Characterizing a standard cell library for large scale design of memristive based signal processing

Abubaker Sasi | Arash Ahmadi | Majid Ahmadi

Abstract
In recent years, the use of memristors in circuits design has rapidly increased and attracted research interest. Advances have been made to both the size and the complexity of memristor designs. Therefore, computer aided design tools are required to handle memristor-based large-scale designs. A comprehensive automatic framework for the design and synthesis of large-scale memristor-complementary metal-oxide-semiconductor (CMOS) circuits is described herein. This framework provides a synthesis approach that can be applied to all memristor-based digital logic designs. In particular, it is a proposal for a characterization methodology of memristor-based logic cells to generate a standard cell library file for large-scale simulation. The proposed architecture is based on RRAM and ReRAM redox-based devices and the memristor ratioed logic design approach. The proposed framework is implemented in the Cadence Virtuoso schematic-level environment and was verified with Verilog-XL, MATLAB, and the electronic design automation synopses compiler after being translated to the behavioral level. The proposed method can be applied to implement any digital logic design. Nevertheless, it is perfectly suitable for signal processing applications that require MATLAB functions to produce text files with hex values in order to overcome the limitations of the simulation environment. A framework is deployed herein for design of the memristor-based parallel 8-bit adder/subtractor and a 2D memristive-based median filter. Both proposed designs memristor-based adder/subtractor and memristive median filter have significant power reductions of 66% and 16% respectively, when compared to the same designs using CMOS technology.

1 | INTRODUCTION

Although the conventional complementary metal-oxide-semiconductor (CMOS) technology scaling limitation was extended using FinFET architecture, FinFET is facing significant challenges for different reasons such as doping damage, restriction in the logic chip design space, limitation of the electrostatics, and integration challenges [1, 2]. Therefore, substitutes to CMOS technology are in high demand. There are several alternative technologies, such as Double-Gate Tunnel FET [3], nanotube programmable devices [4], graphene transistors [5], and memristor devices [6]. Among those technologies, memristor devices are the most promising because of their great scaling ability, long-term data storage, low-power consumption, and CMOS compatibility [7, 8]. It is believed that these two terminal devices will play an essential role in the future fabrication of memory and information processing systems [9, 10]. Nevertheless, the number of memristor-based applications in today’s circuit designs has been increasing exponentially. However, the design and mapping of large-scale memristor-based applications is a challenging task due to the lack of comprehensive high-level design tools and simulation platforms. Currently, circuit design tools like SPICE (H&LT) SPICE, ICAPS, and Cadence Virtuoso are not capable of providing designers with comprehensive design and simulation methodologies for memristors [11].

Xie et al. [12], presented a method for the automatic mapping of large-scale crossbar memristive-based Boolean...
logic circuits. This method involved the use of CMOS to control and drive the design. A programmable architecture for a large-scale neuromorphic-systems-based-memristive crossbar is proposed in [13]. The authors proposed a framework for deep learning networks based on the programming of spin electronics (spintronic devices). The framework mapping blocks consist of memristors and transistors to mimic spindle behaviour. In [14], the authors introduced a design methodology for memristor crossbar architecture-based image compression. The author primary objective is to perform computational operations in a memristive crossbar and store the row-transformed image data in the same crossbar memory array. Therefore, the overall area, timing, and power of this architecture were reduced. The aforementioned methods were implemented based on memristive crossbars. Such design techniques presented real challenges, including those related to sneak path current and signal degradation. Moreover, memristive crossbar circuits require separate circuits to control input signals.

Material implication logic is also implemented to map memristor-based Boolean logic [15, 16]. In these works, implication logic was employed to reduce the number of memristor devices and operating cycles. However, the use of such methods is limited only to Boolean function implementation. Moreover, memristor-based crossbar and implication logic design methods are not synthesizable using computer-aided design (CAD) synthesis tools [17]. In addition, the above-mentioned design methods require sequential computational steps to achieve a logic gate operation. In such a process, execution of one logic computation requires more than one clock cycle.

Considering these challenges, a hybrid memristor/C莫斯 logic design is the most applicable method because it is CMOS compatible, delivers an optimal solution to eliminating signal degradation, and can be synthesised and mapped using CAD tools. However, it is impractical to manually design memristor-based large-scale circuits using currently available methods due to design complexity and the limited number of memristors and transistors that CAD tools support [18].

Herein, a comprehensive automatic framework for the design and synthesis of large-scale memristor-Cmos circuits is proposed. This framework provides a synthesis approach that can be applied to all memristor-based Boolean logic designs. In particular, MATLAB, a high definition language (HDL) simulator, the Cadence Virtuoso environment, and Synopses software were utilised to implement parallel 8-bit adder/subtractor and 2D memristive-based median filter. The filter was manually implemented on the Cadence Virtuoso schematic level and previously published in [19].

