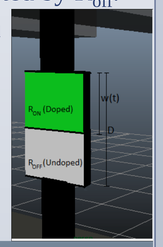
Memristor Crossbars for Parallel Stochastic Computing

Alvaro Velasquez and Sumit Kumar Jha
University of Central Florida
{velasquez, jha}@eecs.ucf.edu

Motivation: The separation of memory and processing units is a bottleneck in von Neumann architectures and therefore an impediment in big data applications where there is a constant need to access memory while computing. The use of memristor crossbars as unified memory and computation units shows a lot of promise by providing an alternative computing paradigm that resolves this bottleneck.

Memristor: A memristor consists of a film with a low-resistance (R_{on}) region containing ions that drift as current is applied. The undoped region of the film has a much higher resistance denoted by R_{off}. The total resistance (memristence) of a Memristor x at time t can be given by:

$$x = \frac{w(t)}{D}R_{on} + \left(1 - \frac{w(t)}{D}\right)R_{off}$$



$$M = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix}$$

Sneak Paths: ON memristors redirect current from one of its connected wire to the other. '1' denotes a turned-on memristor (low resistance) and '0' denotes a turned-off memristor (high resistance).

It has been shown that memristor switching can be represented by a Poisson dis.tribution [1] and is denoted by probability $P(t) = 1 - e^{-t/\tau}$, where t is the voltage pulse duration and τ is the characteristic switching time which is dependent on the magnitude of the voltage pulse V, and is given by $\tau(V) = \tau_0 e^{-V/V_0}$, where τ_0 and V_0 are fitting parameters.

Under a programming voltage of approximately $2V$, scientists at HP have demonstrated that Tantalum Oxide fast-switching memristors have a 1.5 ns characteristic switching time [2]. We use this memristor in the analysis in the following section.

We have proposed a method for synthesizing memristor crossbars for digital computing using sneak paths in [3]. A simple result from said work is the following crossbar for the sum bit $T = (A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) \vee (A \wedge B \wedge C)$. Applying a voltage pulse on the top row wire and grounding the bottom row yields the result when the voltage on the bottom row is measured.

$$M = \begin{pmatrix} \neg B & \neg A & A \\ 1 & \neg C & C \\ B & A & \neg A \end{pmatrix}$$

$A = 0, B = 0, C = 0$
$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

$A = 0, B = 0, C = 1$
$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

$A = 0, B = 1, C = 0$
$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

$A = 0, B = 1, C = 1$
$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

$A = 1, B = 0, C = 0$
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$

$A = 1, B = 0, C = 1$
$\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

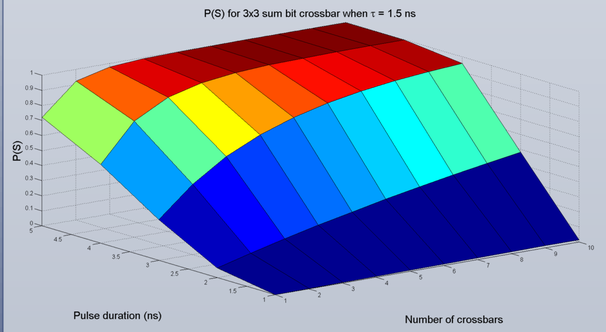
$A = 1, B = 1, C = 0$
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$

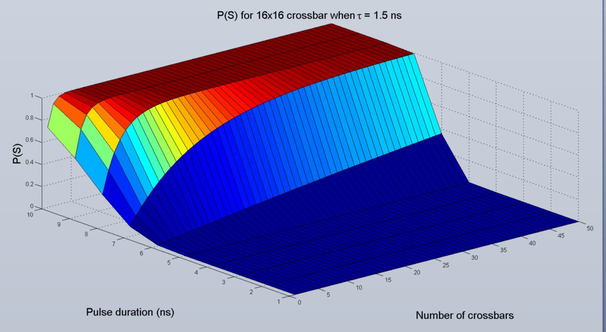
$A = 1, B = 1, C = 1$
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$

Since nanoscale memristor crossbars are susceptible to stochastic noise and computation on each memristor crossbar is only guaranteed to be probabilistically correct, it is reasonable that multiple memristor crossbars can be used to perform the same computation in parallel, thereby amplifying the probability that a correct answer is produced to any desired value.

