



# Hardware Design for Autonomous Bayesian Networks

Rafatul Faria<sup>1\*</sup>, Jan Kaiser<sup>1</sup>, Kerem Y. Camsari<sup>2</sup> and Supriyo Datta<sup>1</sup>

<sup>1</sup> Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, United States, <sup>2</sup> Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA, United States

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), analogous to binary stochastic neurons of stochastic artificial neural networks. 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. The autonomous massively parallel operation of our proposed stochastic hardware has biological relevance since neural dynamics in brain is also stochastic and autonomous by nature.

**Keywords:** Bayesian network, probabilistic spin logic, binary stochastic neuron, magnetic tunnel junction, inference

## OPEN ACCESS

### Edited by:

Jonathan Mapelli,  
University of Modena and Reggio  
Emilia, Italy

### Reviewed by:

Francesco Maria Puglisi,  
University of Modena and Reggio  
Emilia, Italy  
Alexantrou Serb,  
University of Southampton,  
United Kingdom

### \*Correspondence:

Rafatul Faria  
rafatul.faria@gmail.com

**Received:** 18 July 2020

**Accepted:** 26 January 2021

**Published:** 08 March 2021

### Citation:

Faria R, Kaiser J, Camsari KY and  
Datta S (2021) Hardware Design for  
Autonomous Bayesian Networks.  
*Front. Comput. Neurosci.* 15:584797.  
doi: 10.3389/fncom.2021.584797

## 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 (Nikovski, 2000), genomic data analysis (Friedman et al., 2000; Jansen et al., 2003; Zou and Conzen, 2004), forecasting (Sun et al., 2006; Ticknor, 2013), robotics (Premevida et al., 2017), image classification (Arias et al., 2016; Park, 2016), neuroscience (Bielza and Larrañaga, 2014), 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* (Heckerman and Breese, 1996; Koller and Friedman, 2009; Pearl, 2014; Russell and Norvig, 2016). The CPTs can be obtained from expert knowledge and/or machine learned from data (Darwiche, 2009). Each node and edge in a BN have meaning representing specific probabilistic events and their conditional dependencies and they are easier to interpret (Correa et al., 2009) than neural networks where the hidden nodes do not necessarily have meaning. Unlike neural networks where useful information is extracted only at the output nodes for prediction purposes, BNs are useful for both prediction and inference by looking at not only the output nodes but also other nodes of interest. 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 implementations of BNs for efficient inference (Rish et al., 2005; Chakrapani et al., 2007; Weijia et al., 2007; Jonas, 2014; Querlioz et al., 2015; Zermani et al., 2015; Behin-Aein et al., 2016; Friedman et al., 2016; Thakur et al., 2016; Tylman et al., 2016; Shim et al., 2017). In this article, we have elucidated the design criteria for an 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 (Camsari et al., 2017a) based on a building block called probabilistic bits (p-bits) that are analogous to binary stochastic neurons (BSN) (Ackley et al., 1985; Neal, 1992) of the artificial neural network (ANN) literature. p-bits can be interconnected to solve a wide variety of problems such as optimization (Sutton et al., 2017; Borders et al., 2019), inference (Faria et al., 2018), an enhanced type of Boolean logic that is invertible (Camsari et al., 2017a; Faria et al., 2017; Pervaiz et al., 2017, 2018), quantum emulation (Camsari et al., 2019), and *in situ* learning from probability distributions (Kaiser et al., 2020).

Unlike conventional deterministic networks built out of deterministic, stable bits, stochastic or probabilistic networks composed of p-bits (**Figure 1A**), can be correlated by interconnecting them to construct p-circuits defined by two equations (Ackley et al., 1985; Neal, 1992; Camsari et al., 2017a): (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:

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

where  $\text{rand}(-1, +1)$  is a random number uniformly distributed between  $-1$  and  $+1$ , and  $\tau_N$  is the neuron evaluation time.

The synapse generates the input  $I_i$  from a weighted sum of the states of other p-bits. 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:

$$I_i(t + \tau_S) = I_0 \left( h_i + \sum_j J_{ij} m_j(t) \right) \quad (1b)$$

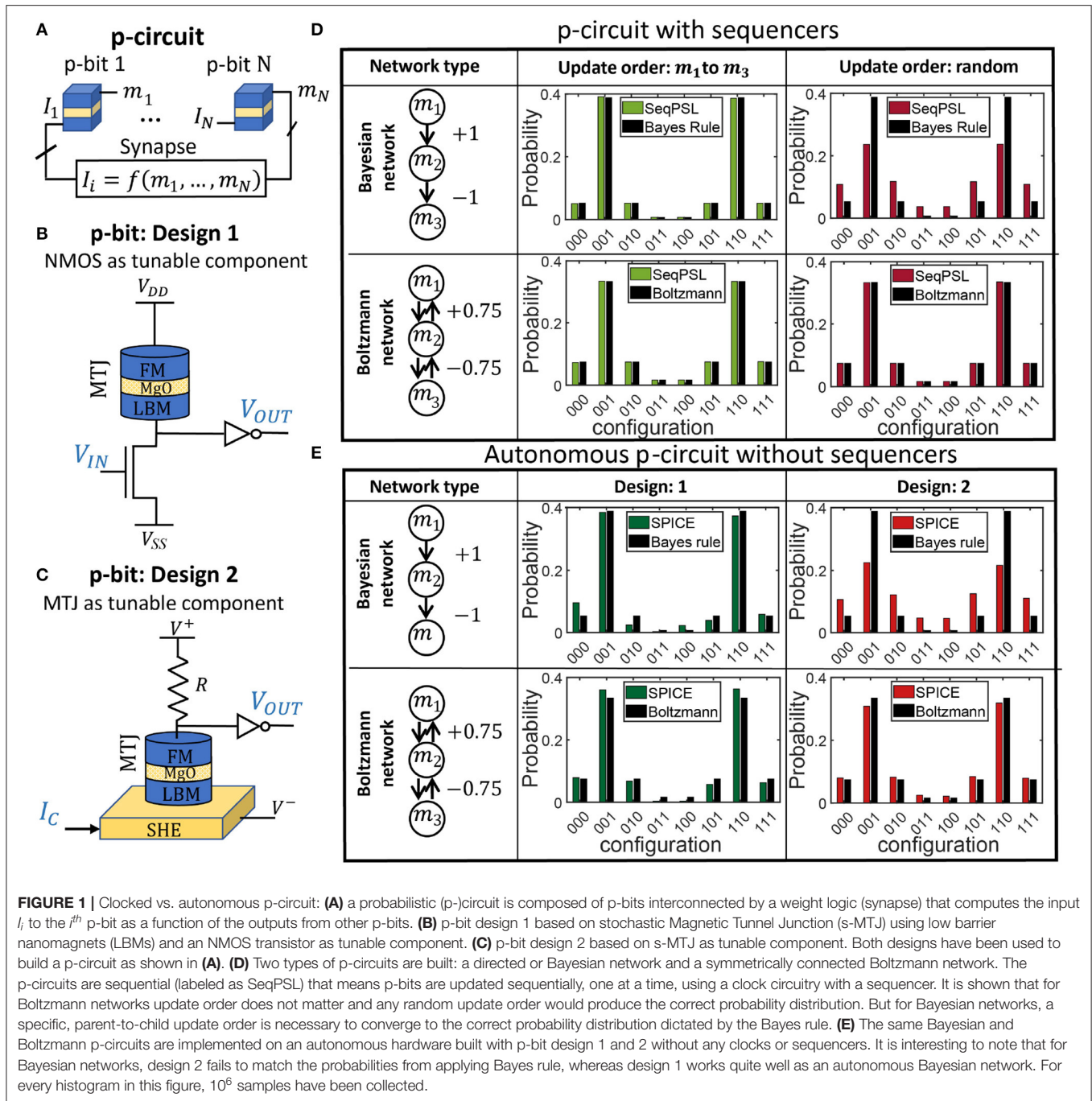
where  $h_i$  is the on-site bias and  $J_{ij}$  is the weight of the coupling from  $j^{\text{th}}$  p-bit to  $i^{\text{th}}$  p-bit,  $I_0$  parameterizes the coupling strength between p-bits, and  $\tau_S$  is the synapse evaluation time. Several hardware designs of p-bits based on low barrier nanomagnet (LBM) physics have been proposed and also experimentally demonstrated (Ostwal et al., 2018; Borders et al., 2019; Ostwal and Appenzeller, 2019; Camsari et al., 2020; Debashis, 2020). The thermal energy barrier of the LBM is of the order of a few  $k_B T$  instead of  $40\text{--}60 k_B T$  used in the memory technology to retain stability. Because of thermal noise the magnetization of the LBM keeps fluctuating as a function of time with an average retention time  $\tau \sim \tau_0 \exp(E_B/k_B T)$  (Brown, 1979), where  $\tau_0$  is a material-dependent parameter called attempt time that is experimentally found to be in the range of nanosecond or less and

