Voltage-driven Building Block for Hardware Belief Networks

Orchi Hassan,1 Kerem Y. Camsari,1 and Supriyo Datta1

1School of Electrical and Computer Engineering, Purdue University, IN, 47907

Probabilistic spin logic (PSL) based on networks of binary stochastic neurons (or p-bits) has been shown to provide a viable framework for Ising computing, Bayesian inference, invertible Boolean logic and image recognition. This paper presents a hardware building block for the PSL architecture, consisting of an embedded MTJ and a capacitive voltage adder of the type used in neuMOS. We use SPICE simulations to show how identical copies of these building blocks (or weighted p-bits) can be interconnected with wires to design and solve a small instance of the NP-complete Subset Sum Problem fully in hardware.

Keywords - Probabilistic computing, Embedded MTJ, p-bits, p-circuits, Invertible Boolean logic, Subset Sum Problem

I. INTRODUCTION

Probabilistic spin logic (PSL) has been shown to provide a viable framework for Ising computing [1–3], Bayesian inference, invertible Boolean logic [5], and image recognition [6]. The PSL model is defined by two equations [5] loosely analogous to a neuron and a synapse. The former is what we call the p-bit whose output $m_i$ is related to its dimensionless input $I_i$ by the relation

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

(1a)

where $\text{rand}(-1, 1)$ 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

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

(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 and $I_0$, a dimensionless constant. The objective of this paper is to present a voltage-driven hardware building block using present day device technologies such as embedded MRAM [7] and Floating-Gate MOS transistors, such that identical copies of the same block can be interconnected with wires to implement Eqs. 1.

The paper is organized as follows: We first show a complete hardware mapping for the weighted p-bit by augmenting a recently introduced Magnetic Random Access Memory (MRAM) type stochastic unit with a floating gate MOS-based capacitive network. We then show how the results of a fully interconnected $W_p$-bit circuit closely approximate the the ideal equations using an example of an “invertible” Full Adder that can perform 1-bit addition and subtraction. Finally, we show how such invertible Full Adders can be interconnected to solve a simple instance of the NP-complete Subset Sum Problem.

Each example in this paper has been obtained using the full SPICE model which simply uses transistors, capacitors and resistors without any additional complex circuitry or processing.

II. BUILDING BLOCK

Our building block has two components corresponding to the two Eqs. 1a,b. The one on the right is an embedded low-barrier unstable MTJ [5] which provides a stochastic output whose average value is controlled by the input voltage:

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

(2a)

where $\pm V_{DD}/2$ are the supply voltages, and $V_0$ is a parameter ($\sim 22 \text{ mV}$) describing the width of the sigmoidal response.

The value of $V_0$ depends on the details of the 1T/1 MTJ in the embedded MRAM structure [9]. $G_0$ of the MTJ is matched to center the transfer characteristics of the whole $W_p$-bit as shown in Fig.1d. To do that a DC analysis is performed, where an input voltage of “0” ($V_{GS} = 0.4 \text{ V}$ to turn ON the transistor) is applied at $V_i$ and $G_0$ is swept to observe $V_{\text{OUT+}}$ and $V_{\text{OUT-}}$. The value of $G_0$ for which $V_{\text{OUT+}} = V_{\text{OUT-}} = 0$ (when $V_{DS} = 0.4 \text{ V}$) is the value picked for the MTJ conductance. In this case it is $1/G_0 \approx 62 \text{ k}\Omega$, and it seems reasonable considering the RA-products of modern MTJs. The value of $V_0$ depends on the choice of $G_0$ among other factors including transistor characteristics.

