Robust neural logic block (NLB) based on memristor crossbar array

Djaafar Chabi, Weisheng Zhao, Damien Querlioz, Jacques-Olivier Klein

Robust Neural Logic Block (NLB) based on Memristor Crossbar Array

Djaafar Chabi, Weisheng Zhao, Damien Querlioz, Jacques-Olivier Klein
Institut d'Electronique Fondamentale
Univ. Paris-Sud ,CNRS
Orsay, France
djaafar.chabi@u-psud.fr

***Abstract*—Neural networks are considered as promising candidates for implementing functions in memristor crossbar array with high tolerance to device defects and variations. Based on such arrays, *Neural Logic Blocks (NLB)* with learning capability can be built to replace *Configurable Logic Block (CLB)* in programmable logic circuits. In this article, we describe a neural learning method to implement Boolean functions in memristor NLB. By using Monte-Carlo simulation, we demonstrate its high robustness against most important device defects and variations, like static defects and memristor voltage threshold variability.**

Keywords-component; memristors, neural network, on-chip learning, defect and variation tolerance, supervised learning.

I. INTRODUCTION

Nanodevices synthesized by low cost technologies (e.g. nanoimprint [1], bottom-up approach [2]) require post-fabrication functionalization. However, the high fault rate of those devices requires special programming methods. Neural networks are often considered as a promising candidate to integrate nanodevices [8] thanks to their fault tolerance and adaptability.

Neural networks have been intensely studied since the beginnings of integrated circuits as an alternative computing architecture beyond Von-Neumann [9-10]. However, little success has been met partly due to the lack of simple synapse-like devices [11-12]. In 2008, the memristor predicted by L. Chua was physically demonstrated and presents synapse-like behaviors [3-13]. It can be fabricated through nanoimprint and is scalable to some nanometers [14]. Thanks to their two-terminal structure, memristors can be integrated in a crossbar array and promise extremely high densities [15]. Different types of research aim at exploiting the synapse-like behavior of memristors. Research focused on unsupervised learning using Spike Timing Dependent Plasticity (STDP) falls into the category of long-term research [16-17]. Shorter-term, applications can be achieved with supervised learning rules. Based on the crossbar array, we show that Neural Logic Blocks (NLB) could be built with learning capability and good tolerance against device defects and variations. This fundamental change could allow significant improvement of area and power efficiency. The NLBs could perform some basic logic functions and thus play the same role as Configurable Logic Blocks (CLB) in FPGA circuits (with the difference that would be configured by training). A number of interconnected NLBs could be assembled in a Field Trainable Neural Array (FTNA), as illustrated in Fig. 1, in order to compute complex functions. The NLBs contain several inputs connected with outputs trough memristors and functions as single-layer perceptrons. The FTNA would thus act as a complex multi-layer neural network [18], where the number of NLBs used to learn the expected function depends on the complexity of the latter. The logic synthesis of the expected function can be done off-line using standard tools to instantiate linearly separable logic functions (e.g. [19]) as primitives.

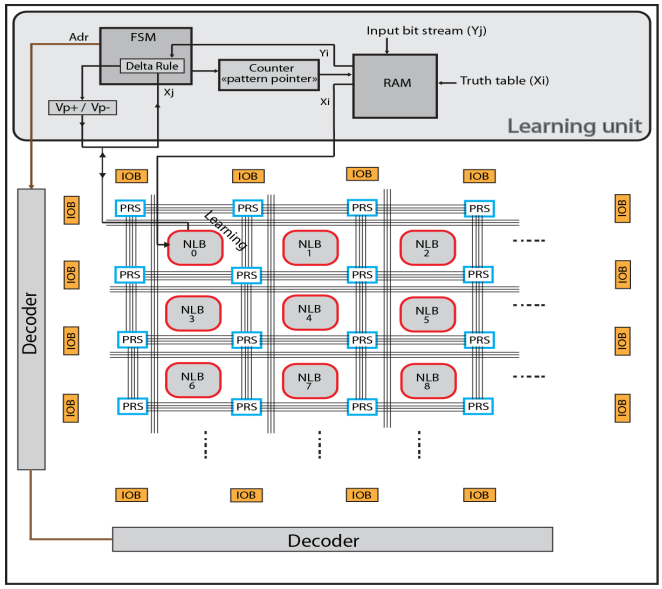

Figure 1. Architecture of a Field Trainable Neural Array (FTNA) based on interconnected Neural Logic Block (NLB) and Programmable Routing Switch (PRS) with on-chip learning. In order to perform function learning on-chip, a learning unit should be associated with an FTNA, which commands the address and learning process of each NLB. The pattern information including truth table and expected results for learning are stored in RAM. A Finite State Machine (FSM) drives then the learning process.

In this article, we describe a neural learning method to implement Boolean functions in NLB, which is robust against device defects and variations. The rest of paper is organized in the following manner. First, we will show a simple memristor model for NLB design and simulation. Second, we present a supervised learning method dedicated for memristor crossbar array. Finally, we focus on robustness study of NLB by using

Djaafar Chabi, Weisheng Zhao, Damien Querlioz and Jacques-Olivier Klein are with IEF, Univ Paris-Sud, UMR 8622, Orsay, F-91405 and CNRS, Orsay F-91405.

some probabilistic predictive models of defects. The most important faults of memristor such as stuck-open, stuck-close defects and its threshold variation have been taken into account in our simulation.

