of the inputs B, theremaining input gives the correct difference $A = S - B$ (Fig. 4). Even more surprisingly, the overall system seems to act like a BM when all magnets are allowed to fluctuate. Each set of magnets A, B and S fluctuates randomly over a wide range of values. But the quantity A+B−S shows a sharp peak around zero (Fig. 5), showing that the interconnected network reflects the desired truth table.

Acknowledgment

The authors gratefully acknowledge many helpful discussions with Behtash Behin-Aein, Vinh Quang Diep, and with Ernesto E. Marinero. This work was supported in part by C-SPIN, one of six centers of STARnet, a Semiconductor Research Corporation program, sponsored by MARCO and DARPA, in part by the Nanoelectronics Research Initiative through the Institute for Nanoelectronics Discovery and Exploration (INDEX) Center, and in part by the National Science Foundation through the NCN-NEEDS program, contract 1227020-EEC.

4. IMPLEMENTING BAYESIAN NETWORKS WITH EMBEDDED STOCHASTIC MRAM

Materials in this chapter have been extracted verbatim from the paper: "Implementing Bayesian networks with embedded stochastic MRAM", R. Faria, K. Y. Camsari, and S. Datta, publised in AIP Advances, 2018. Reprinted with permision from [97].

Magnetic tunnel junctions (MTJ's) with low barrier magnets have been used to implement random number generators (RNG's) and it has recently been shown that such an MTJ connected to the drain of a conventional transistor provides a three-terminal tunable RNG or a p-bit. In this letter we show how this p-bit can be used to build a p-circuit that emulates a Bayesian network (BN), such that the correlations in real world variables can be obtained from electrical measurements on the corresponding circuit nodes. The p-circuit design proceeds in two steps: the BN is first translated into a behavioral model, called Probabilistic Spin Logic (PSL), defined by dimensionless biasing (h) and interconnection (J) coefficients, which are then translated into electronic circuit elements. As a benchmark example, we mimic a family tree of three generations and show that the genetic relatedness calculated from a SPICE-compatible circuit simulator matches well-known results.

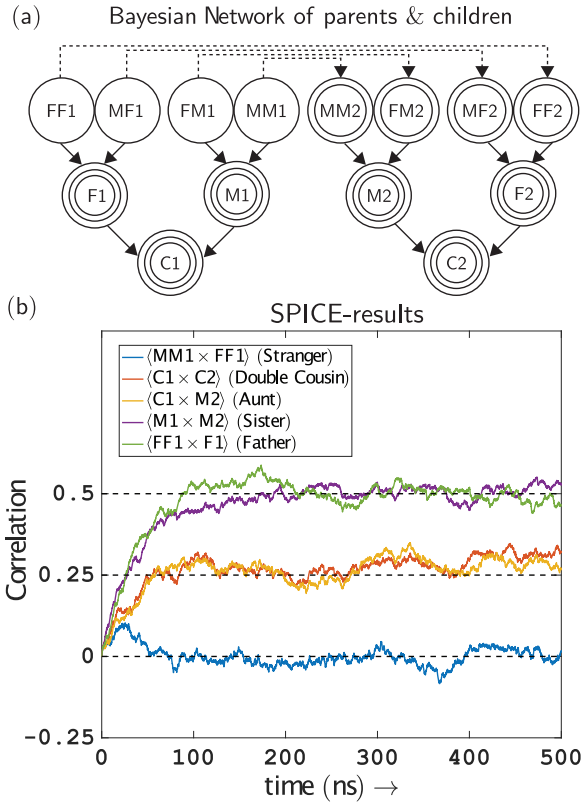
Magnetic tunnel junctions (MTJ's) with low barrier magnets have been used to implement random number generators (RNG's) [18,25,98,99] and it has recently been shown that such an MTJ connected to the drain of a conventional transistor provides a three-terminal tunable RNG or a *p-bit* [7] with applications to optimization [100] and an enhanced type of Boolean logic, that is invertible [4,101–103]. In this paper we show how this p-bit can be used to build a p-circuit that emulates a Bayesian network (BN) [104] defined in terms of conditional probability tables (CPT) that describe how each *child node* is influenced by its *parent nodes*. BN's are widely used to understand causal relationships in real world problems such as forecasting, diagnosis, automated

vision, manufacturing control and so on [105]. For deep and complicated networks where each child node has many parent nodes, the computation of the joint probability becomes impractical [106] and different hardware implementations of BN's have been proposed [10, 107–114].

In this letter we present a systematic approach for translating a BN into an electronic circuit such that the stochastic node voltages mimic the real world variables whose correlations can be obtained from electrical measurements on the corresponding circuit nodes. The proposed electronic circuit and the hardware building blocks are based on present day Magnetoresistive Random Access Memory (MRAM) technology whose MTJs are built out of thermally unstable nanomagnets [7] (Stochastic MRAM), obviating the need for the development of a new device.

As a benchmark example, consider a BN (Fig. 4.1) consisting of three generations of a family, where each child (C) inherits half the genes from the father (F) and the other half from the mother (M), so that $C = 0.5F + 0.5M$, where C, F and M can each be viewed as a bipolar random variable: (-1,$+1$). A well-known concept in genetics is that of *relatedness*. For example, the relatedness $\langle C1 \times C2 \rangle$ of two siblings, with the same parents is 50% : $\langle C1 \times C2 \rangle = .25\big(\langle F \times F \rangle + \langle F \times M \rangle + \langle M \times F \rangle + \langle M \times M \rangle\big) = .25\big(1 + 0 + 0 + 1\big) = 0.5$. On the other hand two cousins whose fathers are siblings have a relatedness of only 12.5%: $\langle C1 \times C2 \rangle = .25\big(\langle F1 \times F2 \rangle + \langle F1 \times M2 \rangle + \langle M1 \times F2 \rangle + \langle M1 \times M2 \rangle\big) = .25\big(0.5 + 0 + 0 + 0\big) = 0.125$. Fig. 4.1b compares the well-known relatedness of different family members (see for example, Ref. [115]) with that calculated from a behavioral model which we call probabilistic spin logic, PSL, and from a simulation of the actual circuit using a SPICE-based circuit simulator.

The behavioral PSL model represents an intermediate step in the translation of BN's to electronic circuits. It is a network whose nodes are abstract p-bits denoted by m (see Fig. 4.2) connected to other nodes and biased through dimensionless constants J, h respectively. The p-bits described by Eq. 4.1a is analogous to a binary stochastic neuron and their interconnection described by Eq. 4.1b is analogous to a synapse. The PSL model is then translated into a circuit model whose nodes are actual circuit



(c)

Relation	Bayes	PSL	SPICE
Stranger	0	−0.0002	0.0003
Sister	0.5	0.4999	0.4641
Father	0.5	0.4994	0.4732
Aunt	0.25	0.2502	0.2580
Grandfather	0.25	0.2499	0.251
Half-Sister	0.25	0.2498	0.272
Double-Cousin	0.25	0.2504	0.2646
Cousin	0.125	0.1247	0.129
Second-Cousin	0.031	0.0309	0.042

Fig. 4.1.: **(a) An example Bayesian Network (BN)** showing three generations with children, parents and grandparents. The grandparent generation has no explicit parents, but we can introduce their correlations implicitly by making the second set of grandparents (MM2,FM2,MF2,FF2) conditionally dependent on the first set (FF1,MF1,FM1,MM1) as shown. The rest of the nodes (C1, F1, M1, C2, F2, M2) are each conditionally dependent on two parents. **(b) Representative SPICE-results** from the full hardware circuit of Fig. 4.4 when the circuit is set up so that FF1 ≈ FF2, MF1 ≈ MF2, FM1 ≈ FM2, MM1 ≈ MM2. In this scenario, $C1$ and $C2$ are double cousins. **(c) Relatedness of family members** calculated from three different models: a behavioral model, PSL, a SPICE model for the corresponding circuit and the well-known result from standard statistical arguments applied to BN. Single, double and triple encirclements indicate a zero-parent node, one parent node,

elements denoted by M connected to other nodes and biased through conductances G and voltages V_{bias}. Fig. 4.1b shows that the relatedness from the PSL model (second column) as well as that obtained from the SPICE model (third column) agree well with the standard BN result (first column), thus providing confidence that the circuits obtained following our procedure can be used to study BN's in general.

Genetic relatedness is a textbook concept that provides a good benchmark for a hardware circuit emulator, but the principles presented here can be used to emulate more complicated BN's as well, involving more complex CPT's, as well as more complex nodes with $N > 2$ parents, reflecting the presence of more than two factors influencing the occurrence of an event.

4.1 Probabilistic Spin Logic: Behavioral Model

PSL is defined by two equations [4] loosely analogous to a neuron and a synapse. The former is a binary stochastic neuron, or what we call a p-bit, whose output m_i is related to its dimensionless input I_i by the relation

$$m_i(t + \Delta t) = \text{sgn}\big(\text{rand}(-1, 1) + \tanh\ I_i(t)\big) \tag{4.1a}$$

where rand $(-1, +1)$ is a random number uniformly distributed between -1 and $+1$, and t is the normalized time unit. The synapse generates the input I_i from a weighted sum of the states of other p-bits according to the relation

$$I_i(t) = I_0\left(h_i(t) + \sum_j J_{ij} m_j\right) \tag{4.1b}$$

where, h_i is the on-site bias and J_{ij} is the weight of the coupling from j^{th} p-bit to i^{th} p-bit.

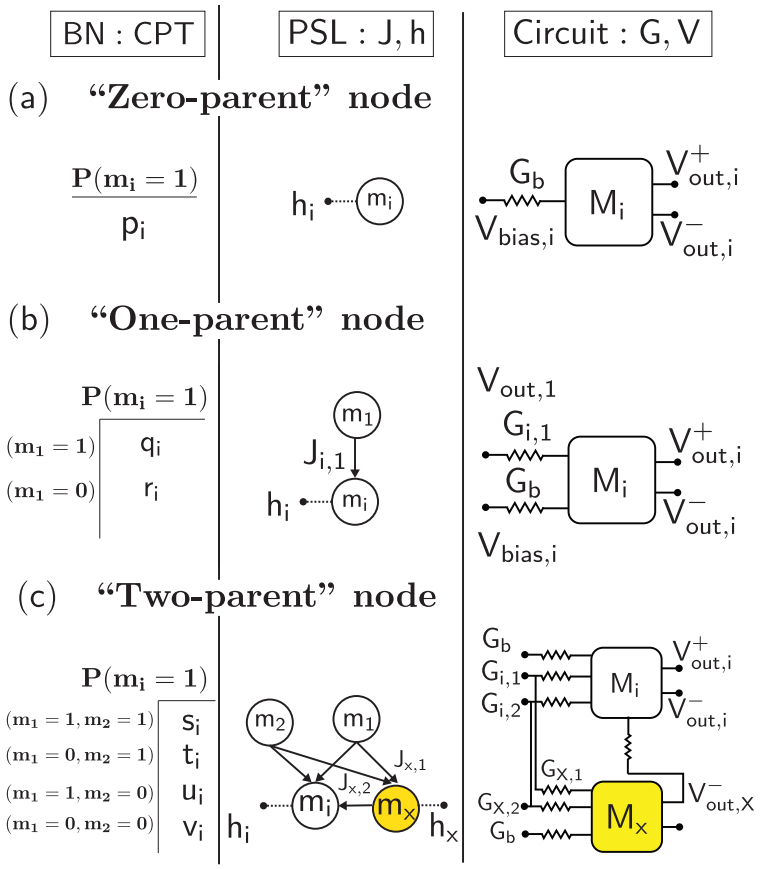


Fig. 4.2.: **Translating nodal information from BN to PSL to circuit**: Each node of a BN is described by a conditional probability table (CPT), that of a PSL network is described by dimensionless constants J, h, and that of circuit is described by conductances G and voltage V_{bias}. The text describes how the CPT is translated to J, h and then to G, V_{bias} for (a) zero-parent node, (b) one-parent node and (c) two-parent node.

4.2 From BN nodes to PSL nodes

To relate I_i to the conditional probability P_i for m_i to be 1, we note from Eq. 4.1a that the average value of m_i is $\tanh(I_i)$ and this must equal $P_i \times (+1) + (1 - P_i) \times (-1) = 2P_i - 1$. Making use of Eq. 4.1b we can write

$$I_0(h_i + \sum_j J_{ij} m_j) = \tanh^{-1}(2P_i - 1) \tag{4.2}$$

We use this relation to translate the P_i from the CPT into J, h in the PSL model, but the details differ depending on the number of "parents" of node i (Fig. 4.2).

Nodes with *no parents* have no connecting weights J_{ij}, only a bias h_i which is related to the specified conditional probability p_i by $h_i = (1/I_0)\ \tanh^{-1}(2p_i - 1)$. Nodes with *one parent* have one connecting weight J_{ij}, and a bias h_i which can be obtained from the two specified conditional probabilities q_i, r_i from the equations

$$h_i + J_{i1}(+1) = \frac{1}{I_0}\tanh^{-1}(2q_i - 1) \tag{4.3a}$$

$$h_i + J_{i1}(-1) = \frac{1}{I_0}\tanh^{-1}(2r_i - 1) \tag{4.3b}$$

Nodes with *two parents* have two connecting weights J_{i1}, J_{i2}, and a bias h_i but there are four equations for these three unknowns. All equations can be satisfied simultaneously only if the equations are not linearly independent. If they are independent then an auxiliary node X is introduced so that:

$$h_i + J_{i1}(+1) + J_{i2}(+1) + J_{iX}m_X = \frac{1}{I_0}\tanh^{-1}(2s_i - 1) \tag{4.4a}$$

$$h_i + J_{i1}(-1) + J_{i2}(+1) + J_{iX}m_X = \frac{1}{I_0}\tanh^{-1}(2t_i - 1) \tag{4.4b}$$

$$h_i + J_{i1}(+1) + J_{i2}(-1) + J_{iX}m_X = \frac{1}{I_0}\tanh^{-1}(2u_i - 1) \tag{4.4c}$$

$$h_i + J_{i1}(-1) + J_{i2}(-1) + J_{iX}m_X = t = \frac{1}{I_0}\tanh^{-1}(2v_i - 1) \tag{4.4d}$$

where $m_X = \tanh(h_X + J_{X1}m_1 + J_{X2}m_2)$ with the parents m_1, m_2 equal to $(\pm 1, \pm 1)$ as appropriate for the four equations. One possibility is to choose h_X, J_{X1}, J_{X2} such that $m_X = m_1 \cap m_2$ and then select the four remaining unknowns $h_i, J_{i1}, J_{i2}, J_{iX}$ to satisfy Eqs. 4.4.

Nodes with N parents have a total of $(N+1)$ unknowns, but there are 2^N equations to satisfy. Depending on the number of linearly independent equations, it is necessary to introduce the appropriate number of auxiliary variables. In this letter we will only present results for the BN in Fig. 4.1 which includes nodes with a maximum of $N = 2$ parents. Moreover, the CPT for the 2-parent nodes is assumed to be of the form $t = u = 0.5$, $s = 1 - \varepsilon$ and $v = \varepsilon$, ε being a small number introduced to avoid the

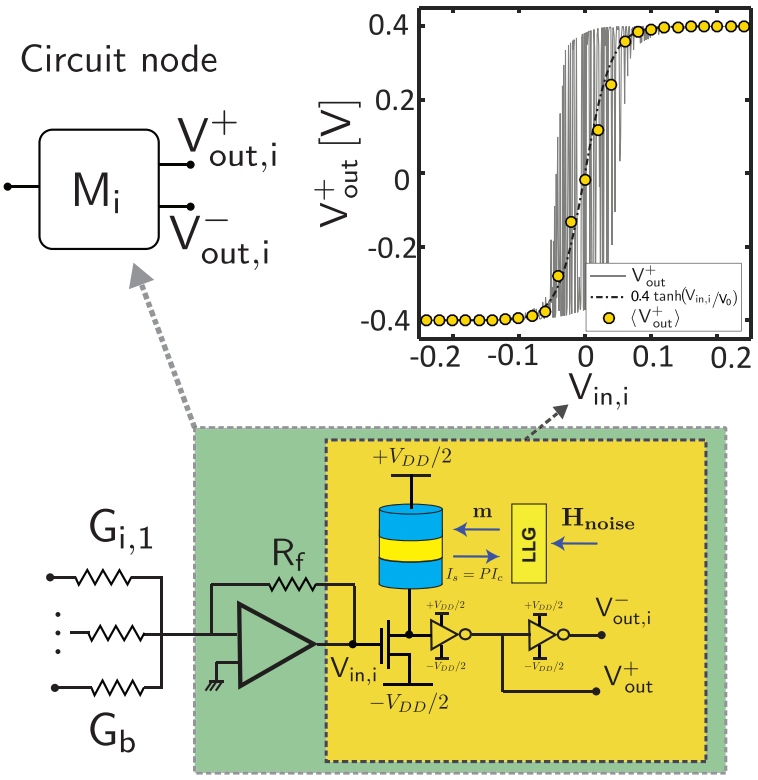


Fig. 4.3.: **Circuit implementation of building block**:The circuit Eqs. 4.5 can be mapped onto the PSL Eqs. 4.1 using Eqs. 4.6 as described in the text. The circuit node M_i is defined to include the transimpedance amplifier along with the p-bit. The details of the embedded MRAM based p-bit are discussed in the text.

singularities associated with the tanh function. With this CPT, no auxiliary node (X) is needed.

4.3 From PSL nodes to circuit nodes

To translate the PSL into a circuit we use the embedded MTJ [4] whose output is related to its input by the relation

$$V_{out,i} = \frac{V_{DD}}{2}\text{sgn}\left(\text{rand}(-1,+1) + \tanh\frac{V_{in,i}}{V_0}\right) \tag{4.5a}$$

where $\pm V_{DD}/2$ are the supply voltages, and V_0 is a parameter (~ 50 mV) describing the width of the sigmoidal response. Although V_0 is a fitting parameter for the

tanh function, it captures the actual sigmoidal response of the MTJ unit quite well. Even if there is a slight deviation with the tanh function due to the skewness of the MTJ response, it will not cause a noticeable difference in the output since PSL is quite robust against noise [4]. The output voltages are connected back through conductances G with a transimpedance amplifier having a feedback resistor R_f, so that (see Fig. 4.3)

$$V_{in,i} = V_{\text{bias,i}} G_b R_f + \sum_j V_{out,j} G_{ij} R_f \tag{4.5b}$$

Eqs. 4.5 can be mapped onto the PSL Eqs. 4.1 by defining

$$m_i = \frac{V_{out,i}}{V_{DD}/2}, \quad I_i = \frac{V_{in,i}}{V_0} \tag{4.6a}$$

$$h_i = \frac{V_{\text{bias,i}}}{V_{DD}/2}, \quad J_{ij} = \frac{G_{ij}}{G_b}, \quad I_0 = G_b R_f \frac{V_{DD}}{2V_0} \tag{4.6b}$$

4.4 SPICE-based p-bit Model

In order to design the basic building block for the BN based on Eq. 4.5a, we are following the p-bit design in Ref. [7] that describes an embedded MTJ structure with a stochastic nanomagnet. We consider the weight logic in Eq. 4.1b to be implemented using ideal transimpedance amplifier with resistors [4] though a capacitive network with a more compact implementation could also be used to implement the weighted sum operation [116]. We use the same parameters for the p-bit as in [7]: A circular (stochastic) in-plane nanomagnet with negligible uniaxial anisotropy ($H_K \sim 0$) [54, 117], damping coefficient for the nanomagnet $\alpha = 0.01$, saturation magnetization $\text{M}_\text{s} = 1100$ emu/cc, with a free layer diameter 22 nm and a thickness of 2 nm. A Tunneling Magnetoresistance (TMR) value of 110% is used based on [118]. The MTJ conductance is assumed to be bias-independent and is given by $G(t) = G_0\big[1 + m_z(t)\ \text{TMR}/(2 + \text{TMR})\big]$, where $m_z(t)$ is provided to the model by a self-consistent solution of the sLLG (stochastic Landau-Lifshitz-Gilbert equation) solver. The device operation is based on the control of the transistor conductance through the input

voltage. The non-linear transistor characteristics with respect to drain, gate and source voltages are captured in simulation by the 14 nm HP-FinFET node from the Predictive Technology Models (PTM) [41]. When the transistor conductance is much greater or less than the MTJ conductance, the output shows little noise but when the MTJ conductance is matched to the transistor ON resistance around $V_{in,i}$=0, there are large fluctuations at the output. In Fig. 4.3, each circular dot in the sigmoid is obtained by averaging 1 μs response of the stochastic output and the dashed lines show a (tanh) fit with a $V_0 = 50$ mV. The solid lines are obtained by sweeping the input voltage rail-to-rail in 100 ns and plotting the input with respect to the output voltage. Within the modular SPICE framework, the magnetization dynamics of the circular stochastic nanomagnet is captured by solving the sLLG equation in the macrospin assumption,

$$(1+\alpha^2)\frac{d\hat{m}}{dt} = -|\gamma|\hat{m}\times\vec{H} - \alpha|\gamma|(\hat{m}\times\hat{m}\times\vec{H}) + \frac{1}{qN}(\hat{m}\times\vec{I_S}\times\hat{m}) + \left(\frac{\alpha}{qN}(\hat{m}\times\vec{I_S})\right) \tag{4.7a}$$

where α is the damping coefficient, γ is the electron gyromagnetic ratio, $N = \mathrm{M_s Vol.}/\mu_B$ is the total number of Bohr magnetons in the magnet volume, $\mathrm{M_s}$ is the saturation magnetization, $\vec{H} = \vec{H_d} + \vec{H_n}$ is the effective field including the out-of-plane ($\hat{x}$ directed) demagnetization field $\vec{H_d} = -4\pi \mathrm{M_s m_x \hat{x}}$, as well as the thermally fluctuating magnetic field due to the three dimensional uncorrelated thermal noise H_n with zero mean $\langle H_n \rangle = 0$ and standard deviation $\langle H_n^2 \rangle = 2\alpha \mathrm{kT}/|\gamma|\mathrm{M_s V}$ along each direction, $I_S = PI_C\hat{z}$ is the spin current along the MTJ fixed layer direction ($\hat{z}$) where P is the polarization of the fixed magnet. The model takes this spin-current (I_S) incident to the free layer into account and for the parameters we have used, this current does not cause appreciable pinning of the free layer. A time step $\Delta t = 1$ ps is used for the circuit simulation which implies a noise bandwidth of $\Delta f = 1$ THz.

