Voltage-Driven Building Block for Hardware Belief Networks

Orchi Hassan, Kerem Y. Camsari, and Supriyo Datta
Purdue University

Editor's note:
In this article, the authors present a hardware building block for probabilistic spin logic (PSL) consisting of a probabilistic bit (p-bit) made from an embedded low-barrier unstable magnetic tunnel junction (MTJ) and a capacitive voltage adder. It was demonstrated through simulation that these interconnected building blocks can be designed to solve a small instance of the NP-complete Subset Sum Problem.
—An Chen, Semiconductor Research Corporation

■ **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 by the relation

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

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

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

where h_i is the on-site bias, J_{ij} is the weight of the coupling from j_{th} 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 article is to present a voltage-driven hardware building block using present-day device technologies such as embedded magnetoresistive random-access memory (MRAM) [6] and floating-gate MOS (FGMOS) transistors, such that identical copies of the same block can be interconnected with wires to implement (1).

This article will first show a complete hardware mapping for the weighted p-bit by augmenting a recently introduced 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 ideal equations using an example of an "invertible" full adder (FA) that can perform 1-bit addition and subtraction. Finally, we show how such invertible FAs can be interconnected to solve a simple instance of the NP-complete subset sum problem (SSP). Each example in this article has been obtained using full SPICE models that simply uses transistors, capacitors, and resistors without any additional complex circuitry or processing.

Digital Object Identifier 10.1109/MDAT.2019.2897964
Date of publication: 06 February 2019; date of current version: 29 May 2019.

Building block

Our building block has two components corresponding to (1a) and (1b). Equation (1a) is implemented by the p-bit shown in Figure 1a, which consists of an embedded low-barrier unstable magnetic tunnel junction (MTJ) coupled to two CMOS inverters that provide a stochastic output whose average value is controlled by the input voltage

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

where $\pm V_{DD}/2$ is the supply voltage, 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 with the transistors in the ^{W}p-bit so that the overall transfer characteristics is centered at zero as shown in Figure 1e. To do that, an input voltage of $\bar{V}_i = 0\text{V}$ is applied at the input of T1 and T2 transistors, turning both of them on ($|V_{GS}| = 0.4\text{V}$), and G_0 is swept to observe the outputs. The value of G_0 for which $V^{+}_{OUT}=V^{-}_{OUT} = 0\text{V}$ is the value chosen to be the MTJ conductance. For minimum sized 14-nm HP-FinFET transistors models with $V_{DD} = 0.8\text{V}$, $1/G_0 \approx 62\ \text{k}\Omega$ and it seems reasonable considering the resistance area (RA) products of modern MTJs [9].

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

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

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

$$\begin{aligned} V_{in,i} &\approx \frac{V_{DD}}{2}\tanh\frac{\bar{V}_i}{v_0}, \\ &\approx \frac{V_{DD}}{2v_0}\bar{V}_i \quad \text{if} \quad \bar{V}_i \ll v_0, \end{aligned} \quad (2c)$$

where v_0 is a parameter characteristic of the inverter. Equations (2a) and (2b) can be mapped onto the PSL equations (1a) and (1b) by defining

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

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

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

$$I_0 = \frac{(V_{DD}/2\,v_0)(V_{DD}/2\,V_0)}{(C_g/C_0) + z_i + b_i\sum_j J_{ij}} \quad (3d)$$

where 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 and b_i are also integers. We adjust the number b_i of bias capacitors to facilitate the 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.

$$I_0 = \frac{(V_{DD}/2\,v_0)(V_{DD}/2\,V_0)}{(C_g/C_0) + K} \quad (4)$$

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 $(V_{DD}/2v_0)(V_{DD}/2V_0)$ to be much greater than 1. This is the reason for using an inverter between the capacitive voltage adder and the p-bit. Our model neglects any leakage resistances associated with the capacitive weights. Modern transistors with thin oxides can have gate leakage currents ~1 nA, with RC approximately from microseconds to milliseconds. This should not affect the weighting because the examples presented here operate at subnanoseconds timescales. For slower neurons, it may be advisable to use thicker oxides for the capacitive weights to ensure lower leakage.

Figure 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 as S, D, and Q, having capacitances C_0, 2 $C0$, and 4 C_0, respectively. Combinations of these are used to implement different weights J_{ij} and different bias h_i. Each block has two outputs V^{+}_{OUT} and 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 $V_{\text{bias},i}$ to be $+V_{DD}/2$ or $-V_{DD}/2$.

