Int. J. Electron. Commun. (AEÜ) 84 (2018) 331–341

Contents lists available at ScienceDirect

Int. J. Electron. Commun. (AEÜ)

journal homepage: www.elsevier.com/locate/aeue

Regular paper

Design of a memristor based fuzzy processor

Mahdi Tarkhan, Mohammad Maymandi-Nejad*

Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Keywords:
Analog storage
Fuzzy processor
Memristor crossbar
Reconfigurable fuzzy logic controller

ABSTRACT

This paper presents a mixed-signal hybrid CMOS/Nano circuit to implement a fully programmable fuzzy processor that performs the zero-order Sugeno's algorithm. The programmability is incorporated in both the rule base and the membership function generator. Membership functions are stored in a memristor-based crossbar array, whereas digital memory is utilized to form a rule base. Each linguistic value whose shape can be independently chosen is stored in each row of the crossbar. Moreover, the adjacent linguistic values can have any order of overlapping ratio. Consequently, a very powerful and flexible fuzzifier is designed.

The analog signal processing blocks are designed in both current and voltage mode to simplify the circuit complexity while the digital circuitry is utilized to add programmability to the whole processor. A fuzzy logic controller with nine rules, two inputs, and one output was successfully simulated using HSPICE with 350 nm technology. Simulation results confirm the proper operation of the processor that can operate up to 10 MFLIPS (Mega Fuzzy Logic Inference Per Seconds) while consuming 3.49 mW (@vdd 3.3v). Furthermore, simulation results show that an equivalent precision of 6-bits is achieved in the output signal generation. The circuit is laid out and it takes an area of approximately 0.614 mm^2.

1. Introduction

As a powerful soft computing tool, fuzzy logic has found its way to be used in a wide variety of applications in which the system behavior is too complex to build any rigorous mathematical model. Control systems, signal processing and consumer electronics are some of these applications in which fuzzy logic has been used [1–5].

The hardware implementation of Fuzzy Processors (FPs) proposed so far can be categorized into the analog, digital and mixed-signal approaches. Analog fuzzy processors are reported both in voltage [6–10] and current mode [11–17]. Despite offering low power processor with small die area, analog fuzzy processors have some limitations from the programmability point of view. For this reason, digital approach is the more commercially successful implementation of fuzzy processors. Digital circuits have much more storage capability compared to analog ones. The storage potential determines the processor capacity to store rules as well as the membership functions. Therefore, digital FPs offer higher programmability and reconfigurability [18–27]. Mixed-signal implementations combine the merits of digital and analog circuits together and present an alternative to fully analog circuits [8,17,28–31].

Analog FPs utilize analog circuitry to produce membership functions, and digital FPs use digital memory to do so. However, digital memories impose high power consumption and data latency to the whole system. As a consequence, it is highly preferable to design an alternative memory block to store the membership function with any desired shape but with a small layout footprint as well as low power consumption. Memristor-based crossbar memory array, and hybrid crossbar/CMOS architectures are promising solutions to address the issues of the conventional transistor based memories [32,33]. Non-volatile property and high storage density in a crossbar array make such an architecture a competent candidate to be used as an analog memory to store membership functions. Since memristors are fabricated in nanoscale and are passive elements, power consumed by the crossbar structure is significantly lower than the conventional CMOS memories, and also they occupy much smaller die area [34]. However, the current flowing through the sneak path is an essential problem for such passive resistive memory arrays and causes the array to be nonfunctional.

Merrikh-Bayat et al. has utilized a crossbar to store fuzzy membership functions [35]. Their proposed method has the capability of storing membership functions in memristors, but their structure suffers from the sneak path problem. Furthermore, acceptable matching between feedback resistor and the memristance of memristor must be ensured, in order to have a satisfactory resolution. Further, Esmaili et al. proposed a novel adaptive real-time fuzzy modeling algorithm based on the Ink Drop Spread (IDS) concept [36]. The proposed algorithm was implemented with memristor crossbar structures.