4.5 SPICE-based Circuit Model

Fig. 4.4a shows the full circuit assembled using the nodes defined in Fig. 4.3 to mimic the Bayesian network in Fig. 4.1a. Fig. 4.4b shows typical nodal voltages obtained from a SPICE simulation, whose correlations can either be calculated in software or measured using an XNOR gate to multiply them as shown and finding the long term average with an RC circuit having a time constant $\gg$ T:

$$\langle C1 \times C2 \rangle = \int_0^T \frac{dt}{T} \, C_1(t) C_2(t)$$

These nodal correlations in the circuit can be used to compute the correlation between causally connected real world variables. For example the relatedness of different family members cited in Fig. 4.1b were all obtained in this manner from circuit simulations. Different relationships between $C1$ and $C2$ are enforced by using different CPT's for their grandparents. For example, if all grandparents are unrelated, the grandchildren $C1$ and $C2$ would show zero correlation. But if we enforce perfect correlation between $FF1$, $FF2$ and between $MF1$,$MF2$ through the corresponding CPT, we effectively make $F1$ and $F2$ into siblings with a correlation of 50%. $C1$ and $C2$ then are first cousins with a correlation of 12.5%.If we further enforce perfect correlation between $FM1$, $FM2$ and between $MM1$, $MM2$, we also make $M1$ and $M2$ into siblings with a correlation of 50%, just like $F1$ and $F2$. $C1$ and $C2$ now are double cousins with a correlation of 25%.

Note that this is an asynchronous circuit with no clocks of any kind. This is particularly interesting since the corresponding PSL simulations require p-bits to be updated sequentially from parent to child node. Such a sequential Bayesian network is also implemented experimenatally by Debashis et al. using stochastic spintronic devices [119] and the experimental results are benchmarked with SPICE simulation of the implemented hardware(see Appendix B). In the SPICE circuit simulation there is no central clock to enforce an updating sequence, but our results show that the correlations are in good agreement with the PSL results and with Bayes theorem. However, such an asynchronous operation works only if the interconnect delays, for

example from node FF1 to FF2, are much shorter than the nanomagnet fluctuations as discussed in Reference [120]. Since magnetic fluctuations occur at ∼ns time scales, this condition is naturally satisfied. The slight mismatch of the Bayes theorem and the PSL model appears to decrease systematically with increasing sample size (N=1e7 for the examples shown in Fig. 4.1b) with the full circuit model closely following them, but the updating issue in asynchronous operation deserves further study. We have not considered variations in the thermal barriers or interconnect delays in this paper, which requires further study.

4.6 Conclusions

In summary, we have used SPICE simulations to show that using existing MRAM technology it should be possible to build p-circuits that mimic Bayesian networks such that each stochastic node is represented by a stochastic p-bit. We show that the *ensemble-averaged* correlations between the actual physical variables can be estimated from the *time-averaged* correlations between the voltages at the corresponding nodes which are measured electronically with XNOR logic gates and long time constant RC circuits, thus requiring no software-based processing of any kind. Our results could open up a new application space for Embedded MRAM technology with minimal modifications.

Acknowledgment

This work was supported by the National Science Foundation (NSF).

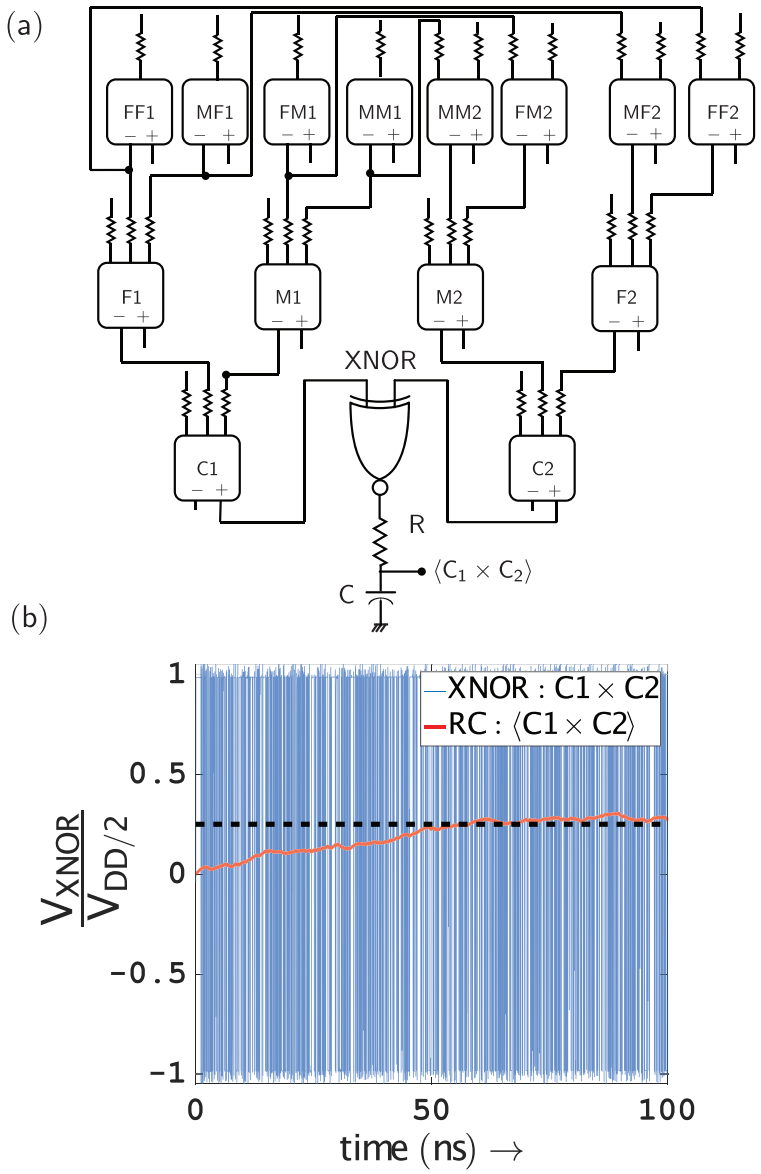


Fig. 4.4.: **SPICE simulation of the full circuit** designed to mimic the Bayesian network in Fig. 4.1a. (a) Circuit diagram, (b) Typical stochastic nodal voltages from which nodal correlations can be obtained using an XNOR gate and a long time constant RC circuit. In the present example, the following parameters are used: The RC circuit uses $R = 200$ kΩ, C=200 fF, $R_f = 150$ kΩ and $I_0 = 1$ with dimensionless weights $J_{ij} = J_0 = 2.3026$ which are then used to obtain conductances G_{ij} from Eq. 4.6. A simulation time of 1 ps is used in HSPICE that combines the self-consistent stochastic LLG with Predictive Technology models (PTM) [41] as in [7]. All transistors use the 14nm HP-FinFET node with minimum fin numbers (nfin=1). The XNOR gate is designed as a standard 14 transistor CMOS circuit, inverting an XOR output.

5. HARDWARE DESIGN REQUIREMENTS FOR AUTONOMOUS BAYESIAN NETWORKS

Most of the materials in this chapter have been extracted verbatim from the paper: "Hardware Design Requirements for Autonomous Bayesian Networks", R. Faria, J. Kaiser, K. Y. Camsari, and S. Datta, arXiv:2003.01767, (in review).

Directed acyclic graphs or Bayesian networks that are popular in many AI related sectors for probabilistic inference and causal reasoning can be mapped to probabilistic circuits built out of probabilistic bits (p-bits) that are analogous to binary stochastic neurons. In order to satisfy standard statistical results, individual p-bits not only need to be updated sequentially, but also in order from the parent to the child nodes, necessitating the use of sequencers in software implementations. In this article, we first use SPICE simulations to show that an autonomous hardware Bayesian network can operate correctly without any clocks or sequencers, but only if the individual p-bits are appropriately designed. We then present a simple behavioral model of the autonomous hardware illustrating the essential characteristics needed for correct sequencer-free operation. This model is also benchmarked against SPICE simulations and can be used to simulate large scale networks. Our results could be useful in the design of hardware accelerators that use energy efficient building blocks suited for low-level implementations of Bayesian networks.

5.1 Introduction

Bayesian networks (BN) or belief nets are probabilistic directed acyclic graphs (DAG) popular for reasoning under uncertainty and probabilistic inference in real world applications such as medical diagnosis [121], genomic data analysis [122–124], forecasting [125,126], robotics [127], image classification [128,129], neuroscience [130]

and so on. BNs are composed of probabilistic nodes and edges from *parent* to *child* nodes and are defined in terms of conditional probability tables (CPT) that describe how each *child node* is influenced by its *parent nodes* [104, 106, 131, 132]. The CPTs can be obtained from expert knowledge and/or machine learned from data [133]. Computation of different probabilities from a BN becomes intractable when the network gets deeper and more complicated with child nodes having many parent nodes. This has inspired various hardware implemenations of BNs for efficient inference [10, 107–114, 134, 135]. In this article we have elucidated the design criteria for a biologically inspired, autonomous (clockless) hardware for BN unlike other implementations that typically use clocks.

Recently a new type of hardware computing framework called Probabilistic Spin Logic (PSL) is proposed [4] based on a building block called probabilistic bits (p-bits) which are analogous to Binary Stochastic Neurons (BSN) [35, 136] in the artificial neural network (ANN) literature. p-bits can be interconnected to solve a wide variety of problems such as optimization [100, 137], inference [97], an enhanced type of Boolean logic that is invertible [4, 101, 120, 138], quantum emulation [139] and machine learning [140].

Unlike conventional deterministic networks built out of stable bits, stochastic or probabilistic networks are composed of p-bits (Fig. 5.1a). p-bits can be correlated by interconnecting them to construct p-circuits defined by two equations [4, 35, 136]: (1) a p-bit/BSN equation and (2) a weight logic/synapse equation. The output of a p-bit, m_i is related to its dimensionless input I_i by the equation:

$$m_i(t + \tau_N) = \text{sgn}\big(\text{rand}(-1, 1) + \tanh\ I_i(t)\big) \tag{5.1a}$$

where rand$(-1, +1)$ is a random number uniformly distributed between -1 and $+1$, and τ_N is the neuron evaulation time.

The synapse generates the input I_i from a weighted sum of the states of other pBits. In general the synapse can be a linear or non-linear function although a common form is the linear synapse described according to the equation:

$$I_i(t+\tau_S) = I_0\left(h_i + \sum_j J_{ij} m_j(t)\right) \tag{5.1b}$$

where, h_i is the on-site bias and J_{ij} is the weight of the coupling from j^{th} pBit to i^{th} pBit and τ_S is the synpase evaluation time. Several hardware designs of p-bits based on low barrier nanomagnet (LBM) physics have been proposed. For example, fig. 5.1a shows two p-bit designs: Design 1 ([7]) and Design 2 ([4]). Design 1 is very similar to the commercially available 1T/1MTJ (T: Transistor, MTJ: Magnetic Tunnel Junction) embedded Magnetoresistive Random Access Memory (MRAM) device where the free layer of the MTJ is replaced by an LBM. Design 2 is similar to the basic building block of SOT-MRAM (SOT: Spin Orbit Torque) device [42, 141] where the thermal fluctuation of the free layer magnetization of the stochastic MTJ (sMTJ) is tuned by a spin current generated in a heavy metal layer underneath the LBM due to SOT effect. Whereas design 2 requires spin current manipulation, design 1 does not rely on that as long as circular in-plane LBMs with continuous valued magnetization states that are hard to pin are used.

In traditional software implementations, p-bits are updated sequentially for accurate operation such that after each $\tau_S + \tau_N$ time interval, only one p-bit is updated. This naturally implies the use of sequencers to ensure the sequential update of p-bits. For symmetrically connected networks ($J_{ij} = J_{ji}$) such as Boltzmann machines, the update order of p-bits does not matter and any random update order produces the standard probability distribution described by equilibrium Boltzmann law as long as p-bits are updated sequentially. But for directed acyclic networks ($J_{ij} \neq 0, J_{ji} = 0$) or Bayesian networks to be consistent with the expected conditional probability distribution, *p-bits need to be updated not only sequentially, but also in a specific update order which is from the parent to child nodes* [136] similar to the concept of forward sampling in belief networks [131, 142, 143]. As long as this parent to child update or-

der is maintained, the network would converge to the correct probability distribution described by probability chain rule or Bayes rule. This effect of update order in a sequential p-circuit is shown on a three p-bit network in fig. 5.1b.

Unlike sequential p-circuits in ANN literature, the distinguishing feature of our probabilistic hardware is that it is *autonomous* where each p-bit runs autonomously in parallel without any clocks or sequencers. This autonomous p-circuit (ApC) allows massive parallelism potentially providing peta flips per second sampling speed [6]. The complete sequencer-free operation of our "autonomous" p-circuit is very different from the "asynchronous" operation of spiking neural networks [13, 14]. Although p-bits are fluctuating in parallel in an ApC, it is very unlikely that two p-bits will update at the exact same time since random noise control their dynamics. Therefore persistent parallel updates are extremely unlikely and are not a concern. So the p-bits update effectively sequentially. But each update has to be *informed* such that when one p-bit updates it has received the up-to-date input I_i based on the latest states of other p-bits m_j that it is connected to. This informed update can be ensured as long as the synapse response time is much faster than the neuron time ($\tau_S \ll \tau_N$) and this is the design rule for an ApC. An ApC works properly for a Boltzmann network without any clock since no specific update order is required in this case. But it is not intuitive at all if an ApC would work for a Bayesian network since a particular parent to child *informed* update order is required in this case as shown in fig. 5.1b. As such, it is not straight-forward that a clockless autonomous circuit can naturally ensure this specific informed update order. In fig. 5.1c, we have shown that it is possible to design hardware p-circuit that can naturally ensure a parent to child informed update order in a Bayesian network without any clocks. In fig. 5.1c, two p-bit designs are evaluated for implementing both Boltzmann and Bayesian network. We have shown that design 1 is suitable for both Boltzmann and Bayesian networks. But design 2 is suitable for Boltzmann networks only and does not work for Bayesian networks in general. The synapse in both types of p-circuits is implemented using a

resistive crossbar architecture [7]. In all the simulations τ_S is assumed to be negligible compared to other time scales in the circuit dynamics.

Further we have provided a behavioral model in section 5.2 for both design 1 and 2 illustrating the essential characteristics needed for correct sequencerfree operation of BNs. Both models are benchmarked against state-of-the-art device/circuit models (SPICE) of the actual devices and can be used for the efficient simulation of large scale autonomous networks.

5.2 Behavioral model for autonomous hardware

5.2.1 Autonomous behavioral model: Design 1

The autonomous circuit behaviour of design 1 can be explained by slightly modifying the two equations (eqns.5.1 a and b) stated in section 5.1. The fluctuating resistance of the low barrier nanomagnet based MTJ is represented by a correlated random number r_{MTJ} with values between -1 and +1 and average dwell time of the fluctuation denoted by τ_N. The NMOS transistor tunable resistance is denoted by r_T and the inverter is represented by a *sgn* function. Thus the normalized output $m_i = V_{OUT,i}/(V_{DD}/2)$ of the i_{th} p-bit can be expressed as:

$$m_i\left(t+\Delta t\right) = \mathrm{sgn}\left(r_{T,i}\left(t+\Delta t\right) - r_{MTJ,i}\left(t+\Delta t\right)\right) \tag{5.2}$$

where, Δt is the simulation time step which is ideally as small as possible, $r_{T,i}$ is the NMOS transistor resistance tunable by the normalized input $I_i = V_{IN,i}/V_0$ where V_0 is a tanh fitting parameter which is $\approx$ 50mV for the chosen parameters and transistor technology and $r_{MTJ,i}$ is a correlated random number generator with an average retention time of τ_N. $r_{T,i}$ as a function of input I_i is approximated by a tanh function with a response time denoted by τ_T modelled by the following equations:

$$r_{T,i}\left(t+\Delta t\right) = r_{T,i}\left(t\right)\exp\left(-\Delta t/\tau_T\right) + \left(1-\exp\left(-\Delta t/\tau_T\right)\right)\left(\tanh\left(I_i(t+\Delta t)\right)\right) \tag{5.3}$$

The synapse delay τ_S in computing the input I_i can be modelled by:

$$I_i\left(t+\Delta t\right)=I_i\left(t\right)\exp\left(-\Delta t/\tau_S\right)+\left(1-\exp\left(-\Delta t/\tau_S\right)\right)\left(I_0\Big(\sum_j J_{ij}m_j(t)+h_j\Big)\right) \tag{5.4}$$

For calculating $r_{MTJ,i}$, at time $t+\Delta t$ a new random number will be picked according to the following equations:

$$r_{flip,i}\left(t+\Delta t\right)=\text{sgn}\left(exp\left(-\frac{\Delta t}{\tau_N}\right)-\text{rand}_{[0,1]}\right) \tag{5.5a}$$

where, $\text{rand}_{[0,1]}$ is a uniformly distributed random number between 0 and 1 and τ_N represents the average retention time of the fluctuating MTJ resistance. If r_{flip} is -1, a new random r_{MTJ} will be chosen. Otherwise the previous r_{MTJ} will be kept which can be expressed as:

$$r_{MTJ,i}\left(t+\Delta t\right)=\frac{r_{flip,i}\left(t+\Delta t\right)+1}{2}r_{MTJ,i}\left(t\right)-\frac{r_{flip,i}\left(t+\Delta t\right)-1}{2}\text{rand}_{[-1,1]} \tag{5.5b}$$

The charge current flowing throught the MTJ branch of p-bit design 1 can get polarized by the fixed layer of the MTJ and generate a spin current I_{MTJ} that can tune/pin r_{MTJ} by modifying τ_N according to:

$$\tau_N=\tau_N^0\text{exp}(r_{MTJ}I_{MTJ}) \tag{5.6}$$

where, τ_N^0 is the retention time of r_{MTJ} when $I_{MTJ}=0$. This pinning effect by I_{MTJ} is much smaller or negligible in in-plane magnets (IMA) than perpendicular magnets (PMA) [144].

Figure. 5.2a shows the benchmarking this behavioral model for p-bit design 1 with SPICE simulation of the actual hardware in terms of fluctuation dynamics, sigmoidal charateristic response, autocorrelation time (τ_{corr}) and step response time (τ_{step}) and in all cases the behavioral model closely matches SPICE simulationsl. SPICE simulation involves experimentally benchmarked modules for different parts of the device, for example solving stochastic Landau-Lifshitz-Gilbert equation (sLLG) for LBM physics,

14 nm Predictive Technology Model (PTM) for transistors etc. The autonmous behavioral model for design 1 is labeled as "PPSL: design 1". The benchmarking is done for two different LBMs: (1) Faster fluctuating magnet 1 with saturation magnetization M_s = 1100 emu/cc, $Diameter$ = 22 nm, $Thickness$ = 2 nm, in-plane easy axis anisotropy $H_k = 1$ Oe, damping coefficient $\alpha = 0.01$, demagnetization field $H_d = 4\pi M_s$ and (2) Slower fluctuating magnet 2 with the same parameters as in magnet 1 except $Diameter = 150$ nm. The fast and slow fluctuations of the normalized output $m_i = V_{OUT,i}/(V_{DD}/2)$ is captured by changing the τ_N parameter in the PPSL model. In the steady state sigmoidal response, V_0 is a tanh fitting parameter that defines the width of the sigmoid and lies within the range of 40 mV to 60 mV reasonably well depending on which part of the sigmoid needs to be better matched. In fig. 5.2, V_0 value of 50 mV is used to fit the sigmoid from SPICE simulation.

There are two types of time responses: (1) Autocorrelation time under zero input condition labeled as τ_{corr} and (2) step response time τ_{step}. The full width half maximum (FWHM) of the autocorrelation function of the fluctuating output under zero input is defined by τ_{corr} which is proportional to the retention time τ_N of the LBM. The step response time τ_{step} is obtained by taking an average of the p-bit output over many ensembles when the input I_i is stepped from a large negative value to zero at time $t = 0$. τ_{step} defines how fast the first statistically correct sample can be obtained after the input is changed. For p-bit design 1, τ_{step} is independent of LBM retention time τ_N and is defined by the NMOS transistor response time τ_T which is much faster (few picoseconds) than LBM fluctuation time τ_N. The effect of this two very different time scales in design 1 ($\tau_{step} \ll \tau_{corr}$) on an autonomous Bayesian network is described in section 5.3.

5.2.2 Autonomous behavioral model: Design 2

The autonomous behavioral model for design 2 is proposed in [6]. In this article, we have benchmarked this model with the SPICE simulation of the single p-bit steady state and time responses shown in fig. 5.2b. According to this model, the normalized output $m_i = V_{OUT,i}/(V_{DD}/2)$ can be expressed as:

$$m_i(t + \Delta t) = m_i(t)\text{sgn}\Big(p_{NOTflip,i}(t + \Delta t) - \text{rand}_{[0,1]}\Big) \tag{5.7a}$$

$$p_{NOTflip,i}(t + \Delta t) = \exp\Big(-\frac{\Delta t}{\tau_N \exp(I_i m_i(t))}\Big) \tag{5.7b}$$

where, $p_{NOTflip,i})(t + \Delta t)$ is the probability of not flipping or probability of retention of the i^{th} p-bit in the next time step that is a function of average neuron flip time τ_N, input I_i and the current p-bit output $m_i(t)$. Figure. 5.2b shows how this simple autonomous behavioural model for design 2 matches reasonably well with SPICE simulation of the device in terms of fluctuation dynamics, sigmoidal charateristic response, autocorrelation time (τ_{corr}) and step response time (τ_{step}). In design 2, τ_{step} and τ_{corr} are both proportional to LBM fluctuation time τ_N unlike design 1.