II. MEMRISTOR MODEL

A memristor features non-volatility, very small size and two-terminal synapse-like behaviors [3, 13]. It is considered as one of the most promising nanodevices to be integrated in a crossbar array to build nano neuromorphic computing systems. The conductance of memristor g(t) = I(t)/V(t) depends on its across current and / or voltage history. In our simulation, we use a simplified model of device behavior inspired by the measurements of [20] for example. The conductance change speed dg(t)/dt is a function of the voltage *V*, shown in Eq.1.

$$\frac{dg(t)}{dt} = f(V(t)) \qquad (1)$$

The sign of *V* depends on the voltage potential difference of the two terminals (see Fig.3). It is positive as the input voltage potential is higher than that of the output.

To implement supervised learning, programming voltage (V_P) pulses will update the conductance progressively. It should remain insensitive to low voltage pulses to keep the history of patterns learned and perform the computing [20, 8]. As a consequence, a memristor for learning circuits requires a "neutral" voltage range separated by two threshold voltages (V_{T-} and V_{T+}) as has been observed experimentally for example in [20, 25]. No conductance change occurs (f(V)=dg/dt=0) as the across voltage of memristor is in this range. When a voltage pulse exceeding those thresholds is applied its conductance changes as shown in Figure 3. This simple model of memristor is used in our learning circuit design and robustness simulation shown in the following sections.

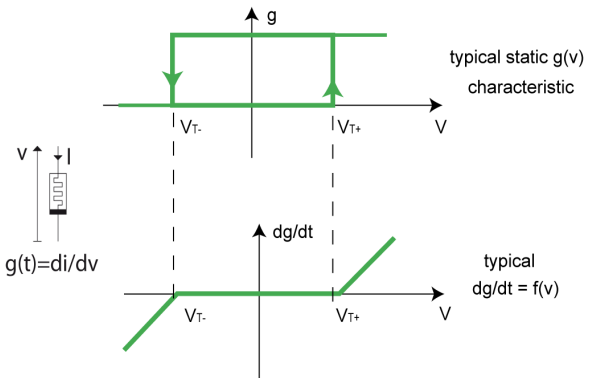

Figure 3. Example of one shape of the characteristic f(v)=dg/dt for memristive devices and the corresponding static g(v) hysteresis.

III. NEURAL LOGIC BLOCK

A. *Neural inspired circuits and architectures*

The architecture of NLB is composed of two parts: a memristor crossbar array (as synapse network), and the CMOS neurons (see Fig. 4). The crossbar array is connected to differential inputs and a usual "bias input", similar to [22] to emulate signed synaptic weights with pairs of positive conductance. There are N synapses or inputs per neuron; i and j represents respectively the number of input and neuron in the crossbar array.

Each logical input (X_i) corresponds to a pair of differential physical wires (X_{i+} and X_{i-}), presenting pre-synaptic potentials, drawn vertically in Fig. 4. Corresponding memristors M_{ij+} and M_{ij} with conductance G_{ij+} and G_{ij-} located at connections between vertical and horizontal lines implement a signed synaptic weight $W_{ij}=(G_{ij+} - G_{ij-}).K_j$ (see Eq. 2). Consequently, the horizontal wires correspond to the post-synaptic potentials described by Eq. 3. These potentials are compared to a common reference voltage (typically the ground) to obtain a binary output state of each neuron cell

$$K_j^{-1} = \sum_{i=1}^{N} G_{ij+} - G_{ij-} \qquad (2)$$

where K_j is a normalized factor the neuron j.

$$V_j = \sum_{i=1}^{N} W_{ij} \times X_{ij} = K_j \sum_{i=1}^{N} \left[G_{ij+} - G_{ij-} \right] \times X_{ij} \qquad (3)$$

V_j is the post-synaptic potential, which is compared to a reference threshold (e.g. ground) to generate the output state X_j (see Eq. 4) of neuron j (1 or -1).

$$X_j = sign\left(V_j\right) \qquad (4)$$

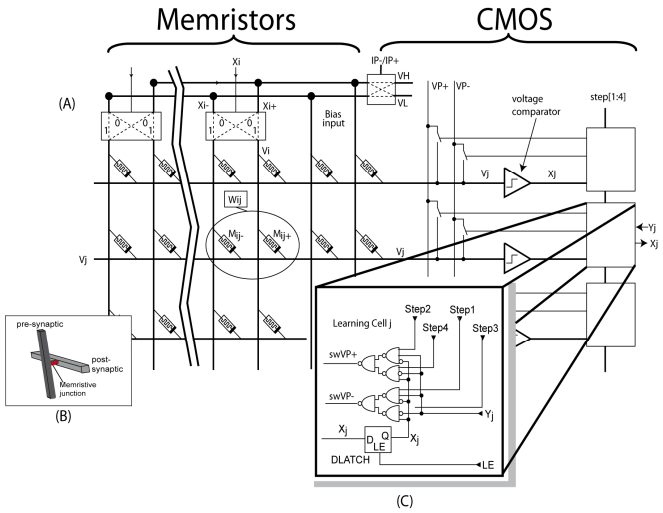

Figure 4. A) Architecture of NLB including a memristor crossbar array and CMOS neuron circuits. B) Memristor is formed at each junction where two nanowires cross. C) Learning cell to send programming voltage pulses (V_{P+}/ V_{P-}) when there is an error (e.g. the calculated output X_j is different to the expected output Y_j). Different steps can be used to eliminate the errors.