In an inspiring research paper, Kim et al. proposed a CMOS compatible crossbar structure [37]. Their proposed method addresses the

* Corresponding author.
E-mail addresses: mahdi.tarkhan@mail.um.ac.ir (M. Tarkhan), maymandi@um.ac.ir (M. Maymandi-Nejad).

https://doi.org/10.1016/j.aeue.2017.10.039
Received 14 August 2017; Accepted 18 October 2017

sneak path problem using the self-current-rectifying property of the memristor.

In this paper, we propose a fuzzy processor architecture, inspired by [37], that takes advantage of the programmability of the crossbar array to store membership functions with any shape and overlapping ratio. The rules are deployed in a digital storage, whereas rule evaluation is performed in an analog domain. The structure of our proposed inference engine makes it possible for the designer to define any rule with any combinations of input variables in the digital storage. Defuzzification, on the other hand, is performed in the analog domain to speed up the output generation process.

This paper is organized as follows; in Section 2, some basic concepts about the fuzzy processor and memristors are summarized. In Section 3, we describe our proposed architecture. Simulation results are presented in Section 4, and finally in Section 5, we draw conclusion.

2. Fuzzy logic processor and memristor

In this section, the relevant principles of the fuzzy processor and memristors are discussed briefly.

2.1. Fuzzy processor (FP)

In fuzzy logic, the definitions of quantities are different from classic math. In fact, a fuzzy number is a tool to approximate quantities by words (linguistic values such as cold, mild, hot) as the humans do. Each linguistic value is defined through a membership function (MF) whose shape can be defined arbitrarily. A fuzzy system represents the dynamic behavior within the system by a set of linguistic rules based on expert knowledge, and produces output signals through fuzzy logic concepts. According to [38], the basic configuration of a fuzzy logic controller (FLC) is composed of four principal components: fuzzifier, rule base, inference engine, and defuzzifier, as shown in Fig. 1.

The fuzzifier measures the numerical values of the input signals, then converts them into corresponding membership degrees using linguistic values defined over the related universe of discourse. The linguistic values are represented by membership functions, and the control strategy is defined by "IF-THEN" rules. The inference engine utilizes the information stored in the rule base, and the membership degrees prepared by the fuzzifier to infer fuzzy control actions employing fuzzy implication and fuzzy logic concept. The defuzzifier does a scale mapping and converts the fuzzy output already generated by the inference engine to a crisp output value.

2.2. Memristor and crossbar array

Resistive random-access memory (RRAM or ReRAM) has shown great potential as a promising candidate for next-generation non-volatile memory. This is due to its sub-10 nm scale, high speed, low power consumption, and potential for 3D integration [39]. The storage elements in RRAMs work based on the resistive switching phenomena which are a reversible change in the metastable resistance of a dielectric material induced by an external electric field. Typically, the change in the resistance is non-volatile. These switching elements differ in switching mechanism, retention time, CMOS process compatibility, switching speed, high to low resistance ratio, and I-V curve characteristic [40].

The existence of memristor was first postulated by Chua in 1971 [41]. A memristor is a particular type of resistive switching element that shows specific behavior characterized by Chua and Kang in [42]. The research interests in memristors have been growing since 2008, once Hewlett-Packard linked the relationship between resistive switching phenomenon and the memristor [33].

The device structure is depicted in Fig. 2. An active material is sandwiched between two electrodes. The electrical property of the film is regulated as the response to the history, and the direction of the current which flows through the device. After forming process, some conductive channels are formed through a special physical and chemical process, and the device is ready to show reversible and reproducible switching behavior. When current flows in one direction, the conducting channel is formed, and the device is entered to low-resistance state (LRS). This process is called "SET" process. On the other hand, a reverse bias ruptures the conducting channel and the device returns to the high-resistance state (HRS). This is called "RESET" process.

The resistance of the device is non-volatile. During the SET process, the resistance of the device can be set to any desired value by limiting the current flowing through the device which is called the *compliance current*.

