μSystems Research Group

School of Engineering

Current-Mode Carry-Free Multiplier Design using a Memristor-Transistor Crossbar Architecture

Shengqi Yu, Ahmed Soltan, Rishad Shafik, Thanasin Bunnam, Domenico Balsamo, Fei Xia, Alex Yakovlev

Technical Report Series

NCL-EEE-MICRO-TR-2019-216

November 2019

Contact: s.yu10@newcastle.ac.uk

NCL-EEE-MICRO-TR-2019-216

μSystems Research Group
School of Engineering
Merz Court
Newcastle University
Newcastle upon Tyne, NE1 7RU, UK

http://async.org.uk/

Current-Mode Carry-Free Multiplier Design using a Memristor-Transistor Crossbar Architecture

Shengqi Yu, Ahmed Soltan, Rishad Shafik, Thanasin Bunnam, Domenico Balsamo, Fei Xia, Alex Yakovlev

November 2019

Abstract

Traditional multipliers consist of complex logic components. They are a major energy and delay contributor in modern compute-intensive applications. As such, designing efficient multipliers has remained a thoroughgoing challenge for reduced energy and faster speed. This paper presents a novel, carry-free multiplier, which is suitable for new-generation of energy-constrained applications. The multiplier circuit consists of an array of memristor-transistor cells that can be selected (i.e., turned ON or OFF) using a combination of DC bias voltages based on the operand values. When a cell is selected it contributes to current in the array path, which is then amplified by current mirrors with variable transistor gate sizes. The different current paths are connected to a node for analogously accumulating the currents to produce the multiplier output directly. This removes the need to have latency-sensitive carry propagation stages, typically seen in traditional digital multipliers. An essential feature of this multiplier is autonomous survivability, i.e., when the power is below this threshold the logic state automatically retains at a zero-cost due to the non-volatile properties of memristors. We conduct a number of experiments to validate the functional and parametric properties. Our experiments showed that proposed multiplier achieves 51.44% savings in energy at a similar accuracy when compared with recently proposed approaches.

1 Introduction

Continued developments in microelectronics technology have led to myriad of new compute-intensive applications at the micro-edge, such as artificial intelligence, signal and image processing. As such, energy efficiency has remained a parimary design objective when powered by batteries or emerging energy harvesters [1]. For many of these applications, multiplication is a crucial arithmetic process. It is a major consumer of energy due to large logic complexities [2]. Performance is also affected my their large logic blocks, primarily due to the latency caused by the long chain of cascaded carry addition requirements [3].

Over the years, researchers have investigated various design methods to minimize the energy and latency, such as approximate and speculative circuit designs [2,4–6]. These are based on marginally shortening the carry chains or truncating carry chains completely. However, achieving the crucial step change in energy efficiency continues to be challenging due to the following two reasons. Firstly, the performance as well as accuracy variations are limited by the constrained length of the carry chains, determined by the design methods. Secondly,

as most of these multipliers use Landauer's logic boundaries defined in the voltage domain, their operating points are limited to a set of permissible voltage and clock frequency pairs above the threshold [7].

As an alternative, analog current-mode arithmetic circuit design has recently gained momentum [8,9]. These circuits operate with a dynamic range of currents (from μA to several mA), thereby providing considerably higher energy efficiency leverage than voltage-mode circuits, with the added advantage of high slew rate and simpler circuitry. For example, using these circuits concurrent additions can be performed by directing current paths into a node, and subtractions can be carried out by controlling current paths away from a node. Due to reduced circuit complexity, current-mode circuits can also operate faster with significantly reduced energy consumption [10, 11].

Despite promises of analog current-mode arithmetic circuit design, the research in this area has remained limited. A major reason for this is the lack of architectural programmability of current paths in ultra-low power mode. For energy efficiency, the programming or switching network will need to offer low conductance with low biasing voltages for the current networks. This is challenging for traditional transistor based current networks as the typical switching current tends to vary marginally based on the biasing voltages. Besides, to reduce the conductance large resistor networks need to be integrated, which can contribute to unexpected parasitic behaviour of the circuits. Recently, memristor-transistor cells have shown excellent current-mode characteristics, which can be positively exploited to accurately program current networks for ultra low-power applications [12].

