**Matrix-to-Vector Multiplication Leveraging RRAM Crossbar**

Matheus Farias1

1Harvard School of Engineering & Applied Sciences (SEAS), Cambridge, MA, USA.

matheusfarias@g.harvard.edu

***Abstract***— Matrix-to-Vector multiplications (MVM) are basis for several applications in modern applications such as neural networks, computer vision and digital signal processing. In this work, I design, with XFAB 130nm PDK technology, an accelerator for 3x3 MVM based on the Resistive RAM (RRAM) crossbar architecture. **RESULTS.** This work enables ground to study directions on how to decrease the energy consumption of A/D and D/A conversions on the architecture, as I configured the whole setup to simulate it.

# I. Introduction

After many years of the proposal of Von Neumann’s architecture for computers [1], the technology used for designing processors had a huge improvement, however, memories did not follow the same trend for performance purposes [2]. In this architecture, the processor must acquire data from the memory which is physically located outside the processor’s chip. Thus, even though the processors are fast, as we always need to acquire data from the memory, we face an overhead called Von Neumann’s bottleneck. See in **Figure 1** a comparison of different operations for a CPU designed with 45nm CMOS technology. To overcome this situation, researchers are studying specific-domain architectures that can leverage processing in memory to fasten operations and decrease the energy consumption such as the Resistive RAM (RRAM) crossbar architecture [4]. In this work I study how to use this architecture to perform MVM in the analog domain as a faster alternative way compared to CPU.

# II. Background and Methodology

Here I consider a specific application of multiplying a vector to a matrix . To perform this multiplication in a RRAM crossbar architecture, we need to define multiply-and-accumulate operations, which in this case are done by using Kirchhoff and Ohm’s laws of circuits analysis theory as shown in **Figure 2**. First, I consider a case that each cell in the crossbar corresponds to a single and fixed resistor to shown that the architecture works as expected, we call this arrangement as 1R (see **Figure 3**). Then, a next step is to replicate the model now using cells composed of one transistor and one resistor (1T1R cells). This way we can control the current that flows through the cells by activating/deactivating the transistor depending on the difference between the gate and source voltage. Finally, I use cells with 1T1R in parallel with the RRAM alongside with a transistor to activate/deactivate the current to this path. RRAM is a non-volatile memory based on memristors (resistors that have memory properties). We can control the resistance of these resistors depending on the voltage input and the time this voltage input is acting on the resistive cell (see **Figures 4 and 5**).

# III. Technical Approach

To design these three circuits, I used XFAB 130nm PDK (xr013) [5]. For the 1R cells, I used resistors of 1 kΩ, 2 kΩ and 3 kΩ (ranging from 500 nm to 1.2 µm of width/length). For the 1T1R cells, I considered NMOS transistors with total width of 4 µm and length 120 nm. Instead of using only one resistor for each cell, I considered a model of 4-bit digital resistors, which means that we can sweep over 16 possible resistance configurations by turning the transistors ON/OFF for each group of resistors (**see Figure 6**). This was made so that we could perform MVM with different matrix *B* and knowing exactly what is going to be the resistance of each cell. Finally, I considered the model attaching a RRAM (see **Figure 7** for the parameters of the RRAM model used). The simulations were made in Spectre and Python and all the code is provided in [6].

# III. Key Results (you can change this title)

Describe the quantifiable results and how you achieved them here. E.g., We ran the DAC at a clock frequency of 100 MHz, since that is the nominal clock frequency of the processor core driving it. The average power consumption is however many Watts. The leakage power accounts for whatever percentage of the total power. Does your design meet the high-level specs that you set out to meet? Could it be used for the application that you intended? How do you know? Which one of your figures demonstrates that you can meet the specifications?

# IV. Conclusion

Summarize the main focus of the work… “we have demonstrated…”. It achieves whatever performance metric, which meets the specifications for whatever application. Describe the future directions that are enabled by the groundwork you have created, for you to pursue, or for others that may be interested in picking up on your work and continuing where you left off.

##### **Acknowledgments**

I would like to thank Prof. Gage Hills for providing the RRAM model as well as for our insightful discussions.

##### **References**

1. J. von Neumann, “First draft of a report on the EDVAC,” in IEEE Annals of the History of Computing, vol. 15, no. 4, pp 27-75, 1993, doi:10.1109/85.238389.
2. Mutlu, O. Ghose, S., Gómez-Luna, J. & Ausavarungnirun, R.. (2020). A Modern Primer on Processing in Memory.
3. Mark Horowitz. Energy table for 45nm process, Stanford VLSI wiki.
4. I. Chakraborty et al., "Resistive Crossbars as Approximate Hardware Building Blocks for Machine Learning: Opportunities and Challenges," in Proceedings of the IEEE, vol. 108, no. 12, pp. 2276-2310, Dec. 2020, doi: 10.1109/JPROC.2020.3003007.
5. X-FAB Silicon Foundries. https://www.xfab.com/technology/rf.

Graphical user interface

Description automatically generated