Autonomous behavioral model for design 2 is also emulated on an FPGA platfrom by Sutton et. al. [6] and the model is benchmarked for a sampling problem on the Sherrington Kirkpatrick model [145] (see Appendix A).

Different time scales in p-bit design 1 and 2 are also reported in [144] in a different context. In this article, we explain the effect of these time scales in designing an autonomous Bayesian network (section 5.3).

5.3 Difference between Design 1 and Design 2 in implementing BN

The behavioral models introduced in section 5.2 are applied to implement a multi layer belief/Bayesian network with 19 p-bits and random interconnection strength between $+1$ and -1 (fig. 5.3a). For simpler understanding purpose, the interconnections are designed in such a way so that although there are no meaningful correlations

between the blue and red colored nodes with random couplings, pairs of intermediate nodes (A, M_1) and (M_1, B) get negatively correlated because of a net $-r^2$ type coupling through each branch connecting the pairs. So it is expected that the start and end nodes (A, B) get positively correlated. Fig. 5.3b shows histograms of four configurations (00, 01, 10, 11) of the pair of nodes A and B obtained from different approaches: Bayes rule (labeled as Analytic), SPICE simulation of design 1 (SPICE: Design 1) and design 2 (SPICE: Design 2), autonomous behavioral model for design 1 (PPSL: Design 1) and design 2 (PPSL: design 2). It is shown that results from SPICE simulation and behavioral model for design 1 matches reasonably well with the standard analytical values showing 00 and 11 states with highest probability whereas design 2 autonomous hardware does not work well in terms of matching with the analytical results and shows approximately all equal peaks. The analytical values are obtained from applying the standard joint probability rule for BNs which is:

$$P(x_1, x_2, ..., x_N) = \prod_{i=1}^{N} x_i | Parents(x_i) \tag{5.8}$$

Joint probability between two specific nodes x_i and x_j can be calcuted from the above equation by summing over all configurations of the others nodes in the network which becomes computationally expensive for larger networks. But one major advantage of our probabilistic hardware is that probabilities of specific nodes can be obtained just by looking at the nodes of interest ignoring all other nodes in the system similar to what Feynman stated about a probabilistic computer imitating the probabilistic laws of nature [146]. Indeed, in the Bayesian network example in fig. 5.3, the probabilities of different configurations of nodes A and B were obtained just by looking at the fluctuating outputs of the two nodes ignoring all other nodes. For the SPICE simulation of design 1 hardware, tanh fitting parameter $V_0 = 57$ mV is used and the mapping principle from dimensionless coupling terms J_{ij} to the coupling resistances in the hardware is described in [97].

The reason why design 1 works for a BN and design 2 does not, is because of the two very different time responses of the two designs shown in fig. 5.2. It is this two

different time scales in design 1 ($\tau_{step} \ll \tau_{corr}$) that naturally ensures a parent to child informed update order in a Bayesian network. The reason is that when τ_{step} is small, each child node can immediately respond to any change of it's parent nodes which has a much larger time scale $\propto \tau_{corr}$, and be conditionally satisfied with the parent nodes very fast. Otherwise, if τ_{corr} gets comparable to τ_{step}, the child node will not be able to keep up with the fast changing parent nodes and will produce substantial number of statistically incorrect samples over the entire time range (Fig. 5.5) thus deviating from the correct probability distribution.

The effect of τ_{step}/τ_{corr} ratio is shown in fig. 5.4 for the same BN presented in fig. 5.3 by plotting the histogram of AB configurations for different τ_T/τ_N ratios. It is shown that when τ_T/τ_N ratio is much small, the histogram converges to the correct distribution. As τ_T gets comparable to τ_N, the histogram begins to diverge from the correct distribution. Thus the very fast NMOS transistor response in design 1 makes it suitable for an autonomous Bayesian network hardware. One thing to note that under certain conditions, results from design 2 can also match with the analytical if the input I_i to each p-bit in the network always fluctuates between large values that ensures a fast step response time.

So apart from ensuring a fast synapse compared to neuron fluctuation time ($\tau_S \ll \tau_N$) which is the design rule for an autonomous probabilistic hardware, the autonomous Bayesian network demands an additional p-bit design rule which is a much faster step response time of the p-bit compared to its fluctuation time ($\tau_{step} \ll \tau_N$) as ensured in design 1. In all the simulations the LBM was a circular IMA with in-plane magnetization along a specific direction spanning all values between +1 and -1 and negligible pinning effect. If the LBM is a PMA magnet with bipolar fluctutations having just two values +1 and -1, design 1 will not provide any sigmoidal response except with substantial pinning effect [137]. Under this condition, τ_{step} of design 1 will be comparable to τ_N again and the system will not work as an autonomous Bayesian network in general. So LBM with continuous range fluctuation is expected for design 1 p-bit to work properly as a Bayesian network.

5.4 Binary p-bit composite as multi-state random variable

Real world applications of BN often involve multi-state random variable having more than two states [147]. Multiple p-bit units can be interconnected to build a composite unit that can represent one multi-state random variable. To demonstrate this, we have solved the famous Monty Hall Puzzle [148] using our proposed autonomous hardware (fig. 5.4).

The problem is stated as:

"*Suppose you're on a game show, and you're given the choice of three doors: Behind one door is a car; behind the others, goats. You pick a door, say No. 1, and the host, who knows what's behind the doors, opens another door, say No. 3, which has a goat. He then says to you, "Do you want to pick door No. 2?" Is it to your advantage to switch your choice?* [149]"

The problem can be mapped to a BN using three tri-state random variables [150]: Prize $P : \{1, 2, 3\}$, player's choice $X : \{1, 2, 3\}$ and Monty opens $M : \{1, 2, 3\}$. The three node network can be translated to another BN where each of the nodes is a binary random variable. For example, the variable P can be represented by a composite of three binary variables: P_1, P_2 and P_3 each having two states: $\{\text{TRUE} : 1, \text{FALSE} : 0\}$. To make sure that only three configurations of (P_1, P_2, P_3) are allowed which are $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$, the interconnection among them can be designed to be either directed acyclic with appropriate CPT or they can be bidirectionally connected to form a Boltzmann machine according to the following energy function:

$$E = P_1 + P_2 + P_3 + 2\bar{P}_1\bar{P}_2\bar{P}_2 \tag{5.9}$$

The BN can be mapped to our proposed autonomous hardware where each random variable can be represented by a p-bit with design 1. The input to each p-bit in the symmetrically connected BM composites can be calculted from the energy equation eq. 5.9 as: $I_{P_i} = -\delta E/\delta P_i$. The CPTs related to different child nodes in the network can be translated to a coupling matrix $[J]$ according to the mapping principles stated

in [97] or they can be leant applying machine learning algorithms [133]. The use of additional auxiliary p-bits to satisfy a given CPT can be reduced by allowing non-linear synapse circuit [137].

The probability of winning if the player switches door option can be obtained by $P_{switch,WIN} = \sum_{i\neq j\neq k} P(P_i = 1, X_j = 1, M_k = 1)$. Joint probabilitities among differnt nodes in the network can be obtained by looking only at the relevant p-bit outputs ignoring other p-bits in the system [80]. The result obtained from the autonomous p-circuit matches with the standard statistical value which is 2/3. Another way of getting the probability of winning by switching option would be to compute $P_{switch,WIN} = P(P_1 = 1|X_2 = 1, M_3 = 1) = P(P_1 = 1, X_2 = 1, M_3 = 1)/P(X_2 = 1, M_3 = 1)$.

5.5 Discussion

In this article we have elucidated the design criteria for an autonomous clockless hardware for Bayesian networks that requires a specific parent to child update order when implemented on a probabilistic circuit. By performing SPICE simulation of two autonomous probabilistic hardwares built out of p-bits (design 1 and design 2 in fig. 5.1), we have shown that the autonomous hardware will naturally ensure a parent to child informed update order without any sequencers if the step response time (τ_{step}) of the p-bit is much smaller than its autocorrelation time (τ_{corr}). This criteria of having two different time scales is met in design 1 as τ_{step} comes from the NMOS transistor response time τ_T in this design which is few picoseconds. We have also proposed an autonomous behavioral model for design 1 and benchmarked it against SPICE simulation of the actual hardware. All the simulations using behavioral model for design 1 is performed ignoring some non-ideal effects listed below:

- Pinning of the sMTJ fluctuation due to spin transfer torque (STT) effect is ignored by assuming $I_{MTJ} = 0$ in eqn. 5.6 which is a reasonable assumption considering circular in-plane magnets with continuous valued fluctuations. This

effect is more prominent in perpendicular anisotropy magnets (PMA) magnets than in-plane anisotropy magnets (IMA). It is important to include the pinning effect in p-bits with bipolar LBM fluctuations since in this case the p-bit does not provide a sigmoidal response without the pinning current. This effect is also experimentally observed in [137] for PMA magnets. Such p-bit design 1 with bipolar PMA and STT pinning might not work for Bayesian networks in general, since in this case τ_{step} will be dependent on magnet fluctuation time τ_N.

- In the proposed behavioral model, the step response time of the NMOS transistor τ_T in design 1 is assumed to be independent of the input I. But there is a functional dependence of τ_T on I in real hardware.

- The NMOS transistor resistance r_T is approximated as a tanh function for simplicity. In order to capture the hardware behavior in a better way, the *tanh* can be replaced by more complicated function and the weight matrix $[J]$ will have to be learnt around that function.

All the non-ideal effects listed above are supposed to have minimal effect on different probability distributions shown in this article, specially the effect of pinning current in devices with circular in-plane LBMs is negligible. The autonomous BN is also quite tolerant to variation in average magnet fluctuation time τ_N as long as $\tau_T \ll min(\tau_N)$. In general the depth of the network would also introduce effective propagation delays at different subsequent layers from the very first parent layer. The general criteria for designing an autonomous Bayesian network would be to choose the transistor and LBM in the pbit design in such a way so that $(\tau_T L) \ll \tau_N$ where L is the number of layers in the network.

Acknowledgment

This work was supported in part by ASCENT, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

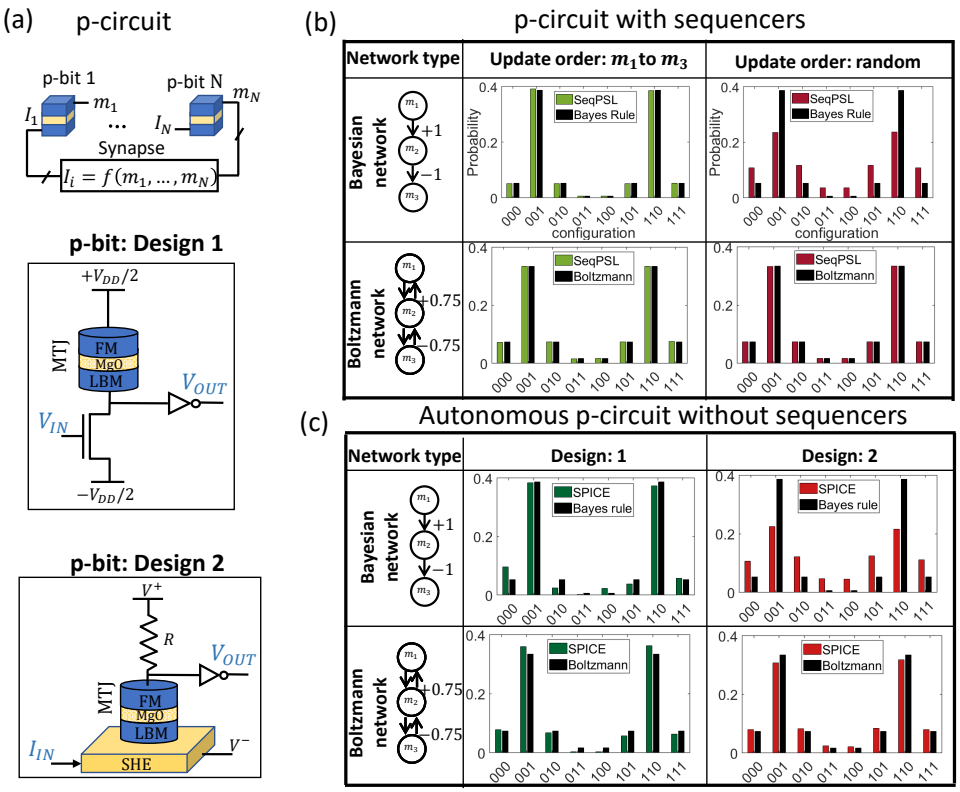


Fig. 5.1.: **Clocked versus Autonomous p-circuit**: (a) a probabilistic (p-)circuit is composed of p-bits interconnected by a weight logic/synapse that computes the input I_i to the i^{th} p-bit as a function of the outputs from other p-bits. Two p-bit designs (design 1 and 2) based on sMTJ using LBM have been used to build a p-circuit. (b) Two types of p-circuits are built: a directed or Bayesian network and a symmetrically connected Boltzmann network. The p-circuits are sequential (labeled as SeqPSL) that means p-bits are updated sequentially one at a time using a clock circuitry/sequencer. It is shown that for Boltzmann networks update order does not matter and any random update order would produce the correct probability distribuiton. But for Bayesian networks, a specific, parent-to-child update order is necessary to converge to the correct probability distribution from applying standard probabilty chain rule or Bayes rule. (c) The same Bayesian and Boltzmann p-circuits are implemented on an autonomous hardware built with p-bit design 1 and 2 without any clocks/sequencers. It is interesting to note that for Bayesian networks, design 2 fails to match the probabilities from applying Bayes rule, whereas design 1 works quite well as an autonomous Bayesian network.

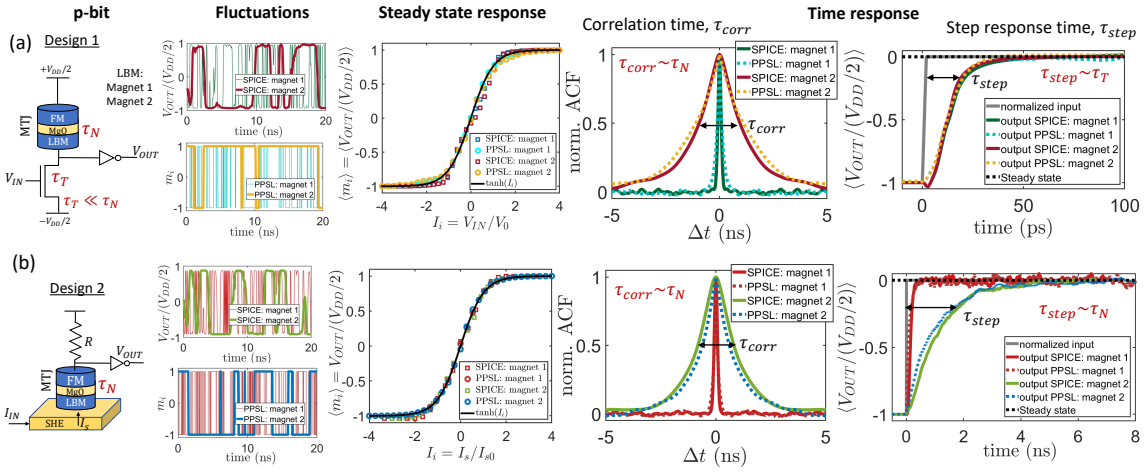


Fig. 5.2.: **Autonomous behavioral model for p-bit: Design 1 and 2**: (a) Behavioral model for the autonomous hardware with design 1 is benchmarked with SPICE simulation of the actual device involving experimentally benchmarked modules. The behavioral model (labeled as 'PPSL') shows good agreement with SPICE in terms of capturing fluctuation dynamics, steady state sigmoidal response, and two different time responses: autocorrelation time of the fluctuating output under zero input condition labeled as τ_{corr} which is proportional to the LBM retention time τ_N in the nanosecond range and the step response time τ_{step} defined by the transistor response time τ_T which is few picoseconds and much smaller than τ_N. The magnet parameters used in the simulations are mentioned in section 5.2 (b) Similar benchmarking for p-bit design 2. In this case τ_{step} is proportional to τ_N.

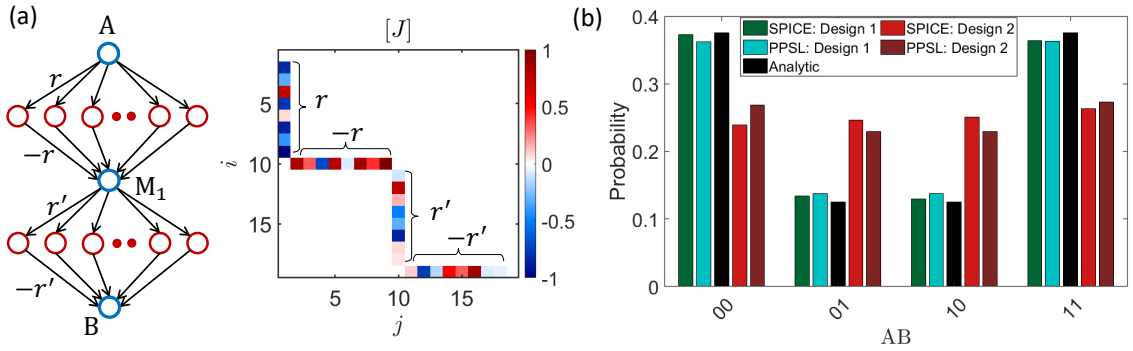


Fig. 5.3.: **Difference between design 1 and design 2:** (a) The behavioral models described in fig. 5.2 are applied to simulate a 19 p-bit BN with random J_{ij} between +1 and -1. The interconnections are designed in such a way so that pairs of intermediate nodes (A, M_1) and (M_1, B) get anti-correlated and (A, B) gets positively correlated. (b) The probability distribution of four configurations of AB are shown in a histogram from different approaches (SPICE, behavioral model and analytic). The behavioral models for two designs (labeled as PPSL) match reasonably well with the corresponding results from SPICE simulation of the actual hardware. Note that While design 1 matches with the standard analytical values quite well, design 2 does not works as an autonomous Bayesian network in general.

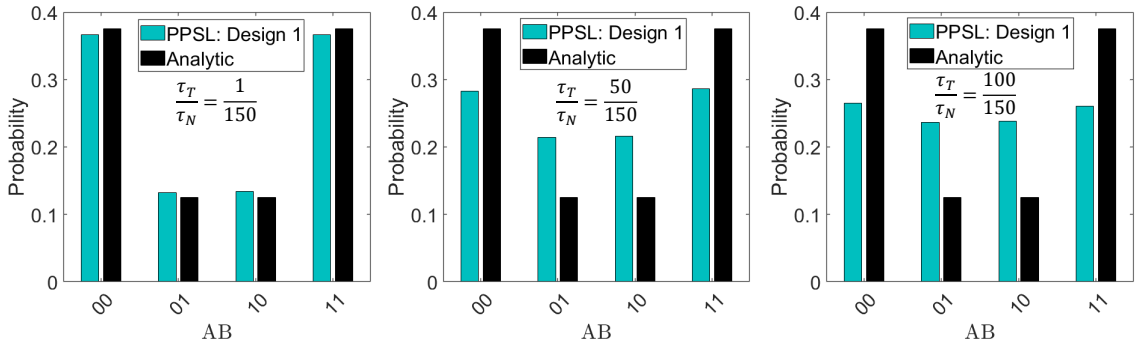


Fig. 5.4.: **Effect of step response time in design 1:**The reason for design 1 to work accurately as an anutonomous Bayesian network as shown in fig. 5.3 is the two different time scales (τ_T and τ_N) in this design with the condition that $\tau_T \ll \tau_N$. The same histogram shown in fig. 5.3 is plotted using the proposed behavioral model for different τ_T/τ_N ratios and compared with the analytical values. It can be seen that as τ_T gets comparable to τ_N, the probility distribution diverges from the standard statistical values.