$E_B$  is the thermal energy barrier (Lopez-Diaz et al., 2002; Pufall et al., 2004). The stochasticity of the LBMs makes them naturally suitable for p-bit implementation.

**Figure 1** shows two p-bit designs: Design 1 (**Figure 1B**) (Camsari et al., 2017b; Borders et al., 2019) and Design 2 (**Figure 1C**) (Camsari et al., 2017a; Ostwal and Appenzeller, 2019). Designs 1 and 2 both are fundamental building blocks of spin transfer torque (STT) and spin orbit torque (SOT) magnetoresistive random access memory (MRAM) technologies, respectively (Bhatti et al., 2017). Their technological relevance motivates us to explore their implementations as p-bits. Design 1 is very similar to the commercially available 1T/1MTJ (T: Transistor, MTJ: Magnetic Tunnel Junction) embedded MRAM device where the free layer of the MTJ is replaced by an in-plane magnetic anisotropy (IMA) or perpendicular magnetic anisotropy (PMA) LBM. Design 2 is similar to the basic building block of SOT-MRAM device (Liu et al., 2012) where the thermal fluctuation of the free layer magnetization of the stochastic MTJ (s-MTJ) (Vodenicarevic et al., 2017, 2018; Mizrahi et al., 2018; Parks et al., 2018; Zink et al., 2018; Borders et al., 2019) is tuned by a spin current generated in a heavy metal layer underneath the LBM due to SOT effect. The in-plane polarized spin current from the SOT effect in the spin hall effect (SHE) material in design 2 requires an in-plane LBM to tune its magnetization, although a perpendicular LBM with a tilted anisotropy axis is also experimentally shown to work (Debashis et al., 2020). However, 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-plane LBMs also provide faster fluctuation than perpendicular ones leading to faster sampling speed in the probabilistic hardware (Hassan et al., 2019; Kaiser et al., 2019).

The key distinguishing feature of the two p-bit designs (designs 1 and 2) is the time scales in implementing Equation (1a). From a hardware point of view, Equation (1a) has two components: a random number generator (RNG) ( $\text{rand}$ ) and a tunable component ( $\tanh$ ). In design 1, the RNG is the s-MTJ utilizing an LBM and the tunable component is the NMOS transistor, thus having two different time scales in the equation. But in design 2, both the RNG and the tunable component are implemented by a single s-MTJ utilizing an LBM, thus having just one time scale in the equation. This difference in time scales in the two designs is shown in **Figure 2**. Note that although the two p-bit designs have the same RNG source, namely a fluctuating magnetization, it is the difference in their circuit configuration with or without the NMOS transistor in the MTJ branch that results in different time dynamics of the two designs.

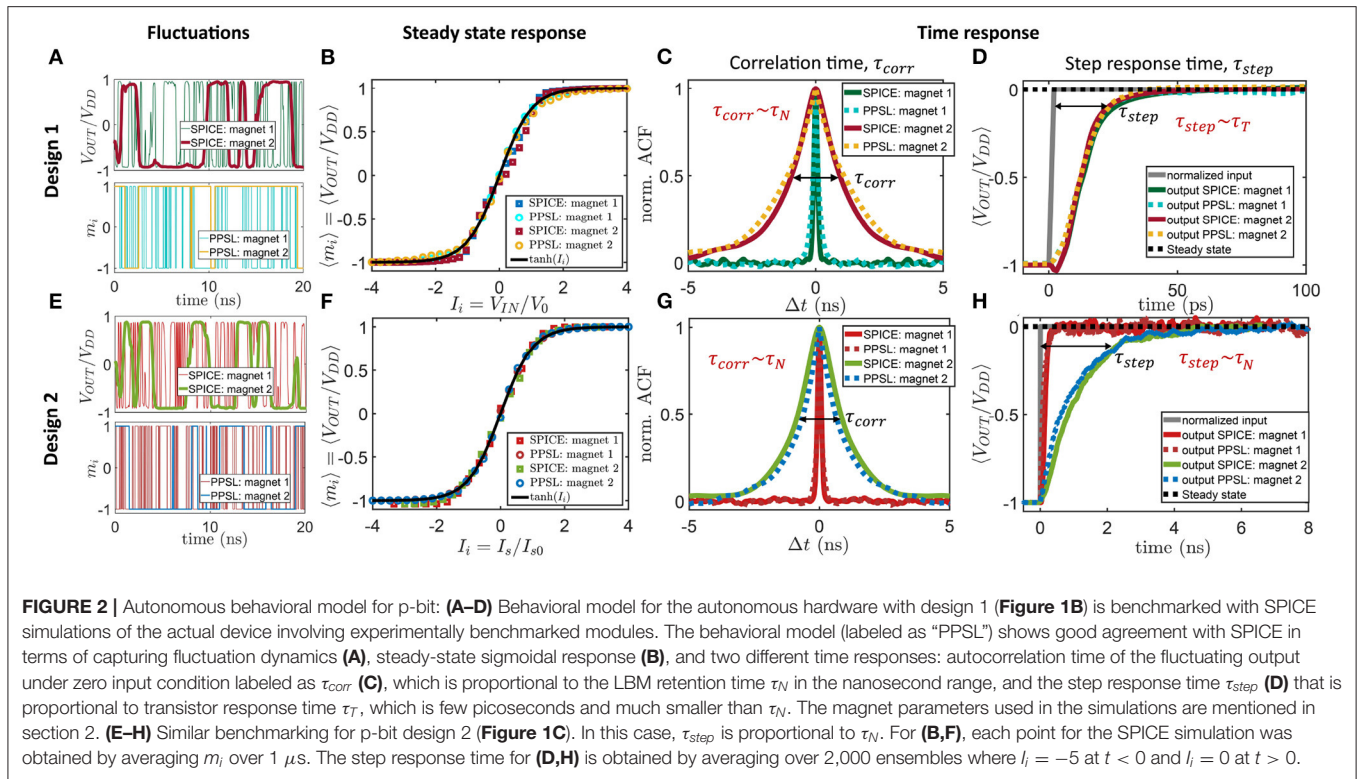
In traditional software implementations, p-bits are updated sequentially for accurate operation such that after each  $\tau_S + \tau_N$  time interval, only one p-bit is updated (Hinton, 2007). This naturally implies the use of sequencers to ensure the sequential update of p-bits. The sequencer generates an *Enable* signal for each p-bit in the network and ensures that no two p-bits update simultaneously. The sequencer also makes sure that every p-bit is updated at least once in a time step where each time step corresponds to  $N \cdot (\tau_S + \tau_N)$ ,  $N$  being the number of p-bits in the network. (Roberts and Sahu, 1997; Pervaiz et al.,



2018). 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 BNs 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 (Neal, 1992) similar to the concept of forward sampling in belief networks (Henrion,

1988; Guo and Hsu, 2002; Koller and Friedman, 2009). As long as this parent to child update order is maintained, the network converges 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 **Figure 1D**. In the **Supplementary Material**, it is shown in an example how the CPT of the BN can be mapped to a p-circuit following Faria et al. (2018).