The non-volatile multi-level storage capability of memristors, makes them a suitable candidate to store membership functions in a fuzzy controller. This way, membership functions with any shape and any overlapping ratio can be defined in the controller. The crossbar structure is the easiest method regarding the fabrication effort to build Nano-scale memristors. Moreover, it offers the maximum storage density that can be achieved. The crossbar consists of an array of the perpendicular bottom (bottom electrodes) and top nanowires (top electrodes) and a resistive switching material which is sandwiched between the electrodes. Each cross-point formed at the intersections of the top and the bottom electrodes corresponds to an individual memory cell [40]. However, the interconnected passive network structure also leads to sneak leakage currents that can severely distort the actual cell data. The read out value of a particular cell (colored green) is the combination of the saved data along with the added distortion as depicted in Fig. 3a. The sneak paths are a result of having open access to any of the cells in the array. To address this issue a "selector" element with rectifying behavior (e.g., diodes) have been used but with limited success [43].

The proposed crossbar array structure in [37] do not use any transistor or diode as a selector device. However, they take advantage of the inherent nonlinear current-voltage characteristic of memristors, which are made of amorphous Si (a-Si), to break the sneak path. The I-V curve of their proposed device is depicted in Fig. 3b. As it can be seen, the device has a self-rectifying behavior which is suitable to break the sneak path since at least one memory cell should be reversed bias in such a path (colored orange in Fig. 3a). Furthermore, their proposed crossbar architecture is CMOS compatible and storage density per unit area can also be maximized by stacking crossbar arrays.

3. Proposed architecture

Multi-level storage capability of memristors is exploited in this work to store membership functions. Each membership function is stored in a

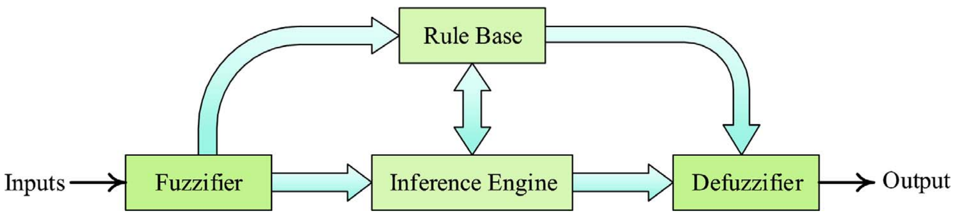


Fig. 1. Graphical representation of a fuzzy system. Inputs and outputs belong to the crisp universe, and internal signal processing is done by fuzzy logic concepts [38].

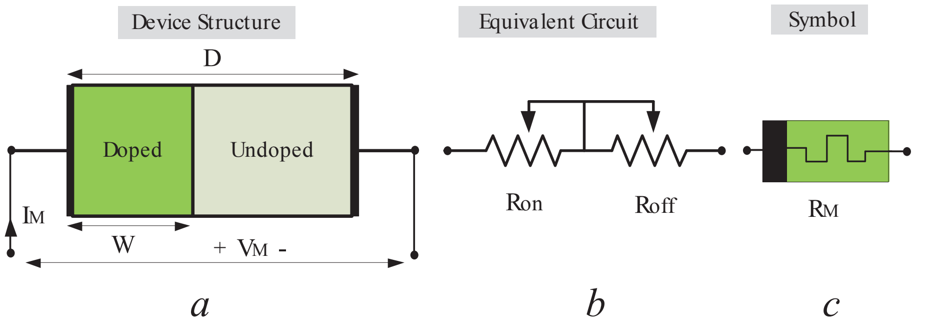


Fig. 2. a memristor structure which consists of doped (low resistance R_{on}) and undoped (high resistance R_{off}) regions. (b) Equivalent circuit of memristor comprising two series resistor. (c) Schematic symbol of memristor [33].