For this architecture, two operating steps have to be distinguished: the first is the learning process where the conductance of memristors is adjusted to define the function of the NLB; the second is the logic computing like other logic blocks, without function change.

B. Conditional learning

In the learning circuit of NLB (see Fig. 4A), the post-synaptic potential V_j is connected to either a positive (V_{P+}) or a negative programming voltage (V_{P-}), which are equal to the threshold voltage (V_{T+} or V_{T-}). In addition, the polarity of all the inputs can be optionally inverted (IP+/IP-). As the amplitude of post-synaptic potential equals to the threshold, the across voltage of the synapses will be greater or lower than the threshold depending on the sign of their inputs. Consequently, the change conductance is conditional (see Eq. 5). If the across voltage of a memristor is greater than V_{T+} and the maximum conductance G_{max} is not reached, its conductance will be increased by an increment value Winc0. On the contrary, if the across voltage is lower than V_{T-} and the minimum conductance G_{min} is not reached, its conductance will be decreased with a decrement value W_{inc0}.

As X_i is active (i.e. $X_i = 1$), $X_{i+}/X_{i-} = V_H / V_L$. V_H and V_L correspond respectively a positive and negative input voltage. On the contrary, $X_{i+}/X_{i-} = V_L / V_H$ as X_i is inactive (i.e. $X_i = 0$). In general, $V_L=-V_H$ and $V_P/V_{P+}=V_T/V_{T+}$.

$$\begin{aligned} &V_H - V_{P-} > V_{T+} \rightarrow \text{G increases (+)} \\ &V_L - V_{P-} < V_{T+} \rightarrow \text{No G change (0)} \\ &V_L - V_{P+} < V_{T-} \rightarrow \text{G decreases (-)} \\ &V_H - V_{P+} > V_{T-} \rightarrow \text{No G change (0)} \end{aligned} \quad (5)$$

The main idea of this method is to apply different sequences (see Fig. 4C) of programming pulses that implement a simplified (Boolean) version of the Windrow-Hoff's [23] Mean Least Square "Delta" rule (Eq. 6).

$$\Delta W_{ij} = \alpha X_i (Y_j - X_j) \quad (6)$$

There are two conditions to implement this rule in memristor crossbar array:

1) Learning conditions:
$V_H - |V_P| > |V_T|$ & $V_L - |V_P| < |V_T|$

2) Operating condition:
$|V_i| < |V_T|$

The first condition ensures the differential effect of inputs; the programming pulses (V_{P+}/ V_{P-}) change only the conductance of one of the couple of memristors (M_{ij+}/M_{ij-}). The second condition ensures that the operating voltage at inputs (V_i) will not change the conductance of memristors.

In order to evaluate the learning method suitable for the NLB, a high level functional model has been developed using Matlab® code. Each memristor is characterized by its conductance state, which depends on the across voltage compared to V_{T+} and V_{T-} as described in Eq. 4. The conductance of memristor is normalized compared to K_j (see Eq. 2). Consequently, in our simulation we assume that each programming voltage pulse V_{P+} or V_{P-} leads to a unitary conductance change. In addition, the conductance is limited at the range from G_{min} to G_{max}. The memristor model considered here is perfectly symmetrical (i.e. $V_{T+} = -V_{T-}$).

IV. ROBUSTNESS SIMULATION

The simulation is based on a model of NLB with one output and several logical inputs (see Fig. 5). The purpose of this work is to study the reliability of a single perceptron that can learn all linearly separable Boolean functions [24]. Most of important memristor synapse defects and variations have been taken into account, supposing that the output neuron can work perfectly. Each defect has been treated individually from the others, which allows us to know their impact on the reliability of the NLBs and to develop an analytical model of their probability of succeeding learning.

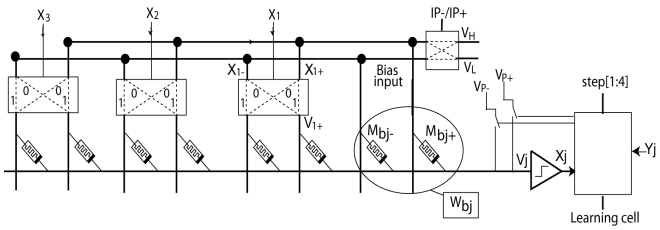

Figure 5. Simplified perceptron model based on crossbar of memristors with one output and several input.

As any nanoscale devices fabricated using low cost technologies [1-2], memristors suffer from a high defect rate and high variability. In order to evaluate their impact during the learning process, we integrated the device defects and variability in the memristor model shown in Fig. 3 for the robustness simulation of NLB architecture.

Fig. 6 shows some possible defects in memristors:

- Defective memristors where the conductance, stuck at one of the extreme value (G_{min}/G_{max}), cannot be changed (see Fig. 6 a).
- Variability of memristor conductance range, (see Fig. 6 b).
- Variability of the threshold V_{T+} and V_{T-} (see Fig. 6 c).
- Variability of the f(v)=dg/dt slope (Fig. 6 d).

In this paper, we focus on the cases of stuck-open, stuck-close defects and the variation of threshold voltage of memristors. Forthcoming works will include all kinds of variability.

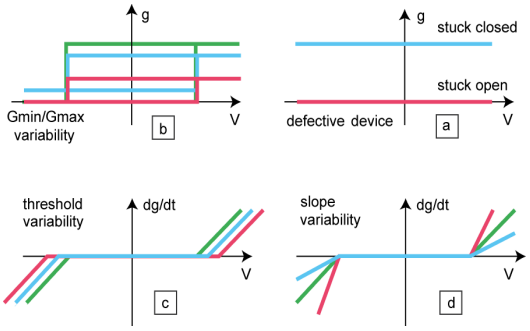

Figure 6. Defect and variability models. (a) The model of defective memristors. (b) The range variability of conductance. (c) The variability of threshold voltage. (d) The f(v) slope variability.