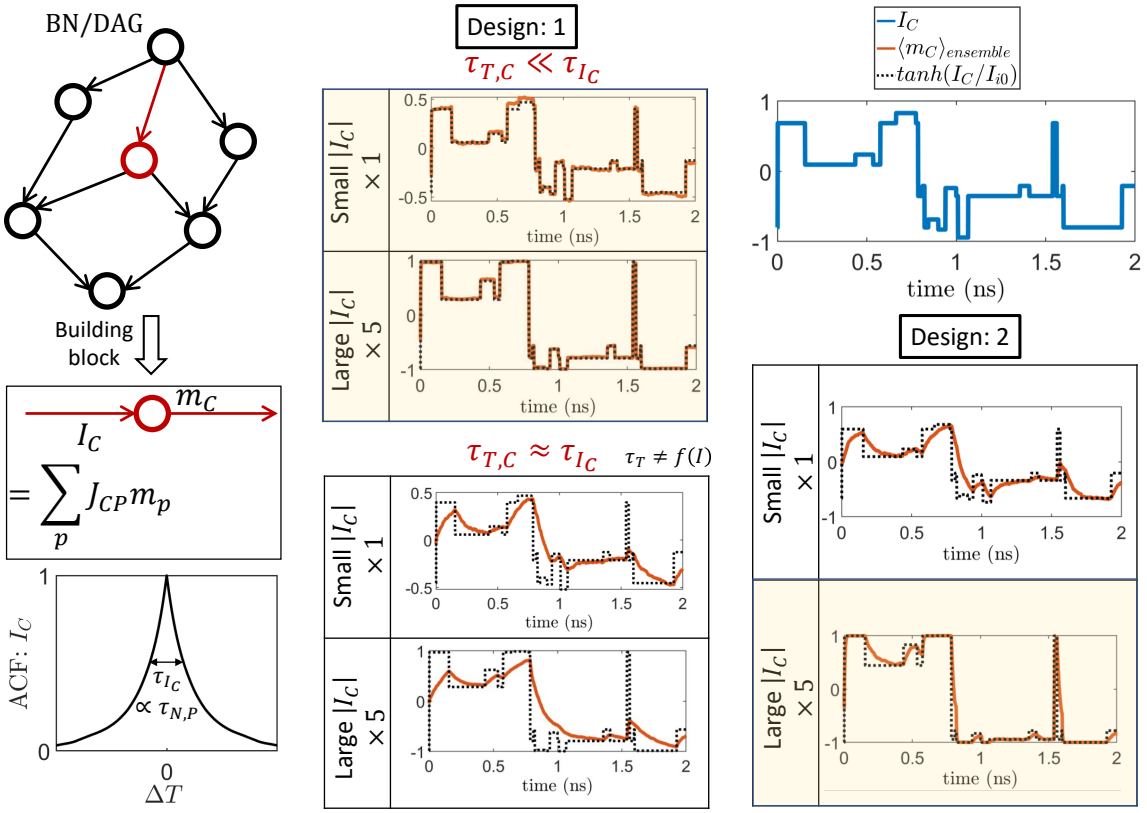
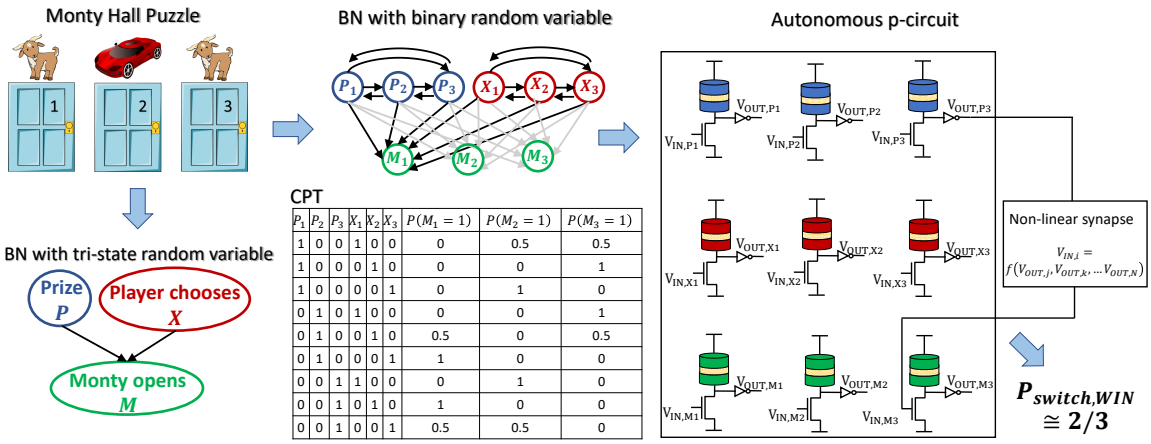


Fig. 5.5.: **Comparing time dynamics of design 1 and 2 in a BN**: The building blocks of a BN/DAG are the child nodes (C) given their input (I_C) as function of parent node outputs (m_p). In design 1, step response time (τ_T) is much smaller than magnet fluctuation time (τ_N) because NMOS response time is usually few picoseconds. That's why any time there is a change in the input I_C, child node can immediately respond to it and be conditionally satisfied always resulting in correct probability distribution consistent with standard Bayes rule. On the other hand, for design 2, τ_T is comparable to τ_N unless I_C is fluctuating between very large values always which is not applicable in general. That's why the child node does not get enough time to respond to a particular I_C value before another new I_C values comes, thus being conditionally unsatisfied majority of the time and fails to match Bayes rule in general.



P_1	P_2	P_3	X_1	X_2	X_3	$P(M_1=1)$	$P(M_2=1)$	$P(M_3=1)$
1	0	0	1	0	0	0	0.5	0.5
1	0	0	0	1	0	0	0	1
1	0	0	0	0	1	0	1	0
0	1	0	1	0	0	0	0	1
0	1	0	0	1	0	0.5	0	0.5
0	1	0	0	0	1	1	0	0
0	0	1	1	0	0	0	1	0
0	0	1	0	1	0	1	0	0
0	0	1	0	0	1	0.5	0.5	0

Fig. 5.6.: **Solving Monty Hall Puzzle using proposed autonomous hardware**: Monty Hall figure is taken from this link.

6. CONFIGURATION MATRIX ANALYSIS OF P-CIRCUIT TIME DYNAMICS

Recently it has been shown that probabilistic circuits or p-circuits based on unstable stochastic units can be used to solve a wide variety of problems: not only non-Boolean logic such as optimization, inference, but also precise Boolean logic with invertibility. In this work we have applied a general formalism based on the transition matrix of the configuration space of a p-circuit to analyze different types of p-circuits with varying degree of bidirectionality. The method can be used for calculating steady state probability distribution of the circuit and estimating convergence delay of the network. Using this method we have quantitatively established the fact that directed network provides faster convergence than bidirectional network. The effect of sequential and random updating order of the nodes in the p-circuit can also be captured by this method. Although this method is suitable for small sized networks because the transition matrix grows as 2^N, where N is the number of nodes in the circuit, it can be used to get a lot of insights about the dynamics of various kinds of p-circuits. Also the results obtained from smaller circuits can be projected to larger circuits.

Probabilistic Spin Logic (PSL) [4] is defined by the two equations:

$$m_i(n+1) = \text{sgn}\Big(\tanh(I_i(n+1)) - r\Big) \tag{6.1}$$

$$I_i(n+1) = I_0\Big(\sum_j J_{ij} m_j(n) + h_i\Big) \tag{6.2}$$

where, m_i is the stochastic output of the p-bit with the stochasticity tuned by the input I_i and r is a uniformly distributed random number between -1 and $+1$. Input I_i is a weighted summation of other p-bit outputs m_j with coupling strengths denoted

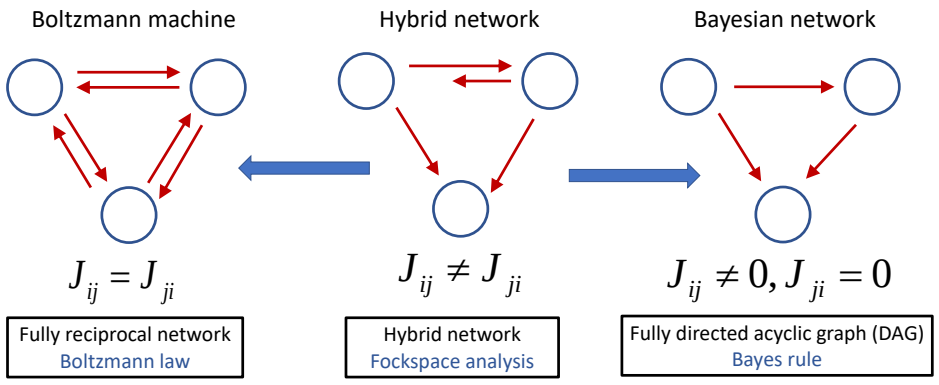


Fig. 6.1.: **Fockspace analysis for p-circuit with varying degree of bidirectionality:** p-circuits or binary stochastic networks can be classified into three catagories: Boltzmann machines (BM) with symmetrical interconnections, Bayesian networks (BN) with directed acycllic connections and hybrid network with both bidirectional and directied connections. While standard Boltzmann law and Bayes rule is applicable for analyzing steady state response of only BMs and BNs respectively, Fockspace analysis is in general applicable to any stochastic network with varying degree of bidirectionality in terms of understanding both steady state and transient behavior.

by J_{ij}. h_i is the local bias to the p-bit and I_0 is a global coupling strength similar to a pseudo inverse temperature.

p-bits can be interconnected according to eqn. 6.2 to construct different types of p-circuits with various degree of bidirectionalilty in the connection scheme, for example symmetrically connected Boltzmann machines, fully directed and acyclic Bayesian networks and hybrid networks with both directed and bidirectional connections. This work describes a general method based on the Transition matrix [151] of the configuration space or Fockspace of the p-circuit adopted from [39] to study the steady state and transient response of different types of p-circuit with varying degree of bidirectionality. Results from this method matches with Boltzmann law for symmetrically connected network and with Bayes rule for directed acyclic networks.

6.1 Method of constructing the configuration space matrix

6.1.1 For sequential updating of p-bits

For sequential updating each p-bit has a configuration space matrix $[W]_i$ associated with it where i stands for the i^{th} p-bit. $[W]_i$ is a $2^N \times 2^N$ matrix where N is the total number of p-bits in the system. The full configuration space matrix $[W]$ is a product of all the individual $[W]_i$s based on the specific update order. For example, for a update order of $i \rightarrow j \rightarrow k$ in a three p-bit network, $[W] = [W]_k[W]_j[W]_i$.

Constructing $[W]_i$ for the i^{th} p-bit: For one specific row in W_i corresponding to one specific p-bit configuration at discrete time point $n+1$, the 2^N columns corresponds to 2^N configurations of N p-bits at discrete time point n with column 1 starting from $\sigma_1 \in (m_1, m_2, ..., m_N)_{\sigma_1} = (-1, -1, ..., -1)$ to column 2^N ending with $\sigma_{2^N} \in (m_1, m_2, ..., m_N)_{\sigma_{2^N}} = (1, 1, ..., 1)$. Each column in $[W]_i$ will have two probabilities associated with it: p_i and $\bar{p_i}$ with $p_i + \bar{p_i} = 1$.

For the k^{th} column,

$$p_{i,\sigma_k} = \frac{1 + \tanh(I_{i,\sigma_k})}{2}$$
$$I_{i,\sigma_k} = I_0\Big(\sum_{\substack{m_j \in \sigma_k \\ j \neq i}} J_{ij} m_j + h_i \Big)$$

The row number related to p_{i,σ_k} in the k^{th} column will be given by:

$$\text{row}_{p_{i,\sigma_k}} = \Big(\sum_{\substack{j=1 \\ (m_{j=i})=1 \\ (m_{j\neq i}) \in \sigma_k}}^{N} 2^{N-j} m_j \Big) + 1$$

Similarly,

$$\overline{p_{i,\sigma_k}} = 1 - p_{i,\sigma_k}$$

And the corresponding row number will be defined by:

$$\text{row}_{\overline{p_i,\sigma_k}} = \Big(\sum_{\substack{j=1 \\ (m_{j=i})=-1 \\ (m_{j\neq i})\in\sigma_k}}^{N} 2^{N-j} m_j \Big) + 1$$

6.1.2 For simultaneous updating of p-bits

In this case each element W_{σ_l,σ_k} corresponding to row configuration σ_l and column configuration σ_k in the configuration matrix $[W]$ will be defined by:

$$W_{\sigma_l,\sigma_k} = \prod_{\substack{j=1 \\ m_j\in\sigma_l}}^{N} \Big(\frac{1+m_j}{2} p_j + \frac{1-m_j}{2}(1-p_j) \Big)$$

$$p_j = \frac{1+\tanh(I_j)}{2}$$

$$I_j = I_0 \Big(\sum_{\substack{m_i\in\sigma_k \\ i\neq j}} J_{ji} m_i + h_j \Big)$$

6.1.3 Steady state response

After constructing the $[W]$ matrix, the steady state probabilities of the 2^N configurations of the network $\{P\}_{\text{steady}}$ will be the eigenvector of $[W]$ corresponsing to the eigenvalue $\lambda = 1$. The fact that each column sums up to 1 in the $[W]$ matrix makes sure that $[W]$ always has the maximum eigenvalue of 1 along with other smaller eigenvalues. In the long run, only the eigenvector corresponding to $\lambda = 1$ will sustain and other eigenvectors will die out. $\{P\}_{\text{steady}}$ will have to be normalized by the sum of all it's elements to make sure that all probabilities add up to 1.

For random (randperm) update order in sequential updating, $[W]$ matrix will have to be generated for all of the $N!$ permulations of update orders and all the $\{P\}_{\text{steady}}$s will have to be averaged.

6.1.4 Transient response

If $\{p_0\}$ is the initial probability distribution at discrete time point $n = 0$, the subsequent probability distributions at n^{th} time point can be found by:

$$\begin{aligned}\{P_n\} &= [W]\{P_{n-1}\} \\ &= [W]^n\{p_0\}\end{aligned}$$

6.2 Results

To benchmark our proposed Configuration/Fockspace matrix analysis method described in section 6.1, we have chosen the Sherrington Kirkpatrick (SK) ising model [152] as an example. In the SK model, the coupling matrix $[J]$ between spins consists of Gaussian distributed random numbers. Our example consists of a $N_m = 7$ p-bit system with coupling strength $J_{ij} \in \text{rand}_{[-1,+1]}$ and local bias $h_i \in \text{rand}_{[-1,+1]}$.

6.2.1 Steady state response

Fig. 6.2 shows the steady state probability distribution of four configurations $(00, 01, 10, 11)$ of (m_1, m_7) for three types of networks: (1) symmetrically connected Boltzmann machine (BM), (2) fully directed acyclic Bayesian network (BN) by considering the lower triangular part of the $[J]$ matrix of BM and (3) a hybrid network (HN) with both bidiectional and directed connections. The probability distributions obtained from the Fockspace method for different update orders of p-bits (specific sequence, random and simultaneous) are compared against the PSL simultion defined by eqn. 6.1 and eqn. 6.2 (labeled as SeqPSL for sequential updating and SimPSL for simultaneous updating) and standard statistical values. For BMs and BNs the standard results are obtained from equllibrium Boltzmann Law and probability chain rule (labeled as Bayes rule) respectively. For HNs, neither Boltzmann law nor Bayes rule will be applicable. Figure 6.2 shows that Fockspace analysis is applicable for any topology of binary stochastic networks with varying degree of bidirectionality.

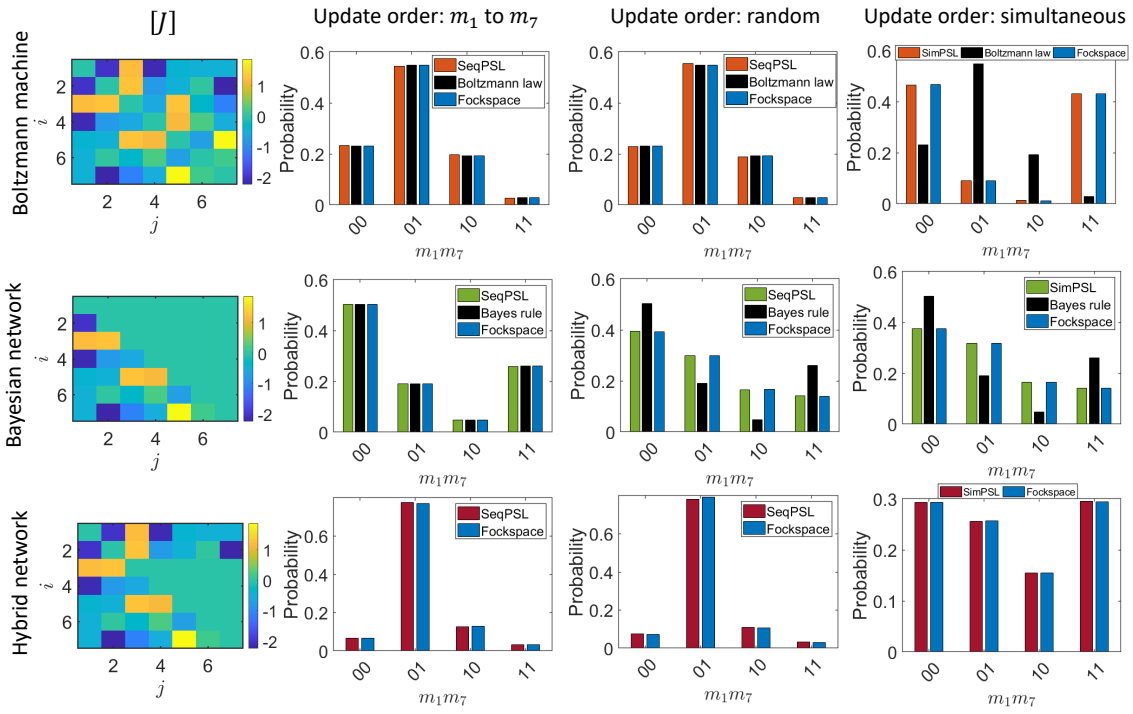


Fig. 6.2.: **Steady state response:** The steady state response of three types of networks (symmetrical, directed acyclic and hybrid) composed of 7 p-bits are shown for different update orders (m_1 to m_7, random and simultaneous) where each histogram shows the probabilities of four configurations of (m_1, m_7). It can be seen that for Boltzmann machines the update order does not matter as long as p-bits are updated sequentially and the Fockspace results match with corrresponding PSL simulaitons and standard Boltzmann law. However the system fails to match Boltzmann law if p-bits are updated simultaneously. In this case also Fockspace analysis nicely captures p-circuit dynamics with simultaneous update . For Bayesian networks a parent to child node update order is important in terms of matching standard Bayes rule [136]. Fockspace analysis matches PSL results for all three types of networks for different update orders.

It can be seen from fig. 6.2 that for Boltzmann networks, update order of p-bits does not matter as long as p-bits are updated sequentially one at a time. The prob-

ability distribution of (m_1, m_7) configurations obtained from the PSL approach over long time averaging matches nicely with those from Boltzmann law and Fockspace analysis with the same update order. For simulataneous updating of the p-bits, results from PSL does not match with Boltzmann law, but Fockspace method nicely captures the simultaneous updating behavior. For Bayesian networks, the update order is important in terms of matching Bayes rule. The steady state probability distribution will match Bayes rule as long as the p-bits are updated from the *parent* to *child* node [136]. Fockspace analysis nicely captures different update order issues and matches with PSL results. For hybrid networks with both bidirectional and directed connections, neither Boltzmann law nor Bayes rule is applicable. But in this case also, Fockspace analysis matches with the PSL results quite well.

6.2.2 Transient response

For the three types of networks described in fig. 6.2 (BM, BN and HN), the correlation of m_1 and m_7 is plotted as a function of discrete time steps for update order from m_1 to m_7. At each time point $\langle m_1 m_7 \rangle$ is obtained from PSL by averaging over 8000 ensembles. For all three networks, PSL time dynamics matches nicely with Fockspace analysis. For BMs and BNs, the steady state correlation value matches well with value from boltzmann law and Bayes rule respectively. Note that the convergence time of BMs is longer than that of BNs. The convergence time of HNs fall in between BMs and BNs as shown in fig. 6.3

6.3 Discussion:

We have shown that Fockspace analysis nicely captures the steady state and time dynamics of binary stochastic networks or p-circuits with any type of coupilng in general and any update order scheme. The Fockspace approach is useful for understanding the behavior of small scale p-circuits, but the approach does not scale up to large p-circuits as the configuration space matrix dimension grows as 2^N. Recently

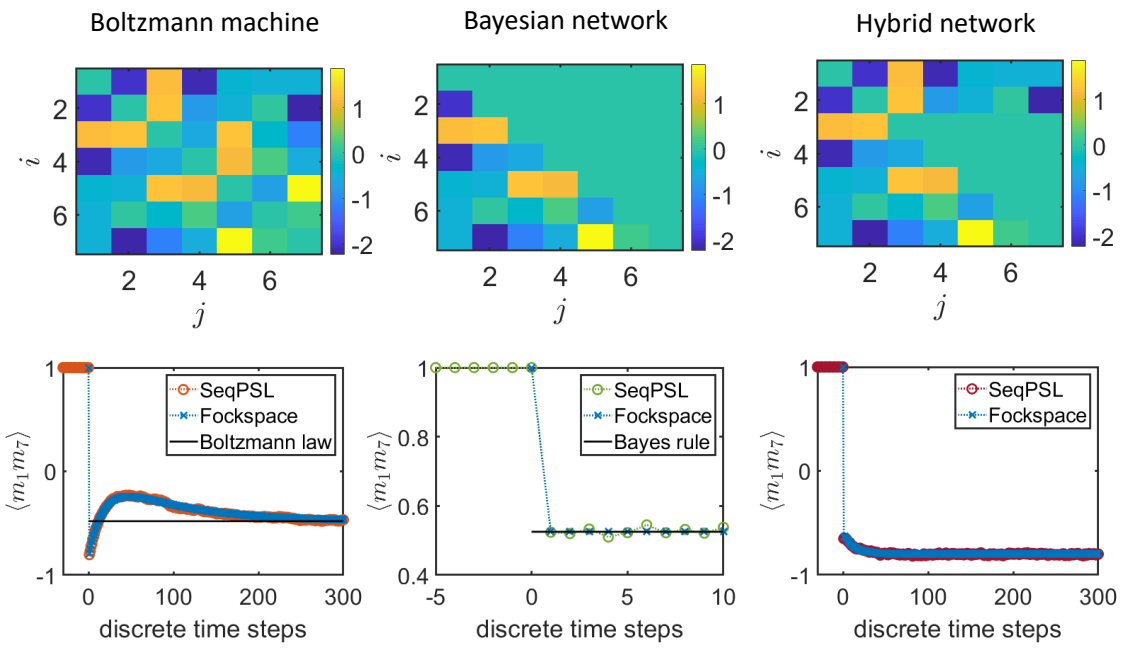


Fig. 6.3.: **Transient response:** The transient response and convergence time of the three types of networks presented in fig. 6.2 are shown from PSL simulation with m_1 to m_7 update order and compared againsed Fockspace analysis. In all cases, Fockspace method nicely captures the PSL time dynamics.

a new type of autonomous probabilistic circuit without any clocks unlike sequential p-circuit is proposed [6]. Whether Fockspace analysis will be applicable to such autonomous p-circuits will be an interesting topic of future research.

