Memristor Crossbar Based Hardware Realization of BSB Recall Function

Miao Hu and Hai (Helen) Li
Dept. of Electrical and Computer Engineering
Polytechnic Institute of New York University
Brooklyn, New York, USA 11201
mhu01@students.poly.edu, hli@poly.edu

Qing Wu and Garrett S. Rose
Air Force Research Laboratory
Information Directorate, 525 Brooks Road,
Rome, New York, USA 13441
qing.wu@rl.af.mil, garrett.rose@rl.af.mil

Yiran Chen
Dept. of Electrical and Computer Engineering
University of Pittsburgh
Pittsburgh, PA, USA 15261
yic52@pitt.edu

***Abstract*—The Brain-State-in-a-Box (BSB) model is an auto-associative neural network that has been widely used in optical character recognition and image processing. Traditionally, the BSB model was realized at software level and carried out on high-performance computing clusters. To improve computation efficiency and reduce resource requirement, we propose a hardware realization by utilizing memristor crossbar arrays. Memristors can remember the historical profiles of the excitations and record them as analog variables. The similarity to biological synaptic behavior has encouraged a lot of research on memristor-based neuro-morphic hardware system. In this work, we explore the potential of a memristor crossbar array as an auto-associative memory. More specifically, the recall function of a multi-answer character recognition based on BSB model was realized. The robustness of the proposed BSB circuit was analyzed and evaluated based on massive Monte-Carlo simulations, considering input defects, process variations, and electrical fluctuations. The physical constraints when implementing a neural network with memristor crossbar array have also been discussed. Our results show that the BSB circuit has a high tolerance to random noise. Comparably, the correlations between memristor arrays introduce directional noise and hence dominate the quality of the circuit.**

Keywords* - *neural network, BSB model, memristor, crossbar array, process variation.

I. INTRODUCTION

As demand on high performance computation increases, the traditional Von Neumann computer architecture becomes less efficient. In recent years, neuromorphic hardware systems have gained great attention. Such systems can potentially provide the capabilities of biological perception and information processing within a compact and energy-efficient platform [1][2]. Many research activities have been carried out on neural network algorithm enhancement [3] and/or system implementations built upon the conventional CPU, GPU, or FPGA [4].

As a highly generalized and simplified abstract of a biological system, an artificial neural network usually uses a *connection matrix* to represent a set of synapse networks. Accordingly, a group or groups of chemically connected or functionally associated neurons can be mathematically transformed into matrix-vector multiplication(s). Similar to the biological systems, the neural network algorithms inherently are adaptive to the environment and resilience to random noise. As a consequence, hardware realizations of neural networks require a large volume of memory and are associated with high design complexity and hardware cost [2]. Algorithm enhancement can alleviate the situation but cannot fundamentally resolve it. More efficient hardware-level solutions become necessary.

The *Brain-state-in-a-box* (BSB) model is a simple, auto-associative, nonlinear, energy-minimizing neural network [5][6][7][8]. A common application of the BSB model is *optical character recognition* (OCR) for printed text [9]. Recently, a multi-answer character recognition method based on the BSB model has been developed to improve reliability and robustness for noisy or occluded text images [10]. An input character image is processed through the BSB models in parallel for the ***recall*** (pattern recognition) operation. When all recalls are completed, a set of candidates are selected based on the convergence speed.

The existence of the memristor was predicted in circuit theory nearly forty year ago [11]. However, it wasn't until 2008 that the first physical realization was demonstrated by HP Lab through a TiO_2 thin-film structure [12]. Afterward, many memristor materials and devices have been reported or rediscovered. The memristor has many promising features, such as non-volatility, low-power consumption, high integration density, and excellent scalability [13][14]. More importantly, the unique property to record the historical profile of the excitations on the device makes it an ideal candidate to realize synapse behavior in electronic neural networks [15][16].

In this paper, we demonstrate a BSB recall circuit built on the memristor crossbar array. The crossbar architecture can naturally transfer the weighted combination of input signals to output voltages. However, due to physical constraints, the connection matrix weights in neuromorphic algorithms might not have the one-to-one mapping to the memristances of a crossbar array. We proposed a fast approximation mapping method so that the connection matrix can be mapped to pure circuit element relations. The validity of the method has been proved. Key design parameters and physical constraints have been extracted and studied. Furthermore, we carried out a detailed analysis to study the weight of each and all noise contributors on the accuracy and robustness of the BSB circuit. Interestingly, even if a large random process variation exists in memristor devices [17], it will not affect the BSB circuit much due to the inherent random noise tolerance of the BSB model. However, the correlation between two memristor networks within one BSB circuit dominates the robustness of the circuit since it introduces directional noise.

The remainder of the paper is organized as follows. In Section 2 we provide background information. Section 3 describes the hardware requirements to realize the BSB model and explains the details of the implementation. Section 4 classifies the types of noise that affect the quality of the BSB recall circuit. Section 5 analyzes the robustness of the BSB circuit design based on the simulation results. At the end, we conclude the paper in Section 6.

II. PRELIMINARY

A. Neural Network and BSB Model

Figure 1 illustrates a simple example of a neural network, in which two groups of neurons are connected by a set of synapses. We define $a_{i,j}$ as the synaptic strength of the synapse connecting the i^{th} neuron in the input group and the j^{th} neuron in the output one. The relationship of the activity patterns **F** of input neurons and **T** of output neurons can be described in matrix form:

$$\mathbf{T}_n = \mathbf{A}_{n\times m} \times \mathbf{F}_m \tag{1}$$

where matrix **A**, denoted as the *connection matrix*, consists of the synaptic strengths between the two neuron groups. The matrix-vector multiplication of Eq. (1) is a frequent operation in neural network theory to model the functionally associated with neurons in brains.