Invertible full adder

In the PSL, any given truth table can be implemented using (1) by choosing an appropriate [J] and [h] matrices [4]. Here, we show how both matrices are mapped onto the physical hardware using our proposed building block using only transistors, resistors, and capacitances.

An FA can be implemented in the PSL using the [J] matrix shown in Figure 2. In this article, we improve the 14 p-bit implementation of the invertible FA using [4] and implement the same functionality using 5 p-bits. This is achieved by first noting that

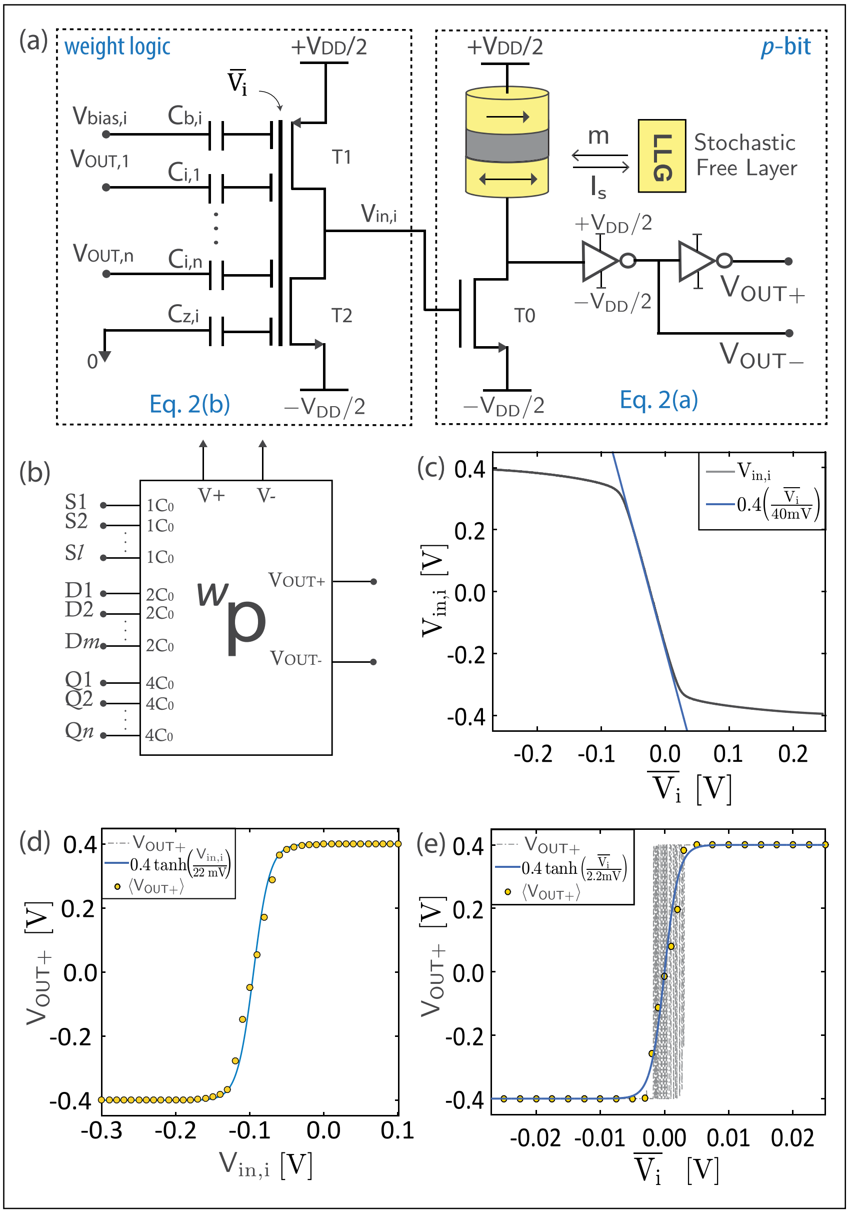

Figure 1. (a) Voltage-driven building block has two components corresponding to (2a) and (2b). The first is the *p*-bit implemented through an embedded low-barrier unstable MTJ with two inverters added to give positive and negative outputs. The low-barrier MTJ can be designed using low barrier or circular nanomagnets. The second is the capacitive voltage adder with an inverter structure on the left similar to the FGMOS 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) Block diagram of ^{w}p-bit. (c) Illustration of 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) Relation of the input gate voltage of the nMOS ($V_{in,i}$) to output (V^{+}_{OUT}). (e) Transfer characteristics of the ^{w}p-bit as a whole. The inputs in each case are swept from −0.4 V to +0.4 V in 1 ps. Yellow dots: time averaged values at each point over 300 ns. Solid blue lines: numerical fits. The magnet used in the simulations is defined by parameters in [7]: M_s = 1100 emu/cc, *D* = 22 nm, *t* = 2 nm, *a* = 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 = 300 K.