7. SUMMARY

In this thesis we have established a new computing framework called Probabilistic Spin Logic [4, 101] built out of p-bits for solving a wide variety of problems such as optimization, inference, invertible Boolean logic and sampling. p-bits are analogous to binary stochastic neurons (BSN) in artificial neural network (ANN) literature. p-bits can be interconnected accroding to a synapse or weight matrix $[J]$ to construct p-circuits. For proper operation of the p-circuits, p-bits need to be updated sequentially one at a time in an informed way so that when one p-bit updates it has the information of the states of other p-bits that it is connected to. To ensure this sequential operation, conventional digital hardware uses clocks/sequencers.

As opposed to the clocked implementation on digital hardware, our proposed probabilistic hardware is completely autonomous that runs without any clocks. To ensure the necessary sequential informed update of p-bits in our autonomous probabilistic hardware, it is important that the synapse delay is much smaller than the BSN fluctuation time. We have demonstrated the operation of autonomous p-circuit on various applications by performing SPICE simulation of the hardware composed of different p-bit desings using experimentally benchmarked modules.

We have shown how a p-circuit defined by the BSN and synapse equation can be mapped to a low barrier nanomagnet (LBM) based probabilistic circuit by solving coupled stochastic Landau-Lifshitz-Gilbert (sLLG) equations and benchmarking the joint probability distribution of the system with a coupled Fokker-Planck equation (FPE). By implementing an invertible 32-bit Adder network out of 32 full adders each being a Boltzmann machine composed of 14 bidirectionally connected p-bits using a total of 448 coupled LBMs, we have shown that large scale correlations are possible even when p-bit outputs are continuous and not perfectly bipolar.

SPICE simulation of our proposed autonomous p-circuits involving sLLG equations for capturing LBM physics is computationally expensive and time consuming limitting the scope of exploring very large scale p-circuits. This limitation necessiates the use of compact models for these hardware to demonstrate their scalability. The standard behavioral models commonly used in digital hardware to simulate stochastic networks use sequencers and thus inadequete for capturing our autonomous clockless hardware behavior. In this regard, we have proposed and carefully benchmarked compact models for two different autonomous p-circuits composed of two different previously proposed p-bit designs that faithfully mimic the SPICE simulation of the real hardware using experimentally benchmarked modules. These two compact models are important for exploring very large-scale p-circuits.

We also talk about the dynamics of different p-circuits: directed (Bayesian) and bidirectional (Boltzmann) and shown that a specific parent to child update order is very important for Bayesian networks unlike Boltzmann machines. We have shown how Bayesian networks defined by conditional probability tables (CPT) can be mapped to PSL coupling parameters J and h and then mapped to an embedded MRAM based autonomous hardware and benchmarked SPICE simulation results against those from standard statistical calculations. We have found that unlike bidirectional networks where update order of p-bits does not matter as long as they are sequential, for Bayesian networks a specific parent to child update order is necessary and proposed autonomous hardware design criteria for naturally ensuring this specific update order without any sequencers.

Lastly, we have applied a configuration space or Fockspace method to understand the dynamics of different types of p-circuits with varying degree of bidirectionality and various update orders of p-bits and benchmarked results from this approach with standard PSL model. We have shown that bidirectional networks converge slower than directed networks. Update order of p-bits does not matter for bidirectional

networks in terms of matching standard statistical results. But for fully directed acyclic networks, a specific parent to child update order is necessary for matching results from applying standard Bayes rule. The Fockspace method captures the PSL steady state and time response for different types of networks with different update orders quite well.

REFERENCES

REFERENCES

[1] D. Bromberg, M. Moneck, V. Sokalski, J. Zhu, L. Pileggi, and J. Zhu, "Experimental demonstration of four-terminal magnetic logic device with separate read-and write-paths," in *2014 International Electron Devices Meeting, San Francisco, CA, Session*, vol. 33, 2014.

[2] S. Datta, S. Salahuddin, and B. Behin-Aein, "Non-volatile spin switch for boolean and non-boolean logic," *Applied Physics Letters*, vol. 101, no. 25, p. 252411, 2012.

[3] D. E. Nikonov and I. A. Young, "Benchmarking of beyond-cmos exploratory devices for logic integrated circuits," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 1, pp. 3–11, 2015.

[4] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, "Stochastic p-bits for invertible logic," *Physical Review X*, vol. 7, no. 3, p. 031014, 2017.

[5] "Binary Stochastic Neuron," 2016. [Online]. Available: https://r2rt.com/binary-stochastic-neurons-in-tensorflow.html

[6] B. Sutton, R. Faria, L. A. Ghantasala, K. Y. Camsari, and S. Datta, "Autonomous probabilistic coprocessing with petaflips per second," *arXiv preprint arXiv:1907.09664*, 2019.

[7] K. Y. Camsari, S. Salahuddin, and S. Datta, "Implementing p-bits with embedded mtj," *IEEE Electron Device Letters*, vol. 38, no. 12, pp. 1767–1770, 2017.

[8] K. Palem and A. Lingamneni, "Ten years of building broken chips: the physics and engineering of inexact computing," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 2s, p. 87, 2013.

[9] S. Cheemalavagu, P. Korkmaz, K. V. Palem, B. E. Akgul, and L. N. Chakrapani, "A probabilistic cmos switch and its realization by exploiting noise," in *IFIP International Conference on VLSI*, 2005, pp. 535–541.

[10] Z. Weijia, G. W. Ling, and Y. K. Seng, "Pcmos-based hardware implementation of bayesian network," in *2007 IEEE Conference on Electron Devices and Solid-State Circuits*. IEEE, 2007, pp. 337–340.

[11] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, "Intrinsic optimization using stochastic nanomagnets," *arXiv preprint arXiv:1608.00679*, 2016.

[12] R. Faria, K. Y. Camsari, and S. Datta, "Low barrier nanomagnets as p-bits for spin logic," *arXiv preprint arXiv:1611.05477*, 2016.

[13] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[14] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[15] N. Locatelli, A. Mizrahi, A. Accioly, R. Matsumoto, A. Fukushima, H. Kubota, S. Yuasa, V. Cros, L. G. Pereira, D. Querlioz *et al.*, "Noise-enhanced synchronization of stochastic magnetic oscillators," *Physical Review Applied*, vol. 2, no. 3, p. 034009, 2014.

[16] M. Bapna, S. K. Piotrowski, S. D. Oberdick, M. Li, C.-L. Chien, and S. A. Majetich, "Magnetostatic effects on switching in small magnetic tunnel junctions," *Applied Physics Letters*, vol. 108, no. 2, p. 022406, 2016.

[17] S. K. Piotrowski, M. F. Matty, and S. A. Majetich, "Magnetic fluctuations in individual superparamagnetic particles," *IEEE Transactions on Magnetics*, vol. 50, no. 11, pp. 1–4, 2014.

[18] W. H. Choi, Y. Lv, J. Kim, A. Deshpande, G. Kang, J.-P. Wang, and C. H. Kim, "A magnetic tunnel junction based true random number generator with conditional perturb and real-time output probability tracking," in *2014 IEEE International Electron Devices Meeting*. IEEE, 2014, pp. 12–5.

[19] A. Mizrahi, N. Locatelli, R. Lebrun, V. Cros, A. Fukushima, H. Kubota, S. Yuasa, D. Querlioz, and J. Grollier, "Controlling the phase locking of stochastic magnetic bits for ultra-low power computation," *Scientific Reports*, vol. 6, 2016.

[20] G. Srinivasan, A. Sengupta, and K. Roy, "Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip stdp learning," *Scientific Reports*, vol. 6, 2016.

[21] A. F. Vincent, J. Larroque, N. Locatelli, N. B. Romdhane, O. Bichler, C. Gamrat, W. S. Zhao, J.-O. Klein, S. Galdin-Retailleau, and D. Querlioz, "Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems," *IEEE transactions on biomedical circuits and systems*, vol. 9, no. 2, pp. 166–174, 2015.

[22] 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, pp. 980–991, 2015.

[23] N. Locatelli, A. F. Vincent, A. Mizrahi, J. S. Friedman, D. Vodenicarevic, J.-V. Kim, J.-O. Klein, W. Zhao, J. Grollier, and D. Querlioz, "Spintronic devices as key elements for energy-efficient neuroinspired architectures," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 994–999.

[24] L. Lopez-Diaz, L. Torres, and E. Moro, "Transition from ferromagnetism to superparamagnetism on the nanosecond time scale," *Physical Review B*, vol. 65, no. 22, p. 224406, 2002.

[25] A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa, and K. Ando, "Spin dice: A scalable truly random number generator based on spintronics," *Applied Physics Express*, vol. 7, no. 8, p. 083001, 2014.

[26] 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.

[27] J. Roychowdhury, private communication.

[28] For an example of the use of "invertible relations", see Ran Canetti and Mayank Varia, "Non-malleable obfusca-tion," inTheory of Cryptography Conference(Springer,2009) pp. 73–90.

[29] B. Behin-Aein, V. Diep, and S. Datta, "A building block for hardware belief networks," *Scientific Reports*, vol. 6, p. 29893, 2016.

[30] Y. Shim, A. Jaiswal, and K. Roy, "Ising spin model using spin-hall effect (she) induced magnetization reversal in magnetic-tunnel-junction," *arXiv preprint arXiv:1609.05926*, 2016.

[31] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature nanotechnology*, vol. 8, no. 1, pp. 13–24, 2013.

[32] V. Q. Diep, B. Sutton, B. Behin-Aein, and S. Datta, "Spin switches for compact implementation of neuron and synapse," *Applied Physics Letters*, vol. 104, no. 22, p. 222405, 2014.

[33] A. Sengupta, Y. Shim, and K. Roy, "Proposal for an all-spin artificial neural network: Emulating neural and synaptic functionalities through domain wall motion in ferromagnets," *IEEE Transactions on Biomedical Circuits and Systems*, 2016.

[34] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, "Ising computer," *Hitachi Review*, vol. 65, no. 6, p. 157, 2016.

[35] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[36] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, "24.3 20k-spin ising chip for combinational optimization problem with cmos annealing," in *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*. IEEE, 2015, pp. 1–3.

[37] T. Inagaki, K. Inaba, R. Hamerly, K. Inoue, Y. Yamamoto, and H. Takesue, "Large-scale ising spin network based on degenerate optical parametric oscillators," *Nature Photonics*, 2016.

[38] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.

[39] D. J. Amit and D. J. Amit, *Modeling brain function: The world of attractor neural networks*. Cambridge university press, 1992.

[40] D. Du, J. Gu, P. M. Pardalos *et al.*, *Satisfiability problem: theory and applications: DIMACS Workshop, March 11-13, 1996*. American Mathematical Soc., 1997, vol. 35.

[41] "Predictive Technology Model (PTM)." [Online]. Available: http://ptm.asu.edu/

[42] L. Liu, C.-F. Pai, Y. Li, H. Tseng, D. Ralph, and R. Buhrman, "Spin-torque switching with the giant spin hall effect of tantalum," *Science*, vol. 336, no. 6081, pp. 555–558, 2012.

[43] A. van den Brink, G. Vermijs, A. Solignac, J. Koo, J. T. Kohlhepp, H. J. Swagten, and B. Koopmans, "Field-free magnetization reversal by spin-hall effect and exchange bias," *Nature communications*, vol. 7, 2016.

[44] Y.-C. Lau, D. Betto, K. Rode, J. Coey, and P. Stamenov, "Spin–orbit torque switching without an external field using interlayer exchange coupling," *Nature nanotechnology*, 2016.

[45] A. K. Smith, M. Jamali, Z. Zhao, and J.-P. Wang, "External field free spin hall effect device for perpendicular magnetization reversal using a composite structure with biasing layer," *arXiv preprint arXiv:1603.09624*, 2016.

[46] S. Fukami, C. Zhang, S. DuttaGupta, A. Kurenkov, and H. Ohno, "Magnetization switching by spin-orbit torque in an antiferromagnet-ferromagnet bilayer system," *Nature materials*, 2016.

[47] J. Heron, J. Bosse, Q. He, Y. Gao, M. Trassin, L. Ye, J. Clarkson, C. Wang, J. Liu, S. Salahuddin *et al.*, "Deterministic switching of ferromagnetism at room temperature using an electric field," *Nature*, vol. 516, no. 7531, p. 370, 2014.

[48] S. Manipatruni, D. E. Nikonov, and I. A. Young, "Spin-orbit logic with magnetoelectric nodes: A scalable charge mediated nonvolatile spintronic logic," *arXiv preprint arXiv:1512.05428*, 2015.

[49] R. H. Koch, G. Grinstein, G. Keefe, Y. Lu, P. Trouilloud, W. Gallagher, and S. Parkin, "Thermally assisted magnetization reversal in submicron-sized magnetic thin films," *Physical review letters*, vol. 84, no. 23, p. 5419, 2000.

[50] S. Urazhdin, N. O. Birge, W. Pratt Jr, and J. Bass, "Current-driven magnetic excitations in permalloy-based multilayer nanopillars," *Physical review letters*, vol. 91, no. 14, p. 146803, 2003.

[51] I. Krivorotov, N. Emley, A. Garcia, J. Sankey, S. Kiselev, D. Ralph, and R. Buhrman, "Temperature dependence of spin-transfer-induced switching of nanomagnets," *Physical review letters*, vol. 93, no. 16, p. 166603, 2004.

[52] A. Khvalkovskiy, D. Apalkov, S. Watts, R. Chepulskii, R. Beach, A. Ong, X. Tang, A. Driskill-Smith, W. Butler, P. Visscher *et al.*, "Basic principles of stt-mram cell operation in memory arrays," *Journal of Physics D: Applied Physics*, vol. 46, no. 7, p. 074001, 2013.

[53] R. Cowburn, "Property variation with shape in magnetic nanoelements," *Journal of Physics D: Applied Physics*, vol. 33, no. 1, p. R1, 2000.

[54] P. Debashis, R. Faria, K. Y. Camsari, J. Appenzeller, S. Datta, and Z. Chen, "Experimental demonstration of nanomagnet networks as hardware for ising computing," in *2016 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2016, pp. 34–3.

[55] K. Y. Camsari, S. Ganguly, and S. Datta, "Modular approach to spintronics," *Scientific reports*, vol. 5, 2015.

[56] L. Liu, T. Moriyama, D. Ralph, and R. Buhrman, "Spin-torque ferromagnetic resonance induced by the spin hall effect," *Physical review letters*, vol. 106, no. 3, p. 036601, 2011.

[57] S. Hong, S. Sayed, and S. Datta, "Spin circuit representation for the spin hall effect," *IEEE Transactions on Nanotechnology*, vol. 15, no. 2, pp. 225–236, 2016.

[58] W. H. Butler, T. Mewes, C. K. Mewes, P. Visscher, W. H. Rippard, S. E. Russek, and R. Heindl, "Switching distributions for perpendicular spin-torque devices within the macrospin approximation," *IEEE Transactions on Magnetics*, vol. 48, no. 12, pp. 4684–4700, 2012.

[59] A. D. Kent and D. C. Worledge, "A new spin on magnetic memories," *Nature nanotechnology*, vol. 10, no. 3, pp. 187–191, 2015.

[60] D. Morris, D. Bromberg, J.-G. Zhu, and L. Pileggi, "mlogic: Ultra-low voltage non-volatile logic circuits using stt-mtj devices," in *DAC Design Automation Conference 2012*. IEEE, 2012, pp. 486–491.

[61] D. Datta, B. Behin-Aein, S. Datta, and S. Salahuddin, "Voltage asymmetry of spin-transfer torques," *IEEE Transactions on Nanotechnology*, vol. 11, no. 2, pp. 261–272, 2011.

[62] J. Biamonte, "Nonperturbative k-body to two-body commuting conversion hamiltonians and embedding problem instances into ising spins," *Physical Review A*, vol. 77, no. 5, p. 052331, 2008.

[63] K.-U. Demasius, T. Phung, W. Zhang, B. P. Hughes, S.-H. Yang, A. Kellock, W. Han, A. Pushp, and S. S. Parkin, "Enhanced spin–orbit torques by oxygen incorporation in tungsten films," *Nature communications*, vol. 7, no. 1, pp. 1–7, 2016.

[64] C.-F. Pai, L. Liu, Y. Li, H. Tseng, D. Ralph, and R. Buhrman, "Spin transfer torque devices utilizing the giant spin hall effect of tungsten," *Applied Physics Letters*, vol. 101, no. 12, p. 122404, 2012.

[65] Q. Hao and G. Xiao, "Giant spin hall effect and switching induced by spin-transfer torque in a w/co 40 fe 40 b 20/mgo structure with perpendicular magnetic anisotropy," *Physical Review Applied*, vol. 3, no. 3, p. 034009, 2015.

[66] T. J. Sejnowski, P. K. Kienker, and G. E. Hinton, "Learning symmetry groups with hidden units: Beyond the perceptron," *Physica D: Nonlinear Phenomena*, vol. 22, no. 1-3, pp. 260–275, 1986.

[67] S. Patarnello and P. Carnevali, "Learning networks of neurons with boolean logic," *EPL (Europhysics Letters)*, vol. 4, no. 4, p. 503, 1987.

[68] L. Personnaz, I. Guyon, and G. Dreyfus, "Collective computational properties of neural networks: New learning mechanisms," *Physical Review A*, vol. 34, no. 5, p. 4217, 1986.

[69] S. V. Aiyer, M. Niranjan, and F. Fallside, "A theoretical investigation into the performance of the hopfield model," *IEEE transactions on neural networks*, vol. 1, no. 2, pp. 204–215, 1990.

[70] H. Suzuki, J.-i. Imura, Y. Horio, and K. Aihara, "Chaotic boltzmann machines," *Scientific reports*, vol. 3, p. 1610, 2013.

[71] G. E. Hinton, "Boltzmann machine," *Scholarpedia*, vol. 2, no. 5, p. 1668, 2007, revision #91076.

[72] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[73] B. Sutton, K. Y. Camsari, R. Faria, and S. Datta, "Probabilistic spin logic simulator," 2017.

[74] J. Liu, S. Zhou, H. Zhu, and C.-K. Cheng, "An algorithmic approach for generic parallel adders," in *ICCAD-2003. International Conference on Computer Aided Design (IEEE Cat. No. 03CH37486)*. IEEE, 2003, pp. 734–740.

[75] R. Uma, V. Vijayan, M. Mohanapriya, and S. Paul, "Area, delay and power comparison of adder topologies," *International Journal of VLSI Design & Communication Systems*, vol. 3, no. 1, p. 153, 2012.

[76] D. E. Knuth and L. T. Pardo, "Analysis of a simple factorization algorithm," *Theoretical Computer Science*, vol. 3, no. 3, pp. 321–348, 1976.

[77] F. L. Traversa and M. Di Ventra, "Polynomial-time solution of prime factorization and np-complete problems with digital memcomputing machines," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 2, p. 023107, 2017.

[78] M. Di Ventra, F. L. Traversa, and I. V. Ovchinnikov, "Topological field theory and computing with instantons," *arXiv preprint arXiv:1609.03230*, 2016.

[79] A. Ekert and R. Jozsa, "Quantum computation and shor's factoring algorithm," *Reviews of Modern Physics*, vol. 68, no. 3, p. 733, 1996.

[80] R. P. Feynman, "Simulating physics with computers," *International journal of theoretical physics*, vol. 21, no. 6, pp. 467–488, 1982.

[81] C. Pan and A. Naeemi, "A proposal for energy-efficient cellular neural network based on spintronic devices," *IEEE Transactions on Nanotechnology*, vol. 15, no. 5, pp. 820–827, sep 2016.

[82] M. G. Mankalale, Z. Liang, and S. S. Sapatnekar, "Stem: A scheme for two-phase evaluation of majority logic," *arXiv preprint arXiv:1609.05141*, 2016.

[83] J. Z. Sun, "Spin-current interaction with a monodomain magnetic body: A model study," *Physical Review B*, vol. 62, no. 1, p. 570, 2000.

[84] Y. Bai and M. Lin, "Stochastic-based spin-programmable gate array with emerging mtj device technology," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016, pp. 279–279.

[85] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, "Stochastic p-bits for probabilistic spin logic," *arXiv preprint arXiv:1610.00377*, 2016.

[86] G. Bertotti, C. Serpico, I. D. Mayergoyz, A. Magni, M. d'Aquino, and R. Bonin, "Magnetization switching and microwave oscillations in nanomagnets driven by spin-polarized currents," *Phys. Rev. Lett.*, vol. 94, p. 127206, Apr 2005.

[87] D. Pinna, A. Kent, and D. Stein, "Thermally-assisted spin-transfer torque magnetization reversal of uniaxial nanomagnets in energy space," *IEEE transactions on magnetics*, vol. 49, no. 7, pp. 3144–3146, 2013.

[88] K. G. Murty and S. N. Kabadi, "Some np-complete problems in quadratic and nonlinear programming," *Mathematical programming*, vol. 39, no. 2, pp. 117–129, 1987.

[89] F. L. Traversa and M. Di Ventra, "Polynomial-time solution of prime factorization and np-hard problems with digital memcomputing machines," *arXiv preprint arXiv:1512.05064*, 2015.

[90] B. Behin-Aein, D. Datta, S. Salahuddin, and S. Datta, "Proposal for an all-spin logic device with built-in memory," *Nature nanotechnology*, vol. 5, no. 4, pp. 266–270, 2010.