Brief details about choosing a proper memristor model are given in Section 2. Section 3 contains a description of cell library characterisation. Section 3.1 contains a discussion of the CAD tools used for the automatic implementation of the proposal. Section 4 provides case study and a discussion of the proposed simulation results, and finally, Section 5 concludes this paper.

2 | MEMRISTOR BASED-LOGIC DESIGN

2.1 Memristor modeling

All designs, simulations and cell characterisations for a memristor-based standard cell library in proposal were implemented using a metal-oxide-based resistive random access memory (RRAM) devices model [20] and reox-based resistive switching memories (ReRAM) model which was presented in [21]. The accuracy levels of both memristive models provides the realistic required switching behaviour. Both models are simulated using the Verilog-A model in the Cadence Virtuoso environment. ReRAM module is the first module used in this proposal, but during the logic cell characterization for delay, power dissipation, and input capacitance. The results were not that good as expected due to the using of CMOS transistor and due to the following factors that were considered when differentiating between the choice of RRAM and ReRAM devices:

[1] Device size and resistive layer: Both devices are designed based on metal oxides that consume power relatively little. RRAM device is preferred due to its small size, which is <10 nm, while the size the ReRAM device is 11 nm. In addition, the size of the metal has a direct effect on the capacitance of the device, which has a significant impact on the power dissipation and delay performance of the circuit.

[2] The amplitude of input voltage: It is important to utilize an appropriate supply voltage to obtain low power consumption and ensure high performance. However, having low voltage led to significant increases in the propagation delay, and significantly decreases the power consumption. Thus, the RRAM device only requires 2 V of input voltage supply, which is low compared to ReRAM which requires 4 V.

For circuit testing and simulation, both Verilog-A models of RRAM that were presented in [20] and the ReRAM device that was presented in [21] were utilised to obtain the desired logic behaviour for the proposed design. The accuracy levels of both memristive models provide the realistic required switching behaviour. Both are simulated using the Verilog-A model in the Cadence Virtuoso environment. Due to the lack of real physical memristor device layout tools, it is important to choose an accurate memristor model [20] that simplifies the implementation of memristor-based applications and study cases for the creation of reliable simulations.

RRAM and ReRAM devices were simulated based on the parameters shown in Table 1. In this proposal, two factors were considered when differentiating and choosing between RRAM and ReRAM (Pt/TaOx/Ta) devices to implement memristor-based logic gates at the behavioral level.

The first factor is device size and resistive layer: both devices were designed based on the small size of metal oxides...
T A B L E 1 The simulation parameters for ReRAM and RRAM devices

| Parameters | N_{min} (m^{-2}) | N_{max} (m^{-2}) | N_{init} (m^{-2}) | C_{l} (pAm/V) | L (nm^2) | L_{Disc} (nm) |
|------------|-----------------|-----------------|-----------------|--------------|----------|--------------|
| ReRAM PT/TaOx/Ta device [21] | 0.308 | 5 | 5 | 6 | 3.14 | 4 |
| RRAM device [20] | Value | 6.14 (e^{-3}) | 2.75 (e^{-10}) | 6 (e^{-12}) | 3.14 (e^{-14}) | 5 | 150 |

Abbreviations: ReRAM, redox-based resistive switching memories; RRAM, resistive random access memory.

2.2 Logic design approach

Memristor ratioed logic (MRL) is a hybrid CMOS-memristor-based logic [23]. It is a voltage-based design approach, unlike MAD [24] and Mirrored [25] logics, which are memristive-based. The compatibility of memristor devices with CMOS increases circuit density and offers the best way to eliminate signal degradation in memristor logic of AND and OR gates. The CMOS inverter is added to output of memristor-based OR and AND gates to achieve the desired NOR and NAND logic [26]. In MRL, the voltages are perceived as OR and AND gates to achieve the desired NOR and NAND logic [26]. In MRL, the voltages are perceived as OR and AND gates to achieve the desired NOR and NAND logic [26].

\[ P = C_L V^2 f \]  

where \( C_L \), \( V \), and \( f \) are load capacitance, voltage amplitude and frequency, respectively. The second factor is the amplitude of the input voltage. It is important to utilise an appropriate supply voltage to obtain low-power consumption and ensure high performance. Although having low voltage leads to significant increases in propagation delay, it significantly decreases power consumption. Thus, the RRAM device only has 2 V of input voltage supply, which is low compared to the ReRAM (Pt/TaOx/Ta) device, which has 4 V as shown in its I–V curve in Figure 1.

that consume less power. Small devices consume less power than large ones [22]. Therefore, the RRAM device is preferred due to its small size, which is <10 nm, while the size of the ReRAM device is 11 nm. In addition, as seen in Equation (1), the size of the metal has a direct effect on the capacitance of the device, which has a significant impact on the power dissipation and delay performance of the circuit

\[ V_{out} = \frac{R_{on}}{R_{on} + R_{off}} V_{high} \cong 0 \]  

and when if low voltage '0' is applied to both inputs terminals then \( V_{out} \) can be calculated as:

The MRL logic design approach was exploited to implement the proposed circuit designs.