A. Robustness of NLB to device defect

At first, we investigated the robustness of the learning process against defective synapses (stuck-open or stuck-close). There are at most three inputs in these functions, and then a NLB 8×1 with one output and eight inputs is sufficient to implement these functions, where there are six differential inputs (X_{1+} X_{1-}, X_{2+} X_{2-}, X_{3+} X_{3-}) and two differential input for the bias (X_{b+} X_{b-}). The learning of three Boolean functions F_1 = X_1, F_2 = X_1 and X2, F3 = (X_1 and X_2)and(notX_3) has been experimented in presence of synaptic defect, injected randomly in the NLB for each simulation and the learning successfully is measured with the average of 100 trials (see Fig. 7).

The choice of the functions F_1, F_2 and F_3 is based on the number of synaptic weights (memristors) N_m needed to learn these functions. There is only a subset of synaptic connections that are programmed during the learning stage of these functions. This subset depends on the function to learn. The three functions require different numbers of memristors. For example, to learn the function F_1, it is necessary to have the synaptic weight of the input X_1 higher than the others, why the memristor M_{X1+} should work correctly. To learn the function F_2, the neuron needs a positive synaptic weight of the input X_1, a positive synaptic weight of the input X_2 and a negative synaptic weight of the bias input, then three memristors (N_m=3) are required to learn this function {M_{X1+}, M_{X2+}, M_{Xb-}}. For the function F_3, five memristors are needed (N_m=5) for learning {M_{X1+}, M_{X2+}, M_{b+}, M_{b-}}. Knowing the number of memristors needed to learn a given function F, it is possible to calculate the probability of success to learn this function. If P_f is the probability to have one defective memristors, the probability to find N_m memristors to learn a function F is given by the function $(1\text{-}P_f)^{Nm}$.

We compare in Fig. 7 the probability of success for N_m =1 (F_1), N_m=3 (F_2) and N_m=5 (F_3) to the simulation curves. The increasing of defect rate P_f decreases the successful of the NLB. The theoretical curves are close to the simulation curves.

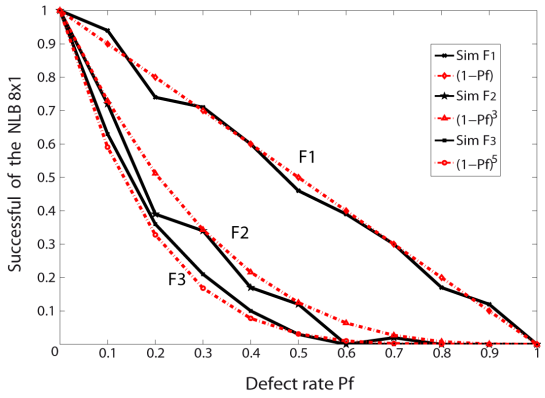

Figure 7. Success convergence of one neuron with eight physical inputs for different rate of defect (stuck close). The solid curves obtained by simulation for 100 trials, and the dashed curves obtained by the calcul of probability.

In order to understand this hypothesis, we analyzed more deeply the results of the learning stage for the function F_2=(X_1)And(X_2) by inspecting influence of stuck defects in each memristor. The set-up is to put one of memristors stuck open or close, and see if the learning succeeds or not. By this way, we can locate the memristors responsible for the failure to learn function F_2. Table 1 and 2 show respectively the conductance of memristors after learning, in the case of stuck-open and stuck-close.

To learn the function F_2, the synaptic weights (W) should be updated following Eqs. 7, extended from Eq. 3 and the truth table of three inputs with complementary binary convention (i.e. low level=-1 and high level=+1):

$$\begin{aligned} &W_b - W_{X1} - W_{X2} - W_{X3} < 0 \\ &W_b + W_{X1} - W_{X2} - W_{X3} < 0 \\ &W_b - W_{X1+} W_{X2} - W_{X3} < 0 \\ &W_b + W_{X1+} W_{X2} - W_{X3} > 0 \\ &W_b - W_{X1} - W_{X2} + W_{X3} < 0 \\ &W_b + W_{X1} - W_{X2} + W_{X3} < 0 \\ &W_b - W_{X1} + W_{X2} + W_{X3} < 0 \\ &W_b + W_{X1} + W_{X2} + W_{X3} > 0 \end{aligned} \qquad (7)$$

TABLE 1. LOCATION OF THE CRITICAL MEMRISTORS TO LEARN F_2 IN CASE OF STUCK-OPEN DEFECT (I.E. G_{Xi}=0)