Unlike sequential p-circuits in ANN literature, the distinguishing feature of our probabilistic hardware is that



it is *autonomous* where each p-bit runs in parallel without any clocks or sequencers. This autonomous p-circuit (ApC) allows massive parallelism potentially providing peta flips per second sampling speed (Sutton et al., 2020). The complete sequencer-free operation of our “autonomous” p-circuit is very different from the “asynchronous” operation of spiking neural networks (Merolla et al., 2014; Davies et al., 2018). 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. Note that even if p-bits update sequentially, 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 a key design rule for an ApC. If the input of the p-bit is based on old state of neighboring p-bits or on time-integrated synaptic inputs, the ApC operation declines in functionality or fails completely. However, for  $\tau_S \ll \tau_N$ , the ApC works properly for a Boltzmann network without any clock because no specific update order is required in this case. But, it is not intuitive at all if an ApC would work for a BN because a particular parent to child *informed* update order is required in this case, as shown in Figure 1D. As such, it is not straightforward that a clockless autonomous circuit can naturally ensure this specific informed update order. In Figure 1E, 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 BN without any clocks. In

Figure 1E, two p-bit designs are evaluated for implementing both Boltzmann network and BN. We have shown that design 1 is suitable for both Boltzmann network and BN. But design 2 is suitable for Boltzmann networks only and does not work for BNs in general. The synapse in both types of p-circuits is implemented using a resistive crossbar architecture (Alibart et al., 2013; Camsari et al., 2017b), although there are also other types of hardware synapse implementations based on memristors (Li et al., 2018; Mahmoodi et al., 2019; Mansueto et al., 2019), magnetic tunnel junctions (Ostwal et al., 2019), spin orbit torque driven domain wall motion devices (Zand et al., 2018), phase change memory devices (Ambrogio et al., 2018), and so on. In all the simulations,  $\tau_S$  is assumed to be negligible compared to other time scales in the circuit dynamics.

Our proposed probabilistic hardware for BNs shows significant biological relevance because of the following reasons: (1) The brain consists of neurons and synapses. The basic building block called “p-bit” of our proposed hardware mimics the neuron and the interconnection among p-bits mimics the synapse function. (2) The components of brain are stochastic or noisy by nature. p-bits mimicking the neural dynamics in our proposed hardware are also stochastic. (3) Brain does not have a single clock for synchronous operation and can perform massively parallel processing (Strukov et al., 2019). Our autonomous hardware also does not have any global clock or sequencers and each p-bit fluctuates in parallel allowing massively parallel operation.

Further, we have provided a behavioral model in section 2 for both designs 1 and 2, illustrating the essential characteristics

needed for correct sequencer-free 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.

## 2. BEHAVIORAL MODEL FOR AUTONOMOUS HARDWARE

In this section, we will develop an autonomous behavioral model that we will call parallel probabilistic spin logic (PPSL) for design 1 (**Figure 1B**) and revisit the behavioral model for design 2, which was proposed by Sutton et al. (2020). The term “Parallel” refers to all the p-bits fluctuating in parallel without any clocks or sequencers. These behavioral models are high-level representations of the p-circuit and p-bit behavior and connect Equations (1a) and (1b) to the hardware p-bit designs. Please note the parameters introduced in these models will represent certain parts of the p-bit and synapse behavior like MTJ resistances ( $r_{MTJ}$ ) and transistor resistances ( $r_T$ ) but are generally dimensionless apart from time variables (e.g.,  $\tau_T, \tau_N$ ). The advantage of these models is that they are computationally less expensive to use than full SPICE simulations while preserving the crucial device and system characteristics.

### 2.1. Autonomous Behavioral Model: Design 1

The autonomous circuit behavior of design 1 can be explained by slightly modifying the two equations (Equations 1a,b) stated in section 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 an average dwell time of the fluctuation denoted by  $\tau_N$ . The NMOS transistor tunable resistance is denoted by  $r_T$  and the inverter is represented by a  $sgn$  function. Thus, the normalized output  $m_i = V_{OUT,i}/V_{DD}$  of the  $i_{th}$  p-bit can be expressed as:

$$m_i(t + \Delta t) = sgn(r_{T,i}(t + \Delta t) - r_{MTJ,i}(t + \Delta t)) \quad (2)$$

where  $\Delta t$  is the simulation time step,  $r_{T,i}$  represents the NMOS transistor resistance tunable by the normalized input  $I_i = V_{IN,i}/V_0$  (compare Equation 1a) where  $V_0$  is a fitting parameter which is  $\approx 50$  mV for the chosen parameters and transistor technology (compare **Figure 2B**) and  $r_{MTJ,i}$  is a correlated random number generator with an average retention time of  $\tau_N$ . For design 1, the transistor represents the tunable component that works in conjunction with the unbiased stochastic signal of the MTJ.  $r_{T,i}$  as a function of input  $I_i$  is approximated by a tanh function with a response time denoted by  $\tau_T$  modeled by the following equations:

$$r_{T,i}(t + \Delta t) = r_{T,i}(t) \exp(-\Delta t/\tau_T) + (1 - \exp(-\Delta t/\tau_T)) (\tanh(I_i(t + \Delta t))) \quad (3)$$

where it can be clearly seen that the dimensionless quantity  $r_{T,i}$  representing the transistor resistance is bounded by  $-1 \leq r_{T,i} \leq 1$  for all synaptic inputs. **Figure 2B** shows by utilizing SPICE simulation how  $I_i$  influences the average output  $m_i$  and shows

that the average response of the circuit is in good agreement with the tanh-function used in Equation (3).

The synapse delay  $\tau_S$  in computing the input  $I_i$  can be modeled by:

$$I_i(t + \Delta t) = I_i(t) \exp(-\Delta t/\tau_S) + (1 - \exp(-\Delta t/\tau_S)) \left( I_0 \left( \sum_j J_{ij} m_j(t) + h_j \right) \right) \quad (4)$$

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

$$r_{flip,i}(t + \Delta t) = sgn \left( \exp \left( -\frac{\Delta t}{\tau_N} \right) - \text{rand}_{[0,1]} \right) \quad (5a)$$

where  $\text{rand}_{[0,1]}$  is a uniformly distributed random number between 0 and 1 and  $\tau_N$  represents the average retention time of the fluctuating MTJ resistance. If  $r_{flip,i}$  is  $-1$ , a new random  $r_{MTJ}$  will be chosen between  $-1$  and  $+1$ . Otherwise, the previous  $r_{MTJ}(t)$  will be kept in the next time step ( $t + \Delta t$ ), which can be expressed as

$$r_{MTJ,i}(t + \Delta t) = \frac{r_{flip,i}(t + \Delta t) + 1}{2} r_{MTJ,i}(t) - \frac{r_{flip,i}(t + \Delta t) - 1}{2} \text{rand}_{[-1,1]} \quad (5b)$$

where  $-1 \leq r_{MTJ,i}(t) \leq 1$ .