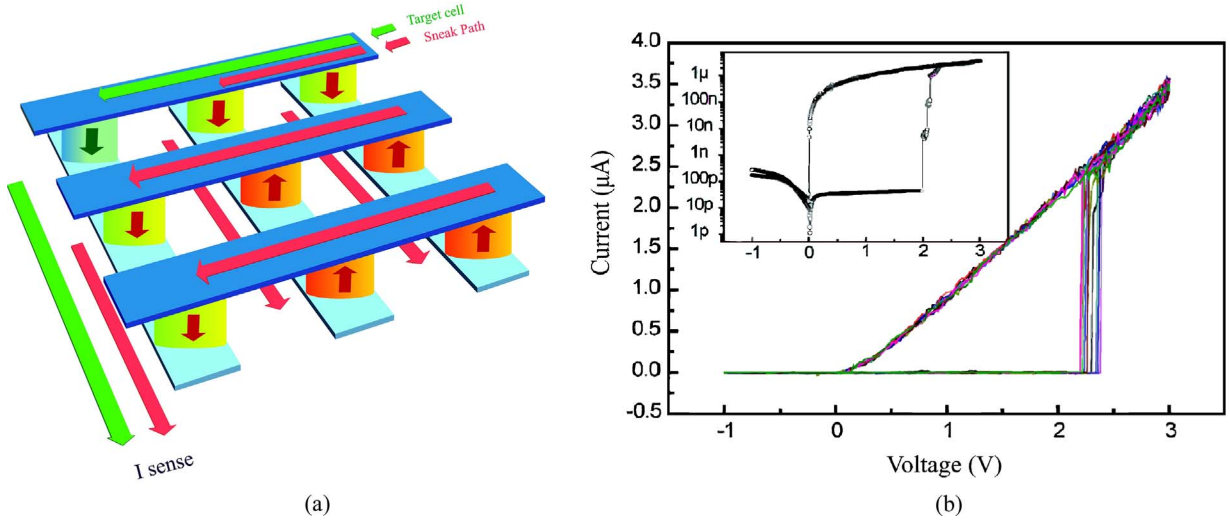


Fig. 3. The crossbar structure and the sneak path (a). The I-V characteristics of the memristor used in [37]. The state threshold voltage is 2.3 V while its reset voltage is −1.5 V. The device shows a rectifying behavior for negative voltages lower than the reset voltage (b).

row in the crossbar, hence, the resolution of the input variables is determined by the number of columns in the crossbar. The number of rows of the crossbar restricts the total membership functions which can be defined in the controller. For example, a controller with 6 × 64 crossbar can have input variables up to 6-bit resolution and also can store at most 6 membership functions. The number of input variables, on the other hand, is limited by the number of rows in the crossbar as well as the number of membership functions defined for each input variable.

As it was mentioned earlier, memristors are used to produce membership functions, and the membership degree is stored as resistance of the memristor. Therefore, a transformation from memristance which is in the range of $[R_{on},R_{off}]$ to a membership degree (which is in the range of [0,1]) is required. Since the number of members with partial or full membership (>0) is limited in each linguistic value, R_{off} and R_{on} are selected to produce zero and full membership degrees, respectively. This selection has a significant impact on the power consumption of the whole controller. In the next section, we will discuss the method of membership function generation in more details.

3.1. Fuzzifier

The structure of the proposed fuzzifier section is depicted in Fig. 4. The input variables are applied to the crossbar, one by one, waiting for a dedicated time interval determined by the counter. Meanwhile, one column in the crossbar corresponding to the selected input variable is activated by the column decoder. Column Control Block (CCB) and Row Control Block (RCB) are special circuits that bias the crossbar with appropriate voltages based on the operational mode of the controller. The controller operates in two modes, i.e. write and read, which are selected via a $R/\overline{W}$ signal. Fig. 5. shows the internal structure of CCB and RCB for a single column and row. Such a circuit should be duplicated for each row and column of the array.