This paper presents a novel multiplier design using a memresistive crossbar architecture for current path controls. The array consists of memristor-transistor cells, which can be turned ON or OFF using a combination of DC bias voltages based on the operand values. When a number of cells in a column are turned ON, they provide with a sum of product terms without involving any carry propagation at all. The resulting current is then amplified by current mirrors with variable transistor gate sizes to suitably multiply these paths with powers of 2. The different current paths are then directed to a node for analogously accumulating the currents using Kirchhoff's current law (KCL), again without the need of carry propagation unlike traditional digital multipliers.

Specifically, this work makes the following ***contributions***:

1. a mixed-signal carry-free multiplier using current-mode principles in a crossbar architecture; and
2. extensive validation and analysis demonstrating the multiplier's improved latency and energy efficiency.

To the best of our knowledge, this is the first complete work demonstrating the memristor-transistor based multiplier design in a current-mode configuration. The rest of the paper is organised as follows. Section.2 provides preliminary concepts and background of the proposed design approach. Section.3 describes the multiplier design using crossbar architecture. In addition, details of operation and overarching principles are also given. Section.4 presents the results of two different multiplication cases and discusses the future work. Finally, Section.5 concludes the paper.

2 Background and Proposed Approach

This section provides relevant background of the traditional binary multiplier and then develops the rationale of the proposed multiplier design using current-mode memristor-transistor crossbar arrays. In a traditional ($N \times N$) binary multiplier, two unsigned integers can be multiplied using N^2 logic AND operations followed by up to 2N

ADD operations. For example, consider the multiplication of two 4-bit unsigned integers, where the multiplier $M_1 : \{m_3m_2m_1m_0\}$ and the multiplicand is $M_2 : \{n_3n_2n_1n_0\}$, as illustrated in Fig.1. As can be seen, the N^2

$$
\begin{array}{cccccccc}
 & & & m_3 & m_2 & m_1 & m_0 & \\
 & & \times & n_3 & n_2 & n_1 & n_0 & \\
\hline
 & & & m_3n_0 & m_2n_0 & m_1n_0 & m_0n_0 & \leftarrow PP \\
 & & m_3n_1 & m_2n_1 & m_1n_1 & m_0n_1 & 0 & \leftarrow PP \\
 & m_3n_2 & m_2n_2 & m_1n_2 & m_0n_2 & 0 & 0 & \leftarrow PP \\
m_3n_3 & m_2n_3 & m_1n_3 & m_0n_3 & 0 & 0 & 0 & \leftarrow PP \\
\hline
\mathrm{P}_7 & \mathrm{P}_6 & \mathrm{P}_5 & \mathrm{P}_4 & \mathrm{P}_3 & \mathrm{P}_2 & \mathrm{P}_1 & \mathrm{P}_0 \quad \leftarrow FP
\end{array}
$$

Figure 1: Binary multiplication algorithm with 4-bit operands

logic AND operations produce partial product (PP) terms (i.e. bits), which can be generated in parallel and fast. These terms are then column-wise added with variable number of PP terms. For the given example, the column-wise sums of the product terms can be expressed as follows:

$$
\begin{aligned}
P_0 &= m_0n_0; \\
P_1 &= m_1n_0 + m_0n_1; \\
P_2 &= m_2n_0 + m_1n_1 + m_0n_2; \\
P_3 &= m_3n_0 + m_2n_1 + m_1n_2 + m_0n_3; \\
P_4 &= m_3n_1 + m_2n_2 + m_1n_3; \\
P_5 &= m_3n_2 + m_2n_3; \\
P_6 &= m_3n_3.
\end{aligned}
\tag{1}
$$

From Eq.(1), note that when the number of bits in each column is two or more, carry propagation becomes more likely depending on the operand bit values. For example, if $m_1 = m_0 = n_1 = n_0 = 1$, P_1 is expected to produce a carry into P_2. When both operands have all bits set to 1, i.e., M_1=$\{1111\}$ and M_2=$\{1111\}$, the multiplier experiences the largest chain of carry propagation between the columns, starting from the least significant to the most significant bits in the multiplier output.

In traditional multipliers, the maximum delay between the longest PP addition (P_3 in the example in Fig. 1) and carry propagation between column-wise additions determine the critical path (i.e., latency) and the energy consumption of the circuit. This latency can be reduced by using carry look-ahead (CLA) adders [13] or their approximate equivalents [14]. However, when higher precision multipliers are used, the larger carry chains can prove challenging in these approaches as a delay bottleneck.