The second of Eqs. 1 is implemented by the component on the left of Fig. 1, which is a capacitive voltage adder just like those used in neuMOS devices [8, 10]. We can write

$$V_i = \frac{V_{\text{bias}, i} C_{b, i} + \sum_j V_{\text{out}, j} C_{ij}}{C_y + C_{z, i} + C_{b, i} + \sum_j C_{ij}}$$

(2b)

Note that the capacitive voltage divider typically attenuates the voltage $V_i$ at its output, and the inverter is intended to scale it up to $V_{\text{in}, i}$, the two being related.
where \( v_0 \) is a parameter characteristic of the inverter. Eqs. 2a, b can be mapped onto the PSL Eqs. 1a, b by defining

\[
m_i = \frac{V_{out,i}}{V_{DD}/2}, \quad I_i = \frac{V_{in,i}}{V_0}
\]

\[
C_{h,i} = b_i C_0 \quad C_{z,i} = z_i C_0
\]

\[
h_i = b_i \frac{V_{bias,i}}{V_{DD}/2} \quad J_{ij} = \frac{C_{ij}}{C_0}
\]

\[
I_0 = \frac{(V_{DD}/2v_0)(V_{DD}/2V_0)}{(C_g/C_0) + z_i + b_i + \sum_j J_{ij}}
\]

The significance of \( C_0 \) is that we assume the input is composed of many identical capacitors \( C_0 \), and that the weights \( J_{ij} \) have been designed to have integer values such that \( C_{ij} \) can be implemented by connecting \( J_{ij} \) elementary capacitors in parallel. The other coefficients \( z_i, b_i \) are also integers and we adjust the number \( z_i \) of grounded capacitors to make \( z_i + b_i + \sum_j J_{ij} = K \) a constant, so that \( I_0 \) is independent of index \( i \):

\[
I_0 = \frac{(V_{DD}/2v_0)(V_{DD}/2V_0)}{(C_g/C_0) + K}
\]

Note that \( K \) is usually a fairly large number equal to the sum of all the weights, and to implement an \( I_0 \approx 1 \) it is important to keep the factor \((V_{DD}/2v_0)(V_{DD}/2V_0)\) to be much greater than 1. This is the reason for using an inverter between the capacitive voltage adder and the \( p \)-bit.

Fig. 1b shows the icon we use to represent our building block which we call a weighted \( p \)-bit. The input consists of four types of inputs designated S, D and Q having capacitances \( C_0, 2 C_0, 4 C_0 \). Combinations of these are used to implement different weights \( J_{ij} \) and different bias \( h_i \). Each block has two outputs \( V_{OUT}^p, V_{OUT}^- \). The choice of output depends on the sign of the corresponding \( J_{ij} \). Similarly different signs of \( h_i \) are implemented by choosing \( V_{bias,i} \) to be \( +V_{DD}/2 \) or \( -V_{DD}/2 \).

### III. INVERTIBLE FULL ADDER

In PSL, any given truth table can be implemented using Eq. 1 by choosing an appropriate \([J]\) and \([h]\) matrices [5]. Here we show how that \([J]\) and \([h]\) are mapped onto physical hardware using our proposed building block using only transistors, resistors and capacitances.

A Full Adder can be implemented in PSL using the \([J]\) matrix shown in Fig. 2. \([J]\) defines the interconnection between the 14 \( W_p \)-bits to design the Full Adder in hardware. Each row of the \([J]\) matrix are realized in terms of capacitive coupling to the gate of the associated terminal.
FIG. 2. Invertible Full Adder with $W_p$-bit: (a) $J$ matrix for implementing a Full Adder based on [5]. (b) Explicitly shows the hardware connections made to one of the inputs (A) (row 12) from the other p-bits where $1C_0$, $2C_0$, and $4C_0$ represent capacitors in units of $C_0 = 100\, aF$. (c) Shows the subcircuit representation of the Full Adder with its input/output terminals. $C_i, B, A$ input and $S, C_0$ output read terminals and separate corresponding clamping terminals $h_{C_i}, h_B, h_A, h_S, h_{C_0}$. We used $8C_0$ for the clamping terminals to ensure input / outputs follow what is dictated from the external signals.