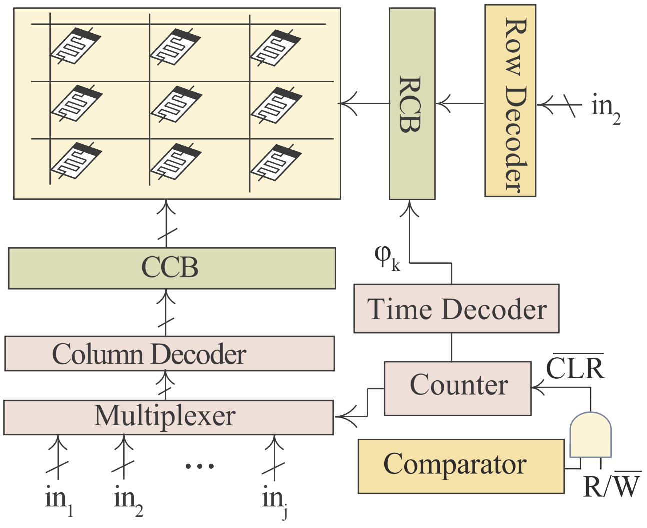


Fig. 4. The structure of our proposed fuzzifier. Digital inputs ($in_1 \cdots in_j$) are picked out by the multiplexer one by one, at the same time, column selection is done by the column decoder. The crossbar is biased by CCB and RCB.

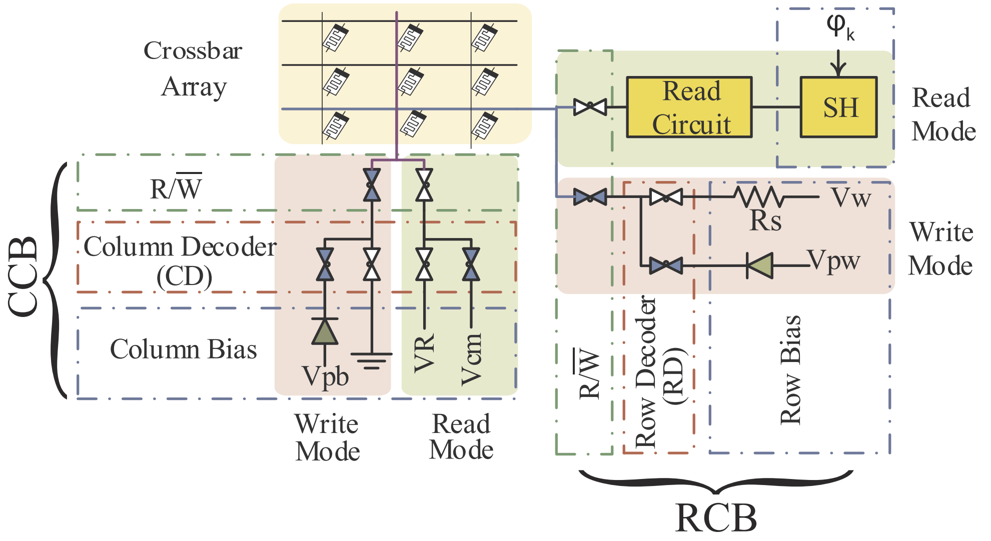


Fig. 5. The internal structure of CCB and RCB. In the read mode, CCB applies V_R to the selected column, and unselected columns are connected to V_{cm}. The information stored in the selected column is read simultaneously by the Read Circuit and is captured by the Sample and Hold (SH) in the proper time generated by the time decoder. In the write mode, the target memristor is selected by the row and column decoders and is connected to the write circuit.

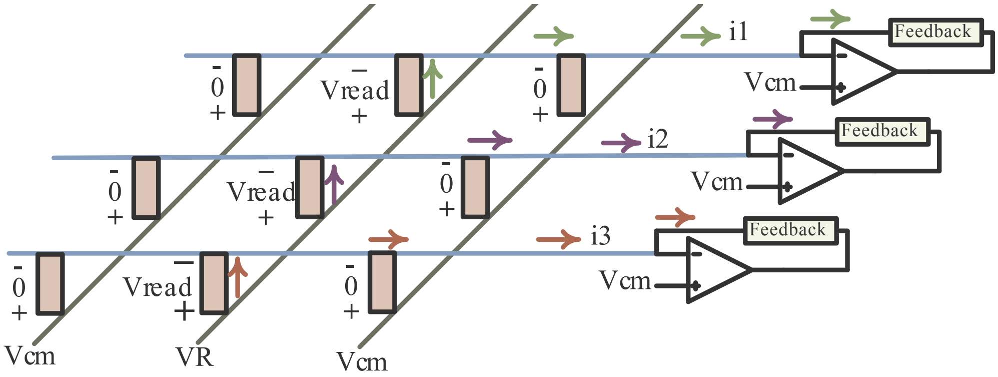


Fig. 6. Measurement diagram in reading mode. The membership degree of the selected member in the membership functions stored in the rows of the crossbar is extracted independently. Each row has its dedicated read circuit to catch the extracted membership values.