[91] W. Scholz, T. Schrefl, and J. Fidler, "Micromagnetic simulation of thermally activated switching in fine particles," *Journal of Magnetism and Magnetic Materials*, vol. 233, no. 3, pp. 296–304, 2001.

[92] J. Z. Sun, "Spin angular momentum transfer in current-perpendicular nanomagnetic junctions," *IBM journal of research and development*, vol. 50, no. 1, pp. 81–100, 2006.

[93] W. F. Brown Jr, "Thermal fluctuations of a single-domain particle," *Journal of Applied Physics*, vol. 34, no. 4, pp. 1319–1320, 1963.

[94] V. Q. Diep, ""Transistor-like" spin nano-switches: Physics and applications," Ph.D. dissertation, Purdue University, Jan. 2015.

[95] A. Accioly, N. Locatelli, A. Mizrahi, D. Querlioz, L. G. Pereira, J. Grollier, and J.-V. Kim, "Role of spin-transfer torques on synchronization and resonance phenomena in stochastic magnetic oscillators," *Journal of Applied Physics*, vol. 120, no. 9, p. 093902, 2016.

[96] J.-P. Adam, S. Rohart, J. Ferré, A. Mougin, N. Vernier, L. Thevenard, A. Lemaître, G. Faini, and F. Glas, "Macrospin behavior and superparamagnetism in (ga, mn) as nanodots," *Physical Review B*, vol. 80, no. 15, p. 155313, 2009.

[97] R. Faria, K. Y. Camsari, and S. Datta, "Implementing bayesian networks with embedded stochastic mram," *AIP Advances*, vol. 8, no. 4, p. 045101, 2018.

[98] H. Lee, F. Ebrahimi, P. K. Amiri, and K. L. Wang, "Design of high-throughput and low-power true random number generator utilizing perpendicularly magnetized voltage-controlled magnetic tunnel junction," *AIP Advances*, vol. 7, no. 5, p. 055934, 2017.

[99] B. Parks, M. Bapna, J. Igbokwe, H. Almasi, W. Wang, and S. A. Majetich, "Superparamagnetic perpendicular magnetic tunnel junctions for true random number generators," *AIP Advances*, vol. 8, no. 5, p. 055903, 2018.

[100] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, "Intrinsic optimization using stochastic nanomagnets," *Scientific reports*, vol. 7, p. 44370, 2017.

[101] R. Faria, K. Y. Camsari, and S. Datta, "Low-barrier nanomagnets as p-bits for spin logic," *IEEE Magnetics Letters*, vol. 8, pp. 1–5, 2017.

[102] A. Z. Pervaiz, L. A. Ghantasala, K. Y. Camsari, and S. Datta, "Hardware emulation of stochastic p-bits for invertible logic," *Scientific reports*, vol. 7, no. 1, pp. 1–13, 2017.

[103] A. Z. Pervaiz, B. M. Sutton, L. A. Ghantasala, and K. Y. Camsari, "Weighted p-bits for fpga implementation of probabilistic circuits," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 6, pp. 1920–1926, 2018.

[104] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[105] D. Heckerman, A. Mamdani, and M. P. Wellman, "Real-world applications of bayesian networks," *Communications of the ACM*, vol. 38, no. 3, pp. 24–26, 1995.

[106] D. Heckerman and J. S. Breese, "Causal independence for probability assessment and inference using bayesian networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 26, no. 6, pp. 826–831, 1996.

[107] I. Rish, M. Brodie, S. Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, and K. Hernandez, "Adaptive diagnosis in distributed systems," *IEEE Transactions on neural networks*, vol. 16, no. 5, pp. 1088–1109, 2005.

[108] L. N. Chakrapani, P. Korkmaz, B. E. Akgul, and K. V. Palem, "Probabilistic system-on-a-chip architectures," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 12, no. 3, p. 29, 2007.

[109] S. Zermani, C. Dezan, H. Chenini, J.-P. Diguet, and R. Euler, "Fpga implementation of bayesian network inference for an embedded diagnosis," in *2015 IEEE Conference on Prognostics and Health Management (PHM)*. IEEE, 2015, pp. 1–10.

[110] W. Tylman, T. Waszyrowski, A. Napieralski, M. Kamiński, T. Trafidło, Z. Kulesza, R. Kotas, P. Marciniak, R. Tomala, and M. Wenerski, "Real-time prediction of acute cardiovascular events using hardware-implemented bayesian networks," *Computers in biology and medicine*, vol. 69, pp. 245–253, 2016.

[111] Y. Shim, S. Chen, A. Sengupta, and K. Roy, "Stochastic spin-orbit torque devices as elements for bayesian inference," *Scientific reports*, vol. 7, no. 1, p. 14101, 2017.

[112] J. S. Friedman, L. E. Calvet, P. Bessière, J. Droulez, and D. Querlioz, "Bayesian inference with muller c-elements," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 6, pp. 895–904, 2016.

[113] D. Querlioz, O. Bichler, A. F. Vincent, and C. Gamrat, "Bioinspired programming of memory devices for implementing an inference engine," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1398–1416, 2015.

[114] B. Behin-Aein, V. Diep, and S. Datta, "A building block for hardware belief networks," *Scientific reports*, vol. 6, p. 29893, 2016.

[115] "Relatedness Calculator." [Online]. Available: http://apps.nolanlawson.com/relatedness-calculator/index

[116] O. Hassan, K. Y. Camsari, and S. Datta, "Voltage-driven building block for hardware belief networks," *IEEE Design & Test*, vol. 36, no. 3, pp. 15–21, 2019.

[117] R. Cowburn, D. Koltsov, A. Adeyeye, M. Welland, and D. Tricker, "Single-domain circular nanomagnets," *Physical Review Letters*, vol. 83, no. 5, p. 1042, 1999.

[118] C. Lin, S. Kang, Y. Wang, K. Lee, X. Zhu, W. Chen, X. Li, W. Hsu, Y. Kao, M. Liu *et al.*, "45nm low power cmos logic compatible embedded stt mram utilizing a reverse-connection 1t/1mtj cell," in *2009 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2009, pp. 1–4.

[119] R. F. S. D. J. A. P. Debashis, V. Ostwal and Z. Chen, "Hardware implementation of bayesian network building blocks with stochastic spintronic devices," *in review*, 2020.

[120] A. Z. Pervaiz, L. A. Ghantasala, K. Y. Camsari, and S. Datta, "Hardware emulation of stochastic p-bits for invertible logic," *Scientific reports*, vol. 7, no. 1, p. 10994, 2017.

[121] D. Nikovski, "Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics," *IEEE Transactions on Knowledge & Data Engineering*, no. 4, pp. 509–516, 2000.

[122] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein, "A bayesian networks approach for predicting protein-protein interactions from genomic data," *science*, vol. 302, no. 5644, pp. 449–453, 2003.

[123] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," *Journal of computational biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[124] M. Zou and S. D. Conzen, "A new dynamic bayesian network (dbn) approach for identifying gene regulatory networks from time course microarray data," *Bioinformatics*, vol. 21, no. 1, pp. 71–79, 2004.

[125] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 1, pp. 124–132, 2006.

[126] J. L. Ticknor, "A bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5501–5506, 2013.

[127] C. Premebida, D. R. Faria, and U. Nunes, "Dynamic bayesian network for semantic place classification in mobile robotics," *Autonomous Robots*, vol. 41, no. 5, pp. 1161–1172, 2017.

[128] D.-C. Park, "Image classification using naïve bayes classifier," *Int J Comp Sci Elec Eng*, vol. 4, no. 3, pp. 135–139, 2016.

[129] J. Arias, J. Martinez-Gomez, J. A. Gamez, A. G. S. de Herrera, and H. Müller, "Medical image modality classification using discrete bayesian networks," *Computer vision and image understanding*, vol. 151, pp. 61–71, 2016.

[130] C. Bielza and P. Larrañaga, "Bayesian networks in neuroscience: a survey," *Frontiers in computational neuroscience*, vol. 8, p. 131, 2014.

[131] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[132] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[133] A. Darwiche, *Modeling and reasoning with Bayesian networks*. Cambridge university press, 2009.

[134] 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, p. 104, 2016.

[135] E. M. Jonas, "Stochastic architectures for probabilistic computation," Ph.D. dissertation, Massachusetts Institute of Technology, 2014.

[136] R. M. Neal, "Connectionist learning of belief networks," *Artificial intelligence*, vol. 56, no. 1, pp. 71–113, 1992.

[137] W. A. Borders, A. Z. Pervaiz, S. Fukami, K. Y. Camsari, H. Ohno, and S. Datta, "Integer factorization using stochastic magnetic tunnel junctions," *Nature*, vol. 573, no. 7774, pp. 390–393, 2019.

[138] A. Z. Pervaiz, B. M. Sutton, L. A. Ghantasala, and K. Y. Camsari, "Weighted p-bits for fpga implementation of probabilistic circuits," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 6, pp. 1920–1926, 2018.

[139] K. Y. Camsari, S. Chowdhury, and S. Datta, "Scaled quantum circuits emulated with room temperature p-bits," *arXiv preprint arXiv:1810.07144*, 2018.

[140] J. Kaiser, R. Faria, K. Y. Camsari, and S. Datta, "Probabilistic circuits for autonomous learning: A simulation study," *Frontiers in Computational Neuroscience*, vol. 14, p. 14, 2020. [Online]. Available: https://www.frontiersin.org/article/10.3389/fncom.2020.00014

[141] S. Bhatti, R. Sbiaa, A. Hirohata, H. Ohno, S. Fukami, and S. Piramanayagam, "Spintronics based random access memory: a review," *Materials Today*, vol. 20, no. 9, pp. 530–548, 2017.

[142] M. Henrion, "Propagating uncertainty in bayesian networks by probabilistic logic sampling," in *Machine Intelligence and Pattern Recognition*. Elsevier, 1988, vol. 5, pp. 149–163.

[143] H. Guo and W. Hsu, "A survey of algorithms for real-time bayesian network inference," in *Join Workshop on Real Time Decision Support and Diagnosis Systems*, 2002.

[144] O. Hassan, R. Faria, K. Y. Camsari, J. Z. Sun, and S. Datta, "Low-barrier magnet design for efficient hardware binary stochastic neurons," *IEEE Magnetics Letters*, vol. 10, pp. 1–5, 2019.

[145] D. Sherrington and S. Kirkpatrick, "Solvable Model of a Spin-Glass," *Physical Review Letters*, vol. 35, no. 26, pp. 1792–1796, Dec. 1975.

[146] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys*, vol. 21, no. 6/7, 1999.

[147] M. Scutari, "Learning bayesian networks with the bnlearn r package," *arXiv preprint arXiv:0908.3817*, 2009.

[148] J. S. Rosenthal, "Monty hall, monty fall, monty crawl," *Math Horizons*, vol. 16, no. 1, pp. 5–7, 2008.

[149] V. Savant, "M. ask marilyn," *Parade Magazine*, p. 15, 9 September 1990.

[150] F. Taroni, A. Biedermann, P. Garbolino, and C. G. Aitken, "A general approach to bayesian networks for the interpretation of evidence," *Forensic Science International*, vol. 139, no. 1, pp. 5–16, 2004.

[151] M. J. Schervish and B. P. Carlin, "On the convergence of successive substitution sampling," *Journal of Computational and Graphical statistics*, vol. 1, no. 2, pp. 111–127, 1992.

[152] D. Sherrington and S. Kirkpatrick, "Solvable model of a spin-glass," *Physical review letters*, vol. 35, no. 26, p. 1792, 1975.

[153] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A Survey of Neuromorphic Computing and Neural Networks in Hardware," *arXiv:1705.06963 [cs]*, May 2017, arXiv: 1705.06963.

[154] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A Learning Algorithm for Boltzmann Machines*," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, Jan. 1985.

[155] R. M. Neal, "Connectionist learning of belief networks," *Artificial Intelligence*, vol. 56, no. 1, pp. 71–113, Jul. 1992.

[156] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. New York, NY, USA: John Wiley & Sons, Inc., 1989.

[157] S. S. Haykin *et al.*, *Neural networks and learning machines/Simon Haykin.* New York: Prentice Hall,, 2009.

[158] K. Y. Camsari, S. Ganguly, and S. Datta, "Modular Approach to Spintronics," *Scientific Reports*, vol. 5, p. 10571, Jun. 2015.

[159] L. E. Reichl, "A modern course in statistical physics," 1999.

[160] Q. Liu, J. Peng, A. Ihler, and J. Fisher III, "Estimating the partition function by discriminance sampling," in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2015, pp. 514–522.

[161] L. Liu, O. Lee, T. Gudmundsen, D. Ralph, and R. Buhrman, "Current-induced switching of perpendicularly magnetized magnetic layers using spin torque from the spin hall effect," *Physical review letters*, vol. 109, no. 9, p. 096602, 2012.

[162] I. M. Miron, K. Garello, G. Gaudin, P.-J. Zermatten, M. V. Costache, S. Auffret, S. Bandiera, B. Rodmacq, A. Schuhl, and P. Gambardella, "Perpendicular switching of a single ferromagnetic layer induced by in-plane current injection," *Nature*, vol. 476, no. 7359, p. 189, 2011.

[163] P. Debashis, R. Faria, K. Y. Camsari, and Z. Chen, "Design of stochastic nanomagnets for probabilistic spin logic," *IEEE Magnetics Letters*, vol. 9, pp. 1–5, 2018.

[164] P. Debashis and Z. Chen, "Experimental demonstration of a spin logic device with deterministic and stochastic mode of operation," *Scientific reports*, vol. 8, no. 1, p. 11405, 2018.

[165] D. Bhowmik, L. You, and S. Salahuddin, "Spin hall effect clocking of nanomagnetic logic without a magnetic field," *Nature nanotechnology*, vol. 9, no. 1, p. 59, 2014.

APPENDICES

A. BENCHMARKING AUTONOMOUS BEHAVIORAL MODEL (PPSL: DESIGN 2) FOR FPGA IMPLEMENTATION

Materials in this chapter have been extracted verbatim from the paper: "Autonomous Probabilistic Coprocessingg with Petaflips per Second", B. Sutton, R. Faria, L. A. Ghantasala, K. Y. Camsari, and S. Datta, arXiv preprint arXiv:1907.09664, in review (*Applied Physics Reviews*). B.S. emulated the autonomous probabilistic hardware behavioral model on an FPGA platform and showed two large scale applications of p-circuits ($\sim 10K$ spin): optimization and quantum annealing. R.F. benchmarked the behavioral model with SPICE simulation of the actual low barrier nanomagnet based hardware to establish the validity of the behavioral model.

Stochastic neural networks (SNN) are widely used for machine learning, inference and many other emerging applications [153]. A common version of such algorithms is based on the concept of a binary stochastic neuron (BSN) [154, 155] which fluctuates between -1 and +1 with probabilities that can be controlled through an input, I_i, constructed from the outputs of other BSNs, m_j. The synaptic function, $I_i(\{m\})$, can have many different forms depending on the desired functionality, but we will restrict this discussion to linear functions defined by a set of weights W_{ij} such that

$$I_i(t+\tau_S) = \beta \sum_j W_{ij} m_j(t) \tag{A.1}$$

where β is a constant and τ_S is the 'synapse time', that is the time it takes to recompute the inputs $\{I\}$ everytime the outputs $\{m\}$ change. In software implementations, each BSN is updated repeatedly according to

$$m_i(t+\tau_N) = \text{sgn}\left[\tanh\left(I_i(t)\right) - r_{[-1,+1]}\right] \tag{A.2}$$

where $r_{[a,b]}$ represents a random number in the range $[a, b]$, and τ_N is the 'neuron' time, that is the time it takes for a neuron to provide stochastic output m_i with the correct statistics dictated by a new input I_i.

It is well-known [156] that to ensure fidelity of operation it is important to avoid *simultaneous* updates of two BSNs that are causally connected through a non-zero W_{ij}. The standard approach is to update each BSN sequentially according to Eq. (A.2), recomputing the input from Eq. (A.1) after each update, a procedure known as Gibbs sampling [157]. By contrast, the objective of this paper is to explore the feasibility of ultrafast operation through an autonomous architecture whereby each BSN continually fluctuates between -1 and +1 with probabilities that are controlled by the input I_i. We refer to this autonomous BSN as a *p-bit* to highlight its role as the key element of an autonomous p-computer (ApC), similar to the role of a q-bit in a quantum computer.

Since digital platforms are inherently synchronous, we mimic autonomous operation by replacing Eq. (A.2) with a new hardware-inspired model, Eq. (A.3) (PPSL), that we benchmarked against established state-of-the-art physical models as described in the Methods section. These equations are based on SPICE simulations of Boltzmann networks where the update order of p-bits is irrelevant given symmetric coupling between connected p-bits. However, for certain networks such as those with directed connections, the update ordering of p-bits may be important and other hardware models more appropriate for these systems are likely required. These models are not discussed in this paper, but the overall FPGA architecture was designed to support the exploration of different hardware models and network topologies, hence they can be included here with minor effort.

At each time step, all p-bits are free to flip and they do so with a probability $\sim s$ that is controlled by the input I_i having a zero-input value $s(I_i = 0) = s_0 \ll 1$.

$$m_i(n+1) = m_i(n) \times \text{sgn}[e^{-s} - r_{[0,1]}] \tag{A.3a}$$

$$s = s_0 e^{-m_i(n) I_i(n)} \tag{A.3b}$$

In each time step the p-bit flips with a probability $\sim s$, so that the average time taken for it to respond is $1/s$. Since time steps are measured in units of τ_S, we have $\tau_N = (1/s) \times \tau_S$ as stated earlier. Unlike Eq. (A.2), Eq. (A.3) can be used to update all p-bits in parallel without explicitly worrying about simultaneous updates. With small values of s_0, the fraction of simultaneous updates is sufficiently small such that Eq. (A.3) in an unsequenced mode gives results equivalent to those obtained from Eq. (A.2) with careful sequencing.

Benchmarking Eq. (A.3) with stochastic LLG

A coupled stochastic Landau-Lifshitz-Gilbert (sLLG) equation is solved and benchmarked against the autonomous p-bit model of (A.3a) and (A.3b) (PPSL). Magnetization dynamics of a circular stochastic nanomagnet are captured by solving the sLLG equation in the macrospin assumption within a modular [158] SPICE framework ,

$$(1+\alpha^2)\frac{d\hat{m}}{dt} = -|\gamma|\hat{m}\times\vec{H} - \alpha|\gamma|(\hat{m}\times\hat{m}\times\vec{H}) + \frac{1}{qN_s}(\hat{m}\times\vec{I}_S\times\hat{m}) + \left(\frac{\alpha}{qN_s}(\hat{m}\times\vec{I}_S)\right) \tag{A.4a}$$

where α is the damping coefficient, γ is the electron gyromagnetic ratio, $N_s = \mathrm{M_s Vol.}/\mu_B$ is the total number of Bohr magnetons in the magnet, $\mathrm{M_s}$ is the saturation magnetization, $\vec{H} = \vec{H}_d + \vec{H}_n$ is the effective field including the out-of-plane ($\hat{x}$ directed) demagnetization field $\vec{H}_d = -4\pi M_s m_x \hat{x}$, as well as the thermally fluctuating magnetic field due to the three dimensional uncorrelated thermal noise H_n with zero mean $\langle H_n\rangle = 0$ and standard deviation $\langle H_n^2\rangle = 2\alpha \mathrm{kT}/|\gamma|\mathrm{M_s Vol.}$ along each direction, $\vec{I}_S$ is the applied spin current to the nanomagnet.

Individual p-bits are coupled according to:

$$I^z_{s,i}(t+\Delta_t) = \beta I_{s0} \sum_j W_{ij}\ \mathrm{sgn}(m^z_j(t)) \tag{A.5}$$

where, I_{s0} is the tanh fitting parameter of the sigmoidal response ($\mathrm{sgn}(m_z)$ versus spin current I^z_s along z-direction). In the benchmark, a circular disk magnet with

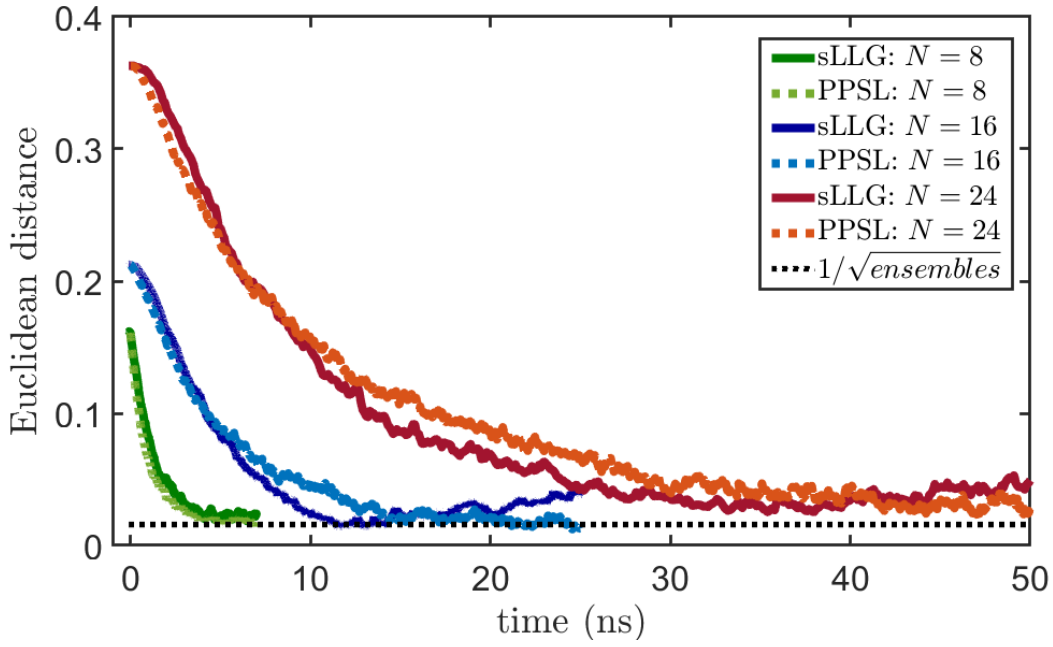


Fig. A.1.: **Benchmarking the PPSL Model with sLLG using Euclidean distance**: Using a random Sherrington-Kirkpatrick spin glass instance for different network sizes, N, the PPSL model is benchmarked against sLLG as a function of time. Each point on the graph represents the Euclidean distance from the ideal Boltzmann distribution and the ensemble solution obtained from PPSL and sLLG. The steady state error will depend on the number of ensembles as shown by the black dotted line.

a vanishing anisotropy (H_K) is used with the parameters: diameter $D = 150$ nm and thickness $t = 2$ nm, $\alpha = 0.01$, $M_s = 1100$ emu/cc, $H_K = 1$ Oe resulting in an autocorrelation time of $\tau_{corr} = 1.372$ ns and $I_{s0} = 1$ mA. A fitting parameter of 1.4 is used in the PPSL model for τ_N, i.e. $\tau_N = 1.4\tau_{corr}$.