In a current-mode circuit, addition operations can be implemented by converging the current paths into a node. When the chain of add operands becomes larger, more paths can be added or enabled without any significant changes in the delay of the circuit. This sets up the **key motivation** of designing a mixed-signal multiplier circuit in our proposed approach, which is described next.

In our proposed multiplier, the column-wise terms (as shown in Fig. 1) are expressed as direct decimal equivalents, programmed as current paths. Updating Eq.(1) the decimal equivalent values of each column-wise

term (P_i) can be expressed as follows:

$$
\begin{aligned}
P_0 &= 2^0 \times (m_0 n_0)\,; \\
P_1 &= 2^1 \times (m_1 n_0 + m_0 n_1)\,; \\
P_2 &= 2^2 \times (m_2 n_0 + m_1 n_1 + m_0 n_2)\,; \\
P_3 &= 2^3 \times (m_3 n_0 + m_2 n_1 + m_1 n_2 + m_0 n_3)\,; \\
P_4 &= 2^4 \times (m_3 n_1 + m_2 n_2 + m_1 n_3)\,; \\
P_5 &= 2^5 \times (m_3 n_2 + m_2 n_3)\,; \\
P_6 &= 2^6 \times (m_3 n_3)\,.
\end{aligned}
\tag{2}
$$

The sum of all column-wise terms in Eq.(2) will then produce the multiplier output as:

$$
M_1 \times M_2 = \sum P_c \quad (c = 0, 1 \ldots 2N)\,, \tag{3}
$$

where P_c is the sum of products on the c_{th} column.

The multiplication algorithm in Eq.(2) and Eq.(3) can thus be simplified into three steps: 1) partial product terms can be generated in parallel by switching ON or OFF the current paths, 2) each current path, defined by the column-wise term in Eq.(2), is amplified in current-mode, and 3) the final output shown in Eq.(3) can be generated by summing the currents from all paths. In the following, we briefly outline the design approach for these three steps;

2.0.1 Partial Product Generation

The product terms are generated by switching the current paths through memristor-transistor cells organised in a crossbar array as shown in Fig.2. The low-level circuit layout of the cell is further detailed in Fig. 2. In the

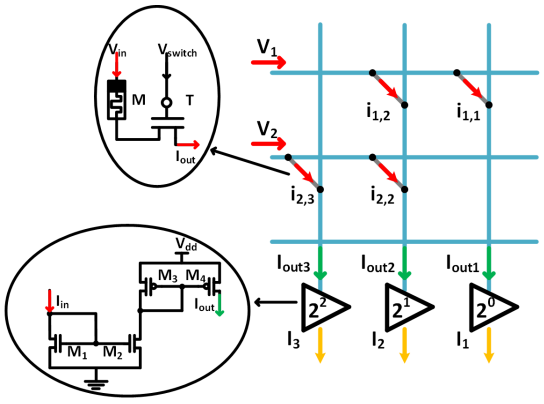


Figure 2: Multiplier product generation and accumulation circuits

crossbar architecture, the row and column lines are connected using the cross point wire using this cell. Such an arrangement allows for switching ON or OFF the current paths based on the multiplier bit-wise operand values. One of the operands is used as the control signal V_{switch}, which switches all the cell transistors on same column. Another operand is used to control the input signal V_{in}, which powers all the cell memristors on

same row. Concurrent switching of the cells using V_{switch} and V_{in} produces bit-wise AND-like operation at each corresponding cell, generating the partial product terms in the multiplier.

In the current-mode switching arrangement, the current paths defining the partial product terms are generated using the Ohm's law. Using this law, current in each pathway, denoted as $I_{k,i}$ (k means k_{th} row, i means i_{th} column), is defined as below

$$I_{k,i} = V_{in_k} \times G_{k,i} \quad , \tag{4}$$

where $G_{k,i}$ is the conductance of cell at the pathway between k_{th} row and i_{th} column.