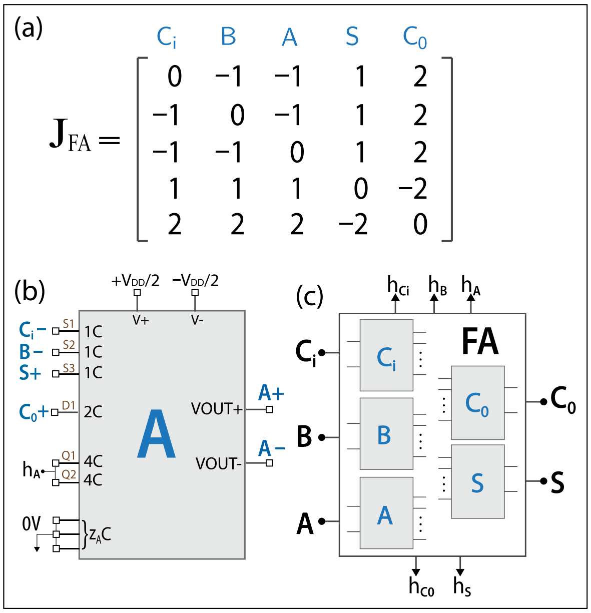

Figure 2. Invertible FA with ^{W}p-bit. (a) [*J*] matrix for implementing an FA. (b) Hardware connections made to one of the input *p*-bit (A) from the other *p*-bits where 1*C*, 2*C*, and 4*C* represent capacitors in units of *C* = C_0 = 100 *aF*. (c) Subcircuit representation of the FA with its input/output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

the first half of the FA truth table is complementary to the second half for the FA (Figure 3a inset). The first four lines in the truth table are turned into an orthonormal set by the Gram–Schmidt process and the [*J*] matrix is obtained using (12) [4] which is finally rounded off to integer values, with diagonal entries replaced by zeros. This [*J*] matrix defines the interconnection between the five ^{W}p-bits of the FA in the hardware. Each row of the [*J*] matrix is realized in terms of capacitive coupling to the gate of the associated terminal.

To ensure that a uniform I_0 is applied to each *p*-bit (4), the same weighting factor *K* needs to be used for all the ^{W}p-bits. To apply the given I_0, we first find $\max(b_i + \sum J_{ij})$ for any given [*J*], and then find the 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. Figure 2b shows that the explicit connections are made to one of the inputs "A" and Figure 2c shows the subcircuit of the FA with *Ci*,*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.

Figure 4 shows the operation of an FA 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

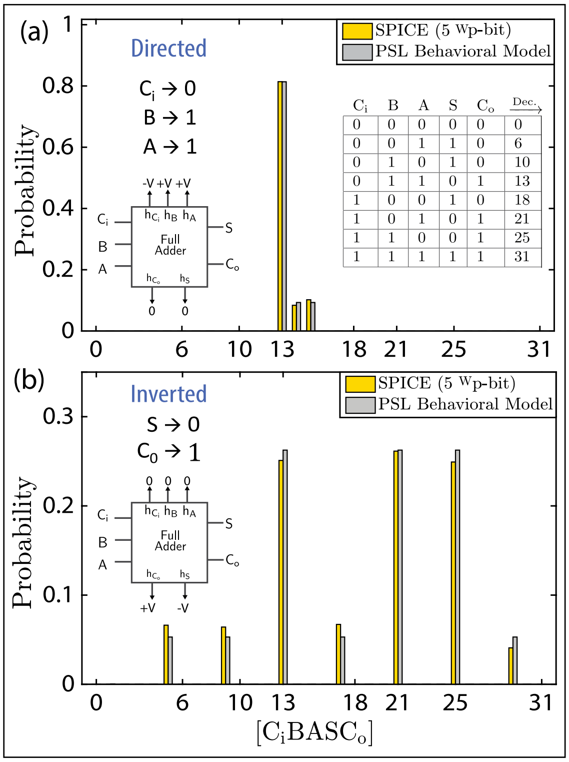

Figure 3. Complete SPICE implementation of an invertible FA (5^{W}*p*-bit). The 5^{W}*p*-bit invertible FA circuit is simulated in (a) directed and (b) inverted modes. The clamping values are indicated. All biasing terminals that are not clamped to 1 or 0 are grounded. The histogram of [CiBASC0] is obtained after thresholding voltages [(*V* < 0) = −1, (*V* > 0) = +1]. The SPICE model is run for 1 ps 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.

