Memristor Hardware Accelerator of Quantum Computations

Iosif-Angelos Fyrigos*, Vasileios Ntinas*†, Georgios Ch. Sirakoulis*, Panagiotis Dimitrakis‡, Ioannis Karafyllidis*
*Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece
†Department of Electronic Engineering, Universitat Polytécnica de Catalunya, Barcelona, Spain
‡Institute of Microelectronics, NCSR "Demokritos", Athens, Greece

***Abstract*—Quantum computing and quantum computers are a major part of the second quantum revolution. Existing quantum algorithms can natively solve complex problems, such as the prime number factorization and searching of unstructured databases, in a fast and efficient way. The main obstacle towards building large and efficient quantum computers is decoherence, which produces errors that have to be continuously corrected using quantum error correcting codes. Beyond the realisation of quantum computing systems with actual quantum hardware, quantum algorithms have been developed based on quantum logic gates that can be described and utilised by classical computers and proper interfaces based on linear algebra operations. Furthermore, memristive grids have been proposed as novel nanoscale and low-power hardware accelerators for the time-consuming matrix-vector multiplication and tensor products. In this work, given that for quantum computations simulation, the matrix-vector multiplication is the dominant algebraic operation, we utilize the unprecedented characteristics of memristive grids to implement circuit-level quantum computations. Since all quantum computations can be mapped to quantum circuits, memristive grids can also be used as efficient quantum simulators, as classical/quantum interfaces and also as accelerators in mixed classical-quantum computing systems.**

I. Introduction

Quantum computers are suitable for various problems that the power of classic computing systems is not sufficient enough to provide practical solutions for real-life applications. Problems such as simulating quantum mechanical processes in physics, chemistry and biology, performing database search [1] and prime number factorization [2] can be efficiently solved utilizing the unprecedented quantum properties, i.e. state superposition and quantum entanglement [3]. To be able to investigate further various problems that quantum computers could provide us with an efficient solution, an important step is the development of quantum simulators [4]. In quantum simulators qubits are represented by vectors in Hilbert space and quantum gates are Hilbert space operators represented by matrices. The effect of the Quantum gates on qubits and quantum registers is described by matrix-vector multiplications.

A novel device named Memristor [5] is able to perform matrix-vector multiplication in a crossbar configuration. Memristor is characterized by its non-volatility, low dimensions and low power consumption. As a circuit element, Memristor corresponds to a two terminal resistor with variable resistance, controlled by the applied signal and depended on its states history. Based on the multilevel resistance characteristic of many memristor devices and the cross-point structure, memristor crossbar array can use the input voltage signal as the vector data and save the matrix data into the memristor cells, which realizes matrix-vector multiplication efficiently with O(1) time complexity merging all cells current in each row through Ohm's law [6], [7], [8]. Taking advantage of the reprogrammable nature of the memristor, different Quantum gates can be implemented utilizing the same crossbar.

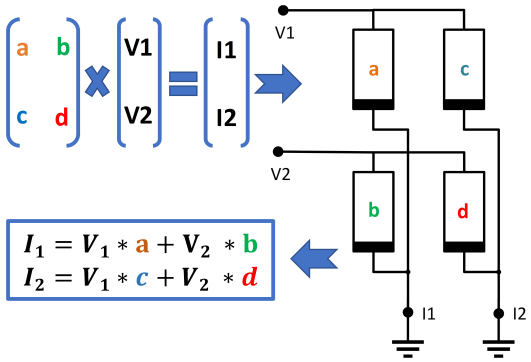

Fig. 1. Crossbar Vector-Matrix Multiplication Example

In this work exploiting the beforementioned function of the memristor crossbar, implementation of a universal set of quantum gates has been developed. The suitability and functionality of a differential memristor crossbar configuration has been tested successfully for the purpose of implementing the Hadamard and CCNOT gate.

