Title

Voltage-Driven Building Block for Hardware Belief Networks

Journal

IEEE Design and Test, 36(3)

ISSN

2168-2356

Authors

Hassan, Orchi

Camsari, Kerem Y

Datta, Supriyo

Publication Date

2019-06-01

DOI

10.1109/mdat.2019.2897964

Peer reviewed

Voltage-driven Building Block for Hardware Belief Networks

Orchi Hassan,[1] Kerem Y. Camsari,[1] and Supriyo Datta[1]

[1] *School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA*

Probabilistic spin logic (PSL) based on networks of binary stochastic neurons (or p-bits) has been shown to provide a viable framework for many functionalities including 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 [2], invertible Boolean logic [4], and image recognition [5]. The PSL model is defined by two equations [4] 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

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

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

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

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 is a dimensionless constant. These two equations constitute the behavioral model of PSL. The objective of this paper is to present a voltage-driven hardware building block using present day device technologies such as embedded MRAM [6] 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 Magnetoresistive Random Access Memory (MRAM) type stochastic unit [7] with a floating gate MOS-based capacitive network [8]. 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 full SPICE models 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. Eq. 1a is implemented by the p-bit in Fig.1a which consists of an embedded low-barrier unstable MTJ coupled to two CMOS inverters which provides a stochastic output whose average value is controlled by the input voltage:

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

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

The value of V_0 depends on the details of the 1T/1MTJ in the embedded MRAM structure [7] and the transistor characteristics. The conductance, G_0 of the MTJ is chosen to match the MTJ switching characteristics to the transistors in the ^{W}p-bit so that the overall transfer characteristics is centered at zero as shown in Fig. 1e. To do that, an input voltage of $\overline{\mathrm{V_i}}$ =0V is applied at the input of T1 and T2 transistors turning both of them ON ($|\mathrm{V_{GS}}| = 0.4$V) and G_0 is swept to observe the outputs. The G_0 value for which $\mathrm{V^+_{OUT}}$=$\mathrm{V^-_{OUT}}$ = 0V is the value chosen to be the MTJ conductance. For minimum sized 14nm HP-FinFET transistors models with $\mathrm{V_{DD}} = 0.8$V, $1/G_0 \approx 62$ kΩ and it seems reasonable considering the RA-products of modern MTJs [9].

Eqs. 1b is implemented by the weighted synapse portion of Fig. 1a , which is a capacitive voltage adder just like those used in neuMOS devices [8, 10]. We can write

$$\overline{V}_i = \frac{V_{bias,i}C_{b,i} + \sum_j V_{out,j}C_{ij}}{C_g + C_{z,i} + C_{b,i} + \sum_j C_{ij}} \qquad (2b)$$