3 SYNTHESIS METHODOLOGY AND IMPLEMENTATION

Creating a memristor-based standard cell library is essential to exploring the potential of memristors in digital design using available CMOS synthesis tools. Using such tools requires an accurate cell characterisation method for memristor-based logic gates. Synthesis tools involve the use of characterised gates library files to facilitate logic optimisation, enhance design speed, and determine the area, timing, and power consumption. The characterisation process for any memristor-CMOS cells can be described as follows.

3.1 Input/output capacitance

The measured capacitance values at each cell pin is the main factor used to estimate dynamic power and delay using synthesis tools. Input capacitance is calculated by measuring the charge flows into or out of each cell pin divided by the magnitude of the power supply. It can be mathematically formulated as follows:

\[ C_{pin} = \frac{1}{V_{DD}} \int_{0}^{t_i} i(t) dt \]  

where \( i(t) \) is the current flow into the pin and \( C_{pin} \) is the pin capacitance, measured as the amount of charge passing through the pin at the input voltage (rising swing from 0 to VDD and from VDD to 0) divided by the voltage supply. In the memristor-based logic cells characterisation method, the characterising simulation utilises a net of inverters as standard capacitive load, which is serially connected to the output pin of the cell under characterisation.
3.2 | Power measurement

The logic transition of cell input pins which are deployed in the proposed method consumes energy. The value of energy consumed by the proposed circuit was measured by calculating the current passing through the zero-DC source that was connected to the VDD. Then the consumed current was integrated over each time transition using the Cadence Virtuoso calculator. The library table of each cell in the proposed design only contains energy values measured in joules, and the rest of the power consumption calculation was accomplished by the Synopsys synthesis compiler. The only measured power consumption in this method is dynamic power, which is mathematically described as follows:

\[ P_{\text{Dynamic}} = \alpha CV_{DD}^2f \]  

(5)

where \( \alpha, C, V_{DD} \) and \( f \) are the switching activity factor, capacitance, voltage source, and operating frequency, respectively.

3.3 | Delay measurement

The non-linear delay simulation method was utilised to measure the propagation delay. With fan-out consideration, the delay measurement depends on the transition time at the cell input pin and the capacitance of the output pin. The specified slew threshold for the cell is set to be between 30% and 70% of the power supply magnitude. In addition, it was defined as the time the signal rises from 30% to 70% and falls from 70% to 30% of its VDD.

3.4 | Area estimation

As illustrated in [27, 28] the memristor device can be fabricated on the top of the CMOS transistors. Therefore, the area was estimated depending on the size of the inverters utilised in each cell.

3.5 | CAD tools for automatic implementation

To prove the functionality of the framework and test the feasibility of the automatic implementation of a memristor-based digital design approach, several steps were taken, as illustrated in the design flow in Figure 3.

4 | MEMRISTOR BASED-LOGIC DESIGN

4.1 | Memristor modelling

In the first step, the behavioural functions of the implemented cells at the schematic level were described using Verilog HDL and simulated using the Cadence NC-Verilog-XL simulator. The Verilog language can be used to read/write files from a storage environment. This feature makes it possible to design a test bench to read data from a storage device, generate stimulus signals for the Verilog test module, and write the results to a storage device. In the proposed framework, the signal processing applications require a MATLAB encoder and decoder. MATLAB function is needed to convert input signals into the form of hex arrays because Verilog only reads and writes ASCII character files, and then another MATLAB function is used to import the processed data encoded by the Verilog test bench to reconstruct it. In the second step, as shown in Figure 3a, after testing the design at the behavioural level, the implemented register-transfer level (RTL) was synthesised to the gate netlist level with the aid of a Synopsys Design Vision compiler. The design compiler uses a standard library that contains all information about the characteristics of logic cells to generate the final CMOS-based gate netlist file.

In the third step, the generated CMOS gate netlist was carefully inspected to realise the logic cells used to build the CMOS-based design. After the logic cells are produced by the Synopsys synthesis compiler, equivalent memristor-based logic gates are implemented at the schematic level, tested,

**FIGURE 1** Memristor device I-V curve for redox-based resistive switching memories (ReRAM) Pt/TaOx/Ta valence change memory device with a bipolar triangular input voltage of 5 V [21]

**FIGURE 2** (a) MRI-based AND gate and its resistance progression. (b) MRI-based OR gate and its resistance progression. MRI, memristor ratioed logic
and characterised using the MRL design method. Hence, at this stage of the design, the characterisation process for memristor-based logic cells was obtained to build a standard memristor-based library for the Synopsys synthesis compiler, as presented in Figure 3b.