Memristors / Stuck open defect	M_{X1+}	M_{X1-}	M_{X2+}	M_{X2-}	M_{X3+}	M_{X3-}	M_{b+}	M_{b-}	Convergence (Nb epochs)
none	2.05	1.02	4.02	0.08	1.03	1.01	0.05	3.03	Success(3)
G_{X1+}=0	**0**	0.01	2.01	0.01	1.00	1.08	1.06	1.02	**Fail (50)**
G_{X1-}=0	3.02	**0**	3.05	1.05	1.06	1.04	0.07	3.00	Success(4)
G_{X2+}=0	2.03	0.09	**0**	0.04	1.09	1.03	1.05	1.09	**Fail (50)**
G_{X2-}=0	2.06	1.01	3.07	**0**	1.09	1.02	0.01	3.03	Success (4)
G_{X3+}=0	1.06	0.08	1.05	0.05	**0**	0.00	0.06	1.03	Success (2)
G_{X3-}=0	1.01	0.01	1.03	0.04	0.1	**0**	0.00	1.04	Success (2)
G_{b+}=0	2.04	1.01	3.06	0.03	1.00	1.05	**0**	3.01	Success (4)
G_{b-}=0	3.00	0.01	3.02	1.02	0.03	3.09	0.05	**0**	**Fail (50)**
G_{X1-}=0, G_{X2-}=0, G_{X3-}=0,G_{X3+}=0, G_{b+}=0	0.09	**0**	1.07	**0**	**0**	**0**	**0**	1.01	Success (2)

TABLE 2. LOCATION OF CRITICAL MEMRISTORS TO LEARN F_2 IN CASE OF STUCK-CLOSE DEFECT (I.E. $G_{Xi}=G_{MAX}$ =12).

Memristors / Stuck-close defect	M_{X1+}	M_{X1-}	M_{X2+}	M_{X2-}	M_{X3+}	M_{X3-}	M_{b+}	M_{b-}	Convergence (Nb epochs)
none	4.02	0.03	3.08	0.02	1.08	1.09	1.02	2.04	Success(3)
$G_{X1+}= G_{max}$	**12**	5.06	5.05	0.08	1.01	0.04	0.09	5.03	Success(4)
$G_{X1-}= G_{max}$	11.99	**12**	3.04	0.03	2.05	1.02	1.09	3.04	**Fail (50)**
$G_{X2+}= G_{max}$	5.04	0.08	**12**	5.08	1.04	0.06	0.05	5.02	Success(4)
$G_{X2-}= G_{max}$	3.07	0.04	11.99	**12**	2.08	1.05	1.06	3.02	**Fail (50)**
$G_{X3+}= G_{max}$	4.06	1.05	6.05	0.09	**12**	10.05	0.07	5.03	Success (4)
$G_{X3-}= G_{max}$	5.01	0.01	5.02	0.01	11.08	**12**	1.09	5.04	Success (4)
$G_{b+}= G_{max}$	4.00	0.01	4.07	1.02	0.01	3.02	**12**	11.99	**Fail (50)**
$G_{b-}= G_{max}$	5.03	0.07	5.07	0.06	0.09	1.00	5.09	**12**	Success (4)
$G_{X1+}=G_{max}$,$G_{X2+}=G_{max}$ $G_{X3-}=G_{max}$,$G_{X3+}=G_{max}$	**12**	1.02	**12**	0.06	**12**	**12**	0.04	1.07	Success (2)
$G_{X1+}=G_{max}$,$G_{X2+}=G_{max}$ $G_{X3-}=G_{max}$,$G_{X3+}=G_{max}$ $G_{b-}= G_{max}$	**12**	0.03	**12**	0.09	**12**	**12**	0.02	**12**	Success (1)

The first row of table 1 shows the conductance of the different memristors in the case of no defect:

$$W_{X1}=G_{X1+} - G_{X1-}= 2.05-1.02=1.03$$
$$W_{X2}= G_{X2+} - G_{X2-}=4.02-0.08=3.94$$
$$W_{X3} = G_{X3+} - G_{X3-}=1.03-1.01=0.02$$
$$W_b = G_{b+} - G_{b-}=0.05-3.03=-2.98$$

We note that these values confirm the previous equations and the inactive synapses during function learning like W_{X3} are neutralized and their weight are always very low nearly to zero.

As shown in table 1, the cases where the learning failed confirm that there is a set of critical memristors to determine the success of function learning. For F_2, the set is {M_{X1+}, M_{X2+}, M_{b-}}. They should be functional to ensure the NLB learning.

In the case of stuck-close defect, table 2 shows the successful convergence of neuron and the final conductance of memristors after learning. The first row of table 2 shows that the neuron converges quickly with only 03 learning epochs. There are three cases {M_{X1-}, M_{X2-}, M_{b+}}, where the neuron diverges due to the defects. As these memristors aren't in defect, the learning process can adjust the conductance of the other functional memristors to compensate these defects, and learn the functions successfully.

The simulation results of both type of defects stuck-open and stuck-close, showed that the neuron can learn the expected function F_2 even in the presence of several defects on memristors (synapses). We have seen that the number of defects in the memristors is not very important as long as the neuron found correct memristor to compensate these defects. However the position of defect performs a critical role. When a defect affects one of the crucial memristors, it prevents the synaptic weight to move in the right direction in order to learn the expected function, and the neuron will be certainly going to fail. This confirms that the neuron cannot tolerate the stuck-open defects affecting the memristors {M_{X1+} M_{X2+} M_{b-}}, on the contrary, the stuck-close defects affecting the complementary memristors {M_{X1-}, M_{X2-}, M_{b+}}. The existence of critical memristors in order to learn a given function can be seen as an issue, since a defect on one of these critical memristors prevents the NLB from converging. It is also advantageous, because not all synapses are necessary to learn the expected function. To improve the tolerance against defects of the NLB, a possible solution is to duplicate the inputs to improve the likelihood to find N_m memristors. It is also possible to duplicate the output neurons, to have several neurons to learn the same function in same time and the winner takes the expected function ("competitive learning"), as we demonstrated in [27].