The charge current flowing through the MTJ branch of p-bit design 1 can get polarized by the fixed layer of the MTJ and generate a spin current  $I_s$  that can tune/pin the MTJ dynamics by modifying  $\tau_N$ . This effect is needed for tuning the output of design 2 but is not desired in design 1. However, the developed behavioral model can account for this pinning effect according to

$$\tau_N = \tau_N^0 \exp(r_{MTJ} I_{MTJ}), \quad (6)$$

where  $\tau_N^0$  is the retention time of  $r_{MTJ}$  when  $I_{MTJ} = 0$ . The dimensionless pinning current  $I_{MTJ}$  is defined as  $I_{MTJ} = I_s/I_{s,0}$  where  $I_{s,0}$  can be extracted by following the procedure of **Figure 2F**. This pinning effect by  $I_{MTJ}$  is much smaller in in-plane magnets (IMA) than perpendicular magnets (PMA) (Hassan et al., 2019) and is ignored for design 1 throughout this paper.

**Figures 2A–D** shows the comparison of this behavioral model for p-bit design 1 with SPICE simulation of the actual hardware in terms of fluctuation dynamics, sigmoidal characteristic response, autocorrelation time ( $\tau_{corr}$ ), and step response time ( $\tau_{step}$ ) and in all cases the behavioral model closely matches SPICE simulations. The SPICE simulation involves experimentally benchmarked modules for different parts of the device. The SPICE model for the s-MTJ model solves the stochastic Landau–Lifshitz–Gilbert equation for the LBM physics. For the transistors, 14 nm Predictive Technology Model<sup>1</sup> is used. As simulator *HSPICE*

<sup>1</sup><http://ptm.asu.edu/>

is utilized with the `.trnoise` function and a time step of 1 ps. The simulating framework was benchmarked experimentally and by using standard simulation tools in the field (Datta, 2012; Torunbalci et al., 2018). The autonomous 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  $D = 22$  nm, thickness  $th = 2$  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  $D = 150$  nm. The supply voltage was set to  $V_{DD} = -V_{SS} = 0.4$  V. The fast and slow fluctuations of the normalized output  $m_i = V_{OUT,i}/V_{DD}$  are captured by changing the  $\tau_N$  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 40–60 mV reasonably well depending on which part of the sigmoid needs to be better matched. In **Figure 2B**,  $V_0$  value of 50 mV is used to fit the sigmoid from SPICE simulation. The following parameters have been extracted from the calibration shown in **Figure 2**, where  $\Delta t = 1$  ps was used:  $\tau_N = 150$  ps (magnet 1),  $\tau_N = 1.5$  ns (magnet 2),  $\tau_T = 3$  ps,  $I_{s,0} = 120$   $\mu$ A (magnet 1),  $I_{s,0} = 1$  mA (magnet 2).

There are two types of time responses: (1) Autocorrelation time under zero input condition labeled as  $\tau_{corr}$  and (2) step response time  $\tau_{step}$ . The full width half maximum (FWHM) of the autocorrelation function of the fluctuating output under zero input is defined by  $\tau_{corr}$ , which is proportional to the retention time  $\tau_N$  of the LBM. The step response time  $\tau_{step}$  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  $t = 0$  and measuring the time it takes for the ensemble averaged output to reach its statistically correct value consistent with the new input.  $\tau_{step}$  defines how fast the first statistically correct sample can be obtained after the input is changed. For p-bit design 1,  $\tau_{step}$  is independent of LBM retention time  $\tau_N$  and is defined by the NMOS transistor response time  $\tau_T$ , which is much faster (few picoseconds) than LBM fluctuation time  $\tau_N$ . The effect of these two very different time scales in design 1 ( $\tau_{step} \ll \tau_{corr}$ ) on an autonomous BN is described in section 3.

## 2.2. Autonomous Behavioral Model: Design 2

The autonomous behavioral model for design 2 is proposed in Sutton et al. (2020). In this article, we have benchmarked this model with the SPICE simulation of the single p-bit steady state and time responses shown in **Figures 2E–H**. According to this model, the normalized output  $m_i = V_{OUT,i}/V_{DD}$  can be expressed as:

$$m_i(t + \Delta t) = m_i(t) \operatorname{sgn} \left( p_{NOTflip,i}(t + \Delta t) - \operatorname{rand}_{[0,1]} \right) \quad (7a)$$

$$p_{NOTflip,i}(t + \Delta t) = \exp \left( - \frac{\Delta t}{\tau_N \exp(I_i m_i(t))} \right) \quad (7b)$$

where  $p_{NOTflip,i}(t + \Delta t)$  is the probability of retention of the  $i^{\text{th}}$  p-bit (or “not flipping”) in the next time step that

is a function of average neuron flip time  $\tau_N$ , input  $I_i$ , and the current p-bit output  $m_i(t)$ . **Figure 2** shows how this simple autonomous behavioral model for design 2 matches reasonably well with SPICE simulation of the device in terms of fluctuation dynamics (**Figure 2E**), sigmoidal characteristic response (**Figure 2F**), autocorrelation time ( $\tau_{corr}$ ) (**Figure 2G**), and step response time ( $\tau_{step}$ ) (**Figure 2H**). In design 2,  $\tau_{step}$  and  $\tau_{corr}$  are both proportional to LBM fluctuation time  $\tau_N$  unlike design 1.

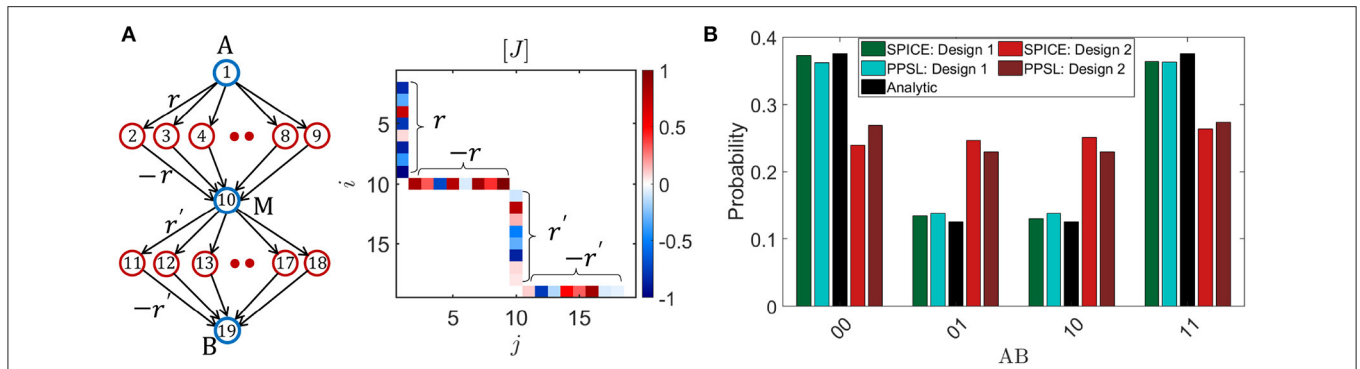
Different time scales in p-bit designs 1 and 2 are also reported in Hassan et al. (2019) in an energy-delay analysis context. In this article, we explain the effect of these time scales in designing an autonomous BN (section 3).