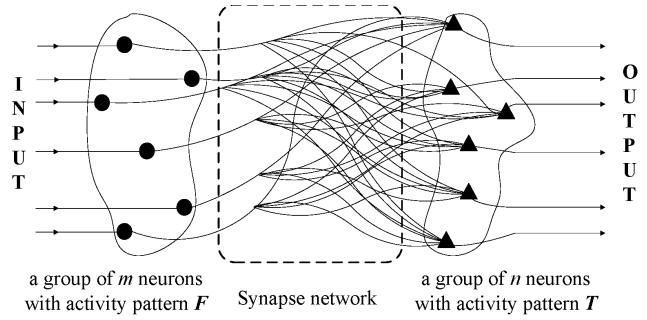

Figure 1: A simple example of neuron network.

The BSB model is a simple auto-associative neural network with two main operations – *training and recall* [5][6][7][8]. In this paper, we will focus on the hardware realization of the BSB recall operation. Its mathematical model can be represented as [10]:

$$\mathbf{x}(t+1) = S\big(\alpha \cdot \mathbf{A} \times \mathbf{x}(t) + \lambda \cdot \mathbf{x}(t)\big) \tag{2}$$

where, **x** is an *N* dimensional real vector, and **A** is an *N*-by-*N* connection matrix. $\mathbf{A} \times \mathbf{x}(t)$ is a matrix-vector multiplication, which is the main function of the recall operation. α is a scalar constant feedback factor. λ is an inhibition decay constant. $S(y)$ is the "squash" function defined as:

$$S(y) = \begin{cases} 1, \ if & y \geq 1 \\ y, \ if & -1 < y < 1 \\ -1, \ if & y \leq -1 \end{cases} \tag{3}$$

For a given input pattern **x**(0), the recall function computes Eq.(2) iteratively until *convergence*, that is, when all entries of **x**(t+1) are either "1" or "-1".

Recently, Wu et al. [10] developed a multi-answer character recognition method based on the BSB model for occluded text images. It processes an input character image through all the BSB models and selects multiple candidates with the fastest convergence speed for word-level or sentence-level language model. This method will be used to evaluate the reliability and robustness of the memristor-based BSB recall circuit proposed in this paper.

B. Memristor

Based on circuit theory, an ideal memristor with memristance M builds the relationship between the magnetic flux φ and electric charge q that passes through the device, that is, $d\varphi = M \cdot dq$. Since the magnetic flux and the electric charge are time dependent parameters, the instantaneous memristance varies with time and reflects the historical profile of the excitations through the device.

When developing actual memristive devices, many materials have demonstrated memristive behavior in theory and/or experimentation via different mechanisms. In general, a certain energy (or threshold voltage) is required to enable a state change in a memristor [18]. When the electrical excitation through a memristor is greater than the threshold voltage, e.g., $|v_{in}| > |v_{th}|$ the memristance changes (training). Otherwise, the memristor behaves like a resistor.

By nature, the memristor crossbar array is attractive for the implementation of neural networks. First of all, it supports a large number of signal connections within a small footprint, which is a basic characteristic of the synapse network in Figure 1. Secondly, the dominant operation in a neural network model is the weighted combination of input signals, which mimic the so-called dendritic potential [19]. As we shall show in Section 3.1, the crossbar array inherently provides capabilities for this operation. Moreover, it is required that the connection matrix in a neural network can be adjusted based on the learning process. Memristive devices are good at "learning" for the historical behavior [20].

In this work, we will focus on a hardware realization of the BSB recall function by assuming all of the memristors have been pre-programmed or already trained. During recall operations, the voltages across the memristors are constrained below the threshold voltage so that all the memristance values remain unchanged.

III. METHODOLOGY

A. Crossbar Array vs. Connection Matrix

Let's use the *N*-by-*N* memristor crossbar array illustrated in Figure 2 to demonstrate its matrix computation functionality.

Here, we apply a set of input voltages $\mathbf{V_I^T} = \{v_{I,1}, v_{I,2}, \cdots, v_{I,N}\}$ on the *word-lines* (*WL*) of the array, and collect the current through each *bit-line* (*BL*) by measuring the voltage across a sensing resistor. The same sensing resistors are used on all the BLs with resistance r_s, or conductance $g_s = 1/r_s$. The output voltage vector is $\mathbf{V_O^T} = \{v_{O,1}, v_{O,2}, \cdots, v_{O,N}\}$. Assume the memristor sitting on the connection between WL_i and BL_j has a memristance of $m_{i,j}$. The corresponding conductance is $g_{i,j} = 1/m_{i,j}$. Then the

relation between the input and output voltages can be represented by:

$$\mathbf{V_O} = \mathbf{C} \times \mathbf{V_I} \tag{4}$$

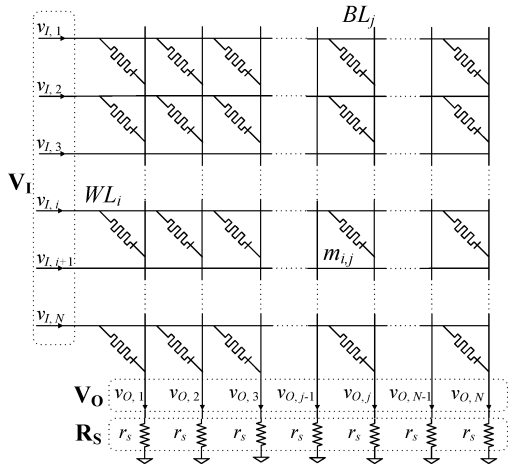

Figure 2: A memristor crossbar array.

