Programmable Memristive Threshold Logic Gate Array

Olga Krestinskaya
Department of Electrical and Computer Engineering
Nazarbayev University, Astana
Email: ok@ieee.org

Akshay Kumar Maan
Queensland Microelectronic Facility
Griffith University
Queensland 4111, Australia
Email: akshay.maan@gmail.com

Alex Pappachen James
Department of Electrical and Computer Engineering
Nazarbayev University, Astana
Email: apj@ieee.org

Abstract—This paper proposes the implementation of programmable threshold logic gate (TLG) crossbar array based on modified TLG cells for high speed processing and computation. The proposed TLG array operation does not depend on input signal and time pulses, comparing to the existing architectures. The circuit is implemented using TSMC 180nm CMOS technology. The on-chip area and power dissipation of the simulated architectures [1], [4] depend on time and inputs, comparing to the existing architectures [1], [4]. Therefore, in this paper, we propose the implementation of flexible programmable TLG array for high speed processing, where the memristor state is independent on the time and input to TLG cell.

There are several works that show the implementation of memristive TLG cell and TLG array is an open problem that attracted a lot of interest in recent years [1], [2], [3], [4]. The memristive TLG array is an alternative solution for FPGA based processing that may allow to speed up computation and ensure small on-chip area and low power consumption [4]. Therefore, in this paper, we propose the implementation of flexible programmable TLG array for high speed processing, where the memristor state is independent on the time and input to TLG cell.

The contributions of this work are the following:

- We implement the programming circuit to update the threshold logic cell and analyze the limitations of non-ideal single transistor switches and their effect on the performance of the cell.
- We present the design of flexible programmable TLG array with the proposed TLG cells and programming circuits that can be used for large-scale complex multiple threshold logic operations.

I. INTRODUCTION

The implementation of memristive TLG cell and TLG array is an open problem that attracted a lot of interest in recent years [1], [2], [3], [4]. The memristive TLG array is an alternative solution for FPGA based processing that may allow to speed up computation and ensure small on-chip area and low power consumption [4]. Therefore, in this paper, we propose the implementation of flexible programmable TLG array for high speed processing, where the memristor state is independent on the time and input to TLG cell.

There are several works that show the implementation of memristive TLG [5], [6], [3], [2], [2] and memristive TLG arrays architecture [1], [4]. However, most of the existing designs are time-dependent, where the outputs are obtained in a sequence of pulses [1], [3] or memristor state depends on the inputs and should be reprogrammed while processing [7]. Depending on the memristor material, the time to update a resistance of a memristor can vary from micro-seconds to seconds, and high voltage pulses are required to update the memristor value in a short time period [8], which cause the increase in power consumption. Therefore, the minimization of the number of pulses required for memristive TLG operation is an important issue in memristive TLG design [9]. To reduce the power dissipation required for processing and to speed up the computation, this paper proposes the implementation of programmable TLG array, where the operation does not depend on time and inputs, comparing to the existing architectures [1], [4].

The contributions of this work are the following:

- We proposed a modified implementation of threshold logic gates shown in [7], which have proven to be effective for various applications [10], [11]. The modified TLG cell in independent on time and inputs.

II. MODIFIED THRESHOLD LOGIC GATES

A. Threshold logic gates cell

Figure 1 (left side) illustrates the implementation of TLG cell inspired from [7]. The architecture is similar to the previously proposed TLG cell [7]; however, the operating principle is modified. In previously proposed TLG cell, the memristor $R_{c2}$ and control signal $V_c$ is updated during the cell processing and change according to the input signal, which limits the processing speed, especially when the time required to update the memristor is large. We proposed to implement unified time-independent TLG cell that can be easily integrated into TLG array, where memristors can be preprogrammed during the single step programming cycle, and the resistance and control voltage should not be changed based on the input. To implement time independent TLG, we modified the transistor parameter and control voltage signal. The threshold of the inverter $M3−M4$ is lowered by adjusting $W/L$ ratio of the transistors. New transistor parameters are the following: $M1 = 0.72\mu/0.18\mu$, $M2 = 0.36\mu/0.18\mu$, $M3 = 0.72\mu/0.18\mu$, $M4 = 20.36\mu/0.18\mu$. The memristors $R_1$, $R_2$, $R_3$ and $R_4$ are always programmed to high resistance $R_{off}$. The parameters of control memristors $R_{c1}$ and $R_{c2}$ that control the operation of TLG are shown in Table I. The memristor resistances used in this paper are $R_{on} = 3k\Omega$, $R_{off} = 60k\Omega$.