II. Quantum Gates and their Matrix Representation

The representation of a qubit for quantum computing simulations can be performed by a column vector:

$$= \alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (1)$$

where $|0\rangle$ and $|1\rangle$ are the basis vectors and α and β the corresponding probability amplitudes, where $|\alpha|^2 + |\beta|^2 = 1$ has to be fulfilled. In the same manner, the action of quantum gates on qubits can be described by the corresponding matrix and performed through a matrix-vector multiplication. For

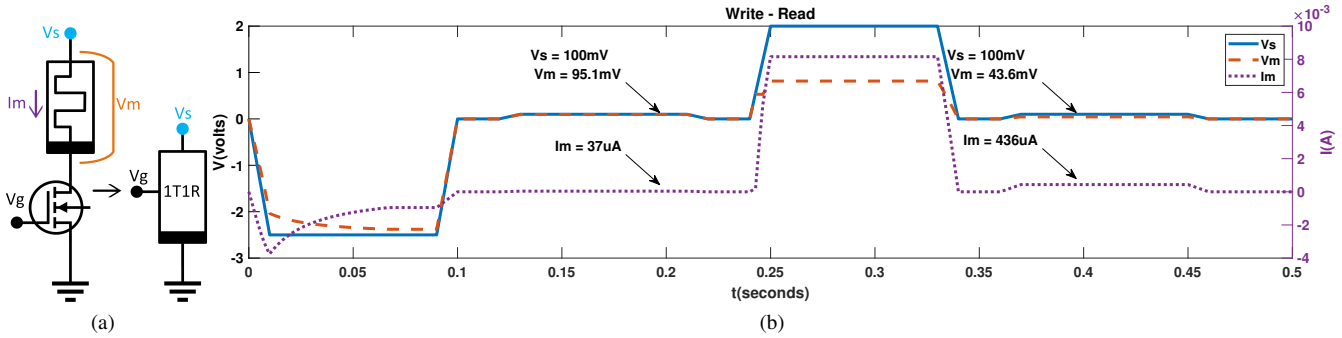

Fig. 2. (a) Transistor in Series with Memristor schematic (b) Write - Read operation of R_{off} and R_{on}

example, the action of a Hadamard gate on a qubit, which maps $|0\rangle \rightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|1\rangle \rightarrow \frac{|0\rangle-|1\rangle}{\sqrt{2}}$, can be performed as:

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \quad (2)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \quad (3)$$

III. Quantum Gates in Memristor Crossbar

An example of matrix-vector multiplication is demonstrated in Fig. 1. The values of matrix elements are mapped on memristor conductance values (Siemens) so the current of each memristor is calculated according to Ohm's law ($I = V \times G$). Subsequently, the currents flowing through the memristors of each column are added together when they reach the ground according to Kirchhoff's law. Consequently, the output currents I_1, I_2 are the result of the matrix vector multiplication and constitute the output vector.

Utilizing the aforementioned matrix-vector multiplication function of the memristor crossbar, a universal set of quantum gates has been emulated using memristive grids. Memristors despite being multi-state devices, programming them to multiple discrete resistance states can end up a challenging task. Binary programming utilizing only R_{on} and R_{off} state is easier and more accurate. A universal set of quantum gates capable of being implemented through binary memristor crossbar arrays was found in [9], [10]. This set comprises of the Hadamard and CCNOT quantum gates. Hadamard gate is able to create the superposition of basis states while CCNOT can evolve the superposition state to an entangled state. The only missing functionality in this set is the generation of imaginary numbers. Bernstein and Vazirani [11] showed that, by adding one extra qubit to the circuit, only real valued matrices are required to cover all the quantum computing operations. The additional qubit indicates whether the systems state is in the real or imaginary part of the Hilbert space.

IV. Simulation Results

In this work, two differential 1T1R crossbars have been simulated that implement the Hadamard and CCNOT gate accordingly. The programming process of the crossbar, as well the matrix-vector multiplication (computation) process are demonstrated here. All simulations were held in Cadence's Virtuoso Suite.

Memristor model. The simulated crossbar consists of 1T1R (transistor in series with memristor) devices shown in Fig. 2a. For our simulations the VTEAM memristor model was used [12] while the transistor was simulated by the BSIM3V3 transistor model [13]. VTEAM is a flexible model that has been proven to effectively fit a wide variety of fabricated memristors. It is easily configurable and Verilog-A compatible.

Utilizing the VTEAM model, an already fabricated Pt-Hf-Ti memristor device [14] was simulated, while the fitting parameters were adopted from [12]. The write as well as the read process of the memristor can be examined in Fig. 2b. The memristor is driven to R_{off} and R_{on} state by applying negative ($V_{RST} = -2.5$V) and positive ($V_{SET} = 2$V) pulses, respectively with a pulse width of 100ms, while rising and falling edge duration was set to $\tau_{r/f} = 10$ ms. The memristor is read by applying $V_{READ} = 100$mV. The output current I_m is 37uA during R_{off} and 436uA during R_{on} as it is shown in Fig. 2b.