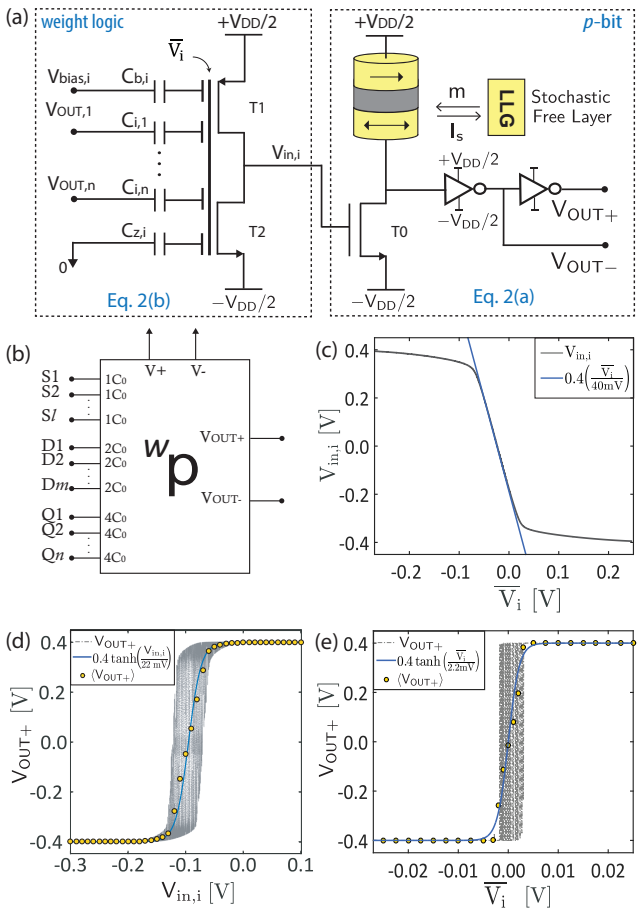

FIG. 1. (a) **Voltage-driven building block** has two components corresponding to Eqs. 2a,b. The first is the p-bit implemented through an embedded low-barrier unstable MTJ [4] with two inverters added to give positive and negative outputs. The low-barrier MTJ can be designed using low barrier or circular nanomagnets. The second is the capacitive voltage adder with an inverter structure on the left similar to the floating gate MOS transistors used in neuMOS devices [8]. We call this combination of p-bit and its weight logic a weighted p-bit ($^W p$-bit). (b)Shows the the block diagram of $^W p$-bit. (c) Shows how an inverter helps amplify the input ($\overline{V}_i$) of the capacitive network to give $V_{in,i}$ at the gate of the p-bit's NMOS transistor T0. (d) Shows the relation of the input gate voltage of the NMOS ($V_{in,i}$) to output (V^+_{OUT}). (e) Shows the transfer characteristics of the $^W p$-bit as a whole. The inputs in each case is swept from -0.4V to $+0.4$V in 1 μs. The yellow dots are time averaged values at each point over 300 ns and the solid blue lines are numerical fits. The magnet used in the simulations is defined by parameters in[7]: $M_s = 1100 emu/cc, D = 22nm, t = 2nm, \alpha = 0.01$. All transistors were modeled using minimum size (nfin=1) 14 nm HP-FinFET Predictive Technology Models with $V_{DD} = 0.8$V and T = 300K.

Note that the capacitive voltage divider typically attenuates the voltage $\overline{V}_i$ at its output, and the inverter scales it up to $V_{in,i}$ as shown in Fig. 1c, the two being related approximately by

$$V_{in,i} \approx \frac{V_{DD}}{2}\tanh\frac{\overline{V}_i}{\nu_0}$$
$$\approx \frac{V_{DD}}{2\nu_0}\,\overline{V}_i \quad \text{if} \quad \overline{V}_i \ll \nu_0 \tag{2c}$$

where ν_0 is a parameter characteristic of the inverter. Eqs. 2a,b can be mapped onto the PSL Eqs. 1a,b by defining

$$m_i = \frac{V_{out,i}}{V_{DD}/2}, \quad I_i = \frac{V_{in,i}}{V_0} \tag{3a}$$

$$C_{b,i} = b_i C_0 \quad C_{z,i} = z_i C_0 \tag{3b}$$

$$h_i = b_i \frac{V_{bias,i}}{V_{DD}/2}, \quad J_{ij} = \frac{C_{ij}}{C_0} \tag{3c}$$

$$I_0 = \frac{(V_{DD}/2\nu_0)(V_{DD}/2V_0)}{(C_g/C_0) + z_i + b_i + \sum_j J_{ij}} \tag{3d}$$

C_g is the intrinsic gate capacitance of the neuMOS inverter. 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. We adjust the number b_i of bias capacitors to facilitate external biasing and 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:

$$I_0 = \frac{(V_{DD}/2\nu_0)(V_{DD}/2V_0)}{(C_g/C_0) + K} \tag{4}$$

Note that K is usually a fairly large number equal to the sum of all the weights, and to implement an $I_0 \sim 1$ it is important to keep the factor $(V_{DD}/2\nu_0)(V_{DD}/2V_0)$ to be much greater than 1. This is the reason for using an inverter between the capacitive voltage adder and the p-bit. Our model neglects any leakage resistances associated with the capacitive weights. Modern transistors with thin oxides can have gate leakage currents ~1nA, with RC ~ μs-ms. This should not affect the weighting, since the examples presented here operate at sub-ns time scales. For slower neurons, it may be advisable to use thicker oxides for the capacitive weights to ensure lower leakage.