| Configuration | $R_{c1}$ | $R_{c2}$ | $V_c$ |
|---------------|---------|---------|------|
| NAND          | $R_{on}$ (low) | $R_{off}$ (high) | 0.8 V |
| NOR           | $R_{off}$ (high) | $R_{on}$ (low) | 0.8 V |
| XNOR          | $R_{off}$ (high) | $R_{on}$ (low) | 0.8 V |

TABLE I
PARAMETERS OF CONTROL MEMRISTORS $R_{c1}$ AND $R_{c2}$ AND CONTROL SIGNAL $V_c$ FOR MODIFIED TLG CELL
B. Programming circuit

The implementation of the programming circuit for the proposed TLG cell is shown in Fig. 1 (right side). The switches for read and write operation are implemented as single NMOS transistors. We use single transistor switch to keep on-chip area of the programming circuit small. Transistors $S_r$ are responsible for the read operation, and transistors $S_w$ are involved in the write operation and update of the value of the memristors. The $W/L$ ratio of the read and write transistors is $M = 10.36\mu/0.18\mu$. The width is increased to ensure linear performance.

This switch configuration is not ideal and the complete ON or OFF state is not possible. The non-ideal behavior of the switch affects the performance of the cell and voltage drop occurs. Therefore, we adjust $W/L$ ratio of the transistor $M2 = 5.36\mu/0.18\mu$, which improves the performance of NMOS transistor for 180nm CMOS technology. However, this parameter should be adjusted based on the design technology.

To turn on the read transistors the voltage $1V$ for $S_{r1}$-$S_{r2}$ and $0.7V$ for $S_{r3}$-$S_{r4}$ is applied to the transistor gate. This voltage also causes the particular voltage drop, which is compensated by lowering the threshold of the inverter $M1$-$M2$, comparing to the cell without programming circuit. Both read and write operations are controlled by $V_{sc}$, which is either $2.5V$ or $0V$ for ON state and OFF state, respectively.

To update the memristors from high to low state, the $S_{w4}$ and $S_{w7}$ for $R_{c1}$ and $S_{w10}$ and $S_{w13}$ for $R_{c2}$ should be in ON state. For changing memristor state from low to high, the $S_{w5}$ and $S_{w6}$ for $R_{c1}$ and $S_{w11}$ and $S_{w12}$ for $R_{c2}$ should be in ON state. The update process of both memristors can be performed in parallel, as the switches in programming circuit allow to separate the control memristors.

III. THRESHOLD LOGIC GATE ARRAY ARCHITECTURE

A. Cell arrangement in threshold logic gate crossbar array

The proposed TLG array architecture is shown in Fig. 2. Each cell can be programmed to perform XNOR, XOR or NAND operation. The switching between rows and columns is performed by single transistor switches with $W/L = 10.36\mu/0.18\mu$. Each row consists of two lines corresponding to a particular pair of inputs to the gates, which should be processed together. The inputs can be either connected or disconnected depending on the configuration. Each row corresponds to the particular inputs, this approach ensures the flexibility of the array programming, when the number of operations performed with the same pair of inputs is large. The outputs of the TLG cells are connected to two columns;
this allows the number of possible configurations in the connection of neighboring cells. The number of lines in each row and column may be increased, if more complex functionality is required or the high array density should be achieved.