The most important characterised cells involved in the proposal are AND, NAND, OR, NOR, multiplexer, and other defined Boolean function circuits, as shown in Table 2. The built library provides a synthesis tool with information about cell logic function, area, input/output capacitance, delay, and power consumption.

### 4.2 Case study 1

In this section, a memristor-based parallel 8-bit adder/subtractor is designed and analysed using the proposed framework. It was implemented at both the schematic and behavioural levels. In other words, the implementation was done to establish and validate the design of the adder/subtractor at the schematic level using a Cadence Spice Spectre simulator, NC-Verilog at the behavioural level, and Synopsys Synthesis tools using Design Vision at the synthesis level. As part of the design, the adder/subtractor was chosen to clarify the proposed framework, design, and simulation. The following is a brief description of the framework.

The memristor-based 8-bit adder/subtractor was implemented with seven cascaded combinations of 1-bit memristor-based full adders. The schematic design of the adder/subtractor circuit is exhibited in Figure 4a. The 1-bit memristor-based full adder logic circuit consists of two memristor-based AND gates, one memristor-based OR gate, and two memristor-based XOR gates, as shown in Figure 4b.

The functionality of the designed adder/subtractor was proven by the simulation results in Figure 5. As illustrated in Figure 4a, the Sel line acts as a control signal to decide whether to use the adder or subtractor circuit modes. When Sel = 1, the Sel line acts as carry-in (Cin). Thus, all inputs of B will be reversed and 1 will be added to the LSB to determine the 2's complement. In addition, when Sel = 0, B XOR 0 will always produce B. Therefore, A and B will be added.

The adder/subtractor Verilog description was verified in the Cadence NC-Verilog simulator, and the Synopsys compiler synthesises the RTL description and then converts the synthesised description to the optimised gate-level. The produced

---

**FIGURE 3** Flow chart displaying design flow based on Synopsys EDA tool for proposed framework. CMOS, complementary metal-oxide-semiconductor; EDA, electronic design automation; HDL, hardware description language.

**TABLE 2** List of logic cells involved in the design of the adder/subtractor

| Cells     | Description                          | Equation       | MOSFET | Memristors |
|-----------|--------------------------------------|----------------|--------|------------|
| AND2X1    | Logic AND for two inputs             | Z = (A \& B)   | 0      | 2          |
| OR2X1     | Logic OR for two inputs              | Z = A \& B     | 0      | 2          |
| NOR2X1    | Logical NOR                           | Z = \overline{A \& B} | 1      | 2          |
| MX2X1     | Two inputs multiplexer with inverted output | Z = (S \& B) \& (\overline{S} \& \overline{B}) | 2      | 6          |
| XOR2X1    | Logic exclusive OR for two inputs     | Z = (A \oplus B) | 5      | 8          |
| XNOR2X1   | Logic exclusive NOR for two inputs    | Z = (A \oplus B) \& (\overline{A} \oplus \overline{B}) | 4      | 8          |
| AOI2X1    | Logical inverted OR of one AND gate and an additional input | Z = (A_0 \& A_1) \& B_0 | 1      | 4          |
| NAND2X1   | Logical NAND of two inputs            | Z = \overline{A \& B} | 1      | 2          |
| NOR3X1    | Logical NOR of three inputs           | Z = \overline{A \& B \& C} | 2      | 4          |
| INVX      | Logical inversion of single input     | Z = \overline{A}  | 1      | 0          |

Abbreviation: MOSFET, Metal-Oxide-Semiconductor Field Effect Transistor.
gates file consists of several logic cells some of which are listed in Table 2. The characterisation procedure for this proposed design was implemented at the schematic level by utilising Cadence spice Spectre. This characterisation process provides the required information for the memristor-based library that has been used by the Synopsys synthesis compiler to estimate the design area, delay, and power consumption. This information represented the design logic function and area. It also includes measurements of the design's input/output capacitance, delay, and power consumption. All this information was generated from the simulation of the memristor-based cells at the schematic level.

4.3 | Case study 2

In this case study, the proposed framework was applied to implement a memristor-based median filter, which was manually implemented and tested only at the Cadence Virtuoso schematic level and previously published in [19]. Image processing is very useful and has been extensively used in the areas of medicine, film and video production, photography, remote sensing, military target analysis, and manufacturing automation and control [29, 30]. These applications usually require bright and clear images or pictures. Hence, corrupted, or degraded images need to be processed to improve human interpretation, enhance visual pictorial information, and modify the data structure used for image representation to optimise it for data storage, transmission, or other representations for autonomous machine perception.