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}}. Figure 4 shows the steady-state (t = 1 μs) histogram plots of the FA operation in the direct and inverted mode side by side with the results from the PSL behavioral model.

A good agreement between the ideal PSL behavioral model and the coupled SPICE simulation that solves predictive technology model (PTM)-based transistors models with stochastic Landau–Lifshitz–Gilbert (LLG) validates the hardware mapping of the ideal p-bit equations with the weighted p-bits.

3SUM problem

3SUM is a decision problem in a 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 FAs can be utilized 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 SSP.

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

Figure 4a shows the circuit constructed out of FAs to solve a 4-bit 3SUM problem. Each of the FAs in the circuit is the five p-bit invertible adders that were shown in Figure 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 the 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 external circuitry could be used to detect only the results that belong to the given set. Figure 4b shows how A, B, and C is fluctuating between the values that satisfy the clamped sum 15.

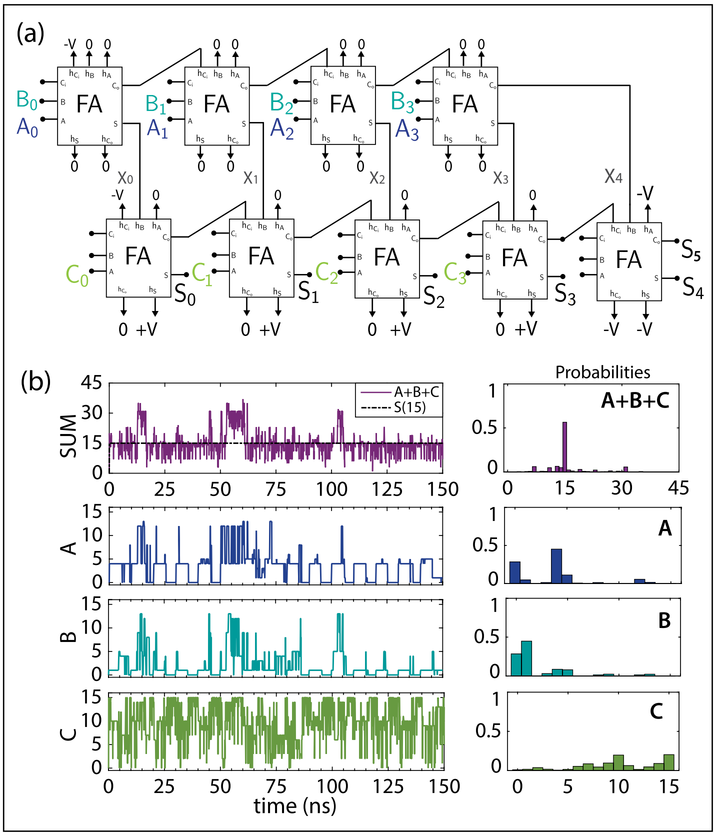

Figure 4. SPICE simulation of a 4-bit 3SUM problem (9 × 5 = 45 $^W p$-bit network). (a) Circuit is constructed by interconnecting two rows of invertible FAs to construct a three-number 4-bit adder. The sum S is clamped to the desired value, and A, B, C resolve themselves to create all the possible three-number subsets out of all positive numbers from 0 to 24 – 1 that satisfy A + B + C = S. (b) Results when S is clamped to 15. Here, A, B, and C get correlated to satisfy the sum with different combinations. In this example, the inputs A, B, and C are unconstrained and can take any value between 0 and 15.

Subset sum problem

In this section, we show how the hardware circuit that was designed for 3SUM problem could be modified to solve a small instance of 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' ⊆ G whose elements sum to a specified target. For example, Figure 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)