Here, **C** can be represented by the memristances and the load resistors as:

$$\mathbf{C} = \mathbf{D} \times \mathbf{G} = diag(d_1, \cdots, d_N) \times \begin{bmatrix} g_{1,1} & \cdots & g_{1,N} \\ g_{2,1} & & g_{2,N} \\ \vdots & \ddots & \vdots \\ g_{N,1} & \cdots & g_{N,N} \end{bmatrix} \tag{5}$$

where, $d_i = 1/(g_s + \sum_{k=1}^{N} g_{i,k})$. The conductance matrix **G** and the memristor matrix **M** have a relation of: $\mathbf{G} = 1 \cdot /\mathbf{M}$[1].

Eq. (4) indicates that a trained memristor crossbar array can be used to construct the connection matrix **C**, and transfer the input vector $\mathbf{V_I}$ to the output vector $\mathbf{V_O}$. However, Eq. (5) shows that **C** is not a direct one-to-one mapping of conductance matrix **G** of the memristor crossbar array, since the diagonal matrix **D** is related to **G**. Though we can use a numerical iteration method to obtain the exact mathematical solution of **G**, it is too complex and impractical. We assume any $g_{i,j} \in \mathbf{G}$ satisfies $g_{min} \le g_{i,j} \le g_{max}$, where g_{min} and g_{max} respectively represent the minimum and the maximum conductance of all memristors in the crossbar array. Instead, we propose a simple and fast approximation to the mapping problem by allowing:

$$g_{i,j} = c_{i,j} \cdot (g_{max} - g_{min}) + g_{min} \tag{6}$$

In the following, we will prove that by using this mapping method, a decayed version of the connection matrix $\hat{\mathbf{C}}$ can be approximately mapped to the conductance matrix **G** of the memristor array.

PROOF. By plugging Eq. (6) in Eq. (5), we have:

$$\hat{c}_{i,j} = \frac{c_{i,j} \cdot (g_{max} - g_{min}) + g_{min}}{g_s + (g_{max} - g_{min}) \cdot \sum_{k=1}^{N} c_{i,j} + N \cdot g_{min}} \tag{7}$$

Note that many memristor materials, such as TiO_2 memristor, demonstrate a large g_{max}/g_{min} ratio [12]. Thus, a memristor at the high resistance state under a low voltage excitation can be regarded as an insulator, that is, $g_{min} \cong 0$. Moreover, the BSB recall matrix **A** is a special matrix with a small $\sum_{k=1}^{N} c_{i,j}$. For example, all the BSB models used for character recognition in our experiments have $\sum_{k=1}^{N} c_{i,j} < 5$ when $N = 256$. And $\sum_{k=1}^{N} c_{i,j}$ can be further reduced by increasing the ratio of g_s/g_{max}. As a result, the impact of $\sum_{k=1}^{N} c_{i,j}$ can be ignored. These two facts indicate that Eq. (7) can be further simplified as:

$$\hat{c}_{i,j} = c_{i,j} \cdot \frac{g_{max}}{g_s} \tag{8}$$

In summary, with the proposed mapping method, the memristor crossbar array performs as a decayed connection matrix $\hat{c}_{i,j}$ between the input and output voltage signals. □

B. Transformation of BSB Recall Matrix

To construct a memristor-based BSB recall circuit, our first task is to transfer the matrix **A** in the mathematical BSB recall model to a memristor array with physical meaning. A memristor is a physical device with conductance $g > 0$. Therefore, all elements in matrix **C** must be positive as shown in Eq. (5). However, in the original BSB recall model, $a_{i,j} \in \mathbf{A}$ could be positive or negative. We propose to split the positive and negative terms of **A** into two matrixes $\mathbf{A}^+$ and $\mathbf{A}^-$ as:

$$a_{i,j}^{+} = \begin{cases} a_{i,j}, & if\ a_{i,j} > 0 \\ 0, & if\ a_{i,j} \le 0 \end{cases}, \text{ and} \tag{9a}$$

$$a_{i,j}^{-} = \begin{cases} 0, & if\ a_{i,j} > 0 \\ -a_{i,j}, & if\ a_{i,j} \le 0 \end{cases}. \tag{9b}$$

As such, Eq. (2) becomes:

$$\mathbf{x}(t+1) = S\big(\mathbf{A}^+ \times \mathbf{x}(t) - \mathbf{A}^- \times \mathbf{x}(t) + \mathbf{x}(t)\big) \tag{10}$$

Here, for the default case we set $\alpha = \lambda = 1$. The two connection matrices A^+ and A^- can be mapped to two memristor crossbar arrays $\mathbf{M}_1$ and $\mathbf{M}_2$ in a decayed version $\hat{\mathbf{A}}^+$ and $\hat{\mathbf{A}}^-$, respectively, by following the mapping method in Eq. (6).