2.0.2 Current Amplification

As previously shown in Eq.(2), the column-wise term (P_i) is then generated by amplifying the output current of the i_{th} column, I_{out_i}, by a gain factor g_i. Hence, P_i can be expressed as:

$$P_i = g_i \times I_{out_i} \quad . \tag{5}$$

In crossbar array, the column current I_{out_i} is the sum of currents from selected cells depending on the multiplier row operand values, given by Kirchhoffâs current law (KCL) as:

$$I_{out_i} = \sum_{\substack{k=1 \\ i=1}}^{N} a_{k,i} i_{k,i} \quad , \tag{6}$$

where $a_{k,i}$ is the number of cells that contributed to the partial product term, i.e. the current on the i_{th} column. The gain g_i follows the relationship as follows:

$$g_i = 2^{i-1} \quad . \tag{7}$$

In current-mode, the amplification of output current is achieved using suitably chosen current mirror ratios (see Section IV). Using Eq.(7) the column-wise term P_i can be expressed as:

$$P_i = g_i \times I_{out_i} = 2^{i-1} \sum_{\substack{k=1 \\ i=1}}^{N} a_{k,i} V_{in_k} G_{k,i} \quad . \tag{8}$$

2.0.3 Current Accumulation Instances

The final product of multiplication is the accumulation (i.e., sum) of all the column-wise terms as shown in Eq.(3). To facilitate a completely carry-free accumulation of the current using KCL, the column-wise terms after amplification are connected in parallel. As such, the final product I can be written as:

$$\begin{aligned} I &= \sum_{i=1}^{N} P_i = \sum_{i=1}^{N} g_i \times I_{out_i} \\ &= \sum_{i=1}^{N} (2^{i-1} \sum_{\substack{k=1 \\ i=1}}^{N} a_{k,i} V_{in_k} G_{k,i}) \quad . \end{aligned} \tag{9}$$

For illustration purposes, consider the following two examples:

$$\text{Ex. 1}: \quad M_1 \times M_2 = 1110 \times 1111 = 11010010b\ (210d) \tag{10}$$

$$\text{Ex. 2}: \quad M_1 \times M_2 = 1101 \times 0110 = 01001110b\ (78d) \tag{11}$$

For the above examples, the respective cell numbers for each of these cases are illustrated in Fig.3.

case	$a_{k,i}$ / i / k	1	2	3	4	5	6	7
Ex.1	1	\	\	\	1	1	1	0
	2	\	\	1	1	1	0	\
	3	\	1	1	1	0	\	\
	4	1	1	1	0	\	\	\
a_i		1	2	3	3	2	1	0
Ex.2	1	\	\	\	0	0	0	0
	2	\	\	1	1	0	1	\
	3	\	0	0	0	0	\	\
	4	1	1	0	1	\	\	\
a_i		1	1	1	2	0	1	0

Figure 3: Cell values and path currents in Ex. 1 and Ex. 2; a_i is the number of cells contributing to the final product

Assuming $G_{k,i} = m$ (i.e., the conductance of memristor in ON state) and $V_{in_k} = n$ (i.e., the switching voltage on the cell row) partial product currents and the corresponding transformations are given below for both examples. The $a_{k,i}$ values are used from Fig. 3 to derive the output current I.

Ex. 1:

$$I = 0 \times 2^0 \times mn + 1 \times 2^1 \times mn + 2 \times 2^2 \times mn + 3 \times 2^3 \times mn + 3 \times 2^4 \times mn + 2 \times 2^5 \times mn + 1 \times 2^6 \times mn + 0 \times 2^7 \times mn = 210mn\ (Amp)$$

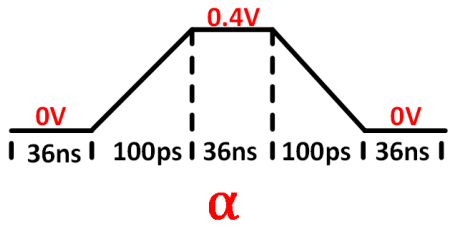
Ex. 2: $I = 0 \times 2^0 \times mn + 1 \times 2^1 \times mn + 1 \times 2^2 \times mn + 1 \times 2^3 \times mn + 2 \times 2^4 \times mn + 1 \times 2^5 \times mn + 0 \times 2^6 \times mn + 0 \times 2^7 \times mn = 78mn\ (Amp)$

As can be seen, the above results match the expected outcomes for the multiplication operations. In the following section, the implementation details are presented, and later the experimental results are delineated comparing with the traditional multiplier circuits.