To ensure a uniform $I_0$ is applied to each p-bit (Eq. 4), the same weighting factor K needs to be used for all $W_p$-bits. To choose a given $I_0$, we first find $\max(b_i + \sum J_{ij})$ for any given $J[i]$, and then ground $z_i = M - b_i + \sum J_{ij}$ ($z_i \geq 0, z_i \in N$) unit capacitances for all terminals where $M$ is a number that can be used to control $I_0$, a larger $M$ causing a smaller $I_0$. Fig. 2b shows explicit connections made to one of the inputs "A" and Fig. 2c shows the subcircuit of the Full Adder with $C_i, B, A$ as inputs, $S, C_0$ as the outputs, and $h_{C_i}, h_B, h_A, h_S, h_{C_0}$ as the clamping pins.

Fig. 4 shows the operation of a Full Adder in the usual forward mode with $C_i, B, A$ clamped to values $(0,1,1)$ which forces the $S$ and $C_0$ to $(0,1)$ according to the truth table. In the invertible mode $S$ and $C_0$ are clamped to $(0,1)$ and the circuit stochastically searches consistent combinations of $C_i, B, A$ to satisfy the truth table: $\{C_i, B, A\} = \{(0,1,1), (1,0,1), (1,1,0)\}$. Fig. 4 shows steady state ($t = 1\, \mu s$) histogram plots after thresholding of the Full Adder operation in direct and inverted mode side by side with results from the PSL model based on Eq. 1.

The good agreement between the ideal PSL model and the coupled SPICE simulation that solves PTM-based transistors models with stochastic LLGs validates the hardware mapping of the ideal p-bit equations with the weighted p-bits.
IV. 3SUM PROBLEM

3SUM is a decision problem in complexity theory that asks whether three elements of a given set can sum up to zero. A variant of the problem is when the set of three numbers have to add up to a given constant number. This problem has a polynomial time solution and is not in NP. In this section, we show how the invertibility feature of the Full Adders can be exploited to design a hardware 3SUM solver. In the next section, we show how the 3SUM hardware can be modified to design a general solver for the NP-complete Subset Sum Problem.

The invertibility property of the Full Adders ensure that given the sum, it can provide the possible input combinations for that sum as shown in Fig. 4a. So an n-bit 3 number adder circuit implemented in PSL can essentially provide solution sets for the 3SUM problem when the sum is clamped to a given value.

Fig. 4a shows the circuit constructed out of Full Adders to solve a 4-bit 3SUM problem. Each of the Full Adders in the circuit are the 14 p-bit invertible adders that were shown in Fig. 3. The first row of adders adds the two 4-bit numbers A and B, and feeds its output X, to the next row of adders which adds X and C to give the sum \( S = C + X = C + B + A \). Because p-circuits are invertible, if we clamp the sum S, the circuit naturally explores through all possible sets and multisets of the set of all integers from 0 to \( 2^4 - 1 \) that add up to S. The given set for the problem could be implemented through clamping certain bits of A, B and C or externally circuitry could be used to detect only the results that belong to the given set. Fig. 4b shows the how A, B, C is fluctuating between values that satisfy the clamped sum 15.

V. SUBSET-SUM PROBLEM (SSP)

In this section we show how the hardware circuit that was designed for 3SUM problem could be modified to solve a small instance of subset-sum problem (SSP) [11] which is believed to be a fundamentally difficult problem
in computer science (NP-complete). In the SSP, a set \( G \) with a finite number of positive numbers is defined. And then the decision problem is to ask whether there is a subset \( S' \) such that \( S' \subseteq G \) whose elements sum to a specified target. For example, Fig. 5 shows a circuit that is programmed to choose a set, \( S=\{1,2,4\} \) and a target that is defined by 4-bits. In the 3SUM circuit the input bits (A, B, C) were left “floating”, here, the inputs are constrained to a given number (1,2, 4) by clamping the remaining bits of an input. For example, the inputs \( A_1 \) and \( A_0 \) are clamped to zero to make A either 4 or 0. Under these conditions, clamping the output to a specified target makes the circuit search for a consistent input combination to find a subset that satisfies the clamped target. Fig. 5c shows three example targets where the inputs get correlated to satisfy the clamped sum. The invertibility feature that is utilized to solve the SSP in this hardware is similar to those discussed in the context of memcomputing [12], however the physical mechanisms are completely different.