B. Robustness of NLB to the variability of V_T

In this section, we will address one of the most important issues in the memristor crossbar array: the threshold variation between memristors. Experimental measurements [25-26] demonstrated that the threshold voltage of nanodevices like memristor is far from being uniform and follows a normal distribution around a mean value V_{T0}. Although the situation is expected to improve when the technology matures, a high level of variability is probably intrinsic due to the small size of the devices and the physical nature of switching. This variability can lead to errors during the learning process and cause the divergence of neuron. The purpose of our study is to investigate the variation rate of threshold voltage that the neuron can tolerate, which allows us to develop a model for convergence prediction in presence of V_T variation. Two sets of simulations have been performed to explore respectively the reliability impact of the extreme V_T values and the deviation of normal distribution.

1) Impact of low and high V_T

As the threshold voltage V_T of memristors follows a normal distribution around the mean $V_{T0,}$ only a minority of memristors presents a particularly high threshold or low threshold. However these extreme values could be the source of failures in learning. As mentioned in III. II, NLB is based on conditional learning (Eq. 7). For this study, we take normalized units such that $V_P/V_{P+}=V_{T0}=1$ (V_{P-} and V_{P+} are chosen equal to V_{T0} as explained in section III.B). We chose V_H/V_L=0.4/-0.4, which ensures the operating conditions for a large range of V_T values. In case of a memristor having an especially low V_T value (for example V_{T+}/V_{T-}=0.5/-0.5), the learning conditions can be described as following:

$V_H-V_{P-}>V_{T+}$ → 0.4+1=1.4>0.5→ G increases (+)

$V_L-V_{P-}< V_{T+}$ → -0.4+1=0.6>0.5→ G increases (+)

$V_L-V_{P+} < V_{T-}$ → -0.4-1=-1.4<-0.5→ G decreases (-)

$V_H-V_{P+} > V_{T-}$ → 0.4-1=-0.6<-0.5 → G decreases (-)

This configuration does not meet completely the conditional requirement to achieve a good learning. The second and fourth conditions should not change the conductance. The direction of conductance change depends only on the sign of V_P, it is positive (negative) for a negative (positive) $V_{P.}$ As a

result, the conductance may oscillate around the initial value throughout a learning stage.

In the case of high threshold, $V_{T+}/V_{T-}=2/-2$, if $V_H/V_L=0.4/-0.4$, $V_P/V_{P+}=V_{T0}=1$, the learning conditions can be described as following:

$V_H-V_{P-}>V_{T+}$ → 0.4+1=1.4 <2 → No G change

$V_L-V_{P-}< V_{T+}$ → -0.4+1=0.6 <2 → No G change

$V_L-V_{P+} < V_{T-}$ → -0.4-1=-1.4 >-2 →No G change

$V_H-V_{P+} > V_{T-}$ → 0.4-1=-0.6 >-2 → No G change

In this configuration, the conductance of memristor cannot be updated during the learning stage, as V_P will never reach V_T. The impact of high V_T is similar to that of stuck-open defect, when the conductance is initialized around zero. On the contrary, when the conductance is initialized around G_{max}, the impact of high V_T is similar to the effect of stuck-close defect.

The above discussions have been confirmed by the simulation of a NLB 8×1 (see Fig. 5) to learn the function F_2. Table 3 shows the average rate of successful convergence for 100 trials. Each time one of eight memristors presents high or low V_T and the initial conductance is set to either G_{min}(G_{init}≈0) or G_{max}(G_{init}≈12). The results demonstrate that the successful learning rate becomes very low (e.g. 0.02) as the critical memristor is affected by the low or high V_T. As $G_{init}=G_{min}$, the set of critical memristors causes the divergence of the neuron correspond to those found in the case of stuck-open defect {M_{X1+}, M_{X2+}, M_{b-}}. As G_{init}≈G_{max}, the set of critical memristors correspond to those found in the case of stuck-close {M_{X1-}, M_{X2-}, M_{b+}}. These results confirm our theoretical conclusion mentioned in the previous paragraphs.

TABLE 3. AVERAGE RATE OF SUCCESSFUL CONVERGENCE FOR 100 TRIALS. THE LOW AND HIGH V_T IS RESPECTIVELY 0.5V AND 2V.

	M / V_T	M_{X1+}	M_{X1-}	M_{X2+}	M_{X2-}	M_{X3+}	M_{X3-}	M_{b+}	M_{b-}
$G_{init}=G_{min}$	low	0.02	1	0.02	1	0.99	1	0.95	0.06
	high	0	1	0	1	0.99	0.99	0.97	0.07
$G_{init}=G_{max}$	low	1	0.01	1	0.02	1	1	0.01	1
	high	1	0	1	0	1	1	0.02	1