## 3. DIFFERENCE BETWEEN DESIGNS 1 AND 2 IN IMPLEMENTING BAYESIAN NETWORKS

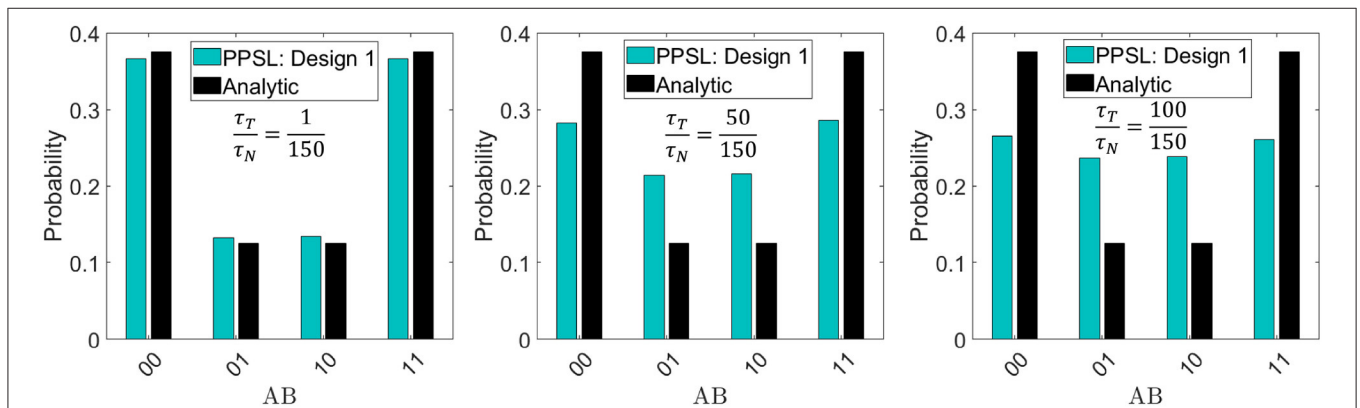
The behavioral models introduced in section 2 are applied to implement a multi-layer belief/BN with 19 p-bits and random interconnection strengths between +1 and -1 (**Figure 3A**). For illustrative purposes, the interconnections are designed in such a way that although there are no meaningful correlations 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. **Figure 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), and 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. We have tested this basic conclusion for other networks as well with more complex topology as shown in **Supplementary Figure 1**. The analytical values are obtained from applying the standard joint probability rule for BNs (Pearl, 2014; Russell and Norvig, 2016), which is:

$$P(x_1, x_2, \dots, x_N) = \prod_{i=1}^N P(x_i | \text{Parents}(x_i)) \quad (8)$$

Joint probability between two specific nodes  $x_i$  and  $x_j$  can be calculated from the above equation by summing over all configurations of the other 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 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 (Feynman, 1982). Indeed, in the BN example in



**FIGURE 3 |** Difference between designs 1 and 2: **(A)** The behavioral models described in **Figure 2** are applied to simulate a 19 p-bit BN with random  $J_{ij}$  between +1 and -1. The indices  $i$  and  $j$  of  $J_{ij}$  correspond to the numbers inside each circle. 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 work as an autonomous BN in general. For each histogram,  $10^6$  samples have been collected.



**FIGURE 4 |** Effect of step response time in design 1: The reason for design 1 to work accurately as an autonomous Bayesian network as shown in **Figure 3** is the two different time scales ( $\tau_T$  and  $\tau_N$ ) in this design with the condition that  $\tau_T \ll \tau_N$ . The same histogram shown in **Figure 3** is plotted using the proposed behavioral model for different  $\tau_T/\tau_N$  ratios and compared with the analytical values. It can be seen that as  $\tau_T$  gets comparable to  $\tau_N$ , the probability distribution diverges from the standard statistical values. For each histogram  $10^6$  samples have been collected.

**Figure 3**, the probabilities of different configurations of nodes  $A$  and  $B$  were obtained 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 Faria et al. (2018). An example of this mapping is given in the **Supplementary Material**.

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 **Figure 2** due to the fact that the tunable component is the transistor in design 1 ( $\tau_{step} \propto \tau_T$ ) and the MTJ in design 2 ( $\tau_{step} \propto \tau_N$ ). It is these two different time scales in design 1 ( $\tau_{step} \ll \tau_{corr}$ ) that naturally ensures a parent to child informed update order in a BN. The reason is that when  $\tau_{step}$  is small, each child node can immediately respond to any change of its parent nodes that happens due to a random event, which have a much larger time scale  $\propto \tau_{corr}$ . Thus, due to that fast step response, information about changing p-bits at the parent

node can propagate quickly through the network and the output of the child nodes can be conditionally satisfied with the parent nodes very fast. Otherwise, if  $\tau_{corr}$  gets comparable to  $\tau_{step}$ , the child nodes will not be able to keep up with the fast changing parent nodes since the information of the parent p-bit state has not been propagated through the network. As a result, the child nodes will produce a substantial number of statistically incorrect samples over the entire time range, thus deviating from the correct probability distribution. This effect is especially strong for networks where the coupling strength between p-bits is large.

To illustrate this point, the effect of  $\tau_{step}/\tau_{corr}$  ratio is shown in **Figure 4** for the same BN presented in **Figure 3** by plotting the histogram of  $AB$  configurations for different  $\tau_T/\tau_N$  ratios. It is shown that when  $\tau_T/\tau_N$  ratio is small, the histogram converges to the correct distribution. As  $\tau_T$  gets comparable to  $\tau_N$ , 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 BN hardware. One thing to note that under certain conditions, results from design 2 can also match

the analytical results if spin current bias is large enough to drive down the fast step response time to ensure  $\tau_{step} \ll \tau_{corr}$ .

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 BN demands an additional p-bit design rule that 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 in-plane magnet whose magnetization spans all values between +1 and -1 and negligible pinning effect. If the LBM is a PMA magnet with bipolar fluctuations having just two values +1 and -1, design 1 will not provide any sigmoidal response except with substantial pinning effect (Borders et al., 2019). Under this condition,  $\tau_{step}$  of design 1 will be comparable to  $\tau_N$  again and the system will not work as an autonomous BN in general. Therefore, LBM with continuous range fluctuation is expected for design 1 p-bit to work properly as a BN.