The main goal of any enhancement method is to obtain a more suitable result compared to the original.

Digital images are represented as 2D arrays of numbers, where the value of each entry corresponds to the grey-scale value of a pixel, ranging between 0 and 255 (255 being white). Thus, image enhancement techniques are transformed into 2D filtering operations. The 2D median filter replaces the value of each element based on the median value of its neighbour. The Sxy is a neighbourhood concept with eight elements immediately surrounding the median element. Thus, the mathematical representation of an image g (x; y) in the median filtering process is described as follows:

\[
S_{xy} = median\{g_s\} \quad (s, t) \in S_{xy}
\] (6)

The implementation of a memristor-based median filter has two phases. The first phase is the schematic-level implementation. At this stage, the median filter is manually designed and a 3 × 3 window is applied to verify the functionality of the proposal. In the second phase, due to the high design complexity, automated synthesis tools are required to make reliable and accurate simulations. Therefore, using a standard memristor cells library is essential to improving the accuracy of synthesis tools when they estimate power, area, and delay. Thus, the memristor cells involved in schematic
implementation are characterised to create a memristor-based standard cell library.

4.3.1 Schematic level implementation

The sorting mechanism in this technique is to find the median pixel from the surrounding neighbourhood pixels. The execution steps of memristive median circuit detects the median pixel in a $3 \times 3$ window, and the simulation results for the circuit are shown in Figure 6. This design was implemented using seven three-input 8-bit memristor-based comparators. Each of these comparators consists of three memristor-based two-input 8-bit magnitude comparators.

The two input 8-bit comparators were implemented as illustrated in Figure 7, with two 4-bit memristor-based magnitude comparators to compare between two pixels (eight bits for each input). The schematic of this comparator is displayed in Figure 8, and it was implemented based on a memristor-based MRL logic structure as shown in Figure 9. The outputs of the two 4-bit comparators were compared again with those of the 2-bit comparator to find the largest pixel value between the two inputs. Then only one output value from the 2-bit comparator was split between two multiplexers and the other output was connected to the selector of the first multiplexer to decide which pixel had the maximum value, and the same output was inverted and connected to the second multiplexer to select the pixel with the minimum value.

The proposed filter proceeds with nine inputs and determines the median value among them. This proposed architecture of the memristor-based median filter design was implemented and tested with Cadence Virtuoso environment at the schematic level using the memristor model presented in Verilog-A [21] and the parameters utilised for this model are shown in Table 1.

4.3.2 Automatic implementation

To prove the functionality of the proposed filter, the first step was to describe the behaviour of the median filter algorithm that was implemented at the schematic level using Verilog HDL and simulate it using the Cadence NG-Verilog-XL simulator. Unfortunately, Verilog only reads and writes ASCII character files. Therefore, it is not capable of reading images in standard formats, such as BITMAP or JPEG, directly from disk [30]. To resolve this problem, it is necessary to define a new image format to be used with a design test bench. The new image must be a HEX file that only contains information about RGB/greyscale vectors for each pixel of the input image. The data from hex-files are applied as stimuli to the point operations blocks described in Verilog language. The HEX characters are then elegantly converted to binary format by the Verilog HDL simulator.

In this part, the median filter implemented in Verilog was a behavioural model that removes the ‘salt and pepper’ noise of an input image and outputs the filtered image. The filtered image is then compared to an expected result that was created using the same filtering process in MATLAB for verification. The proposed filter flow chart of design verification is described as shown in Figure 10. The development steps of the proposed method in the Verilog behavioural model include four modules:

M1, M2, M3, and M4. M1 and M4 were set to load image data to memory and to write the filtered image data to a file. However, M2 was designated to buffer pixels, read addresses,
filter input pixels, and sends out noise-free data, while M3 was designed to generate addresses. In addition, to achieve an efficient processing time and reduce power dissipation, all these implementation stages were pipelined under a unique clock signal. The flow chart of the verification process for the design is shown in Figure 10. In the second step, as shown in Figure 3a, after testing the design at the behavioural level, the implemented RTL is synthesised to the gate netlist level with the aid of a Synopsys Design Vision compiler. The design compiler uses a standard library that contains all the information about the characteristics of logic cells to generate a final CMOS-based gate netlist file.