As in non-ideal switches the voltage drop may effect the performance and outputs of the TLG cells, we introduce the thresholding block consisting of two inverters with the following parameters: $M5 = 0.72 \mu/0.18 \mu$, $M6 = 20.36 \mu/0.18 \mu$, $M7 = 0.72 \mu/0.18 \mu$, and $M8 = 0.36 \mu/0.18 \mu$. The threshold of this circuit is about 0.4 V. This reduces the effect of voltage drop across the switches to the array performance. The array is flexible in programming, and switches allow the routing of the TLG cell outputs in different directions.

B. Cell modification to improve TLG array scalability

To improve ensure scalability of TLG array and reduce on-chip area of the programming circuit, the voltage control approach with single programmable memristor $R_{c2}$ and changing $V_c$ can be used. The truth table for this approach is shown in Table I. The memristors $R_1$, $R_2$ and $R_{c1}$ and inverter $M1 - M2$ (Fig. 1) can be separated from the architecture and switches corresponding to the programming of $R_{c2}$ can be removed. The modified cell architecture with the control of all lines is shown in Fig. 3. The programming of $R_{c2}$ can be performed by row and column transistors.

The architecture is also appropriate for modular approach, when the TLG array’s cells are arranged into smaller arrays. This approach is useful when the processed data has a large number of various combinations of inputs that should be processed separately. In addition, the other possibility to use TLG cell illustrated in Fig. 3 is to preprogram the cells and control the functionality of the cell selecting particular rows and columns.

![Fig. 3](image)

**Fig. 3.** Modification of the cell for programming circuit reduction and improvement of array arrangement.

**Fig. 4.** Simulation of the proposed TLG cell with programming circuit.

| Configuration | $R_{c1}$ | $R_{c2}$ | $V_c$ |
|---------------|---------|---------|-------|
| NAND          | $R_{off}$ (high) | $R_{off}$ (high) | 1 V   |
| NOR           | $R_{off}$ (high) | $R_{off}$ (high) | 0.8 V |
| XNOR          | $R_{off}$ (high) | $R_{on}$ (low)  | 0.8 V |

| AVAILABLE S SIGNALS FOR TLG CELL WITH A SINGLE PROGRAMMABLE MEMRISTOR AND REDUCED PROGRAMMING CIRCUIT.

| Configuration | $R_{c1}$ | $R_{c2}$ | $V_c$ |
|---------------|---------|---------|-------|
| NAND          | $R_{off}$ (high) | $R_{off}$ (high) | 1 V   |
| NOR           | $R_{off}$ (high) | $R_{off}$ (high) | 0.8 V |
| XNOR          | $R_{off}$ (high) | $R_{on}$ (low)  | 0.8 V |

**IV. SIMULATION RESULTS AND DISCUSSION**

For this simulation, we used modified S memistor model shown in [12] and TSMC 180nm CMOS technology. Also, we use underdrive inverters in the design with $V_{DD} = 1V$ to ensure lower inverter threshold. The transient simulation of the proposed modified TLG cell for read and write operations are shown in Fig. 4. In the read cycle, the outputs of three different cells are shown by red lines, and blue lines correspond to the outputs of the cell with programming circuit, which shows a small deviation from the ideal voltage in NAND and XNOR cells and can be improved using higher $R_{off}$ resistance or improving the performance of the read switches. Also, this deviation is easily removed by the thresholding circuit in the TLG crossbar array. In the writing cycle, the update process of memristors $R_{c1}$ for $R_{off}$ to $R_{on}$ and $R_{c2}$ from $R_{on}$ to $R_{off}$ is performed at the same time. It is shown that the update process of all the memristors for the proposed programming circuit can be done in parallel and does not effect the resistance of $R_1$, $R_2$, $R_3$ and $R_4$.