C. Circuit Realization

To realize the BSB recall function at the circuit level, we first convert the normalized input vector $\mathbf{x}(t)$ to a set of input voltage signals $\mathbf{V}(t)$. The corresponding function for the voltage feedback system can be expressed as:

$$\begin{aligned} \mathbf{V}(t+1) &= S'\big(\mathbf{A}^+ \times \mathbf{V}(t) - \mathbf{A}^- \times \mathbf{V}(t) + \mathbf{V}(t)\big) \\ &= S'\big(\mathbf{V}_{\mathbf{A}+}(t) - \mathbf{V}_{\mathbf{A}-}(t) + \mathbf{V}(t)\big) \end{aligned} \tag{11}$$

We use v_{bn} to represent the input voltage boundary, that is, $-v_{bn} \le v_i(t) \le v_{bn}$ for any $v_i(t) \in \mathbf{V}(t)$. The new saturation boundary function $S'()$ need to be modified accordingly. In implementation, v_{bn} can be adjusted based on requirements for convergence speed and accuracy. Meanwhile, v_{bn} must be smaller than v_{th} so that the memristance values will not change during the recall process.

Figure 3 illustrates the BSB recall circuit built based on Eq. (11). The design is an analog system consisting of three major components. Below is the detailed description.

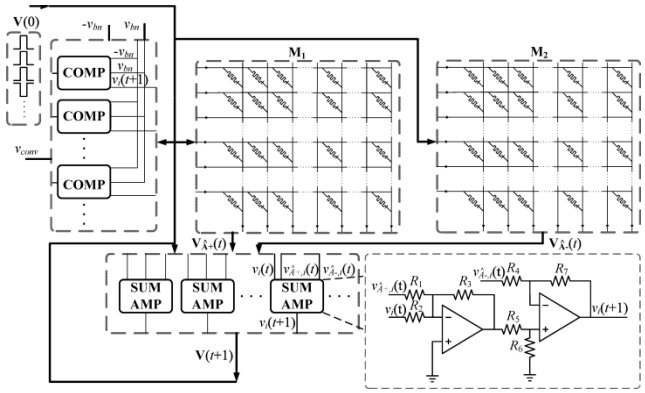

Figure 3: The conceptual diagram of the BSB recall circuit.

Memristor crossbar arrays: As the key component of the overall design, the memristor crossbar arrays are used to realize the matrix-vector multiplication function in the BSB recall operation. To obtain both positive and negative weights in the original BSB algorithm in Eq. (2), two memristor crossbar arrays $\mathbf{M}_1$ and $\mathbf{M}_2$ are required in the design to represent the connection matrices $\widehat{\mathbf{A}}^+$ and $\widehat{\mathbf{A}}^-$, respectively. The memristor crossbar array has the same dimension as the BSB weight matrix $\mathbf{A}$.

Summing amplifier: In our design, the input signal $v_i(t)$ along with $v_{\hat{A}_{+,i}}(t)$ and $v_{\hat{A}_{-,i}}(t)$, the corresponding voltage outputs of two memristor crossbar arrays, are fed into a summing amplifier. The conceptual structure of the summing amplifier can be found in the inner set of Figure 3.

Resulted by the decayed mapping method proposed in Section 3, the required $v_{A_{+,i}}(t)$ should be g_s/g_{max} times of the generated $v_{\hat{A}_{+,i}}(t)$. $v_{A_{-,i}}(t)$ has the same requirement too. In our design, we set $R_1 = R_4 = R_6 = 1/g_s$ and $R_2 = R_3 = R_5 = R_7 = 1/g_{max}$. The resulting output of the summing amplifier:

$$\begin{aligned} v_i(t+1) &= \frac{g_s}{g_{max}} \cdot v_{\hat{A}_{+,i}}(t) - \frac{g_s}{g_{max}} \cdot v_{\hat{A}_{-,i}}(t) + v_i(t) \\ &= v_{A_{+,i}}(t) - v_{A_{-,i}}(t) + v_i(t) \end{aligned} \quad (12)$$

which indicates that the decayed effect has been canceled out. The N dimensional BSB model requires N summing amplifiers to realize the addition/subtraction operation in Eq. (11). Also, for summing amplifiers, we should adjust their power signals to make their maximum/minimum output voltages equal $\pm v_{bn}$, respectively. In implementation, we can adjust the resistances $R_1 \sim R_7$ to match the required α and λ in Eq. (2), if they are not the default value 1.

Comparator: Once a new set of voltage signals $\mathbf{V}(t+1)$ is generated from the summing amplifiers, we send them back as the input of the next iteration. Meanwhile, every $v_i \in \mathbf{V}$ is compared to v_{bn} and $-v_{bn}$ to determine if path i has "converged". The recall operation stops when all the N paths reach convergence. Totally N comparators are needed to cover all the paths.

IV. Robustness of BSB Recall Circuit

Running the BSB recall circuit constructed in Section 3 under ideal conditions should lead to the exact same results as the BSB mathematical algorithm. Unfortunately, the noise induced by process variations and signal fluctuations in implementation can significantly affect circuit performance. In this section, we will address the modeling of this noise at the component level. The impact of physical design constraints will also be discussed.