One striking difference in the design of the SSP we considered, compared to the 3SUM hardware is the direction of information. In 3SUM the connections were from the first layer of Full Adders to the second, as in normal addition (Fig. 4a). In the SSP, we observed that reversing these connections from the second layer of adder to the first layer drastically improves the accuracy of the solution (Fig. 5a). A similar observation regarding the directional flow of information for another inverse problem using p-circuits (integer factorization) was made in [5]. Here we have limited the discussion to a small instance of the SSP which would in general require more layers of Full Adders in both vertical and horizontal directions to account for more numbers of elements in \( S \) and their size. This example illustrates how invertibility can be combined with standard digital VLSI design to construct any general “cost function” for hard problems of computer science in an asynchronously running hardware platform, without any external clocking.

VI. CONCLUSION

In this paper we have proposed a compact building-block for Probabilistic Spin Logic (PSL) combining a recently proposed Embedded MRAM-based p-bit, with a capacitive network that can be implemented using Floating Gate MOS (FGMOS) transistors similar to the neuMOS concept. We have shown by extensive SPICE simulations that the results of the hardware model for the weighted p-bit agree well with the behavioral equations of PSL. Even though an FGMOS-based capacitive network to do the voltage addition seems like a natural option for the p-bit, we note that the device equations for a conductance network \([G]\) would have been essentially the same. Moreover, our discussion was only about static weights, but an FPGA-like reconfigurable weighting scheme can also be employed either by using transistor-based gates or by additional multiplexing circuitry to perform online learning or to redesign p-circuits. Finally, using the basic building block we have shown how a small instance of a hardware solver of the NP-complete Subset Sum Problem can be designed using the unique invertibility feature of p-circuits.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) through E2CDA.

[1] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, Scientific Reports 7 (2017).
[2] B. Behin-Aein, V. Diep, and S. Datta, Scientific reports 6 (2016).
[3] Y. Shim, A. Jaiswal, and K. Roy, Journal of Applied Physics 121, 193902 (2017), http://dx.doi.org/10.1063/1.4983636.
[4] R. Faria, K. Y. Camsari, and S. Datta, arXiv preprint arXiv:1801.00497 (2018).
[5] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, Phys. Rev. X 7, 031014 (2017).
[6] R. Zand, K. Y. Camsari, I. Ahmed, S. D. Pyle, C. H. Kim, S. Datta, and R. F. DeMara, arXiv preprint arXiv:1710.00249 (2017).
[7] C. Lin, S. Kang, Y. Wang, K. Lee, X. Zhu, W. Chen, X. Li, W. Hsu, Y. Kao, M. Liu, et al., in Electron Devices Meeting (IEDM), 2009 IEEE International (IEEE, 2009) pp. 1–4.
[8] T. Shihata and T. Ohmi, IEEE Transactions on Electron devices 39, 1444 (1992).
[9] K. Y. Camsari, S. Salahuddin, and S. Datta, IEEE Electron Device Letters 38, 1767 (2017).
[10] N. Nakamura, K. Shimada, T. Matsuda, and M. Kimura, in Future of Electron Devices, Kansai (IMFEDK), 2015 IEEE International Meeting for (IEEE, 2015) pp. 90–91.
[11] T. H. Cormen, Introduction to algorithms (MIT press, 2009).
[12] F. L. Traversa and M. Di Ventra, Chaos: An Interdisciplinary Journal of Nonlinear Science 27, 023107 (2017).