The simulated network is a Sherrington-Kirkpatrick [145] spin glass with a random coupling matrix and random bias between -1 and +1. The benchmarking of the proposed PPSL model with the coupled sLLG network, analogous to the probabilistic circuit proposed in [4], is accomplished by comparing two different quantities: (1) Euclidean distance and (2) Free energy.

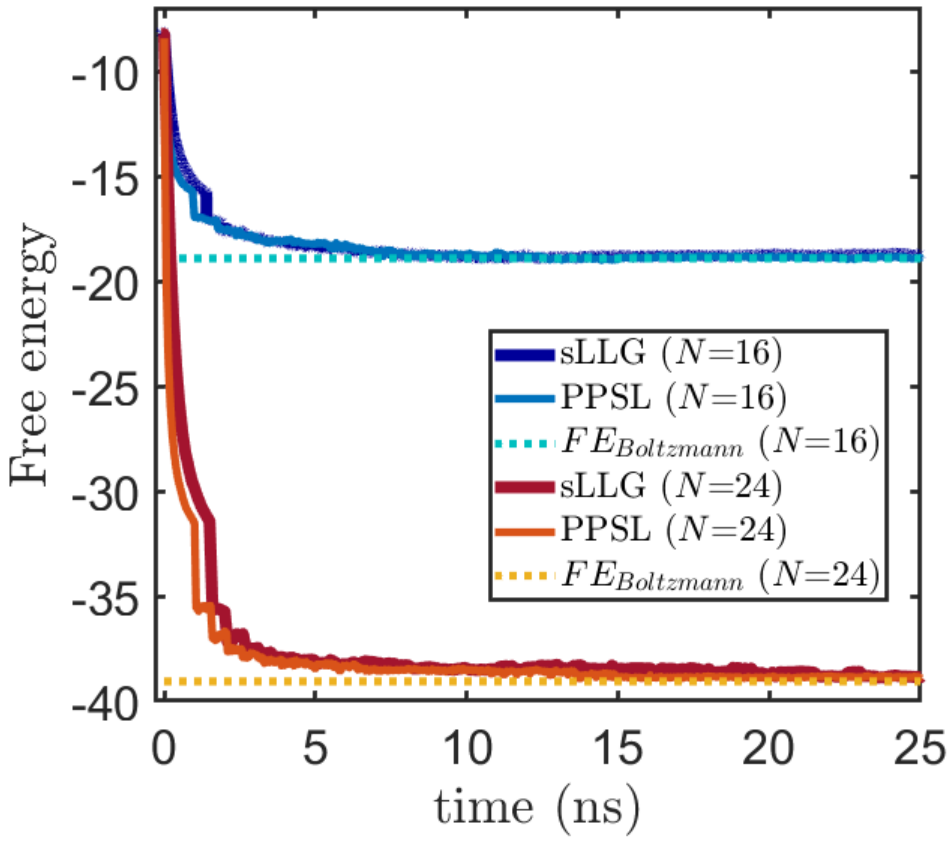


Fig. A.2.: **Benchmarking the PPSL Model with sLLG using Free Energy**: The free energy calculated for the random Sherrington-Kirkpatrick spin glass instance of Fig. A.1 from the PPSL model is benchmarked against sLLG as a function of time for network sizes $N = 16$ and $N = 24$, showing convergence to the free energy obtained from Boltzmann law.

Euclidean distance is defined by:

$$ED = \sqrt{\sum_{i=1}^{2^N} (P_i - P_{\text{i,Boltzmann}})^2} \tag{A.6}$$

where P_i is the probability of occurrence of the i-th configuration computed out of 4000 ensembles at each time step of the simulation. $P_{\text{i,Boltzmann}}$ is computed from the joint probability distribution obtained from a Boltzmann law.

The second benchmark approach is based on a comparison of the free energy of the system with what is expected from the principles of statistical mechanics. Free energy is defined by [159] the partition function Z:

$$FE = \frac{\ln(Z)}{-\beta} \tag{A.7}$$

where, β is the pseudo-inverse temperature. Partition function Z is given by:

$$Z = \sum_{k} \exp(-\beta E_k) \tag{A.8}$$

where k represents different configurations of the network. Energy of a specific configuration is defined by:

$$E_k = -0.5 \sum_{\substack{i,j \\ i \neq j}} W_{ij} m_i m_j - \sum_{i} h_i m_i \tag{A.9}$$

When numerically calculating free energy from the sLLG data, the following steps have been applied (similar to the importance sampling method described in [160]):

1. The probability of different configurations, P_i, are calculated out of 4000 ensembles for each time step
2. For each P_i larger than a certain threshold value P_{th}, the partition function $Z_i = \exp(-I_0 E_i)/P_i$ is calculated, so that outliers are excluded
3. For each Z_i, the free energy $FE_i = -\ln(Z_i)/I_0$ is calculated.
4. Finally the mean of all FE_i is computed.

The above method is suitable for small examples, but may not scale to large examples due to the difficulty in empirically calculating different probabilities P_i as the network size grows. The striking agreement between the sLLG model and the behavioral model given by Eq. (A.3) shown in Figs. A.1 and refA.7 establishes the validity of Eq. A.3 as a suitable model for the projected autonomous, stochastic MTJ-based computer.

B. SPICE BENCHMARKING OF HARDWARE IMPLEMENTATION OF BAYESIAN NETWORK BUILDING BLOCKS WITH STOCHASTIC SPINTRONIC DEVICES

Most of the materials in this chapter have been extracted verbatim from the paper: " Hardware implementation of Bayesian network building blocks with stochastic spintronic devices", P. Debashis, V. Ostwal, R. Faria, S. Datta, J. Appenzeller and Z. Chen, In review (Physical Review Applied), 2019. P.D. and V.O. performed the experiment and R.F. performed the simulation to benchmark the experimental results.

In chapter 4 and chapter 5, it was shown how a Bayesian network (BN) can be mapped to a p-circuit composted of p-bits/Binary Stochastic Neurons (BSN). P. Debashis and V. Ostwal et al. have recently performed the first experimental demonstration of a Bayesian network building block implemented with naturally stochastic spintronic devices. These devices are based on nanomagnets with perpendicular magnetic anisotropy, initialized to their hard axes by the spin orbit torque from a heavy metal under-layer utilizing the giant spin Hall effect, enabling stochastic behavior. P.D. and V,O. have constructed an electrically interconnected network of two stochastic devices and manipulate the correlations between their states by changing connection weights and biases. By mapping given conditional probability tables to the circuit hardware proposed in [97], it is experimentally demonstrated that any two node Bayesian networks can be implemented by the stochastic network. A theoretical benchmarking is also presented by performing the stochastic Landau Lifshitz Gilbert (sLLG) simulation of an example case of a four node Bayesian network using the proposed device model in SPICE, with parameters taken from the experiment. Theoretical results are also compared with those expected from calculating joint probability distributions applyting standard statistical rules.

EXPERIMENTAL RESULTS AND ANALYSIS

Hard axis initialized PMA magnet as p-bit

In the experiment, the stochastic device is based on a hard axis initialized magnet with perpendicular magnetic anisotropy (PMA), whose output probability is controlled by the magnetic field produced by a charge current passing through an isolated metal ring17,18,20. The top left of Fig. 1 (a) shows the schematic of our device. It consists of a nanomagnet island with perpendicular magnetic anisotropy (PMA) shown in orange, on top of a heavy metal (Ta) Hall bar, shown in blue. It is well understood that the magnetization of a PMA magnet can be deterministically switched by the Spin Obit Torque (SOT) of a heavy metal under-layer in the presence of a symmetry breaking in-plane magnetic field [161,162]. However, when the spin current density is large enough, and when this field is absent, the magnetization gets pinned in the direction of the spin polarization, i.e. the magnets hard axis. Once the spin current is removed, ambient thermal noise makes the magnetization relax to either "up" or "down" with equal probability due to the symmetric energy landscape for these two states [163–165] as depicted by the cartoon in the top right of Fig. B.1 (a). The magnetization state is read out by the anomalous Hall effect (AHE), where the transverse V_{OUT} is +ve for a magnetization in the "up" direction and -ve for "down". The probability of relaxing back to the "up" or "down" direction can be controlled by applying a small out-of-plane magnetic field that lifts the degeneracy of the energy landscape. A positive field in the z-direction lowers the energy of the "up" state and raises that of the "down" state, thus making the "up" state more favorable. A negative z-directed field does the exact opposite. This is depicted in the energy landscape diagrams shown in the bottom panel of Fig. B.1 (a). This z-directed field is provided by a ring-shaped electrode called the "Oersted ring" henceforth, shown in yellow in the device schematic. A current I_{IN} passing through the Oersted ring of radius r produces a magnetic field given by $B = \mu_0 * I_{IN}/2r$.

Fig. 1 (b) shows the sLLG simulation of such a device. The top panels show the magnetization dynamics during the pulsing of the device. The current pulse through the GSHE layer is shown in black color in both the panels. The z-component of magnetization (m_z) is shown in blue and red. It can be seen that m_z goes to zero while the current pulse is ON. After the pulse is removed, m_z relaxes to -1 in the first case and it relaxes to +1 in the second, nominally identical case, highlighting the stochastic nature of the process. The time scale of this relaxation is governed by the material parameters of the nanomagnet such as saturation magnetization M_s, anisotropy field H_K and damping. The bottom panel of Fig. 1 (b) shows the average of the magnetization (after the dynamics have settled) in the z-direction (perpendicular easy axis) as a function of the input current, resembling a sigmoidal activation function. For experimental implementation, starting with a stack of Ta(5nm)/CoFeB(1nm)/MgO(2nm)/Ta(1nm) thin film, a Hall bar device with a PMA magnetic island located at the center is fabricated by means of successive e-beam lithography and Ar ion milling steps. To generate the out-of-plane field for tunability, the "Oersted ring" is fabricated on top and electrically isolated from the Hall bar by a dielectric layer. A false colored SEM image of the fabricated device is shown in the inset of Fig. 1(c). For the operation of the device, a Keithley 6221 current source is used to provide a current pulse of duration $100\mu s$ through the Ta Hall bar. This current pulse experimentally implements the required hard axis biasing scheme as shown in the sLLG simulation of Fig. 1 (b). Although the magnet can respond to much faster pulses, as shown in Fig. 1(b), we chose to use $100\mu s$ to be safely within the delay times of the measurement circuit. After the pulsing event, the state of the magnetization is read by a lock-in scheme, with a sinusoidal current provided by the same Keithley current source and an SRS830 lock-in amplifier. The device is pulsed repeatedly, and the state of the magnetization is read after each individual pulse. Fig. 1 (c) shows the average magnetization as a function of the input current I_{IN}. Each data point is obtained by averaging 25 pulsing events, as shown for three representative cases in the bottom panels. These measurements clearly demonstrate

the successful implementation of a device with an electrical input and output, which behaves stochastically for individual events, but produces a sigmoidal curve for the average output. This is the desired characteristic for many probabilistic spin logic applications including hardware BNs.

Theoretical benchmarking on a two node Bayesian network in hardware

We show how the stochastic devices described in the previous section can be used to implement a two node Bayesian network in hardware. The essential characteristic of a BN is captured in the CPT. Fig. B.2 (a) shows the example of a two-node network, with the first or the parent node (m_1) representing the packaging material for blocks of cheese in a dairy farm, and the second node (m_2) representing the probability of finding a stale cheese block. The values a and b in the CPT represent the probability of a cheese block being stale if the packaging material is of low quality ($m_1 = 0$) vs. high quality ($m_1 = 1$). Since the packaging material positively affects the shelf life, in this case, $a > b$. If instead of packaging material, m_1 represents the print design on the package, then the shelf life is not affected by it, and hence, $a = b$ in this case. Similarly, if some other variable, that negatively affects the shelf life is represented by m_1, then the CPT would have $a < b$. Now, for the first case, if the cheese was stored in a cold and dry storage, then the shelf life is increased, irrespective of the packaging material quality. This corresponds to adding a positive value to both a and b in the CPT. Hence, the variables in the CPT can span the entire space between 0 and 1 independently, depending on the problem being modeled. We first demonstrate that the CPT between the two probabilistic random variables in our example can be implemented by design of proper electrical connections between two of our stochastic devices (of the type shown in Fig. B.1). Then, by testing the circuit with designed parameters, we show that the probability of the output device (m_2) follows the probability of finding a stale cheese block, obtained from calculating the joint probability distribution. We also show that the inference about

the potential cause of stale cheese that is evaluated by Bayes theorem is well matched to the directly observed values from the joint distribution of the device outputs. The results are also verified by stochastic LLG simulations with magnet parameters (M_s, H_k and volume) taken to match the sigmoidal activation function obtained from the experiment. Fig. 2 (a) shows the given CPT that represents the relation between the stochastic variables m_1 and m_2. This CPT is translated into the parameters J_{21} and h_2 of the PSL model as shown in Fig. B.2 (b) where J_{21} corresponds to the connection from the first to the second device, m1 corresponds to the state of the first device and h_2 corresponds to the constant bias given to the second device.

The parameters J_{21} and h_2 are then used to design the hardware connection strengths and biases to two stochastic devices. Fig. 2 (c) shows the schematic of our circuit. The output voltage from the first device is amplified by an Op Amp. The output level of the Op Amp is determined by its $+/-V_{DD}$ supply voltages. This output is then connected to the Oersted ring of the second device through a weight resistor "R_{weight}" that determines how much current passes through it, and hence controls the output probability of the second device, corresponding to the J_{21} term in a BN. Additionally, a voltage source "V_{bias}" is connected to the input of the second device through a resistor "R_{bias}" to mimic the fixed bias (h_2) in a BN. The values of the circuit parameters V_{DD}, V_{bias}, R_{weight} and R_{bias} are obtained from the required J_{21} and h_2 by the following design equations shown in fig. B.2. In our circuit as shown in Fig. B.2(c), J_{21} is the magnetic field produced by the Oersted ring of device 2, normalized with the field required to saturate its magnetization in the "up" or "down" state, denoted by B_0. We can span all possible conditional probabilities between two nodes of a BN (given by 'a' and 'b' in the CPT) by changing the circuit parameters R_{weight}, polarity and R_{bias}.

We take five different CPTs with "a" and "b" spanning the range between 0 and 1, shown in Fig. B.3(a). We then calculate J_{21} and h_2 for these five cases and design our circuit according to the equations stated in fig. B.2(c). The designed circuits are then tested by repeating a sequential pulsing scheme. The inset of Fig. B.2(c)

shows the timing diagram of the measurement procedure. The two devices are pulsed sequentially and during the pulsing of the second device a constant DC read current is passed through the first device in order to generate the input voltage to the second device. Then, this sequential pulsing is repeated to generate the required statistics. The two devices produce random outputs, but with correlated statistics, as is required by the CPT between the two random variables. The output after each pulse is measured by a lock-in amplifier and then digitized. Representative sections of the device outputs are shown in Fig. B.3 (b) for three different connection configuration. The probability of finding a stale cheese block can be found from the join probability distribution by using the probability chain rule:

$$\begin{aligned} p(m_2 = 1) &= \sum_{m_1} p(m_1, m_2 = 1) = \sum_{m_1} p(m_2 = 1|m_1)p(m_1) \\ &= p(m_2 = 1|m_1 = 0)p(m_1 = 0) + p(m_2 = 1|m_1 = 1)p(m_1 = 1) \\ &= ap(m_1 = 0) + bp(m_1 = 1) \end{aligned} \tag{B.1}$$

where $p(m_1 = 0 \text{ or } 1)$ is an input parameter. The number of terms in the above expression grows as 2^N where N is the number of parent nodes for the particular child node of interest [106]. Instead of performing this algebra, the required probability can be obtained from the circuit by directly observing the stochastic output of device 2 and obtaining its mean value over several pulsing cycles. Similarly, given that a randomly drawn cheese block from a large lot is stale, the probability that it was caused by a low quality packaging material can be found by using Bayes theorem:

$$\begin{aligned} p(m_1 = 0|m_2 = 1) &= p(m_1 = 0, m_2 = 1)/p(m_2 = 1) \\ &= p(m_2 = 1|m_1 = 0)p(m_1 = 0)/p(m_2 = 1) \\ &= (ap(m_1 = 0))/(ap(m_1 = 0) + bp(m_1 = 1)) \end{aligned} \tag{B.2}$$

The number of terms required in the evaluation of the above expression also grows as $\sim 2^N$ where N is the number of potential binary causes of a particular effect [106]. However, from the hardware BN, this probability can be directly obtained by observing the joint distribution of states of the two devices. It is to be noted

here that this way of performing the inference always involves observing the joint distributions of only two nodes of the BN: nodes corresponding to the effect and the potential cause of interest, irrespective of N. After 100 pulsing cycles, the obtained output probabilities for all the five circuits (representing the five different CPTs of Fig. B.3(a)) is comparable with the expectation from calculating the joint probability distribution and is also verified by stochastic LLG simulations, as shown in Fig. B.3 (c). Similarly, the obtained probabilities from inference is comparable with that from Bayes theorem and stochastic LLG simulations, seen in Fig. B.3(d).

SIMULATION OF A FOUR NODE BAYESIAN NETWORK

In this section, we present a self-consistently coupled sLLG simulation of the more complicated, four node Bayesian network shown in the top left inset of Fig. 4 (a). Here, the BN consists of four nodes: cloud (C), rain (R), sprinkler (S), and wetness of grass (W). In this case, the evaluation of a node probability from the joint probability distribution requires the following evaluation, for example for the W node:

$$\begin{aligned} p(W) &= \sum_C \sum_R \sum_S p(C,R,S,W) \\ &= \sum_C \sum_R \sum_S p(C)p(R|C)p(S|C)p(W|RS) \end{aligned} \tag{B.3}$$

Here the number of terms to be evaluated in the summation is eight, as each of the C, R and S nodes could take two possible values “0” or “1”. Similarly performing inference, for example, what is the probability that it had rained, given that the grass is wet requires the following evaluation:

$$p(R|W) = p(RW)/p(W) \tag{B.4}$$

where both the numerator and the denominator of the right-hand side of the above equation must be evaluated by summing over the joint probability distribution P(C,R,S,W), resulting in the evaluation of four and eight terms respectively. However, by using the hardware, the required node probabilities and the inference can be

obtained in exactly the same way as our previous two-node example: we simply observe the stochastic output of the corresponding node for probability assessment; and observe the joint distribution of only the R and the W node to perform the required inference. This is demonstrated in the simulation study below. The parameters used in the sLLG simulation platform such as the magnet dimensions and the output sigmoidal response are benchmarked with the experimental results from the device in Fig. 1 (c). The coupling and biases are benchmarked with the two node BN network experiments shown in Fig. 2 and 3. Fig. 4 (a) shows the circuit implementation, where each node is represented by a hardware p-bit as described in Fig. 1. It is to be noted here that an auxiliary p-bit (represented by node 'X') is needed to implement this four node Bayesian network. This is because, the CPT capturing the dependency of node 'W' on node 'R' and 'S' has four conditional probabilities, which can take any value between 0 and 1 independent of each other. Therefore, from basic principles of linear algebra, we need four independent physical parameters to implement this CPT. Two of the four required parameters are provided by the two interconnection weights (J_{WR} and J_{WS}) and another parameter is provided by the bias to the node 'W' (h_W). The remaining one parameter is provided by the interconnection to the auxiliary node 'X'. The requirement of auxiliary nodes in designing Bayesian networks from p-bits is described in more detail by Faria et al. 21. The dynamics of the PMA magnet used in the hardware p-bit design is captured by solving the sLLG equation with a monodomain macrospin assumption:

$$(1+\alpha^2)\frac{d\hat{m}}{dt} = -|\gamma|\hat{m}\times\vec{H} - \alpha|\gamma|\hat{m}\times\hat{m}\times\vec{H} - \frac{1}{qN_s}\hat{m}\times\hat{m}\times\vec{I_s} + \frac{\alpha}{qN_s}\hat{m}\times I_s \quad \text{(B.5)}$$

where, $\vec{H}$ is the total internal and external field along with thermal noise field, $\vec{I_s}$ is the spin current, $N_s = M_s V$ is the total magnetic moment with M_s being the saturation magnetization, α is the damping coefficient, γ is the gyromagnetic ratio. Magnet parameters used in the simulation are: $H_k = 200\ Oe$, $M_s = 1000\ emu/cc$, $D_1 = 1\ \mu m$, $D_1 = 3\ \mu m$, $t = 1\ nm$, $\alpha = 0.1$. The average magnetization of each p-bit

can be approximated by $m_z = \tanh(H/H_0)$, where H is the Oersted field generated from the current coil and H_0 is a fitting parameter. For the system simulation, we start with chosen CPTs for each of the nodes. These are shown as the inputs next to the respective nodes in Fig. 4 (c). These values are then

$$H_i = H_0(\sum_j J_{ij} m_j + h_i) \tag{B.6}$$

The coupling and bias component of H_i can be realized through the coupling resistance R_{weight} and R_{bias} respectively with a mapping principle as described in fig. B.2 for the two node case. While solving the coupled sLLG, each p-bit is put along the hard axis by the GSHE current in a sequential order from parent to child node and the magnetizations of all p-bits are recorded after their corresponding pulse is turned off. It is worth noting that the pulse sequence is important for the proper operation of the Bayesian network. The pulsing should start from the first node and move down the hierarchy from parent to corresponding child nodes. The order of pulsing among different nodes on the same hierarchy level (e.g. node R and S in our example) is not critical. Taking these principles into account, the pulsing order for one cycle is shown in Fig. 4 (b). This cycle is repeated several times to generate the probabilities of each of the four nodes. Fig. 4 (c) shows representative data of magnetization of each node for 50 pulses. From this distribution of the magnetization state of each node in 'UP' vs. 'DN' state, probabilities of each node are calculated. For example, the magnetization of the p-bit corresponding to 'sprinkler' node shows more occurrences in the 'DN' state compared to 'UP' state, resulting in a low probability of sprinkler being ON ($P(S) \sim 0.25$ in this case). Similarly, the probability of 'rain': P(R) and the probability of 'grass being wet': P(W) are obtained from the magnetization state distribution. The obtained probabilities are compared with those obtained by calculating the joint probability distribution as shown in the output tables alongside each of the four nodes in Fig. 4 (c). It can be seen that the probabilities obtained from the coupled sLLG result match well with the simple PSL behavioral model and with the values obtained from the evaluation of equation (12). Similarly, the probability of

rain, given that the grass is wet ($P(R|W)$) is obtained from the coupled sLLG result is 0.73, which is well matched with the value of 0.75 obtained from equation (13). It is to be noted that the accuracy in this depends on the number of samples taken to calculate the probabilities. Using experimentally benchmarked sLLG simulations, we have shown that a BN implemented in hardware using the experimentally demonstrated stochastic spintronic devices can generate probabilities that are well matched to the theoretical values from calculating the joint probability distribution.