A. Process Variations

1) Memristor Crossbar Arrays

Due to process variations, the real memristance matrix $\mathbf{M}'$ of a memristor crossbar array could be quite different from the theoretical $\mathbf{M}$. Their difference can be represented by a noise matrix $\mathbf{N_M}$, which includes two contributors – the systematic noise $\mathbf{N}_{\mathbf{M},\text{sys}}$ and the random noise $\mathbf{N}_{\mathbf{M},\text{rdm}}$. Consequently, $\mathbf{M}'$ can be expressed by:

$$\mathbf{M}' = \mathbf{M} \cdot\!* \mathbf{N_M} = \mathbf{M} \cdot\!* (1 + \mathbf{N}_{\mathbf{M},\text{sys}} + \mathbf{N}_{\mathbf{M},\text{rdm}}) \quad (13)$$

The impact of $\mathbf{N_M}$ on the connection matrix $\mathbf{C}$ is too complex to be expressed by a mathematical closed-form solution. But numerical analysis shows that it can be approximated by:

$$\mathbf{C}'_{\mathbf{M}} = \mathbf{C} \cdot\!* \mathbf{N_{CM}} = \mathbf{C} \cdot\!* \frac{1}{\mathbf{N_M}} \cdot\!* \frac{1}{\mathbf{N_M}} \quad (14)$$

Here, $\mathbf{C}'_{\mathbf{M}}$ is the connection matrix after including memristance process variations. $\mathbf{N_{CM}}$ is the corresponding noise matrix.

In the following analysis, we assume $\mathbf{N}_{\mathbf{M},\text{sys}}$ follows a normal distribution. To fully demonstrate the impact of the random process variations, the lognormal distribution is used to generate $\mathbf{N}_{\mathbf{M},\text{rdm}}$. Coefficient Corr_M is used to represent the correlation degree between the two memristor crossbar arrays in the same BSB circuit. When $\text{Corr}_\text{M} = 1$, the two arrays have the same systematic noise.

2) Sensing Resistance

Similar to the analysis of memristance variation, we classify the noise induced by $\mathbf{R_S}$ variations into the systematic noise $\mathbf{N}_{\mathbf{R},\text{sys}}$ and the random noise $\mathbf{N}_{\mathbf{R},\text{rdm}}$. The actual sensing resistance vector becomes:

$$\mathbf{R}'_{\mathbf{S}} = \mathbf{R_S} \cdot\!* \mathbf{N_{R_S}} = \mathbf{N_S} \cdot\!* (1 + \mathbf{N}_{\mathbf{R},sys} + \mathbf{N}_{\mathbf{R},rdm}) \quad (15)$$

$\mathbf{C}'_{\mathbf{R}}$, the connection matrix after including $\mathbf{N_{Rs}}$, is:

$$\mathbf{C}'_{\mathbf{R}} = \mathbf{C} \cdot\!* \mathbf{N_{CR}} = \mathbf{C} \cdot\!* [\mathbf{N_{R_S}}, \mathbf{N_{R_S}}, \cdots, \mathbf{N_{R_S}}] \quad (16)$$

Here, $\mathbf{N_{CR}}$ is the noise matrix of $\mathbf{C}$ after including the process variation of the sensing resistors. The mean value of r_s distribution, which reflects the impact of systematic noise, can be obtained during the post-fabrication testing. When training the memristances in BSB circuit, $\mathbf{N}_{\mathbf{R},\text{sys}}$ should have been included. Hence, in the following analysis, we only consider the random noise $\mathbf{N}_{\mathbf{R},\text{rdm}}$, which follows a normal distribution.

B. Signal Fluctuations

The electrical noise from the power supplies and the neighboring wires can significantly degrade the quality of analog signals. Different from the process variations that remain unchanged after the circuit is fabricated, these signal fluctuations vary during circuit operation. Without loss of generality, we assume the runtime noise of the summing amplifier's output signals follows a normal distribution, same as that of the output of the comparators.

C. Physical Constraints

There are three major physical constraints in the circuit implementation: (1) For any $v_i(0) \in \mathbf{V}(0)$, The voltage amplitude of initial input signal $v_i(0)$ is limited by the input circuit; (2) boundary voltage v_{bn} must be smaller than v_{th} of memristors; and (3) the summing amplifier has finite resolution.

In the BSB recall function, the ratio between boundaries of $S(y)$ and the initial amplitude of $x_i(0)$, $x_i(0) \in \mathbf{x}(0)$ determines the learning space of the recall function. If the ratio is greater than the normalized value, the recall operation takes more iterations to converge with a higher accuracy. Otherwise, the procedure converges faster by lowering stability. Thus, minimizing the ratio of $|v_i(0)|$ and v_{bn} can help obtain the best performance. However, the real amplifier has a finite resolution and v_{bn} is limited within v_{th} of the memristor. Continuously reducing $|v_i(0)|$ eventually will lose enough information in the recall circuit. So the resolution of the summing amplifier is a key parameter to determine the optimal ratio of $|v_i(0)|$and v_{bn} in circuit implementation. Certainly it also affects the design cost of amplifier and the overall design.

V. SIMULATION RESULTS

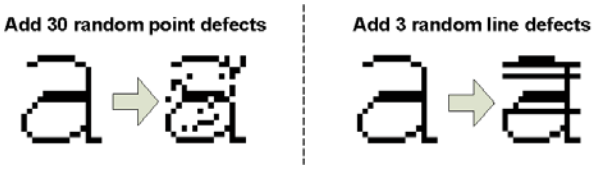

Figure 4: (a) Random line defects; (b) Random point defects

The robustness of the BSB recall circuit was analyzed based on Monte-Carlo simulations conducted with MATLAB. We tested 26 BSB circuits corresponding to the 26 lower case letters from "a" to "z". Each character image consists of 16×16 points and can be converted to a 256-entry vector. Accordingly, the BSB recall matrix has a dimension of 256×256. In each test, we created 500 design samples for each BSB circuit and ran 13,000 Monte-Carlo simulations. Two types of input pattern defects, random point defects and random line defects (see Figure 4), have been evaluated.