3.1.1. Read mode

In this mode, the information stored in the crossbar is read by proper biasing of memory cells. Input selection is done by a counter and a multiplexer. The column biasing is done by a column decoder and the CCB. The CCB connects the selected (unselected) column(s) to V_R (V_{cm}). At the same time rows are connected to V_{cm}, Fig. 6. illustrates the biasing diagram of the crossbar. As it can be seen, memory cells belonging to the selected column are biased by V_{read}, which is the difference between V_R and V_{cm}($V_{read} = V_R - V_{cm}$). Hence, a current will flow through the rows proportional to the memristance of the activated cells. No voltage is dropped on unselected memristors, and therefore, no current is generated by these cells. In this way, the sneak path cannot be created in the read mode. Eventually, the current flowing through the memristor can be calculated by (1)

$$I_{mem} = \begin{cases} \frac{V_R - V_{cm}}{R_{mem}} & \textit{for selected column} \\ 0 & \textit{for unselected columns} \end{cases} \tag{1}$$

The reading voltage (V_{read}) should be lower than the threshold voltage of the memristor, otherwise, it will change the memristance that is not desirable. As it was mentioned earlier, the current flowing through the selected memristors (which corresponds to the membership degree of the selected member) flows into the read circuit. The internal structure of a read circuit is depicted in Fig. 7. Every row has its dedicated read circuit, as a result, the membership degree of the selected input variable in all membership functions, defined in the controller, is read at the same time. However, only those membership degrees are valid which are related to the linguistic values defined for the selected input. The information extracted from other rows belongs to other inputs, and therefore, should be discarded. This process is accomplished by the sample and hold (SH) through the activation signal (φ_k) which is generated by a time decoder (Fig. 4) and is synchronized with the input selection through the shared digital counter. This way, as soon as an input variable is selected by the multiplexer, its associated sample and holds are also activated to catch the corresponding membership degree values.

In Fig. 7 the current mirror is used in the feedback path for signal repetition and transformation. At the last stage, the current is converted to voltage by PMOS transistors and is sampled at a specific time by the SH. Signal conversion from current to voltage leads to a faster transient behavior since gate-source voltage has small voltage swing due to the quadratic relationship between drain current and V_{gs}. Furthermore, the structure of the interface circuit between the fuzzifier and the inference engine in the voltage mode is simpler and more efficient from the routing effort point of view, otherwise, many current mirrors are required to reproduce membership degrees.

The fuzzification phase would be terminated as soon as all the input signals are evaluated. Since the controller has j inputs, j clock cycles are required to complete this phase. This process is controlled with the digital comparator depicted in Fig. 4 that reset the counter after j clock cycles.

3.1.2. Write mode

In this mode, the array is programmed to generate the desired membership functions. Since the memristance is a value within the range of [R_{on}, R_{off}], a linear relationship should be defined between the standard membership degree (in the range of [0, 1]) and the memristance. In order to achieve an architecture with a lower power

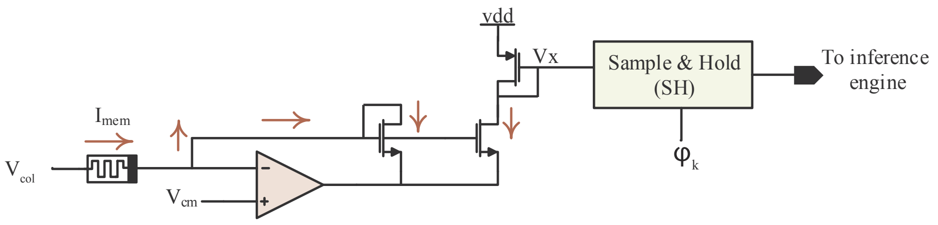


Fig. 7. The schematic of the read circuit. V_{cm} is the common mode voltage of the sense amplifier. V_{col} is the voltage applied by CCB to each column which is either V_R or V_{cm} based on the column decoder output. φ_k is the sample and hold activation signal generated by the time decoder.