## 4. DISCUSSION

In this article, we have elucidated the design criteria for an autonomous clockless hardware for BNs that requires a specific parent to child update order when implemented on a probabilistic circuit. By performing SPICE simulations of two autonomous probabilistic hardware designs built out of p-bits (designs 1 and 2 in **Figure 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 ( $\tau_{step}$ ) of the p-bit is much smaller than its autocorrelation time ( $\tau_{corr}$ ). This criteria of having two different time scales is met in design 1 as  $\tau_{step}$  comes from the NMOS transistor response time  $\tau_T$  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 are performed ignoring some non-ideal effects listed as follows:

- Pinning of the s-MTJ fluctuation due to STT effect is ignored by assuming  $I_{MTJ} = 0$  in Equation (6). This is a reasonable assumption considering circular in-plane magnets that are very difficult to pin due to the large demagnetization field that is always present, irrespective of the energy barrier (Hassan et al., 2019). This effect is more prominent in perpendicular anisotropy magnets (PMA) magnets. It is important to include the pinning effect in p-bits with bipolar LBM fluctuations because in this case the p-bit does not provide a sigmoidal response without the pinning current. This effect is also experimentally observed in Borders et al. (2019) for PMA magnets. Such a p-bit design with bipolar PMA and STT pinning might not work for BNs in general, because in this case  $\tau_{step}$  will be dependent on magnet fluctuation time  $\tau_N$ .
- In the proposed behavioral model, the step response time of the NMOS transistor  $\tau_T$  in design 1 is assumed to be independent of the input  $I$ . But there is a functional dependence of  $\tau_T$  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 a 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 effects on different probability distributions shown in this article. Real LBMs may suffer from common fabrication defects, resulting in variations in average magnet fluctuation time  $\tau_N$  (Abeed and Bandyopadhyay, 2019). The autonomous BN is also quite tolerant to such variations in  $\tau_N$  as long as  $\tau_T \ll \min(\tau_N)$ .

It is important to note that, for design 1 (Transistor-controlled) to function as a p-bit that has a step response time ( $\tau_{step}$ ) much smaller than its average fluctuation time ( $\tau_N$ ), the LBM fluctuation needs to be continuous and not bipolar. It is important to note that while most experimental implementations of low barrier magnetic tunnel junctions or spin-valves exhibit telegraphic (binary) fluctuations (Pufall et al., 2004; Locatelli et al., 2014; Parks et al., 2018; Debashis et al., 2020), theoretical results (Abeed and Bandyopadhyay, 2019; Hassan et al., 2019; Kaiser et al., 2019) indicate that it should be possible to design low barrier magnets with continuous fluctuations. Preliminary experimental results for such circular disk nanomagnets have been presented in Debashis et al. (2016). We believe that a lack of experimental literature on such magnets is partly due to the lack of interest of randomly fluctuating magnets that have long been discarded as impractical and irrelevant. The other experimentally demonstrated p-bits (Ostwal et al., 2018; Ostwal and Appenzeller, 2019; Debashis, 2020) fall under design 2 category with the LBM magnetization tuned by SOT effect and are not suitable for autonomous BN operation in general. It might also be possible to design p-bits using other phenomena such as voltage controlled magnetic anisotropy (Amiri and Wang, 2012), but this is beyond the scope of the present study. Here, we have specifically focused on two designs that can be implemented with existing MRAM technology based on STT and SOT.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

## AUTHOR CONTRIBUTIONS

RF and SD wrote the paper. RF performed the simulations. JK helped setting up the simulations. KC developed the simulation modules for the BSN in SPICE. All authors discussed the results and helped refine the manuscript.

## FUNDING

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



## ACKNOWLEDGMENTS

The authors would like to thank professor Joerg Appenzeller from Purdue University for helpful discussions. This manuscript has been released as a preprint at arXiv (Faria et al., 2020).