A. BSB Circuit under Ideal Condition

For an input pattern, the different BSB circuits have different convergence speeds. Figure 5 shows an example when processing a perfect "a" image through the BSB circuits trained for all 26 lower case letters. The BSB circuits for "a", "l", and "s" reach convergence with the least iteration numbers. The multi-answer character recognition method considers these three letters as winners and sends them to word-level language model [10].

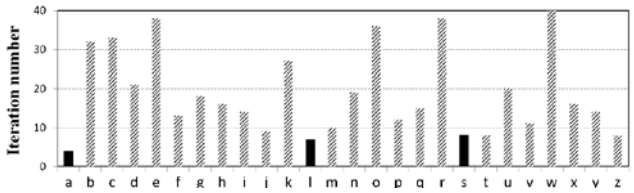

Figure 5: Iterations of 26 BSB circuits for a perfect "a" image.

Figure 6 summarizes the performance of the BSB circuit design under ideal conditions without input defects, process variations, or signal fluctuations. The *x*-axis and *y*-axis represent input images and the BSB circuits, respectively. All the winners are highlighted by the black blocks. Figure 6 shows that a BSB circuit corresponding to its trained input pattern always wins under the ideal condition. However, after injecting noise to input pattern or circuit design, some BSB circuits might *fail* to recognize its trained input pattern. In this work, we use the probability of failed recognitions P_F to measure the performance of a BSB circuit.

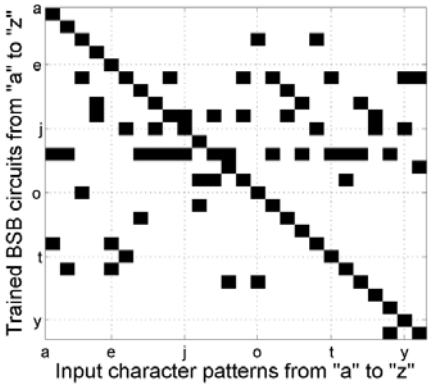

Figure 6: The performance of 26 BSB circuits under ideal condition.

B. Process Variations and Signal Fluctuations

Impact of random noise:

The random noise in the BSB circuit could come from process variations as well as electrical signal fluctuations. We summarize the impact of every single random noise component in Table 1, based on Monte-Carlo simulation results. Here, we assume two memristor crossbar arrays are fully correlated, *i.e.*, $\text{Corr}_M = 1$. The simulation results show that BSB circuit design has a high tolerance on the random noise: compared to the ideal condition without any fluctuation ("IDEAL"), these random noise of circuits cause slight performance degradation. This is because resilience to random noise is one of the most important inherent features for the BSB model as well as other neural networks.

Static Noise vs. Dynamic Noise:

The noise matrices $\mathbf{N_M}$ and $\mathbf{N_{Rs}}$ mainly affect the mapping between connection matrix and memristor crossbar array. Physically, these noise components come from process variations and remain unchanged during the

recall operation. Thus, they can be regarded as ***static noise*** N_S. On the contrary, the noise from the summing amplifiers and comparators are induced by electric fluctuations, which demonstrate a dynamic behavior during the iteration process of BSB recall function. We classify them as ***dynamic noise*** N_D.

Table 1: P_F(%) of 26 BSB circuits for 26 input patterns.

random point numbers	0	10	20	30	40	50
IDEAL	0	2.1	4.2	5.3	10.0	20.8
M (σ_{sys} = 0.1 & σ_{rdm} = 0.1)	0	1.9	4.6	6.5	14.2	24.7
R_S (σ = 0.1)	0	1.8	4.3	6.2	13.7	24.1
SUM-AMP (σ = 0.1)	0	1.9	4.4	7.7	13.5	23.1
COMPARATOR (σ = 0.1)	0	2.3	5.5	5.4	11.1	22.0
$Corr_M$ = 0.6	5.6	10.2	17.2	22.7	30.8	38.6
OVERALL ($Corr_M$ = 0.6)	4.6	8.2	15.2	20.7	32.8	36.6
random line numbers	0	1	2	3	4	5
IDEAL	0	7.3	13.8	21.5	35.8	50.2
M (σ_{sys} = 0.1 & σ_{rdm} = 0.1)	0	7.4	14.8	25.5	38.8	53.6
R_S (σ = 0.1)	0	7.4	14.8	23.3	35.1	51.8
SUM-AMP (σ = 0.1)	0	7.7	15.3	23.4	34.7	52.6
COMPARATOR (σ = 0.1)	0	6.9	14.5	23.3	33.7	53.2
$Corr_M$ = 0.6	5.1	14.4	24.7	34.6	44.2	55.1
OVERALL ($Corr_M$ = 0.6)	6.3	15.4	24.2	34.1	44.0	58.2

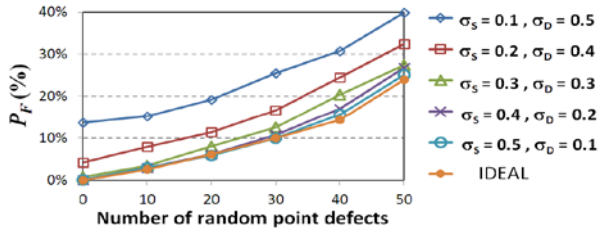

Figure 7: Static noise vs. dynamic noise.