**Figure 8** Schematic view of the implemented 4-bit memristor-based magnitude comparator

**Figure 9** Implemented memristor-based 4-bit magnitude comparator

**Figure 10** Flow chart of memristor based median filter design verification. HDL, hardware description language

**Figure 11** Performance of designed filter for boat, camera man and houses 512 × 512 images contains salt and pepper with different noise density ratios: (a) boat with 30% noise; (b) camera man with 20% noise; (c) houses with 10% noise; (d) filtered boat image; (e) filtered camera man image. (f) filtered houses image
The proposed schematic and behavioral levels of the memristor were drawn using Cadence Virtuoso with a 65 nm cell library. The RRAM and ReRAM median filters at the schematic level were simulated using Verilog results for the implemented memristor.

In this part, the filtering process of the proposed median filter is thoroughly inspected to realise the logic cells that were elaborated on in the CMOS-median-filter-based design. After the logic cells are produced by the Synopsys synthesis compiler, equivalent memristor-based logic gates are implemented at the schematic level, tested, and characterised using the MRL design method. Hence, in this stage of the design, the characterisation process for memristor-based logic cells is obtained to build a standard memristor-based library for the Synopsys synthesis compiler, as presented in Figure 3b. The most important characterised cells involved in the proposal are AND, NAND, OR, NOR, multiplexer, and the other defined Boolean function circuits listed in Table 2. The built library provides a synthesis tool with information about cell logic function, area, input/output capacitance, delay, and power consumption.

### 4.4 Simulation results

#### 4.4.1 Performance results

In this part, the filtering process of the proposed median filter is evaluated on both schematic and behavioural levels. The simulation results for the implemented memristor-based median filter at the schematic level were drawn in Cadence Virtuoso with a 65 nm cell library. The RRAM and ReRAM filters were tested, and characterised using the MRL design method. Simulation results for the implemented memristor were shown in Figure 7. Then the implemented schematic was converted to the behavioral level, where it was simulated with Verilog-XL and NC-Verilog and synthesized with a Synopsys compiler before being tested for image denoising. In this test, standard 512 × 512-pixel images (boat, cameraman, and houses) with various levels of salt and pepper noise density ratios (10% up to 50%) were tested with the designed memristive median filter. The filter successfully removed the noise from the distorted images. Samples with superimposed salt and pepper noise and recovered images are shown in Figure 11. Peak signal to noise ratio (PSNR), mean square error (MSE), and mean absolute error (MAE) were measured to evaluate the quality of the recovered images and calculated as follows:

\[
MSE = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (S_{ij} - F_{ij})^2}{M \times N},
\]

where \( S_{ij} \) represents the noise-free image, \( F_{ij} \) is the recovered image, the numbers of image columns and rows are represented \( M \) and \( N \), respectively, and \( M \times N \) is the total number of pixels.
pixels in the image. Moreover, PSNR can be determined using the following equation:

\[ PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \]  

(8)

The sample word length for each architecture was 8 bits with a 3 × 3 window size. The simulation results in Table 3 show that the memristor-based adder/subtractor can reduce power, area and delay compared with the implemented CMOS-based adder/subtractor. In Table 4 it can be seen that the memristive median filter is the most time- and power-efficient design when compared to an efficient implementation of 1-d median filter (EIMF) [31] and a low-power architecture for the design of a low-power architecture for a one-dimensional median filter (LPAMF) [32]. The power consumption reduced by 16.25% compared with the lowest power consumed by other designs, as shown in Table 4. Compared to the equivalent CMOS design, the area of the proposed architecture is significantly reduced by 16.82%. The visual simulation results for noisy images (10% up to 30%) recovered by the proposal filter are displayed in Figure 11. Tables 5–7 summarise the quantitative restoration results of PSNR, MSE, and MAE for boat, cameraman, and house images, respectively.

4.4.2 Monte Carlo simulation results

In this proposal, the simulation results which were presented in the previous section exhibited an ideal outputs, and also the implemented memristor-based logic cells with the same parameters and function are perfectly matched. However, the process variations on memristor model parameters might be a reason for consequential degradation. Therefore, understanding the impact of process variation for the utilised technology node is important, to know the amount of variation that cells-based memristor RRAM devices can tolerate without any fanout and degradation in power and delay.

| TABLE 7 Performance parameters for different salt and pepper noise variations applied on the house image |
|---|---|---|---|---|---|
| **House image** | 10% | 20% | 30% | 40% | 50% |
| Noise ratio | MSE | PSNR | MAE |
| __ | __ | __ | __ | __ | __ |
| __ | __ | __ | __ | __ |
| __ | __ | __ | __ | __ |

| TABLE 8 The power consumption and delay of memristor based logic gate including the effects of process variation for 180 and 65 nm technology |
|---|---|---|---|---|---|
| **Gate** | **180 nm** | **65 nm** | **180 nm** | **65 nm** |
| | P (μW) | Tc (Ps) | Tr (Ps) | P (μW) | Tc (Ps) | Tr (Ps) |
| __ | __ | __ | __ | __ | __ | __ |
| __ | __ | __ | __ | __ | __ | __ |
| __ | __ | __ | __ | __ | __ | __ |
| __ | __ | __ | __ | __ | __ | __ |
| __ | __ | __ | __ | __ | __ | __ |

Abbreviation: SD, standard deviation.