Hadamard Gate. The first quantum gate of the universal set to be implemented in the crossbar is the Hadamard Gate which matrix representation can be seen below:

$$H = \frac{1}{\sqrt{2}} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (4)$$

The crossbar equivalent of the 1-qubit Hadamard gate is presented in Fig. 3a and consists of a 2x4 memristor configuration and 2 differential amplifiers (Differential Crossbar [15]). To incorporate the representation of negative values of the matrix into the crossbar array, using binary programming, twice the number of columns are needed. In this way, one column is connected to the positive input of the differential amplifier and the other to the negative, corresponding to the positive and negative value of the matrix. Specifically, when both memristors are in the same state the inputs of the differential

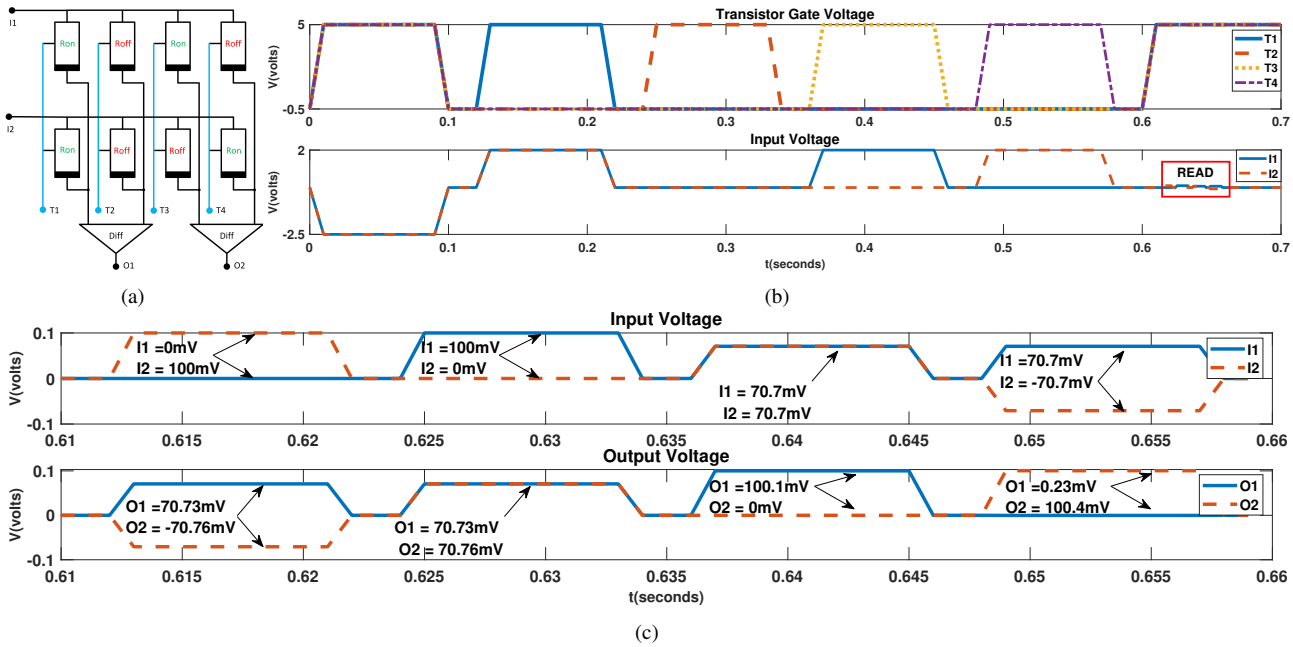

Fig. 3. (a) Hadamard Gate Crossbar (b) Hadamard Programming Process (c) Hadamard Computation (READ) Process

amplifiers are mutually negated, mapped to 0. In order to tune the crossbar array value to 1 (-1), the memristor connected to the positive (negative) input of the differential amplifier has to be in the R_{on} state, while the other to the R_{off}.

Initially the crossbar must be configured through the programming process to represent a Hadamard gate. The exact state of each memristor of the crossbar is depicted in Fig. 3a. The simulation results of the programming process are presented in Fig. 3b. During this process all the memristors are initialized in R_{off} state by applying a negative voltage V_{RST} to every row (I_1, I_2) while the selector-transistors are switched on by applying $V_{SEL} = 5$V to their gates $(T_1 - T_4)$. The crossbar then is programmed column by column. Aiming to switch the memristors of the first column towards R_{on}, T_1 is selected by applying V_{SEL} and V_{SET} is applied to I_1 and I_2. For the second column both memristors must stay in R_{off} so no input voltage is applied from I_1, I_2. For the third column V_{SEL} is applied at T_3 and V_{SET} at I_1 leading to the switch of the upper memristor. Finally for the forth column V_{SEL} is applied at T_4 and V_{SET} at I_2. The programming process occurs only once and is only needed again in case we want to reprogram the same crossbar into a different quantum gate.