consumption, R_{off} and R_{on} are selected to produce a zero and a full membership degree, respectively. In the reading mode, the information stored in the memristors (membership degree) is extracted as a current signal, therefore, the memristance should be adjusted so that the current flowing through the memristor in reading mode generates a proper membership degree. Eqs. (2)–(4) show such relations.

$$I_{min} = \frac{V_{read}}{R_{off}}, I_{max} = \frac{V_{read}}{R_{on}} \tag{2}$$

$$I_{mem} = I_{min} + (I_{max} - I_{min})\mu \tag{3}$$

$$R_{mem} = \frac{V_{read}}{I_{mem}} \tag{4}$$

where I_{min} and I_{max} are the minimum and maximum currents flowing through the memristors in the read mode, V_{read} is the reading voltage, and μ is the standard membership degree (in the range of [0,1]). For any desired membership degree, an appropriate current is calculated from (3), and the memristance is determined by (4). Since each membership function is stored in a row, programming memristors (resided in that row) with proper values calculated from (4) will lead to generation of the desired membership function in the reading phase.

In the write mode, as shown in Fig. 5, the selected row is connected to the writing voltage (V_w) through a series resistor (R_S), and unselected rows are connected to a protective voltage (V_{pw}). On the other hand, the selected column is connected to the ground, and unselected columns are connected to V_{pb} (protective voltage). Fig. 8 shows the voltage distribution of the crossbar in this mode. The parasitic current paths made by the half-selected devices in the unselected columns are blocked by external diodes at the outside of the array.

As a result, the programming current (I_{target}) flows through the selected memristor, and its amplitude is controlled by a series connected resistor (R_S). The resistance of the cell in LRS is determined by the amplitude and duration of the programming pulse. For more information about the writing scheme the reader can refer to [37]. Besides using a series resistor to control the compliance current, writing methods using feedback can also be used to have a more precise writing phase[44–48].

3.2. Inference engine

Inference engine evaluates the rules defined in the rule base utilizing the fuzzy logic concepts and the membership degree information already produced by the fuzzifier. Several inference methodologies with a different level of computational complexity are proposed in the literature [49,50]. However, the zero-order Sugeno's method is very popular for hardware implementation [10]. The rule base of this type of the controller consists of "IF-THEN" rules that relate fuzzy antecedents (IF part) with singleton consequent (THEN part) in the form of:

$$\text{IF } in_1 \text{ } is \text{ } A_1^i \text{ and } in_2 \text{ } is \text{ } A_2^i \text{ and. ..and } in_j \text{ } is \text{ } A_j^i \text{ THEN } y_i = K_i \tag{5}$$

where $in_k(k = 1,2,\cdots,j)$ and y_i are input and output variables, respectively, A_k^i are linguistic values defined by membership functions (fuzzy sets) in the ranges (universes of discourse) of inputs. K_i is a constant number, $i(1,2,\cdots,H)$ is the rule number and "and" is the fuzzy logic intersection operator.

In Sugeno type fuzzy system, the minimum operation is used as t-norm in the inference engine for the calculation of the intersection (and). Membership functions are stored into the crossbar array by proper programming of the memristor, and its information is extracted as current signal in the fuzzifier. Hence, in the inference engine, a minimum circuit calculation in current mode is required. However, as it is described under the fuzzifier section, the information of the membership degree's information is transformed to voltage signal by PMOS transistors and stored by sample and holds. Therefore, inference engine should be designed as a maximum voltage calculation circuit, since a lower current in the memristor means a higher voltage stored in the sample and hold. Fig. 9 depicts the inference circuit used in the proposed fuzzy processor. Such a circuit should be duplicated for every rule defined in the controller, to evaluate all the rules concurrently.