2) *Gaussian distribution of V_T*

According to the learning conditions described in Eq. 7, V_T should meet both the following conditions to ensure successful learning:

$$V_L - V_{P-} < V_{T+} < V_H - V_{P-} \quad (8)$$

$$V_L - V_{P+} < V_{T-} < V_H - V_{P+} \quad (9)$$

If an error ξ is added to the V_T of memristor, the interesting thing to obtain the margin of error that neuron can tolerate. For example, if $V_{T+}/V_{T-}=V_{T0}+\xi/-V_{T0}+\xi$, $V_P=V_{T0}$, $V_H/V_L=V_i/-V_i$ (voltage of the input i), Eq. 10 can be obtained:

$$(8): \quad -V_i + V_{T0} < V_{T0}+\xi < V_i+V_{T0}$$

$$(9): \quad -V_i - V_{T0} < -V_{T0}-\xi < V_i-V_{T0}$$

$$-V_i < \xi < V_i \quad (10)$$

Eq. 10 describes the margin of error ξ, which is defined by the input voltage $-V_i$ and V_i. Consequently, if we want to improve the tolerance of threshold variation, we should increase the input voltage of NLB. Nevertheless, the input voltage is also limited by the operating condition (see III. B), in which the tolerated margin of error becomes as follows:

$$|V_i| < |V_T| \Rightarrow V_i < V_{T+} \ \& \ -V_i > V_{T-}$$

$$\Rightarrow V_i < V_{T0}+\xi \ \& \ -V_i > -V_{T0}-\xi$$

$$\Rightarrow \xi > V_i - V_{T0} \quad (11)$$

The error ξ of V_T is distributed randomly following a Gaussian form Eq. 12 (see Fig. 8).

$$\varphi(V) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(V-V_{T0})^2}{2\sigma^2}\right) \quad (12)$$

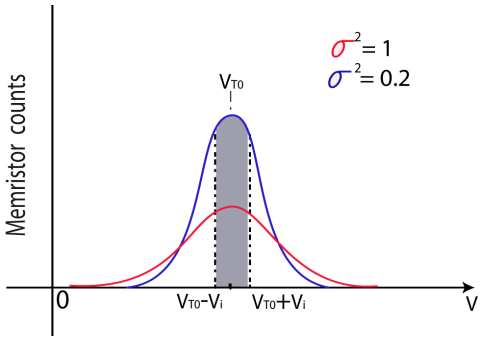

Figure 8. Gaussian distribution of V_T for the memristor crossbar array. The standard deviation value σ can be changed while keeping the same mean value V_{T0}. The gray zone shows the margin of V_T to ensure the intrinsic tolerance of threshold variation (see Eq. 10).

where σ is standard deviation. Based on the Eqs. 10-12, the probability of a functional memristor m denoted $P_{Success}(m)$ can be calculated by the Gaussian error function *erf(x)*:

- If we consider only the first condition Eq. 10:

$$P_{Success}(m) = \left(erf\left(\frac{V_i}{\sigma\sqrt{2}}\right)\right) \quad (13)$$

- If we combine the two conditions Eq.10-11 :

$$P_{Success}(m) = \left(erf\left(\frac{V_i}{\sigma\sqrt{2}}\right)\right) \times \left(\frac{1}{2} erfc\left(\frac{V_i - V_{T0}}{\sigma\sqrt{2}}\right)\right) \quad (14)$$

As shown in IV. A, several critical memristors are required to learn some functions by the neuron. If we consider N_m is the

necessary number to learn a function F. The probability of N_m functional memristors can be given by Eq. 15:

$$P_{Success}(N_m, m) = \left[P(m)\right]^{N_m} \quad (15)$$

Based on the Eqs. 13-15 the impact of deviation σ on successful learning rate can be plotted for N_m=3 (blue dotted curve in Fig. 9). To compare the theoretical calculation and simulation results with same condition (e.g. N_m); the NLB 8×1 was simulated to learn the function F_2. Gaussian distribution of V_T is implemented in all the memristors.

Fig. 9 (red solid curve) shows the simulation result of successful learning rate in function of standard deviation. The two curves are very close, which validates again our predictive model. From these curves, we can observe that the neuron successes 100% as σ≤0.3. For the values of σ greater than 0.3 the successful convergence starts to decrease and reach 90% for σ=0.4. The successful rate becomes lower than 10% from σ=0.4.

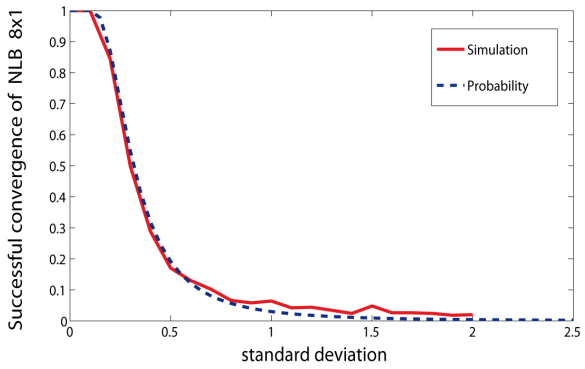

Figure 9. Successful convergence of NLB 8×1 to learn the function F2. $V_{T+}/V_{T-}=V_{T0}+\xi/-V_{T0}-\xi$, $V_{P+}/V_{P-}=V_{T0}/-V_{T0}=1v/-1v$, $V_H/V_L=V_i/-V_i=0.4/-0.4$. The solid curve represents the average rate of successful learning. The dotted curve corresponds to probability calculation of three functional memristors. For each value of σ the learning process was repeated 500 times.

V. Conclusion and Perspectives

In this paper, we presented a new logic block called Neural Logic Block (NLB) to build FTNA. It is based on a memristor crossbar array and capable of learning logic functions. An innovative learning approach for NLB, inspired from artificial neural network, is introduced in order to demonstrate its easy use and fault tolerance. Two types of defects, stuck-open and stuck-close and the threshold variability of memristor have been studied to investigate the fault-tolerant capability. For this purpose, predictive models have been developed to calculate the likelihood of successful learning in presence of certain rate of defect or variation. Simulations have been performed to demonstrate the intrinsic fault tolerance defect of NLB, and confirm the predictive models. These models will allow us to study the performances of the NLB with a large size without being limited by the simulation time-consuming.