During the matrix-vector multiplication process (computation process) all memristors contribute to the calculation so that every column of the crossbar is selected by applying V_{SEL} to $(T_1 - T_4)$. The qubit state is applied as a voltage signal in I_1 and I_2 while the output voltages of the differential amplifiers (O_1 and O_2) are the final result of the calculation and represent the new state of the qubit. The input voltages as well as the output voltages range between $-100mV$ and $100mV$ to map appropriately the qubit basis state coefficients which range between -1 and 1. Use of higher voltage as input signal was avoided to make certain that the memristors do not switch during computation phase and additionally maintaining the power consumption to low levels. The pulse width of the input voltages was reduced to 8ms compared to the 80ms of the programming process. The simulation of the computation process can be seen at the end of Fig. 3b marked with a red square and labelled READ while a zoomed more detailed depiction of the computation process is shown in Fig. 3c. Four different input combinations were tested and the output voltages showcase the correct operation of the memristor crossbar as a Hadamard gate.

Tofoli (CCNOT) Gate. The second Quantum Gate of the universal set that has been implemented in the crossbar grid is the CCNOT Gate, the action of which is represented by the following matrix:

$$CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5)$$

Following the same methodology of Programming-Computing, the CCNOT gate has been implemented and tested on an 8x16 memristor crossbar depicted in Fig. 4a. This gate has only 0 and 1 values, therefore there is no need to map negative values on the memristor crossbar and as such, half of the crossbar's columns remained unaltered during the programming process. The complete programming process is shown in Fig. 4b. Finally in Fig. 4c the simulation of