Acknowledgement

This work was supported by the Center for Probabilistic Spin Logic for Low-Energy Boolean and Non-Boolean Computing (CAPSL), one of the Nanoelectronic ComputingResearch (nCORE) Centers as task 2759.003 and 2759.004, a Semiconductor Research Corporation (SRC) program sponsored by the NSF through CCF 1739635.

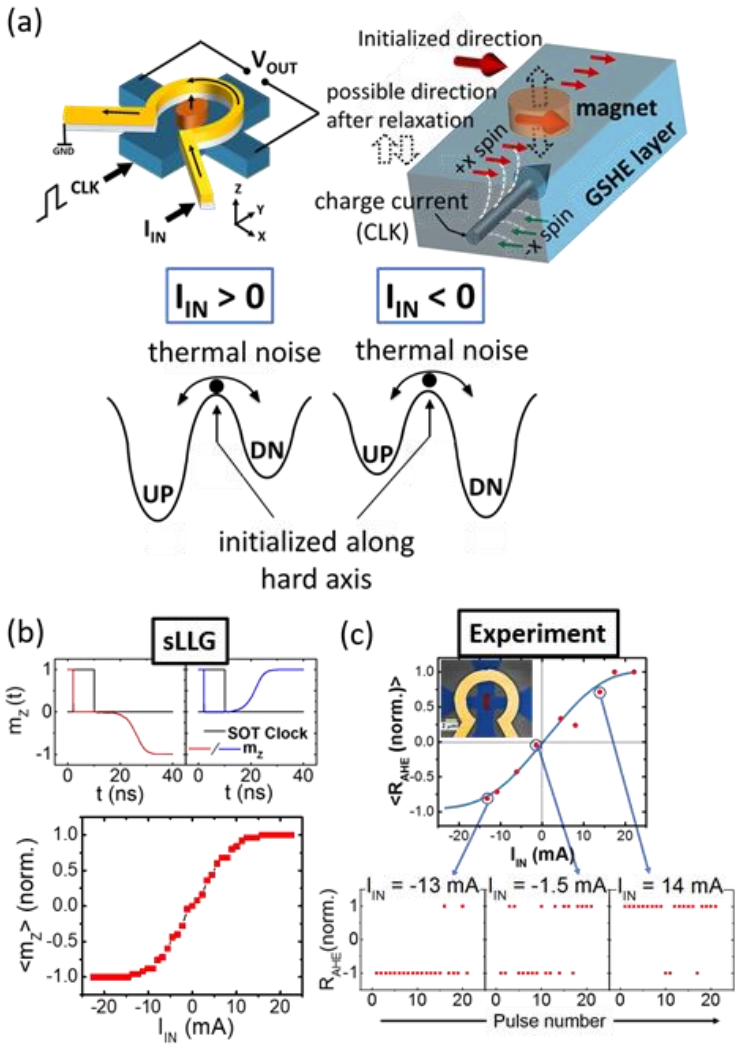


Fig. B.1.: Hardware building block of Bayesian Networks. (a) Schematic of the probabilistic device and illustration of the hard axis initialization by spin orbit torque. (b) Stochastic LLG simulation of 500 ensembles, showing tunable random behavior of the device. The two top panels show representative cases where the magnetization relaxes to the "up" and "down" direction after being released from the hard axis. (c) Experimental measurements on the device showing stochastic behavior with tunability using a charge current through an isolated Oersted ring. The bottom panels show the stochastic outputs, whose averages show the sigmoidal behavior as a function of the input current.

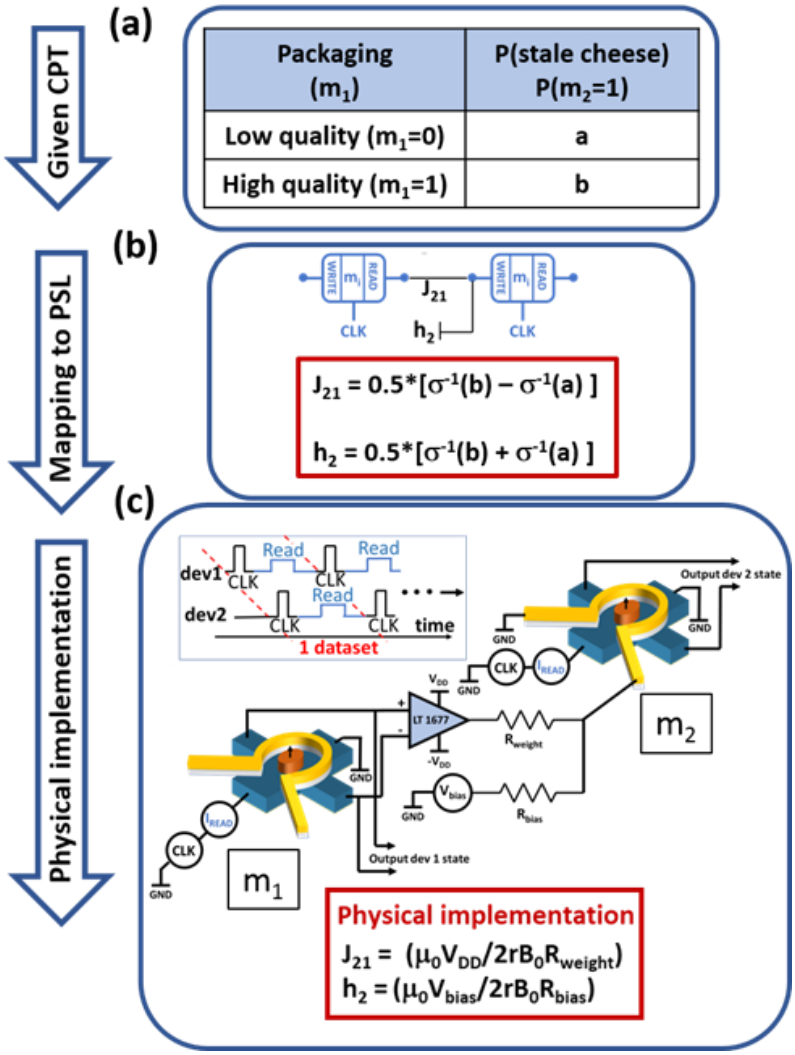
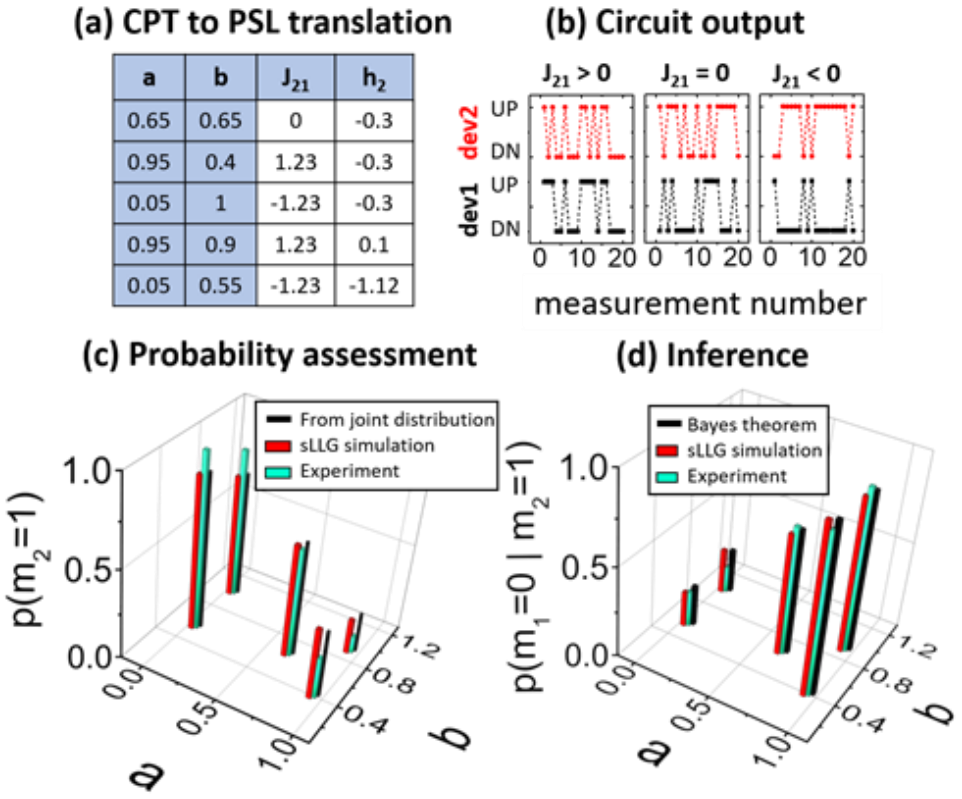


Fig. B.2.: Hardware design of a two-node network. (a) The given conditional probability table (CPT) representing the causal dependency of two probabilistic variables, i.e., the quality of packaging and state of cheese (b) PSL model of the two node BN with the CPT parameters translated to PSL parameters (c) Circuit schematic of two connected devices to implement two coupled Bayesian nodes. Inset on the top left shows the timing diagram of various operations performed on device 1 and 2.



a	b	J_{21}	h_2
0.65	0.65	0	-0.3
0.95	0.4	1.23	-0.3
0.05	1	-1.23	-0.3
0.95	0.9	1.23	0.1
0.05	0.55	-1.23	-1.12

Fig. B.3.: Testing of the two node BN circuit. (a) Five different combinations of the CPT parameters that are experimentally implemented in hardware. (b) Representative sections of the measured data for positive, negative and no connection between device 1 and device 2 as shown in Fig 2(c). (c) Obtained output probabilities of cheese being stale for the five different given CPTs. The experimentally obtained probability values are in good agreement with theory and stochastic LLG simulations. (d) Inference about probability of the packaging being bad quality given that a stale cheese is found is plotted for the different CPTs, showing good match between direct experimental observation, Bayes theorem and stochastic LLG simulations.

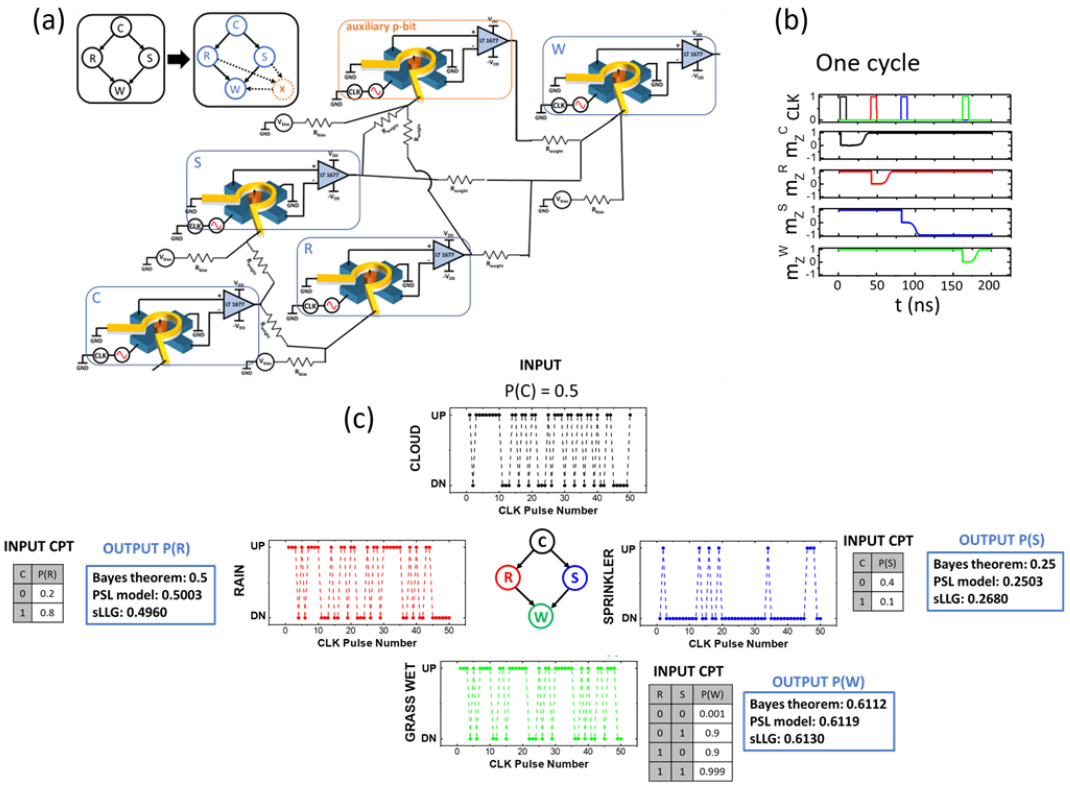


Fig. B.4.: Simulation results of a four node BN. (a) Hardware implementation layout (b) Representative one clock cycle of operation (c) Results obtained from the four node BN with the given CPTs shown as the input tables. Probabilities for each node, generated after 500 clock cycles are shown inside the blue boxes. Representative sections of the state of each node after 50 pulses is shown next to them. The obtained probabilities show good agreement with expectation from calculating the joint probability distribution.

C. ABBREVIATIONS

Table C.1.: Abbreviations used in this thesis

Full form	Abbreviation
Probabilistic Spin Logic	PSL
Probabilistic Bit	p-bit
Probabilistic Circuit	p-circuit
Magnetic Tunnel Junction	MTJ
Bayesian Network	BN
Magnetoresistive Random Access Memory	MRAM
Low Barrier Magnet	LBM
Binary Stochastic Neuron	BSN
Tunable Random Number Generator	TRNG
Artificial Neural Network	ANN
Magnetoresistive Random Access Memory	MRAM
Complementary Metal Oxide Semiconductor	CMOS
Boltzmann Machine	BM
Giant Spin Hall Effect	GSHE
Stochastic Landau Lifshitz Gilbert	sLLG
Artificial Intelligence	AI
Inplane Magnetic Anisotropy	IMA
Perpendicular Magnetic Anisotropy	PMA
Field Effect Transistor	FET
Fokker Planck Equation	FPE
Directed Acyclic Graph	DAG

continued on next page

Table C.1.: *continued*

Full form	Abbreviation
Conditional Probability Table	CPT
Spin Orbit Torque	SOT
Spin Transfer Torque	STT
Autonomous Probabilistic Circuit	ApC
Sherrington Kirkpatrick	SK

D. PUBLICATIONS

Before Preliminary examination

- Camsari K. Y., **Faria R.**, Sutton B. M., & Datta S. (2017). Stochastic p-bits for invertible logic. *Physical Review X*, 7(3), 031014.
- **Faria R.**, Camsari K. Y., & Datta S. (2017). Low-barrier nanomagnets as p-bits for spin logic. *IEEE Magnetics Letters*, 8, 1-5.

After Preliminary examination

- **Faria R.**, Camsari K. Y., & Datta S. (2018). Implementing Bayesian networks with embedded stochastic MRAM. *AIP Advances*, 8(4), 045101 (Featured article).
- **Faria R.**, Kaiser J., Camsari K. Y., & Datta, S. (2019). Hardware design for autonomous Bayesian networks. To be submitted.

Other publications

- Sutton B., **Faria R.**, Ghantasala L. A., Camsari K. Y., & Datta S. (2019). Autonomous Probabilistic Coprocessing with Petaflips per Second. arXiv preprint arXiv:1907.09664. In review (*Applied Physics Reviews*).
- Kaiser J., **Faria R.**, Camsari K. Y., & Datta S. (2019). Probabilistic Circuits for Autonomous Learning: A simulation study. In review (*Frontiers in Neuroscience*).

- Debashis P., Ostwal V., **Faria R.**, Datta S., Appenzeller J., & Chen Z. (2019). Hardware Implementation of Bayesian Networks with Stochastic Spintronic Devices. In review (*Phys. Rev. Applied*).
- Debashis P., **Faria R.**, Camsari K. Y., Datta S., & Chen Z. (2019). Correlated Fluctuations in Spin Orbit Torque-Coupled Perpendicular Nanomagnets. In review (*Phys. Rev. B*).
- Hassan O., **Faria R.**, Camsari K. Y., Sun J. Z., & Datta S. (2019). Low-Barrier Magnet Design for Efficient Hardware Binary Stochastic Neurons. *IEEE Magnetics Letters*, 10, 1-5.
- Sayed, S., Camsari, K. Y., **Faria, R.**, & Datta, S. (2019). Rectification in Spin-Orbit Materials Using Low-Energy-Barrier Magnets. *Physical Review Applied*, 11(5), 054063.
- Camsari K. Y., **Faria R.**, Hassan O., Sutton B. M., & Datta S. (2018). Equivalent circuit for magnetoelectric read and write operations. *Physical Review Applied*, 9(4), 044020.
- Debashis P., **Faria R.**, Camsari K. Y., & Chen Z. (2018). Design of stochastic nanomagnets for probabilistic spin logic. *IEEE Magnetics Letters*, 9, 1-5.
- Ostwal V., Debashis P., **Faria R.**, Chen Z., & Appenzeller J. (2018). Spin-torque devices with hard axis initialization as Stochastic Binary Neurons. *Scientific reports*, 8(1), 16689.
- Camsari K. Y., **Faria R.**, Hassan O., Pervaiz A. Z., Sutton B. M., & Datta S. (2017, September). p-transistors and p-circuits for Boolean and non-Boolean logic. In *Spintronics X* (Vol. 10357, p. 10357).
- Debashis P., **Faria R.**, Camsari K. Y., Appenzeller J., Datta S., & Chen Z. (2016, December). Experimental demonstration of nanomagnet networks as hardware for ising computing. In 2016 *IEEE International Electron Devices Meeting (IEDM)* (pp. 34-3). IEEE.

- Camsari K. Y., Pervaiz A. Z., **Faria R.**, Marinero E. E., & Datta S. (2016). Ultrafast spin-transfer-torque switching of synthetic ferrimagnets. *IEEE Magnetics Letters*, 7, 1-5.
- Camsari K. Y., Pervaiz A. Z., **Faria R.**, Marinero-Caceres E. E., & Datta S. (2019). Spin-transfer-torque synthetic anti-ferromagnetic switching device. *U.S. Patent* Application No. 16/085,450.
- Sutton B., Camsari K. Y., **Faria R.**, & Datta S. (2017). Probabilistic spin logic simulator. .
- Yu Z., Park J. J., Faltens T., **Faria R.**, & Datta S. (2015). MIF generator for OOMMF. .
- **Faria R.** et. al. Fockspace analysis of p-circuit time dynamics. *Unpublished.*
- **Faria R.** et. al. Accelerating machine learning using stochastic embedded MTJ. *Unpublished (submitted in IEDM 2018).*

E. CODES

The codes used for generating the figures in this thesis are available on request to the author (email: rfaria@purdue.edu).

VITA

VITA

Rafatul Faria is a PhD candidate in the school of Electrical and Computer Engineering (ECE) at Purdue University, West Lafayette, Indiana. She is working as a Research Assistant (RA) in Professor Supriyo Datta group. She earned a Bachelor of Science (B.S.) in Electrical and Electronic Engineering (EEE) from Bangladesh University of Engineering and Technology (BUET) in April, 2012. Her PhD research is focused on proposing a new type of hardware computing platform called "Probabilistic Spin Logic (PSL)" for solving a wide variety of complex unconventional problems such as optimization, inference, invertible Boolean logic and sampling. She did theoretical simulation to design "Autonomous probabilistic hardware" that is very different from conventional digital computers and also proposed compact models for the hardware. Her undergraduate research was focused on analyzing Quantum Cascade Lasers (QCL).