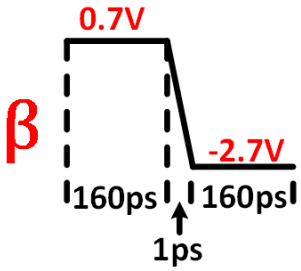
3 Proposed Multiplier Architecture

As shown, the building block for crossbar array is a one-memristor one-transistor (1M1T) cell, which is illustrated in Fig.2. The memristor values represent one set of operands, while the voltage signals in the row lines represent the other set of operands [12]. The 1M1T logic cell uses memristor as the memory unit, and transistor as the switching unit. The memristor remembers its resistance state, even when there is no power supply. When the memristor voltage is over its threshold, the set voltage (SV) will bias the memristor to low resistance state (LRS) and reset voltage (RSV) will bias the memristor to high resistance state (HRS). We denote LRS as logic '1' and HRS as logic '0'. Fig.4 shows how these bias voltages affect the operations. The operation details of logic cell include α (multiplication operand), β (tune resistance state of memristor to low level, write logic '1'), and γ (tune resistance state of memristor to high level, write logic '0'). In α, 0.4V is denoted as a logic '1'

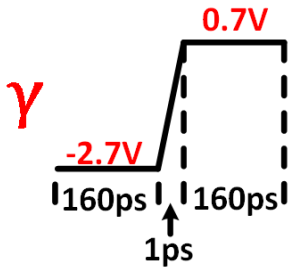
in the multiplier 'x', and 0V is the logic '0'. Then, Fig.5 shows a 4-bit multiplier design using the proposed memristor-transistor crossbar array.



(a) Read operation



(b) Write '1' operation



(c) Write '0' operation

Figure 4: CL operation pulses details

We discuss the implementation details as follows. In the multiplier circuit shown in Fig. 5, the basic 1M1T cell is organized at each cross point (i.e., cell) via the mapping procedure. This design provides a combination of fast operation and accurate selection. Both the input and control signals are applied in the form of a single bar source (SBS). SBS means that the source covers the power supplies of all the 1M1T cells when they are connected to the same row bar, or the control signals of all the 1M1T cells, when they are connected to the same column bar. For the same expression, the row bars that take the input signals are called source lines (SLs), the column bars that take the control signals are called gate lines (GLs) and the column bars that give the output signals are called result lines (RLs). The non-volatile resistive memory (1M1T) applies the threshold voltage memristor model (VTEAM) [15] with the model parameters from [16] which are extracted from the physical device. This ensures our design can be practically implemented, and parameters are listed in Table.1.

Table 1: VTEAM model parameters from [16]

$alpha_{off}$	4	$alpha_{on}$	4
V_{off}(V)	0.3	V_{on}(V)	-1.5
R_{off}(Ohms)	300K	R_{on}(Ohms)	1K
k_{off}(m/s)	0.091	k_{on}(m/s)	-216.2
w_{off}(nm)	3	w_{on}(nm)	0

The proposed design is based on UMC 65nm technology. The transistors are divided into two groups, logic cell (LC), and current mirror (CM) which shown in Fig.2. All LCs contain the same size transistors which are 1000nm width and 60nm length. At the output terminal, NMOS and PMOS CMs are serially connected to perform high ration application. As CMs works as amplifiers with respective gains, their transistor sizes are

different as shown in Table.2.

Table 2: Transistor sizes of the current mirrors

group	NMOS		PMOS	
	M1 (nm)	M2 (nm)	M3 (nm)	M4 (nm)
1	1520/60	400/60	80/60	240/60
2	2720/60	1600/60	80/60	260/60
3	3840/60	2400/60	80/60	720/60
4	5440/60	3200/60	80/60	1680/60
5	4080/60	4800/60	80/60	1920/60
6	2720/60	4800/60	80/60	2680/60
7	1520/60	1840/60	80/60	5120/60

4 Experimental Results and Discussions

In simulation experiment, a 4 by 4 crossbar multiplier will be operated to show the multiplication process in this novel multiplier. The multiplication is operated by two 4 bit binary numbers. The input 'x' is from 0 (0000) to 15 (1111) and the weight 'w' keeps a fix value 0 (0000). Multiplication operands will be denoted in Table.3, the logic '0' of output has two current values 0A and 1.33μA. Both of them are small enough to neglect for the logic '1'.