Figure 5 illustrate the example of the programming of $3 \times 4$ TLG crossbar array programming and performance of the circuit. The exemplar sequence is shown in Fig. 5 (top). The output graphs show the outputs of the cell before (orange line) and after (green line) the thresholding block, which illustrates the thresholding block can compensate the voltage drop in the switches. The on-chip area and maximum power dissipation are shown in Table II. Comparing to the previous implementation [7], the on-chip area of the circuit TLG cell and programming circuit is improved.
On-chip area

The future work will include the improvement of the switch performance, investigation of the possibility to use memristive switch \cite{13}, development of the programming algorithm and analysis of the limitations of the proposed TLG array for large scale simulations. Also, the trade-off between power consumption, on-chip area and programming time will be investigated.

V. CONCLUSION

This work proposes the implementation of memristive programmable TLG crossbar array with modified TLG cell for high speed processing. The TLG cell was modified to be independent on the input and avoid changing the resistance of preprogrammed control memristor during the processing, which allows to speed up the operation. The proposed crossbar arrangement of TLG cells ensures flexibility in the programing. The non-ideal behavior of the devices has been considered. The on-chip area and power dissipation of the simulated $3 \times 4$ TLG array is $1463 \mu m^2$ and $425 \mu W$, respectively. The future work includes the investigation of the array scalability, improvement of the performance in term of on-chip area and power dissipation and development of generalized crossbar programming algorithm.

REFERENCES

\begin{itemize}
  \item \cite{1} S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Memristor-based material implication (imply) logic: Design principles and methodologies,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 10, pp. 2054–2066, Oct 2014.
  \item \cite{2} A. K. Maan, D. A. Jayadevi, and A. P. James, “A survey of memristive threshold logic circuits,” IEEE transactions on neural networks and learning systems, vol. 28, no. 8, pp. 1734–1746, 2017.
  \item \cite{3} S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Magic x2014;memristor-aided logic,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 11, pp. 895–899, Nov 2014.
  \item \cite{4} K. C. Rahman, D. Hammerstrom, Y. Li, H. Castagnaro, and M. A. Perkowski, “Methodology and design of a massively parallel memristive stateful imply logic-based reconfigurable architecture,” IEEE Transactions on Nanotechnology, vol. 15, no. 4, pp. 675–686, July 2016.
  \item \cite{5} L. Gao, F. Alibart, and D. B. Strukov, “Programmable cmos/memristor threshold logic,” IEEE Transactions on Nanotechnology, vol. 12, no. 2, pp. 115–119, March 2013.
  \item \cite{7} S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, “Mtl x2014; memristor ratioed logic,” in 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications, Aug 2012, pp. 1–6.
  \item \cite{8} A. K. Maan and A. P. James, “Voltage controlled memristor threshold logic gates,” in 2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Oct 2016, pp. 376–379.
  \item \cite{9} J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and M. A. Perkowski, “Methodology and design of a massively parallel memristive stateful imply logic-based reconfigurable architecture,” IEEE Transactions on Nanotechnology, vol. 15, no. 4, pp. 675–686, July 2016.
  \item \cite{10} O. Krestinskaya, T. Ibrayev, and A. P. James, “Hierarchical temporal memory features with memristor logic circuits for pattern recognition,” IEEE Transactions on Computers-Aided Design of Integrated Circuits and Systems, vol. 37, no. 6, pp. 1143–1156, June 2018.
  \item \cite{11} N. Dastanova, S. Duisenbay, O. Krestinskaya, and A. P. James, “Binary plane extracted moving-object detection using memristive crossbar-cam arrays for edge computing image devices,” IEEE Access, vol. 6, pp. 18954–18966, 2018.
  \item \cite{12} D. Bialek, Z. Kolka, V. Biolková, Z. Birošek, M. Potoveč, and D. Tošić, “Modeling and simulation of large memristive networks,” International Journal of Circuit Theory and Applications, vol. 46, no. 1, pp. 50–65, 2018.
  \item \cite{13} Y. Zhang, Y. Shen, X. Wang, and L. Cao, “A novel design for memristor-based logic switch and crossbar circuits,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 5, pp. 1402–1411, 2015.
\end{itemize}