Fig. 1b shows the icon we use to represent our building block which we call a weighted p-bit. The input consists of three types of inputs designated S, D and Q having capacitances C_0, 2 C_0 and 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}, 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 $-V_{DD}/2$.

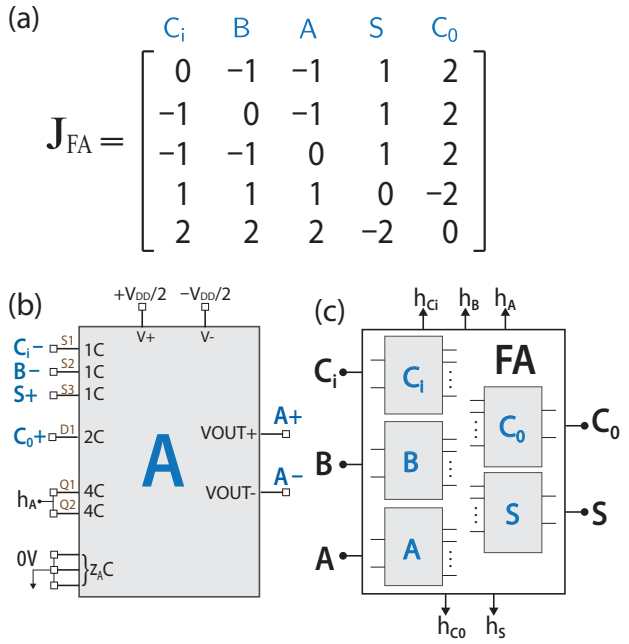

FIG. 2. **Invertible Full Adder with $^W p$-bit**: (a)$[J]$ matrix for implementing a Full Adder. (b) Explicitly shows the hardware connections made to one of the input p-bits (A) from the other p-bits where $1C$, $2C$, and $4C$ represent capacitors in units of $C = C_0 = 100aF$. (c) Shows the subcircuit representation of the Full Adder with its input/output terminals; C_i, B, A input and S, C_o output read terminals and separate corresponding clamping terminals $h_{C_i}, h_B, h_A, h_S, h_{C_0}$. We used $8C$ for the clamping terminals to ensure input / outputs follow what is dictated by the external signals.

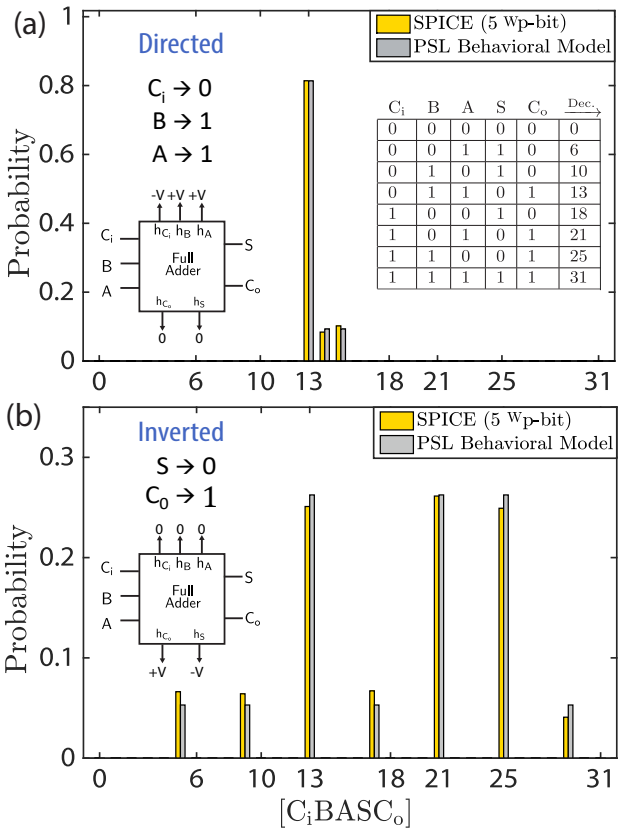

FIG. 3. **Full SPICE implementation of an Invertible Full Adder(5 $^W p$-bit)**: The 5 $^W p$-bit invertible Full Adder circuit is simulated in (a) Directed and (b) Inverted modes. The clamping values are indicated. All biasing terminals that are not clamped to 1 or 0 are grounded. The histogram of $[C_iBASC_0]$ is obtained after thresholding voltages $((V < 0) \equiv -1, (V > 0) \equiv +1)$. The SPICE model is run for 1μs and compared with the PSL equations where each p-bit is updated in random but sequential order [4]. In this example $I_0 \simeq 1$ is chosen to emphasize how the models are in good agreement even in the magnitudes of the minor peaks of the histogram.