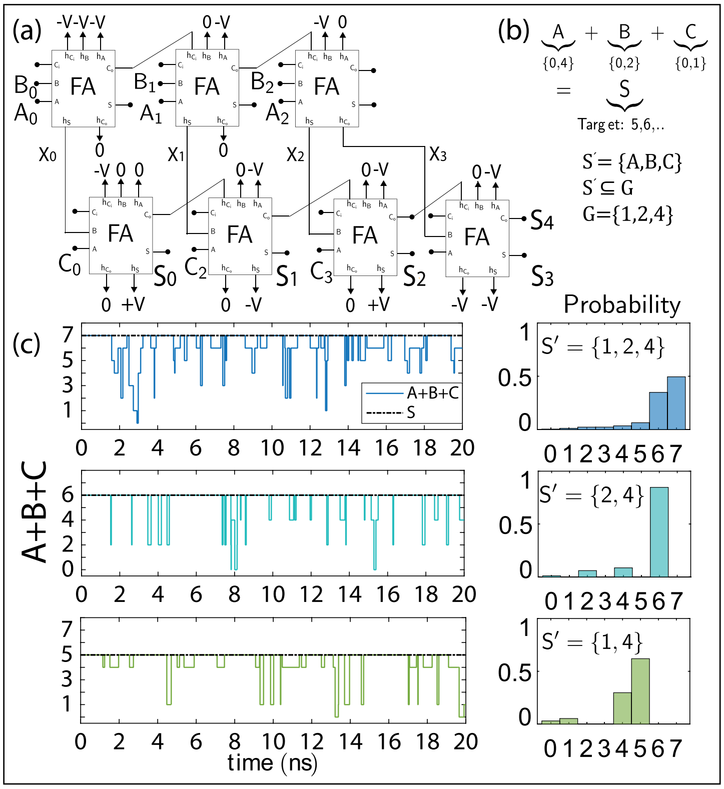

Figure 5. SPICE simulation of a three-input, 3-bit SSP (7 × 5 = 35 ^{W}p-bit network). (a) Three-input 3-bit binary adder that adds three numbers A, B, and C. Unlike the 3SUM, in this case, the inputs are constrained to a given value specified by the set G = {1,2,4}. A target S is selected and the output of the adders are clamped to the target value as shown in (b). (c) 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.

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. Figure 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 FAs to the second, as in normal addition (Figure 4a). In the SSP, we observed that reversing these connections from the second layer of the adder to the first layer drastically improves the accuracy of the solution (Figure 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 of the SSP which would, in general, require more layers of FAs 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 the hard problems of computer science in an asynchronously running hardware platform without any external clocking.

In this article, we have proposed a compact building block for the PSL combining a recently proposed embedded MRAM-based *p*-bit with an integrated capacitive network that can be implemented using 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 [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 $[C_j]$ or conductance network $[G_j]$ 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 simple illustration of the NP-complete SSP hardware solver can be designed using the unique invertibility feature of *p*-circuits. ■

References

[1] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, “Intrinsic optimization using stochastic nanomagnets,” *Sci. Rep.,* vol. 7, p. 44370, 2017.

[2] B. Behin-Aein, V. Diep, and S. Datta, “A building block for hardware belief networks,” *Sci. Rep.*, vol. 6, p. 29893, 2016.

[3] Y. Shim, A. Jaiswal, and K. Roy, “Ising computation based combinatorial optimization using spin-Hall effect (SHE) induced stochastic magnetization reversal,” *J. Appl. Phys.*, vol. 121, p. 193902, 2017.

[4] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, “Stochastic p-bits for invertible logic,” *Phys. Rev.*, vol. 7, p. 031014, 2017.

[5] R. Zand et al., “R-DBN: A resistive deep belief network architecture leveraging the intrinsic behavior of probabilistic devices,” arXiv preprint arXiv:1710.00249, 2017.

[6] C. Lin et al., “45nm low power CMOS logic compatible embedded STT MRAM utilizing a reverse-connection 1T/1MTJ cell,” in *Proc. 2009 IEEE Int. Elect. Dev. Meet.*, pp. 1–4.

[7] K. Y. Camsari, S. Salahuddin, and S. Datta, “Implementing p-bits with embedded MTJ,” *IEEE Elect. Device Lett.,* vol. 38, p. 1767, 2017.

[8] T. Shibata and T. Ohmi, “A functional MOS transistor featuring gate-level weighted sum and threshold operations,” *IEEE Trans. Elect. Dev.*, vol. 39, p. 1444, 1992.

[9] A. Mizrahi et al., “Neural-like computing with populations of superparamagnetic basis functions,” *Nat. Commun.*, vol. 9, p. 1533, 2018.

[10] N. Nakamura, K. Shimada, T. Matsuda, and M. Kimura, “Neuron MOS inverter and source follower using thin-film transistors,” in *Proc. 2015 IEEE Int. Meet. Future Electr. Dev., Kansai*, pp. 90–91.

[11] T. H. Cormen, *Introduction to Algorithms*. MIT Press, 2009.

[12] F. L. Traversa and M. Di Ventra, “Polynomial-time solution of prime factorization and NP-complete problems with digital memcomputing machines,” *Chaos*, vol. 27, p. 023107, 2017.

■ Direct questions and comments about this article to Orchi Hassan, School of Electrical and Computer Engineering, West Lafayette, IN 47907 USA; hassan19@purdue.edu.