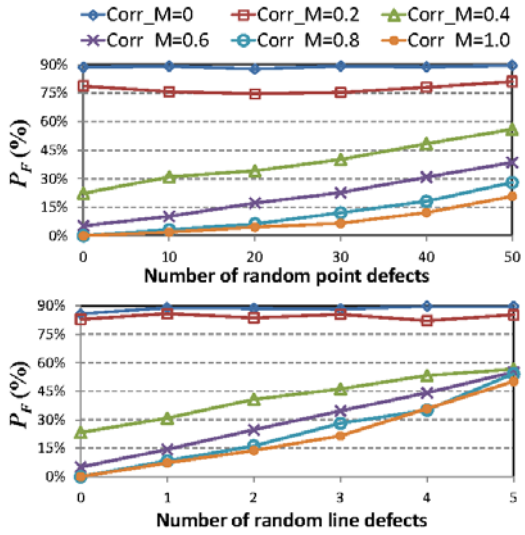

Figure 8: The impact of $Corr_M$.

We can adjust N_S and N_D and observe the combined impact on BSB circuit performance. For simplicity, we set $\sigma_{rdm}(M) = \sigma(R_S) = \sigma_S$ and $\sigma(AMP) = \sigma(COMP) = \sigma_D$. And $Corr_M$ = 1 to exclude the impact of correlations between the two memristor arrays. The result in Figure 7 shows that the dynamic noise dominates P_F. For example, when σ_D = 0.5 and σ_S = 0.1, P_F is high even with a clean input image. Decreasing σ_D but increasing σ_S results in P_F reduction in all regions.

Impact of $Corr_M$:

The BSB circuit implementation uses two memristor crossbar arrays to split the positive and negative elements of **A**. Reducing $Corr_M$ and hence increasing the difference in the systematic noise of two memristor arrays can be regarded as $\mathbf{A}^+$ and $\mathbf{A}^-$ having different overall shifts. This is ***directional noise*** in the recall function. As a consequence, $Corr_M$ demonstrates a higher impact. As shown in Figure 8, when decreasing $Corr_M$ from 1 to 0, the average P_F dramatically increases.

C. Impact of Summing Amplifier Resolution

To achieve the same learning space as the normalized BSB model, based on the amplifier operating voltage we set $v_{bn} = 1.6V$ and all elements of $\mathbf{V}(\mathbf{0})$ to be $\pm 0.1V$. Then we vary the summing amplifiers' resolution under different static and dynamic noise configurations. $Corr_M$ was fixed at 0.6. The simulation results are shown in Figure 9.

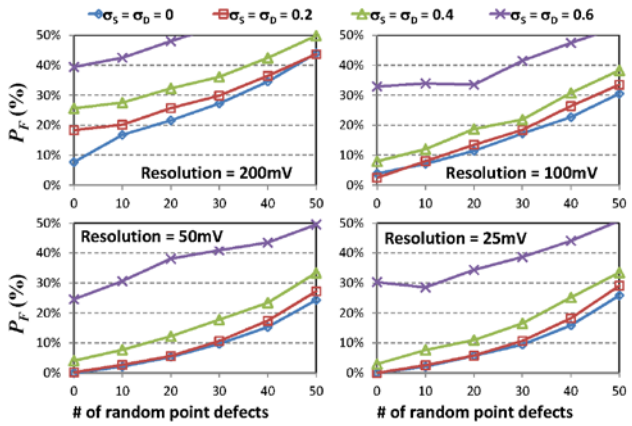

Figure 9: Impact of resolutions of summing amplifiers.

Again the simulation results demonstrate the BSB circuit's high tolerance for random noise: when $\sigma_S = \sigma_D \leq 0.4$, P_F is close to the ideal condition of $\sigma_S = \sigma_D = 0$. A $200mV$ resolution for the summing amplifier is too coarse to be acceptable: the BSB circuit cannot have zero P_F even under the ideal condition when neither input defects nor random noise are included. The resolution of $100mV$ is acceptable when the noise is not significant (e.g., $\sigma_S = \sigma_D \leq 0.2$) and the input pattern defect number is small (e.g., less than 20 random point defects). For the given physical constraints configuration, the $50mV$ and $25mV$ resolutions show similar results when $\sigma_S = \sigma_D \leq 0.2$. Further improving the resolution does not improve robustness of the BSB circuit much but increases design complexity and cost.

D. Overall Performance

In the previous analysis, we use the averaged P_F of all 26 BSB circuits for performance evaluation. One thing of particular interest is whether all BSB circuits degrade in the same way as we inject defects and noise into the system, or perhaps certain BSB circuits perform much better or worse

than the others. In this test, we set $Corr_M = 0.8$ and inject 0 or 30 random points defects for each input image. Figure 10 shows the comparison of P_F of each input character pattern under ideal conditions (noise free) and under the scenario including all process variations and signal fluctuations ($\sigma_S = \sigma_D = 0.1$).

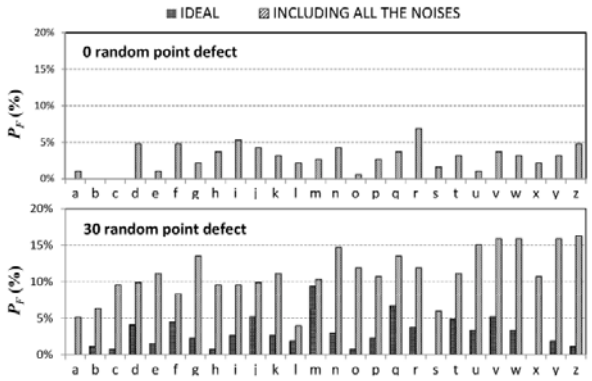

Figure 10: P_F for each character pattern.