III. INVERTIBLE FULL ADDER

In PSL, any given truth table can be implemented using Eq. 1 by choosing an appropriate $[J]$ and $[h]$ matrices [4]. Here we show how those $[J]$ 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. In this paper, we improve the 14 p-bit implementation of the invertible Full Adder (FA) in Ref.[4] and implement the same functionality using 5 p-bits. This is achieved by first noting that the first half of the FA truth table is complementary to the second half for the FA (Fig. 3a inset). The first 4 lines in the truth table is turned into an orthonormal set by a Gram-Schmidt process and a [J] matrix is obtained using Eq.12 in Ref.[4] which is finally rounded to integer values, with diagonal entries replaced by zeros. This $[J]$ defines the interconnection between the 5 $^W p$-bits of 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.

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 apply a given I_0, we first find $\max(b_i + \sum J_{ij})$ for any given $[J]$, 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 $h_{Ci}, h_B, h_A, h_S, h_{Co}$ 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 μs) histogram plots of the Full Adder operation in direct and inverted mode side by side with results from the PSL behavioral model.

The good agreement between the ideal PSL behavioral model and the coupled SPICE simulation that solves PTM-based transistors models with stochastic LLG 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 utilized to design a hardware 3SUM solver, and 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.

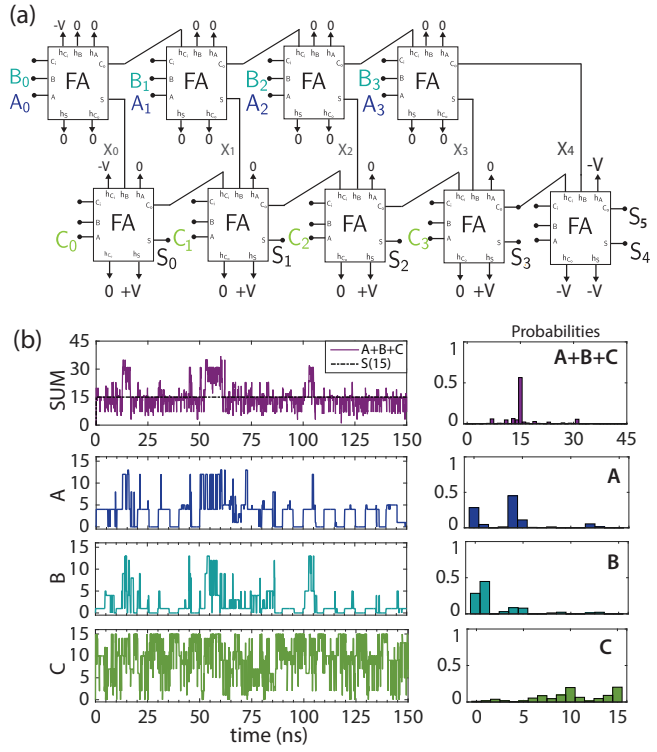

FIG. 4. **SPICE simulation of a 4bit 3-SUM Problem (9 × 5 = 45 ^{W}p-bit network)**: (a) The circuit is constructed by interconnecting two rows of invertible Full-Adders (FA) to construct a 3 number, 4-bit adder. The sum S is clamped to the desired value and A, B, C resolves themselves to create all the possible 3 number subsets out of all positive numbers 0 to 2^4-1 that satisfy A + B + C = S. (b) Shows the results when S is clamped to 15. A, B and C get correlated to satisfy the sum with different combinations. In this example, the inputs A, B, C are unconstrained and can take on any value between $0-15$.