Table 3: The results definition of multiplication

State	w	x
1	300KOhms	0.4V
0	1KOhms	0V
Multiplication Result		
Logic '1'		
0.4V/1KOhms=0.4mA		
Logic '0'		
0V/300KOhms=0A	0V/1KOhms=0A	0.4V/300KOhms=1.33μA

And, the results of multiplication in theory and simulation can be shown in Fig.6 and Fig.7

Fig.7 shows the result in current mode from the proposed multiplier. For an instance, the last stair in Fig.7 is the result of x=1111 times w=0000. On circuit, it means,

1. The input voltage series to the crossbar in Fig.5 is input1=0.4V, input2=0.4V, input3=0.4V, and input4=0.4V.

2. The switch voltage series to the crossbar in Fig.5 is Vswitch1=1.2V, Vswitch2=1.2V, Vswitch3=1.2V, Vswitch4=1.2V, Vswitch5=1.2V, Vswitch6=1.2V, and Vswitch7=1.2V.

3. The memristors in all LCs on the crossbar in Fig.5 are bias to HRS.

It is clear to notice that, in this 4 by 4 crossbar multiplier in Fig.5, the least significant bit (LSB) refers to different items, for input voltage series, the LSB is 'input1' on circuit; for the final product, the LSB is 'out1'

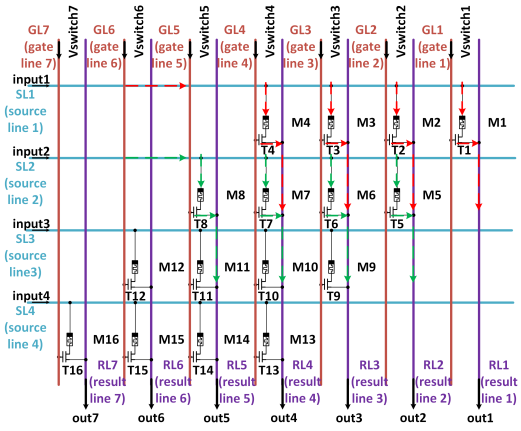


Figure 5: Four by four 1M1T crossbar circuit with three line setting, one row line and two parallel column lines defined to give the circuit the ability to select any cell within the circuit

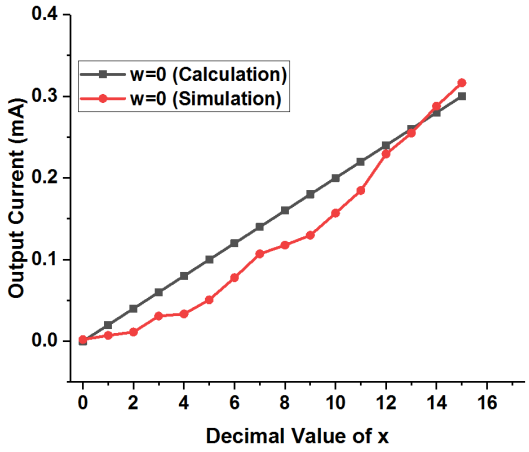


Figure 6: Calculation results comparison with simulation results in the 4bits multiplication of w=0

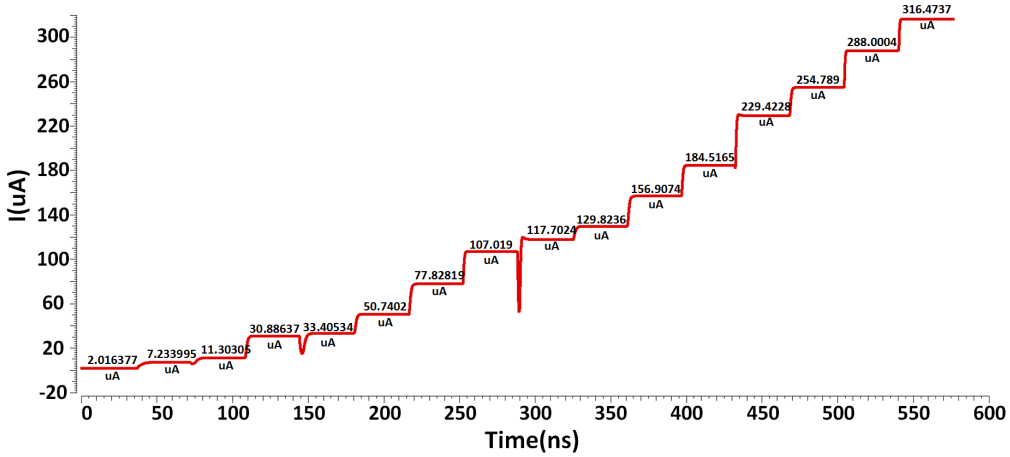


Figure 7: Multiplication outcome of proposed multiplier and comparison with the expected results