**FIGURE 12** Hierarchical procedure for process variation simulation. Simulation executes for 4-bit memristor-based magnitude comparator, 8-bit comparator, 3-input 8-bit comparator, and finally median filter circuit
Table 8 shows the statistical analysis including process and mismatch effects on CMOS inverter and memristor based logic gates. Two technology nodes at 180 and 65 nm were utilised to apply statistical spectre simulation to determine process variation on power consumption and delay for the proposed characterised memristor-based logic cells. An ocean script has been written in Cadence Spectre to perform Monte Carlo simulation on power consumption and delay of the proposed large scale circuits. The Monte Carlo simulation procedure for proposed memristor based median filter circuit is achieved in a hierarchical order, as shown in Figure 12. The first step of the Monte Carlo simulation was performed on memristor based logic gates such as OR, AND, NAND and NOR with 180 and 65 nm technology. In the second step, the mean (ME), standard deviation (SD), and number of iterations runs (N) in power consumption and delay for large scale circuits (8-bit adder/subtractor and 2D memristive median) implemented in this proposal are achieved. The voltage variation results, using Cadence Analog Statistical Analysis for inverter, memristor based AND, NAND, OR, NOR, and XOR are shown in Figure 13. The statistical analysis indicates that a large circuits would not tolerate the fabrication standards and would not function as designed. Therefore, buffers are inserted at the outputs where the voltage dropped to correct the degradation issue. In Figure 14a input voltages for A, B, and Cin pins, and in Figure 14b output voltage variation for optimised 1-bit adder/subtractor circuit, and Figure 14c untolerated output voltage for 1-bit adder/subtractor circuit are exhibited.
4.4.3 | Areas for further research

Further research is required of the characterization level of memristor cells for large-scale designs. Furthermore, characterisation of memristor circuits and the addition of more memristor-based logic cells to our devised cell library is preferred. Moreover, future research should continue to develop the mapping methodology of Boolean logic circuits on memristor crossbar array.

5 | CONCLUSION

In conclusion, this framework is a general methodology for designing large-scale CMOS/memristor-based circuits for digital logic. In particular, in this method MATLAB, a high definition language (HDL) simulator, the Cadence Virtuoso design environment, and Synopsys software were utilised in this framework. A low-power and high-speed memristor-based parallel 8-bit adder/subtractor and 2D memristive median filter were designed with RRAM and ReRAM devices. They were tested and verified in Cadence Virtuoso, Verilog-XL, Synopses Design Vision, and MATLAB. The low-power, low-area, and high-speed performances were achieved by generating a standard memristor-based cell library. The simulation results and verification process proved that the designed memristive behavioural model was able to restore original images from distorted ones with 10%–30% salt and pepper noise. The proposed design shows very significant enhancement in power consumption and delay compared to equivalent CMOS architecture, EIMF, and LPAMF designs. Compared to the equivalent CMOS design, the area of the proposed architecture is significantly reduced by 52.79%.