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 5 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). The SSP asks, given a set G with a finite number of positive numbers, if 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, G={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 [4]. Here we have limited the discussion to a small instance

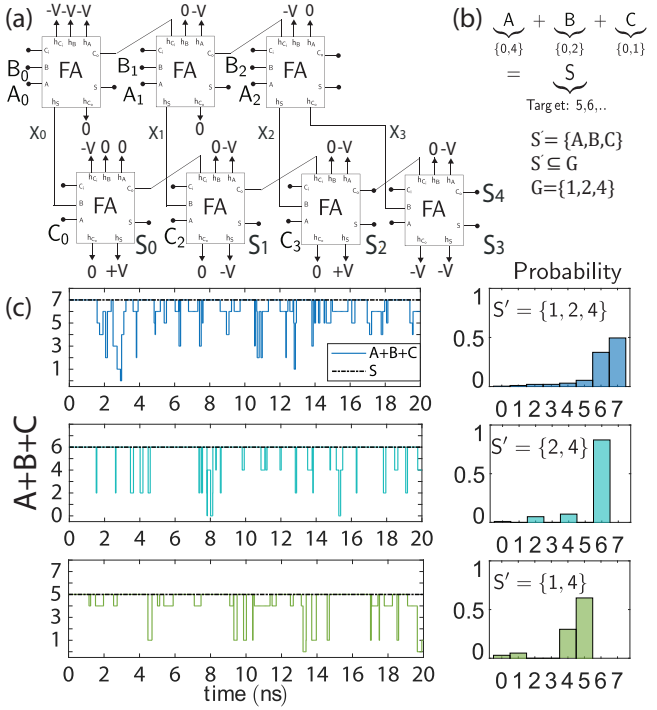

FIG. 5. **SPICE simulation of a 3 input, 3-bit Subset Sum Problem ($7 \times 5 = 35$ ^{W}p-bit network)**: (a) A 3-input 3-bit binary adder that adds three numbers A,B,C. Unlike the 3SUM, in this case the inputs are constrained to a given value specified by the set G $=\{1, 2, 4\}$ in this example. A target S is selected and the output of the adders are clamped to the target value as shown in (b). (c) Shows three different instances of a target where the inputs find a consistent combination (the correct subset of G) to satisfy the target. Histograms show that the highest probable state is the correct subset. An important difference from the 3SUM circuit is that the information flow is *directed* from the target (second layer of adders) to the first layer of adders.

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 G and their size. The purpose of this example is to illustrate 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 an integrated 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. Having dedicated MTJ based hardware stochastic neurons could help minimize the footprint and consume lower power for applications as also indicated by ref.[5, 9]. Even though an FGMOS-based capacitive network for performing the voltage addition seems like a natural option, we note that the device equations for any capacitance $[\mathsf{C}_{ij}]$ or conductance network $[\mathsf{G}_{ij}]$ 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-circuit connectivity. Finally, using the basic building block we have shown how a small instance of the NP-complete Subset Sum Problem hardware solver can be designed using the unique invertibility feature of p-circuits.

ACKNOWLEDGMENT

This work was supported in part by the Center for Probabilistic Spin Logic for Low-Energy Boolean and Non-Boolean Computing (CAPSL), one of the Nanoelectronic Computing Research (nCORE) Centers as task 2759.005, a Semiconductor Research Corporation (SRC) program sponsored by the NSF through ECCS 1739635.

[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] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, Phys. Rev. X **7**, 031014 (2017).
[5] 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).
[6] C. Lin, S. Kang, Y. Wang, K. Lee, X. Zhu, W. Chen, X. Li, W. Hsu, Y. Kao, M. Liu, *et al.*, *Electron Devices Meeting (IEDM), 2009 IEEE International* (IEEE, 2009) pp. 1–4.
[7] K. Y. Camsari, S. Salahuddin, and S. Datta, IEEE Electron Device Letters **38**, 1767 (2017).
[8] T. Shibata and T. Ohmi, IEEE Transactions on Electron devices **39**, 1444 (1992).
[9] A. Mizrahi, T. Hirtzlin, A. Fukushima, K. Hitoshi, Y. Shinji, J. Grollier and D. Querlioz, *Neural-like computing with populations of superparamagnetic basis functions* Nature communications,**9**, 1533 (2018).

[10] N. Nakamura, K. Shimada, T. Matsuda, and M. Kimura, *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).