on circuit; and for memory, the LSB are 'M1', 'M5', 'M9', and 'M13' on circuit. In the same method, the most significant bit (MSB) for input voltage series, final product, and memory on circuit are 'input4', 'out7', 'M4', 'M8', 'M12', and 'M16'. The rising of the stairs means the input 'x' binary value increases from '0000' to '1111' with the respective input voltage series, and generates specific current to output the calculation result. The result graph in Fig.7 shows that the multiplication process well.

Fig.8 illustrates that, in the 4 bits multiplication, proposed design saved 50% energy than Kulkarni's design. And have 84.22% higher rate in error rate than Qiqieh's approach in the min case.

There are still a number of issues in the early stage; for example several stairs raise over the next level. That is because the LRS of LC will cause a higher voltage drop than the HRS. Thus, the current generated for logic '1' is lower than it should be, and the logic '0' current is higher. Moreover, the output current errors can also be amplified by the CM circuit. The gain of CM is also effected by the terminal voltages, the higher voltage drop of logic '1' current case leads the decrease of CM amplifier gain. Simultaneously, the logic '0' current

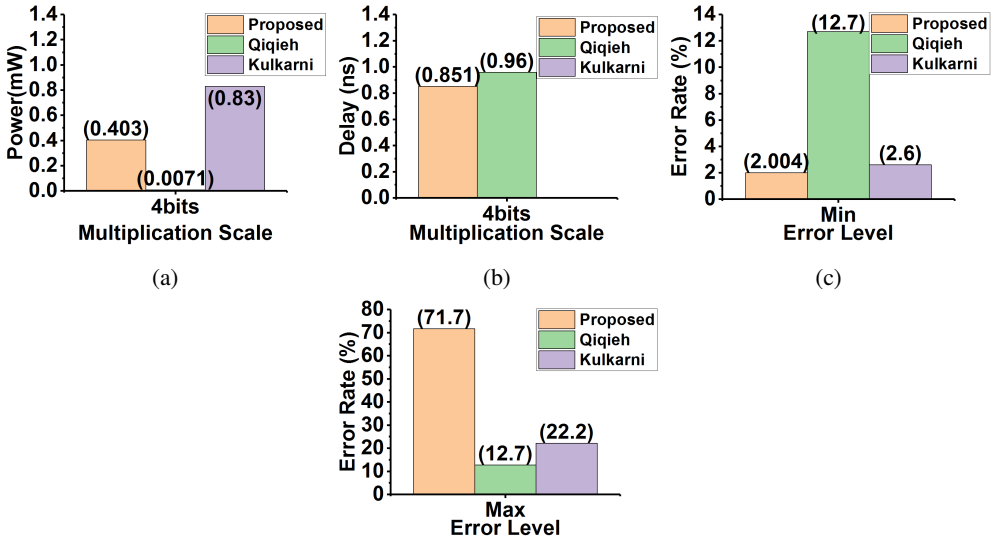


Figure 8: Comparative power, delay and accuracy analysis: (a) shows that the power consumption of proposed design is 403μW, Qiqieh's approach [2] is 7.1μW, and Kulkarni's approach [17] is 830μW, (b) illustrates that proposed multiplier has 0.851ns delay, while Qiqieh's approach has a delay of 0.96ns, (c) and (d) shows that in low error level, the proposed design has the lowest error rate(ER) 2.004%, then follows the Kulkarni 2.6% and Qiqieh is the highest 12.7%. However, situation reversed in the high error level, Qiqieh has the lowest ER 12.7%, Kulkarni still in the mid place 22.2%, the proposed gets the highest ER 71.7%

with lower voltage drop gets higher gain than it should. This leads the current level of LSB logic '1' can be lower than the level of MSB logic '0'. Things will get worse after the amplification, Fig.6 shows that problem. Fig.8c and Fig.8d give a much more direct method to describe the difference between the error rates between high level multiplication and low level multiplication. But the delay of proposed multiplier is the smallest one in these three models, which means the speed can be an outstanding character of proposed model.