## REFERENCES

- Abeed, M. A., and Bandyopadhyay, S. (2019). Low energy barrier nanomagnet design for binary stochastic neurons: design challenges for real nanomagnets with fabrication defects. *IEEE Magnet. Lett.* 10, 1–5. doi: 10.1109/LMAG.2019.2929484
- Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cogn. Sci.* 9, 147–169. doi: 10.1207/s15516709cog0901\_7
- Alibart, F., Zamanidoost, E., and Strukov, D. B. (2013). Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training. *Nat. Commun.* 4, 1–7. doi: 10.1038/ncomms3072
- Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R. M., Boybat, I., di Nolfo, C., et al. (2018). Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* 558, 60–67. doi: 10.1038/s41586-018-0180-5
- Amiri, P. K., and Wang, K. L. (2012). “Voltage-controlled magnetic anisotropy in spintronic devices,” in *Spin*, Vol. 2 (World Scientific), 1240002. doi: 10.1142/S2010324712400024
- Arias, J., Martinez-Gomez, J., Gamez, J. A., de Herrera, A. G. S., and Müller, H. (2016). Medical image modality classification using discrete bayesian networks. *Comput. Vis. Image Understand.* 151, 61–71. doi: 10.1016/j.cviu.2016.04.002
- Behin-Aein, B., Diep, V., and Datta, S. (2016). A building block for hardware belief networks. *Sci. Rep.* 6:29893. doi: 10.1038/srep29893
- Bhatti, S., Sbiaa, R., Hirohata, A., Ohno, H., Fukami, S., and Piramanayagam, S. (2017). Spintronics based random access memory: a review. *Mater. Tdy.* 20, 530–548. doi: 10.1016/j.mattod.2017.07.007
- Bielza, C., and Larrañaga, P. (2014). Bayesian networks in neuroscience: a survey. *Front. Comput. Neurosci.* 8:131. doi: 10.3389/fncom.2014.00131
- Borders, W. A., Pervaiz, A. Z., Fukami, S., Camsari, K. Y., Ohno, H., and Datta, S. (2019). Integer factorization using stochastic magnetic tunnel junctions. *Nature* 573, 390–393. doi: 10.1038/s41586-019-1557-9
- Brown, W. (1979). Thermal fluctuation of fine ferromagnetic particles. *IEEE Trans. Magnet.* 15, 1196–1208. doi: 10.1109/TMAG.1979.1060329
- Camsari, K. Y., Chowdhury, S., and Datta, S. (2019). Scalable emulation of sign-problem-free hamiltonians with room-temperature *p*-bits. *Phys. Rev. Appl.* 12:034061. doi: 10.1103/PhysRevApplied.12.034061
- Camsari, K. Y., Debashis, P., Ostwal, V., Pervaiz, A. Z., Shen, T., Chen, Z., et al. (2020). From charge to spin and spin to charge: Stochastic magnets for probabilistic switching. *Proc. IEEE.* 108:1322–1337. doi: 10.1109/JPROC.2020.2966925
- Camsari, K. Y., Faria, R., Sutton, B. M., and Datta, S. (2017a). Stochastic *p*-bits for invertible logic. *Phys. Rev. X* 7:031014. doi: 10.1103/PhysRevX.7.031014
- Camsari, K. Y., Salahuddin, S., and Datta, S. (2017b). Implementing *p*-bits with embedded mtj. *IEEE Electron Dev. Lett.* 38, 1767–1770. doi: 10.1109/LED.2017.2768321
- Chakrapani, L. N., Korkmaz, P., Akgul, B. E., and Palem, K. V. (2007). Probabilistic system-on-a-chip architectures. *ACM Trans. Design Automat. Electron. Syst.* 12:29. doi: 10.1145/1255456.1255466
- Correa, M., Bielza, C., and Pamiès-Teixeira, J. (2009). Comparison of bayesian networks and artificial neural networks for quality detection in a machining process. *Expert Syst. Appl.* 36, 7270–7279. doi: 10.1016/j.eswa.2008.09.024
- Darwiche, A. (2009). *Modeling and Reasoning With Bayesian Networks*. Cambridge University Press. Available online at: <https://www.cambridge.org/core/books/modeling-and-reasoning-with-bayesian-networks/8A3769B81540EA93B525C4C2700C9DE6> doi: 10.1017/CBO9780511811357
- Datta, D. (2012). *Modeling of spin transport in MTJ devices* (Ph.D. thesis), Purdue University Graduate School; ProQuest. Available online at: <https://docs.lib.purdue.edu/dissertations/AA13556189/>
- Davies, M., Srinivasa, N., Lin, T.-H., China, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359
- Debashis, P. (2020). *Spintronic devices as P-bits for probabilistic computing* (Ph.D. thesis). Purdue University Graduate School. Available online at: [https://hammer.figshare.com/articles/thesis/Spintronic\\_Devices\\_as\\_P-bits\\_for\\_Probabilistic\\_Computing/11950395](https://hammer.figshare.com/articles/thesis/Spintronic_Devices_as_P-bits_for_Probabilistic_Computing/11950395)
- Debashis, P., Faria, R., Camsari, K. Y., Appenzeller, J., Datta, S., and Chen, Z. (2016). “Experimental demonstration of nanomagnet networks as hardware for ising computing,” in *2016 IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA: IEEE), 34. doi: 10.1109/IEDM.2016.7838539
- Debashis, P., Faria, R., Camsari, K. Y., Datta, S., and Chen, Z. (2020). Correlated fluctuations in spin orbit torque coupled perpendicular nanomagnets. *Phys. Rev. B* 101:094405. doi: 10.1103/PhysRevB.101.094405
- Faria, R., Camsari, K. Y., and Datta, S. (2017). Low-barrier nanomagnets as *p*-bits for spin logic. *IEEE Magnet. Lett.* 8, 1–5. doi: 10.1109/LMAG.2017.2685358
- Faria, R., Camsari, K. Y., and Datta, S. (2018). Implementing bayesian networks with embedded stochastic mram. *AIP Adv.* 8:045101. doi: 10.1063/1.5021332
- Faria, R., Kaiser, J., Camsari, K. Y., and Datta, S. (2020). Hardware design for autonomous bayesian networks. *arXiv preprint arXiv:2003.01767*.
- Feynman, R. P. (1982). Simulating physics with computers. *Int. J. Theor. Phys.* 21:467–488. doi: 10.1007/BF02650179
- Friedman, J. S., Calvet, L. E., Bessi re, P., Droulez, J., and Querlioz, D. (2016). Bayesian inference with muller *c*-elements. *IEEE Trans. Circ. Syst. I Regular Pap.* 63, 895–904. doi: 10.1109/TCSI.2016.2546064
- Friedman, N., Linal, M., Nachman, I., and Pe’er, D. (2000). Using bayesian networks to analyze expression data. *J. Comput. Biol.* 7, 601–620. doi: 10.1089/106652700750050961
- Guo, H., and Hsu, W. (2002). “A survey of algorithms for real-time bayesian network inference,” in *Join Workshop on Real Time Decision Support and Diagnosis Systems*. American Association for Artificial Intelligence. Available online at: <https://www.aaai.org/Papers/Workshops/2002/WS-02-15/WS02-15-001.pdf>
- Hassan, O., Faria, R., Camsari, K. Y., Sun, J. Z., and Datta, S. (2019). Low-barrier magnet design for efficient hardware binary stochastic neurons. *IEEE Magnet. Lett.* 10, 1–5. doi: 10.1109/LMAG.2019.2910787
- Heckerman, D., and Breese, J. S. (1996). Causal independence for probability assessment and inference using bayesian networks. *IEEE Trans. Syst. Man Cybernet A Syst. Hum.* 26, 826–831. doi: 10.1109/3468.541341
- Henrion, M. (1988). “Propagating uncertainty in bayesian networks by probabilistic logic sampling,” in *Machine Intelligence and Pattern Recognition*, (Elsevier) 5:149–163. doi: 10.1016/B978-0-444-70396-5.50019-4
- Hinton, G. E. (2007). Boltzmann machine. *Scholarpedia* 2:1668. doi: 10.4249/scholarpedia.1668
- Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., Chung, S., et al. (2003). A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* 302, 449–453. doi: 10.1126/science.1087361
- Jonas, E. M. (2014). *Stochastic architectures for probabilistic computation* (Ph.D. thesis). Massachusetts Institute of Technology, Cambridge, MA, United States.
- Kaiser, J., Faria, R., Camsari, K. Y., and Datta, S. (2020). Probabilistic circuits for autonomous learning: a simulation study. *Front. Comput. Neurosci.* 14:14. doi: 10.3389/fncom.2020.00014
- Kaiser, J., Rustagi, A., Camsari, K., Sun, J., Datta, S., and Upadhyaya, P. (2019). Subnanosecond fluctuations in low-barrier nanomagnets. *Phys. Rev. Appl.* 12:054056. doi: 10.1103/PhysRevApplied.12.054056
- Koller, D., and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press. Available online at: <https://mitpress.mit.edu/books/probabilistic-graphical-models>
- Li, C., Belkin, D., Li, Y., Yan, P., Hu, M., Ge, N., et al. (2018). Efficient and self-adaptive *in-situ* learning in multilayer memristor neural networks. *Nat. Commun.* 9, 1–8. doi: 10.1038/s41467-018-04484-2
- Liu, L., Pai, C.-F., Li, Y., Tseng, H., Ralph, D., and Buhrman, R. (2012). Spin-torque switching with the giant spin hall effect of tantalum. *Science* 336, 555–558. doi: 10.1126/science.1218197
- Locatelli, N., Mizrahi, A., Accioly, A., Matsumoto, R., Fukushima, A., Kubota, H., et al. (2014). Noise-enhanced synchronization of stochastic magnetic oscillators. *Phys. Rev. Appl.* 2:034009. doi: 10.1103/PhysRevApplied.2.034009

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fncom.2021.584797/full#supplementary-material>