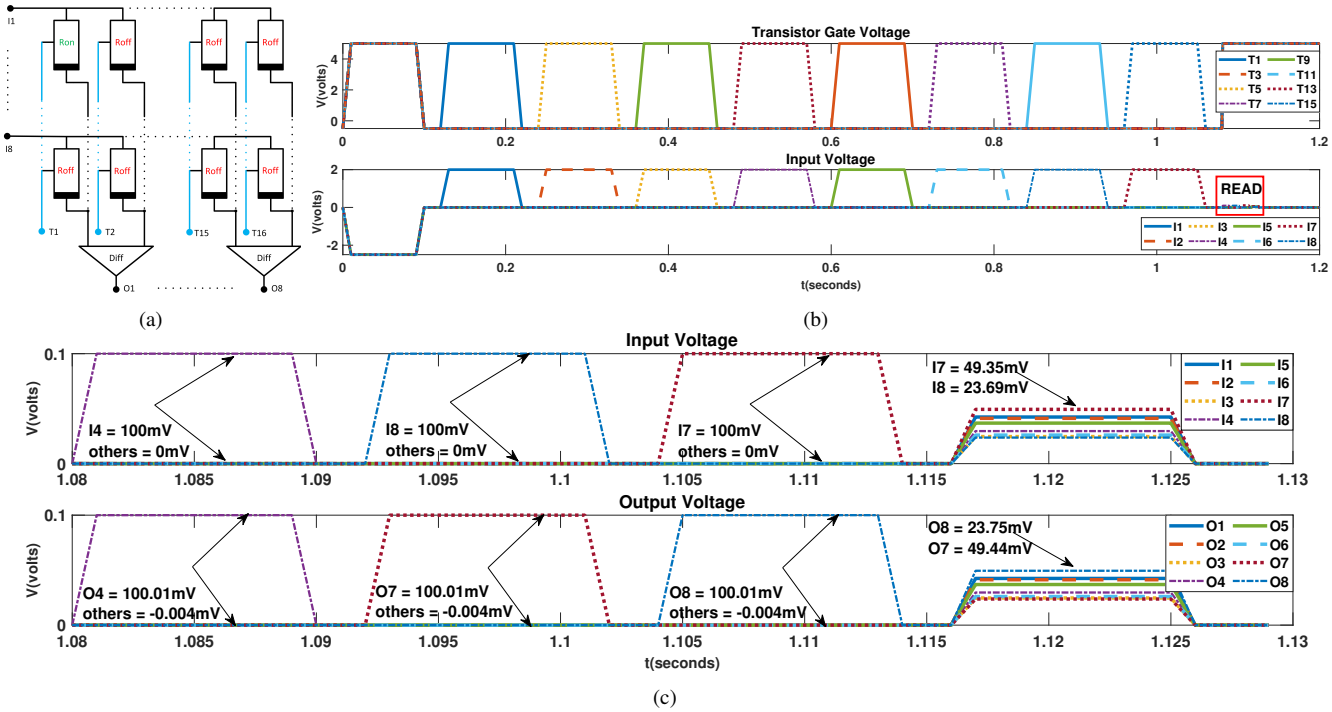

Fig. 4. (a) CCNOT Gate Crossbar (b) CCNOT Programming Process (c) CCNOT Computation (READ) Process

the computation process is demonstrated. Four different input combinations were tested. As it is shown there is reversion of the outputs O_7, O_8 only when at least one of the inputs I_7, I_8 is present. This behavior reflects the proper operation of the CCNOT gate.

V. Conclusions

In this work an approach to emulate quantum gates utilizing reprogrammable memristor crossbars was investigated. More specifically, given that for quantum computations, the matrix-vector multiplication is the dominant algebraic operation, we utilized the unprecedented characteristics of memristive grids to implement circuit-level quantum computations. Hadamard and CCNOT, which comprise a universal quantum set for real probablility amplitudes, were mapped successfully on memristor crossbars and their correct operation was demonstrated. The simulation results presented enable us to further explore the original idea of using memristor devices for hardware acceleration of quantum computations in our future works.

Acknowledgment

We gratefully acknowledge the Greece-Russia bilateral joint research project MEM-Q (proj.no./MIS T4ΔPΩ-00030/5021467) supported by GSRT and funded by National and European funds.

References

[1] L. K. Grover, "A fast quantum mechanical algorithm for database search," *arXiv preprint quant-ph/9605043*, 1996.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[3] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[4] T. H. Johnson, S. R. Clark, and D. Jaksch, "What is a quantum simulator?" *EPJ Quantum Technology*, vol. 1, no. 1, p. 10, Jul 2014.

[5] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, p. 80, 2008.

[6] M. Hu, J. P. Strachan, Z. Li, E. M. Grafals, N. Davila, C. Graves, S. Lam, N. Ge, J. J. Yang, and R. S. Williams, "Dot-product engine for neuromorphic computing: Programming 1t1m crossbar to accelerate matrix-vector multiplication," in *Proceedings of the 53rd annual design automation conference*. ACM, 2016, p. 19.

[7] L. Xia, P. Gu, B. Li, T. Tang, X. Yin, W. Huangfu, S. Yu, Y. Cao, Y. Wang, and H. Yang, "Technological exploration of rram crossbar array for matrix-vector multiplication," *Journal of Computer Science and Technology*, vol. 31, no. 1, pp. 3–19, 2016.

[8] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nature Electronics*, vol. 1, no. 1, p. 52, 2018.

[9] D. Aharonov, "A simple proof that toffoli and hadamard are quantum universal," *arXiv preprint quant-ph/0301040*, 2003.

[10] Y. Shi, "Both toffoli and controlled-not need little help to do universal quantum computation," *arXiv preprint quant-ph/0205115*, 2002.

[11] E. Bernstein and U. Vazirani, "Quantum complexity theory," *SIAM Journal on computing*, vol. 26, no. 5, pp. 1411–1473, 1997.

[12] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, "Vteam: A general model for voltage-controlled memristors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 8, pp. 786–790, 2015.

[13] Y. Cheng and C. Hu, *MOSFET modeling & BSIM3 users guide*. Springer Science & Business Media, 1999.

[14] E. Yalon, A. Gavrilov, S. Cohen, D. Mistele, B. Meyler, J. Salzman, and D. Ritter, "Resistive Switching in HfO_2 Probed by a Metal–Insulator–Semiconductor Bipolar Transistor," *IEEE Electron Device Letters*, vol. 33, no. 1, pp. 11–13, 2011.

[15] T. M. Taha, R. Hasan, and C. Yakopcic, "Memristor crossbar based multicore neuromorphic processors," in *2014 27th IEEE International System-on-Chip Conference (SOCC)*. IEEE, 2014, pp. 383–389.