5 Conclusion

In this paper, a mixed-signal current-mode multiplier has been proposed. The proposed multiplier features carry-free operation using current-mode principles. By making circuit simpler and less complex, computation latency and power consumption are significantly reduced. Using Cadence VHDL-AMS, the efficacy of the proposed multiplier is validated. When compared with existing multipliers, the proposed crossbar array shows deterministic precision and consumes much less power (in some cases showing power savings of well over 51.44%). This makes the proposed device more relevant for applications in which the computation units at the edge are powered with limited energy sources with unpredictable and sporadic supply powers. The use of memristors enables the state of the switches to be retained naturally under power cuts, which we aim to study at a greater detail in our future work. Our further planned works include the development of a fully featured crossbar together with in-situ power delivery and control. We will aim to enhance the functional capability of the crossbar to a hierarchical multiply-accumulate unit, suitable for emerging machine learning applications.

References

[1] R. Shafik, A. Yakovlev, and S. Das, "Real-power computing," *IEEE Trans. Computers*, 2018.

[2] I. Qiqieh, R. Shafik, G. Tarawneh, D. Sokolov, S. Das, and A. Yakovlev, "Significance-driven logic compression for energy-efficient multiplier design," *IEEE J. Emerging and Selected Topics in Circuits and Systems*, 2018.

[3] G. Tagliavini, S. Mach, D. Rossi, A. Marongiu, and L. Benin, "A transprecision floating-point platform for ultra-low power computing," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2018, pp. 1051–1056.

[4] Y. Kim, B.-S. Song, J. Grosspietsch, and S. F. Gillig, "A carry-free 54b/spl times/54b multiplier using equivalent bit conversion algorithm," *IEEE J. Solid-State Circuits*, vol. 36, no. 10, pp. 1538–1545, 2001.

[5] T.-B. Juang and S.-F. Hsiao, "Low-error carry-free fixed-width multipliers with low-cost compensation circuits," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 52, no. 6, pp. 299–303, 2005.

[6] S. Getzlaff and R. Schüffny, "A mixed signal multiplier principle for massively parallel analog VLSI systems," in *Recent Advances in Signal Processing and Communication; Proceedings of the 3rd IMACS Int. Multiconference on Circuits, Systems, Communications and Computers CSCCâ99*. Athens, Greecs, 1999.

[7] A. Yakovlev, *Ch. Enabling Survival Instincts in Electronic Systems: An Energy Perspective, in Transforming Reconfigurable Systems*, 2015, ch. Imperial College Press, pp. 237–263.

[8] M. A. Eldeeb, Y. H. Ghallab, Y. Ismail, and H. El-Ghitani, "A 0.4-v miniature cmos current mode instrumentation amplifier," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 3, pp. 261–265, 2018.

[9] S. Lawanwisut, P. Satthaphol, K. Payakkakul, P. Pipatthitikorn, and M. Siripruchyanun, "A temperature-insensitive current-mode multiplier/divider using only double-output vdta," *Procedia Computer Science*, vol. 86, pp. 156–159, 2016.

[10] G. G. Gielen and R. A. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1825–1854, 2000.

[11] F. Yuan, "Low-voltage cmos current-mode circuits: topology and characteristics," *IEE Proceedings-Circuits, Devices and Systems*, vol. 153, no. 3, pp. 219–230, 2006.

[12] 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.

[13] H. Ling, "High-speed binary adder," *IBM Journal of Research and Development*, vol. 25, no. 3, pp. 156–166, 1981.

[14] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Rap-cla: A reconfigurable approximate carry look-ahead adder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 8, pp. 1089–1093, 2018.

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

[16] S. Kvatinsky, D. Belousov, S. Liman, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, "MAGIC - memristor-aided logic," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 11, pp. 895–899, 2014.

[17] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading accuracy for power with an underdesigned multiplier architecture," in *VLSI Design (VLSI Design), 2011 24th Int. Conf.* IEEE, 2011, pp. 346–351.