- Lopez-Diaz, L., Torres, L., and Moro, E. (2002). Transition from ferromagnetism to superparamagnetism in the nanosecond time scale. *Phys. Rev. B* 65:224406. doi: 10.1103/PhysRevB.65.224406
- Mahmoodi, M., Prezioso, M., and Strukov, D. (2019). Versatile stochastic dot product circuits based on nonvolatile memories for high performance neurocomputing and neurooptimization. *Nat. Commun.* 10, 1–10. doi: 10.1038/s41467-019-13103-7
- Mansueto, M., Chavent, A., Auffret, S., Joumard, I., Nath, J., Miron, I. M., et al. (2019). Realizing an isotropically coercive magnetic layer for memristive applications by analogy to dry friction. *Phys. Rev. Appl.* 12:044029. doi: 10.1103/PhysRevApplied.12.044029
- Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642
- Mizrahi, A., Hirtzlin, T., Fukushima, A., Kubota, H., Yuasa, S., Grollier, J., et al. (2018). Neural-like computing with populations of superparamagnetic basis functions. *Nat. Commun.* 9, 1–11. doi: 10.1038/s41467-018-03963-w
- Neal, R. M. (1992). Connectionist learning of belief networks. *Artif. Intell.* 56, 71–113. doi: 10.1016/0004-3702(92)90065-6
- Nikovski, D. (2000). Constructing bayesian networks for medical diagnosis from incomplete and partially correct statistics. *IEEE Trans. Knowl. Data Eng.* 4, 509–516. doi: 10.1109/69.868904
- Ostwal, V., and Appenzeller, J. (2019). Spin-orbit torque-controlled magnetic tunnel junction with low thermal stability for tunable random number generation. *IEEE Magn. Lett.* 10, 1–5. doi: 10.1109/LMAG.2019.2912971
- Ostwal, V., Debashis, P., Faria, R., Chen, Z., and Appenzeller, J. (2018). Spin-torque devices with hard axis initialization as stochastic binary neurons. *Sci. Rep.* 8, 1–8. doi: 10.1038/s41598-018-34996-2
- Ostwal, V., Zand, R., DeMara, R., and Appenzeller, J. (2019). A novel compound synapse using probabilistic spin-orbit-torque switching for mtj-based deep neural networks. *IEEE J. Explorat. Solid State Comput. Dev. Circ.* 5, 182–187. doi: 10.1109/JXDC.2019.2956468
- Park, D.-C. (2016). Image classification using naive bayes classifier. *Int. J. Comp. Sci. Electron. Eng.* 4, 135–139. Available online at: <http://www.isaet.org/images/extramages/P1216004.pdf>
- Parks, B., Bapna, M., Igbokwe, J., Almasi, H., Wang, W., and Majetich, S. A. (2018). Superparamagnetic perpendicular magnetic tunnel junctions for true random number generators. *AIP Adv.* 8:055903. doi: 10.1063/1.5006422
- Pearl, J. (2014). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier. Available online at: <https://www.elsevier.com/books/probabilistic-reasoning-in-intelligent-systems/pearl/978-0-08-051489-5>
- Pervaiz, A. Z., Ghantasala, L. A., Camsari, K. Y., and Datta, S. (2017). Hardware emulation of stochastic p-bits for invertible logic. *Sci. Rep.* 7:10994. doi: 10.1038/s41598-017-11011-8
- Pervaiz, A. Z., Sutton, B. M., Ghantasala, L. A., and Camsari, K. Y. (2018). Weighted p-bits for fpga implementation of probabilistic circuits. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 1920–1926. doi: 10.1109/TNNLS.2018.2874565
- Premevida, C., Faria, D. R., and Nunes, U. (2017). Dynamic bayesian network for semantic place classification in mobile robotics. *Auton. Robots* 41, 1161–1172. doi: 10.1007/s10514-016-9600-2
- Pufall, M. R., Rippard, W. H., Kaka, S., Russek, S. E., Silva, T. J., Katine, J., et al. (2004). Large-angle, gigahertz-rate random telegraph switching induced by spin-momentum transfer. *Phys. Rev. B* 69:214409. doi: 10.1103/PhysRevB.69.214409
- Querlioz, D., Bichler, O., Vincent, A. F., and Gamrat, C. (2015). Bioinspired programming of memory devices for implementing an inference engine. *Proc. IEEE* 103, 1398–1416. doi: 10.1109/JPROC.2015.2437616
- Rish, I., Brodie, M., Ma, S., Odintsova, N., Beygelzimer, A., Grabarnik, G., et al. (2005). Adaptive diagnosis in distributed systems. *IEEE Trans. Neural Netw.* 16, 1088–1109. doi: 10.1109/TNN.2005.853423
- Roberts, G. O., and Sahu, S. K. (1997). Updating schemes, correlation structure, blocking and parameterization for the gibbs sampler. *J. R. Stat. Soc. Ser. B* 59, 291–317. doi: 10.1111/1467-9868.00070
- Russell, S. J., and Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. Pearson Education Limited. Available online at: <https://www.pearson.com/us/higher-education/product/Norvig-Artificial-Intelligence-A-Modern-Approach-Subscription-3rd-Edition/9780133001983.html>
- Shim, Y., Chen, S., Sengupta, A., and Roy, K. (2017). Stochastic spin-orbit torque devices as elements for bayesian inference. *Sci. Rep.* 7:14101. doi: 10.1038/s41598-017-14240-z
- Strukov, D., Indiveri, G., Grollier, J., and Fusi, S. (2019). Building brain-inspired computing. *Nat. Commun.* 10. doi: 10.1038/s41467-019-12521-x
- Sun, S., Zhang, C., and Yu, G. (2006). A bayesian network approach to traffic flow forecasting. *IEEE Trans. Intell. Transport. Syst.* 7, 124–132. doi: 10.1109/TITS.2006.869623
- Sutton, B., Camsari, K. Y., Behin-Aein, B., and Datta, S. (2017). Intrinsic optimization using stochastic nanomagnets. *Sci. Rep.* 7:44370. doi: 10.1038/srep44370
- Sutton, B., Faria, R., Ghantasala, L. A., Jaiswal, R., Camsari, K. Y., and Datta, S. (2020). Autonomous probabilistic coprocessing with petaflops per second. *IEEE Access*. 157238–157252. doi: 10.1109/ACCESS.2020.3018682
- Thakur, C. S., Afshar, S., Wang, R. M., Hamilton, T. J., Tapson, J., and Van Schaik, A. (2016). Bayesian estimation and inference using stochastic electronics. *Front. Neurosci.* 10:104. doi: 10.3389/fnins.2016.00104
- Ticknor, J. L. (2013). A bayesian regularized artificial neural network for stock market forecasting. *Expert Syst. Appl.* 40, 5501–5506. doi: 10.1016/j.eswa.2013.04.013
- Torunbalci, M. M., Upadhyaya, P., Bhawe, S. A., and Camsari, K. Y. (2018). Modular compact modeling of MTJ devices. *IEEE Trans. Electron Dev.* 65, 4628–4634. doi: 10.1109/TED.2018.2863538
- Tylman, W., Waszyrowski, T., Napieralski, A., Kamiński, M., Trafidlo, T., Kulesza, Z., et al. (2016). Real-time prediction of acute cardiovascular events using hardware-implemented bayesian networks. *Comput. Biol. Med.* 69, 245–253. doi: 10.1016/j.compbiomed.2015.08.015
- Vodenicarevic, D., Locatelli, N., Mizrahi, A., Friedman, J. S., Vincent, A. F., Romera, M., et al. (2017). Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing. *Phys. Rev. Appl.* 8:054045. doi: 10.1103/PhysRevApplied.8.054045
- Vodenicarevic, D., Locatelli, N., Mizrahi, A., Hirtzlin, T., Friedman, J. S., Grollier, J., et al. (2018). “Circuit-level evaluation of the generation of truly random bits with superparamagnetic tunnel junctions,” in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)* (Florence: IEEE), 1–4. doi: 10.1109/ISCAS.2018.8351771
- Weijia, Z., Ling, G. W., and Seng, Y. K. (2007). “PCMOs-based hardware implementation of bayesian network,” in *2007 IEEE Conference on Electron Devices and Solid-State Circuits* (Tainan: IEEE), 337–340. doi: 10.1109/EDSSC.2007.4450131
- Zand, R., Camsari, K. Y., Pyle, S. D., Ahmed, I., Kim, C. H., and DeMara, R. F. (2018). “Low-energy deep belief networks using intrinsic sigmoidal spintronic-based probabilistic neurons,” in *Proceedings of the 2018 on Great Lakes Symposium on VLSI* Chicago IL, 15–20. doi: 10.1145/3194554.3194558
- Zermani, S., Dezan, C., Chenini, H., Diguët, J.-P., and Euler, R. (2015). “FPGA implementation of bayesian network inference for an embedded diagnosis,” in *2015 IEEE Conference on Prognostics and Health Management (PHM)* (Austin, TX: IEEE), 1–10. doi: 10.1109/ICPHM.2015.7245057
- Zink, B. R., Lv, Y., and Wang, J.-P. (2018). Telegraphic switching signals by magnet tunnel junctions for neural spiking signals with high information capacity. *J. Appl. Phys.* 124:152121. doi: 10.1063/1.5042444
- Zou, M., and Conzen, S. D. (2004). A new dynamic bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics* 21, 71–79. doi: 10.1093/bioinformatics/bth463

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Faria, Kaiser, Camsari and Datta. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.