The NLB architecture presents considerable tolerance against defects and V_T variation thanks to the intrinsic fault tolerance of neural network. It could be further improved by adding redundant neurons. Indeed, we showed in [27] that competitive strategy during the learning stage allows increasing significantly the robustness forecast for a full FTNA made of huge number of synapses and NLB. We show also in [28] that the fine grain FTNA and NLB architecture is well suited to implement arithmetic operators efficiently. This present work suggests its potential for use with massively defective technologies.

VI. Acknowledgement

The authors wish to acknowledge support from the French national project ANR PANINI.

VII. References

[1] S.Y. Chou et al., “Imprint of Sub-25 nm Vias and Trenches in Polymers”, Applied Physics Letters, Vol.67, pp. 3114-3116, 1995.

[2] Y. Huang et al., "logic gates and computation from assembled nanowire building blocks", Science, Vol.294, pp. 1313-1317, 2001.

[3] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, “The missing memristor found”, Nature, 453, pp. 80–83, Jan 2008.

[4] G. Agnus et al., “Two-Terminal Carbon Nanotube Programmable Devices for Adaptive Architectures”, Vol. 22, pp. 702-706, 2010.

[5] S.Y. Liao et al., “Design and Modeling of a neuro-Inspired Learning circuit using Nanotube-based memory devices”, IEEE Trans. Circuits and Systems I: Regular Papers, Doi.10.1109/TCSI.2011.2112590, 2011.

[6] W.S. Zhao et al., “Nanotube devices based crossbar architecture: toward neuromorphic computing”, Nanotechnology, Vol.21, 175202, 2010.

[7] Q. Xia, J. J. Yang, W. Wu, X. Li, R. S. Williams, “Self-Aligned Memristor Cross-Point Arrays Fabricated with One Nanoimprint Lithography Step”, Nano Letters, Article 10 (8), pp. 2909–2914, 2010.

[8] T. Hasegawa et al., “Learning abilities achieved by a single solid-state atomic switch”, Advanced Materials, Vol.22, pp. 1831-1834, 2010.

[9] C. Mead, “Neuromorphic electronic systems”, Proceedings of The IEEE, Vol.78, pp. 1629-1636, 1990.

[10] J. Von Neumann, “Probabilistic logics and the synthesis of reliable organisms from unreliable components”, Automata Studies, pp. 43–98, 1955.

[11] C. Mead, “Analog VLSI and Neural Systems. Reading”, MA: Addison-Wesley, 1989.

[12] K.M. Hynna, K. Boahen, “Thermodynamically equivalent silicon models of voltage-dependent ion channels”, Neural Computation, Vol.19, pp. 327-350, 2007.

[13] L.O Chua and S.M Kang, “Memristive devices and systems”, Proc. IEEE, Vol. 64, pp. 209–23, 1976.

[14] J. J. Yang, J. Borghetti, D. Murphy, D. R. Stewart, and R. S. Williams. A family of electronically reconfiguiable nanodevices. Adv Mater, 21, pp. 3754–3758, Jan 2009.

[15] Jo S.H et al, High-Density Crossbar Arrays Based on a Si Memristive System, Nano Letter 9, pp. 870-874, 2009.

[16] G.S Snider, “Spike-timing-dependent learning in memristive nanodevices”, Proc. of IEEE-NANOARCHI, pp. 85-92, 2008.

[17] D. Querlioz, O. Bichler and C. Gamrat, “A memristor-based spiking neural network immune to device variations,” International Joint Conference on Neural Networks, IJCNN 2011.

[18] J. H. Lee and K. K. Likharev, in Proc. WCCI/IJCNN 06, pp. 5026–5034, 2006.

[19] P. Jamieson and J. Rose, “A verilog RTL synthesis tool for heterogeneous FPGAs”, IEEE- International Conference on Field Programmable Logic and Applications, pp. 305–310, 2005.

[20] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, et W. Lu, “Nanoscale Memristor Device as Synapse in Neuromorphic Systems”, Nano Letters, vol. 10, no. 4, pp. 1297-1301, 2010.

[21] S. Jo, and W. Lu, “CMOS Compatible Nanoscale Nonvolatile Resistance Switching Memory”, Nano Lett. 8, pp. 392-397, 2008.

[22] D.W. Tank, J.J. Hopfield, "Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit, and 'a Linear Programming Circuit," IEEE Transactions on Circuits and Systems, vol. 33, No. 5, pp. 533-541, 1986.

[23] B. Widrow and M. Hoff. Adaptive switching circuits. WESCON convention record, pp. 96–104, 1960.

[24] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Washington, DC: Spartan Books, 1962.

[25] J. Borghetti et al, “Memristive switches enable stateful logic operations via material implication”, Nature 464, pp. 873-876, 2010.

[26] Z. Zhong, et al, “Nanowire Crossbar Arrays as Address Decoders for Integrated Nanosystems”, Science 302, pp. 1377, 2003.

[27] D. Chabi, J.O. Klein, "Hight fault tolerance in neural crossbar", Design and Technology of Integrated Systems in Nanoscale Era (DTIS), 5th International Conference on, 23-25 March 2010.

[28] D. Chabi, O. Brousse J.O Klein and M. Paindavoine, “Learning arithmetic operators in nanodevices”, submitted to IEEE Trans on Neural Network. April 2011.