ORCID

Arash Abjadi https://orcid.org/0000-0001-5094-5967

REFERENCES

1. Razavieh, A., Zeitzoff, P., Nowak, E.J.: Challenges and limitations of CMOS scaling for FinFET and beyond architectures. IEEE Trans. Nanotechnol. 18, 999–1004 (2019).
2. Malinowski, A., et al.: What is killing Moore’s law? Challenges in advanced FinFET technology integration. In: Proceedings of the 2019 MIXDES-26th International Conference on Mixed Design of Integrated Circuits and Systems, pp. 46–51. Rzeszow, Poland (2019).
3. Boucart, K., Ionescu, A.M.: Double-gate tunnel FET with high-oxide dielectric. IEEE Trans. Electron Dev. 54(7), 1725–1733 (2007).
4. Agrus, G., et al.: Two-terminal carbon nanotube programmable devices for adaptive architectures. Adv. Mater. 22(6), 702–706 (2010).
5. Kedzierski, J., et al.: Epitaxial graphene transistors on SiC substrates. IEEE Trans. Electron Dev. 55(8), 2078–2085 (2008).
6. Strukov, D.B., et al.: The missing memristor found. Nature. 453(7191), 80 (2008).
7. Zhang, Y., et al.: Synaptic characteristics of ag/agniobse/ta-based memristor for pattern recognition applications. IEEE Trans. Electron Dev. 64(4), 1806–1811 (2017).
8. Gergel-Hackett, N., et al.: A flexible solution-processed memristor. IEEE Electron. Device Lett. 30(7), 706–708 (2009).
9. Esphaghan, K., et al.: Memristor MOS content addressable memory (MCAM): hybrid architecture for future high performance search engines. IEEE Trans. Very Large Scale Integr. Syst. 19(8), 1407–1417 (2011).
10. Li, H., Chen, Y.: An overview of non-volatile memory technology and the implication for tools and architectures. In: Proceedings of the 2009 Design Automation Test in Europe Conference, pp. 731–736. Nice, France (2009).
11. Sarmiento-Reyes, A., et al.: A CAD-oriented simulation methodology for memristor circuits. In: Proceedings of the 2016 IEEE 7th Latin American Symposium on Circuits Systems (LASCAS), pp. 103–106. Florianopolis, Brazil (2016).
12. Xie, L., et al.: A mapping methodology of Boolean logic circuits on memristor crossbar. IEEE Trans. Comput. Aided Des. Integrated. Circ. Syst. 37(2), 311–323 (2018).
13. Ramasubramanian, S.G., et al.: SPINDLE: SPINTRonic deep learning engine for large-scale neuromorphic computing. In: Proceedings of the 2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 15–20. La Jolla, CA (2014).
14. Halawani, Y., et al.: Memristor-based hardware accelerator for image compression. IEEE Trans. Very Large Scale Integr. Syst. 27, 1–5 (2019).
15. Marranghello, F.S., et al.: Four-level forms for memristive material implication logic. IEEE Trans. Very Large Scale Integr. Syst. 36, 2749–2758 (2018).
16. Marranghello, F.S., et al.: Improved logic synthesis for memristive stateful logic using multi-memristor implication. In: Proceedings of the 2015 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 181–184. Lisbon, Portugal (2015).
17. Nguyen, H.A.D., et al.: Synthesising HDL to memristor technology: a generic framework. In: Proceedings of the 2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), pp. 43–48. Beijing, China (2016).
18. Bhurana, B.P., Manohar, B.R., Bhaskaran, V.S.K.: Standard cell characterisation for reversible logic. In: Proceedings of the 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), pp. 534–538. Ghaziabad, India (2016).
19. Sasi, A., et al.: A memristive TaOx-based median filter design for image processing application. In: Proceedings of the 2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), pp. 85–88. Prague, Czech Republic (2018).
20. Chen, P.-Y., Yu, S.: Compact modeling of ram devices and its applications in 1t1r and 1s1r array design. IEEE Trans. Electron Dev. 62(12), 4022–4028 (2015).
21. Siemon, A., et al.: Simulation of TaOx-based complementary resistive switches by a physics-based memristive model. In: Proceedings of the 2014 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1420–1423. Melbourne, VIC (2014).
22. Deng, L., et al.: Energy consumption analysis for various memristive networks under different learning strategies. Phys. Lett. 380(7–8), 903–909 (2016).
23. Kvatinsky, S., et al.: MRL-memristor ratioed logic. In: Proceedings of the 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications, pp. 1–6. Turin, Italy (2012).
24. Guckert, L., Swartzlander, E.E.: MAD gates-memristor logic design using driver circuitry. IEEE Trans. Circ. Syst. II. 64(2), 171–175 (2017).
25. Amirsoleimani, A., Ahmadi, M., Ahmadi, A.: Logic design on mirrored memristive crossbars. IEEE Trans. Circ. Syst. II. 65(1), 65 (2018).
26. Cho, K., Lee, S.-J., Eshraghian, K.: Memristor-CMOS logic and digital computational components. Microelectron. J. 46(5), 214–220 (2015).
27. Cong, J., Xiao, B.: mFPGA: a novel FPGA architecture with memristor-based reconfiguration. In: Proceedings of the 2011 IEEE/ACM International Symposium on Nanoscale Architectures, pp. 1–8. San Diego, CA (2011).
28. Huang, R., et al.: Resistive switching of silicon-rich-oxide featuring high compatibility with CMOS technology for 3D stackable and embedded applications. Appl. Phys. A. 102(4), 927–931 (2011).
29. Gonzalez, R.C., & Woods, R.E.: Digital Image Processing. Prentice Hall, Upper Saddle River, NJ (2002).
30. Esakkirajan, S., et al.: Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Process. Lett. 18(5), 287–290 (2011)

31. Moshnyaga, V.G., Hashimoto, K.: An efficient implementation of 1-D median filter. In: Proceedings of the 2009 52nd IEEE International Midwest Symposium on Circuits and Systems, pp. 451–454. Cancun, Mexico (2009)

32. Chen, R.-D., Chen, P.-Y., Yeh, C.-H.: A low-power architecture for the design of a one-dimensional median filter. IEEE Trans. Circ. Syst. II. 62(3), 266–270 (2015)