Given k $m \times n$ memristor crossbars $M_i, i = 1, \ldots, k$, we can derive the following equations, where $P(S)$ denotes the probability of successfully configuring at least one of the crossbars .

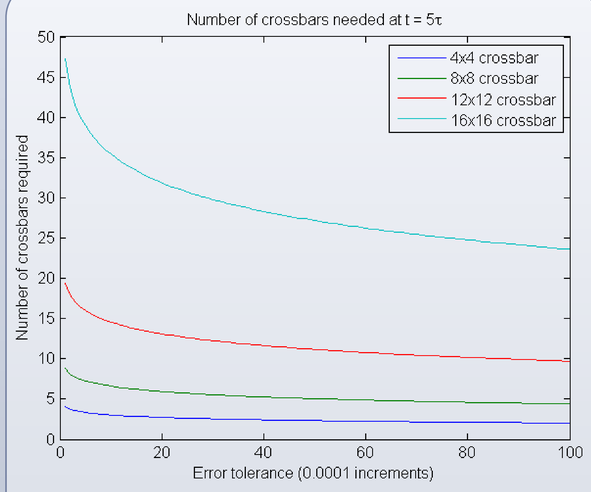
$$P(M_i) = (1 - e^{-t/\tau})^{mn}$$
$$P(\neg M_i) = 1 - (1 - e^{-t/\tau})^{mn}$$
$$P(\bigvee_{i=1}^{k} M_i) = 1 - P(\bigwedge_{i=1}^{k} \neg M_i) = 1 - (1 - (1 - e^{-t/\tau})^{mn})^k = P(S)$$

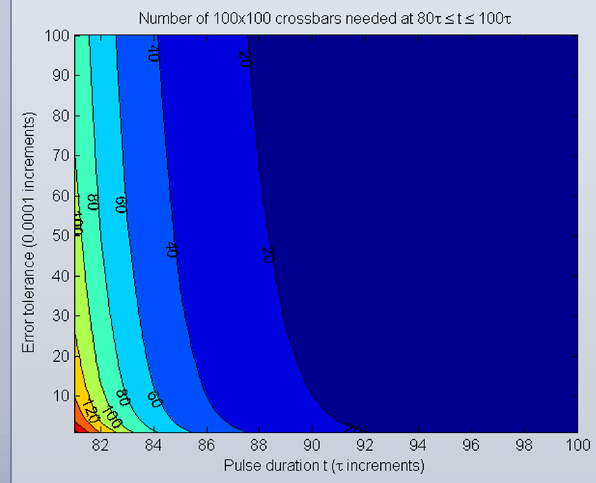




The graphs above are based on the memristor presented in [2]. We can determine how many crossbars are required in order obtain the success probability $P(S)$ to a desired degree. We denote the error tolerance by ϵ. Then we derive the following lower bound on the number of crossbars k.

$$P(S) \geq 1 - \epsilon \implies k = \lceil log_{(1-(1-e^{-t/\tau})^{mn})}\epsilon \rceil$$
$$= \lceil ln(\epsilon)/ln(1 - (1 - e^{-t/\tau})^{mn}) \rceil$$





FUTURE WORK

We will explore the avenue of memristor-based big data applications by expanding on the work presented herein and in [3].

REFERENCES

[1] S. Jo, K. Kim, and W. Lu, "Programmable resistance switching in nanoscale two-terminal devices," *Nano Lett.*, vol. 9, no. 1, pp. 496–500, 2008.

[2] F. Miao, J. P. Strachan, J. J. Yang, M.-X. Zhang, I. Goldfarb, A. C. Torrezan, P. Eschbach, R. D. Kelley, G. Medeiros-Ribeiro, R. S. Williams, 'Anatomy of a Nanoscale Conduction Channel Reveals the Mechanism of a High-Performance Memristor," Advanced Materials, vol. 23, no. 47, pp. 5633-5640, Nov. 2011.

[3] Alvaro Velasquez and Sumit Jha. Automated synthesis of crossbars for nanoscale computing using formal methods, 2015. NANOARCH 2015, July 8 – 10, Boston, MA.