The simulation shows that the performance degradation induced by process variations and signal fluctuations have a constant impact on all BSB circuits. When processing a perfect image under the ideal condition, no BSB circuits fails and hence $P_F = 0$. After including static and dynamic noise, P_F ranges from 1% to 7% for different input characters. When increasing the random point defects to 30 for input images, the range of P_F increases from 0~10% under ideal conditions to 4~16% after including all the noise sources.

VI. CONCLUSION

In this work, we firstly introduce a framework for a hardware realization of neural network algorithms using memristor crossbar arrays. More specific, we transfer the mathematical expression of the BSB recall model to a pure physical device relation and design the corresponding circuit architecture. The multi-answer character recognition method was used to perform robustness analysis for the proposed circuit. The impacts of various noise components induced by process variations and electrical fluctuations have been thoroughly studied. The physical constraints in circuit implementation have also been discussed. We found that the correlation between the two memristor crossbar arrays within a BSB recall circuit dominates the quality of the circuit. The resolution of the summing amplifier is another important factor, which is related to the physical constraints in circuit implementation. Interestingly, the random noise due to process variations and electrical fluctuations do not have obvious correlation with the character pattern which is "trained" and stored in the BSB connection matrix.

VII. ACKNOWLEDGMENT OF SUPPORT AND DISCLAIMER

Received and approved for public release by AFRL on 01/10/2012, case number 88ABW-2012-0192.

Any Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or its contractors.

VIII. REFERENCES

[1] P. Camilleri, M. Giulioni, V. Dante, D. Badoni, G. Indiveri, B. Michaelis, J. Braun, and P. del Giudice, "A neuromorphic avlsi network chip with configurable plastic synapses," in *International Conference on Hybrid Intelligent Systems*, 2007, pp. 296–301.

[2] J. Partzsch and R. Schuffny, "Analyzing the scaling of connectivity in neuromorphic hardware and in models of neural networks," *Neural Networks*, IEEE Transactions on, vol. 22, no. 6, pp. 919–935, 2011.

[3] M. Wang, B. Yan, J. Hu, and P. Li, "Simulation of large neuronal networks with biophysically accurate models on graphics processors," in *Neural Networks (IJCNN), The 2011 International Joint Conference on. IEEE*, 2011, pp. 3184–3193.

[4] H. Shayani, P. Bentley, and A. Tyrrell, "Hardware implementation of a bioplausible neuron model for evolution and growth of spiking neural networks on fpga," in *NASA/ESA Conference on Adaptive Hardware and Systems. IEEE*, 2008, pp. 236–243.

[5] J. Anderson, J. Silverstein, S. Ritz, and R. Jones, "Distinctive features, categorical perception, and probability learning: Some applications of a neural model." *Psychological Review*, vol. 84, no. 5, p. 413, 1977.

[6] E. M. H. Hassoun, "Associative neural memories: Theory and implementation," in *Oxford University Press*, 1993.

[7] J. Park, Y. Park, "An Optimization Approach to Design of Generalized BSB Neural Associative Memories," *Neural Computation, MIT Press Journals*, Vol. 12, No. 6, Jun. 2000, pp. 1449-1462.

[8] Y. Park, "Optimal and Robust Design of Brain-State-in-a-Box Neural Associative Memories," *Neural Networks, Elsevier*, Volume 23, Issue 2, Mar. 2010, pp. 210-218.

[9] A. Schultz, "Collective recall via the brain-state-in-a-box network," *Neural Networks, IEEE Transactions on*, vol. 4, no. 4, pp. 580–587, 1993.

[10] Q. Wu, M. Bishop, R. Pino, R. Linderman, and Q. Qiu, "A multi-answer character recognition method and its implementation on a high-performance computing cluster," in *FUTURE COMPUTING 2011, The Third International Conference on Future Computational Technologies and Applications*, 2011, pp. 7–13.

[11] L. Chua, "Memristor-the missing circuit element," *IEEE Transaction on Circuit Theory*, vol. 18, pp. 507–519, 1971.

[12] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

[13] Y. Ho, G. Huang, and P. Li, "Nonvolatile memristor memory: device characteristics and design implications," in *Proceedings of the 2009 International Conference on Computer-Aided Design. ACM*, 2009, pp. 485–490.

[14] D. Niu, Y. Chen, C. Xu, and Y. Xie, "Impact of process variations on emerging memristor," in *Design Automation Conference (DAC)*, 2010, pp. 877–882.

[15] D. Strukov, J. Borghetti, and S. Williams, "Coupled ionic and electronic transport model of thin-film semiconductor memristive behavior," *SMALL*, vol. 5, pp. 1058–1063, 2009.

[16] Y. Pershin and M. Di Ventra, "Experimental demonstration of associative memory with memristive neural networks," *Neural Networks*, vol. 23, no. 7, pp. 881–886, 2010.

[17] M. Hu, H. Li, Y. Chen, X. Wang, and R. Pino, "Geometry variations analysis of tio 2 thin-film and spintronic memristors," in *Asia and South Pacific Design Automation Conference. IEEE Press*, 2011, pp. 25–30.

[18] Y. Pershin and M. Di Ventra, "Practical approach to programmable analog circuits with memristors," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 8, pp. 1857–1864, 2010.

[19] U. Ramacher and C. V. D. Malsburg, *On the Construction of Artificial Brains*. Springer, 2010.

[20] T. Hasegawa, T. Ohno, K. Terabe, T. Tsuruoka, T. Nakayama, J. K. Gimzewski, and M. Aono, "Learning abilites achieved by a single solid-state atomic switch," *Advanced Materials*, vol. 22, no. 16, pp. 1831–1834, 2010.