Suppose a controller with j input variables and m linguistic values defined for each of them. Then, a maximum voltage detection circuit with j input is required. It is a CMOS WTA[1] maximum circuit with multiple inputs[51–53]. The size of the multiplexer is defined by the number of membership functions (m). Analog multiplexers are controlled by a digital code stored in the rule base. The bit definition of such a code is depicted in Fig. 10. Each rule can be defined by a $(jn + L)$-bits code. Desired membership function defined in a universe of discourse of each input variable is selected by programming the n bits defined specially for each input variable (calculated from (6)). The least significant bits, L, represent the consequent part of the rule (K_i in (5)).

$$n \geqslant \log_2(m + 1) \tag{6}$$

For example, imagine a controller with two inputs, each with at most seven different membership functions (n = 3 from (6)). Then, a code such as 101-011-100 means:

$$\text{IF } in_1 \text{ } is \text{ } A_1^3 \text{ and } in_2 \text{ } is \text{ } A_2^5 \text{ THEN } y = 4$$

The proposed structure, offers the maximum level of programmability, and makes it possible to define every desirable rule in the controller. Furthermore, defining the special type of rules with some missing input variables in the antecedent part of the rule is also possible by setting the corresponding bits to zero. For example, A code such as 000-110-010 means:

$$\text{IF } in_1 \text{ } is \text{ } A_1^6 \text{ THEN } y = 2$$

Setting zero to a rule connects the output of the corresponding multiplexer to the ground (Fig. 9), which has no effect in the maximum voltage calculation. At the last stage of the inference engine (after maximum voltage calculation), the activation level of the rule is converted to current signal (IR_i in Fig. 9) by a PMOS transistor.

3.3. Defuzzifier

Each rule generates an action signal based on its activation level and the fuzzy singleton which is defined in the consequent part. However,

[1] winner-take-all

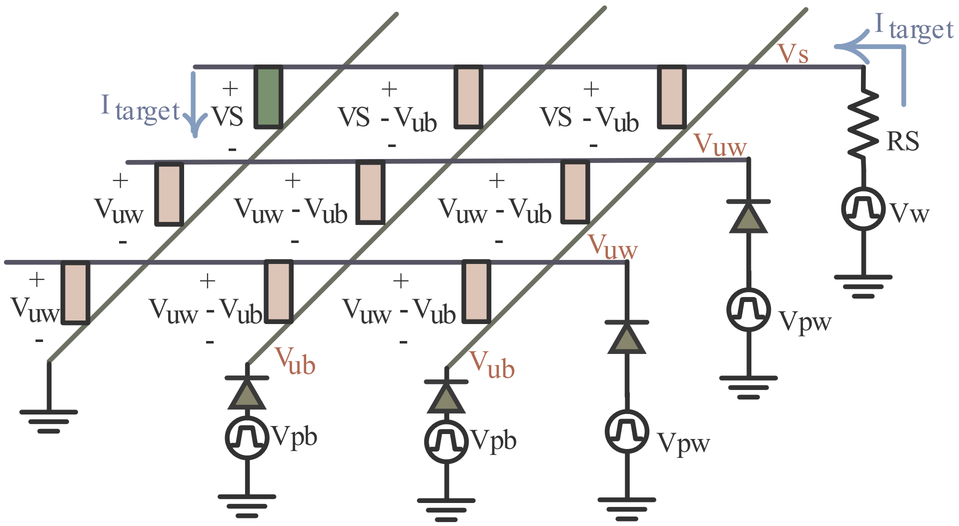


Fig. 8. Voltage distribution in the write mode. The selected row is connected to V_w (programming voltage) through R_S and the unselected rows are connected to V_{pw} (protective voltage). The selected column is connected to the ground and the unselected ones are connected to V_{pb}. V_{pw} and V_{pb} are protective voltages and are chosen carefully to diminish disturbances during the programming phase. External diodes block the current flowing through parasitic parallel paths [37].

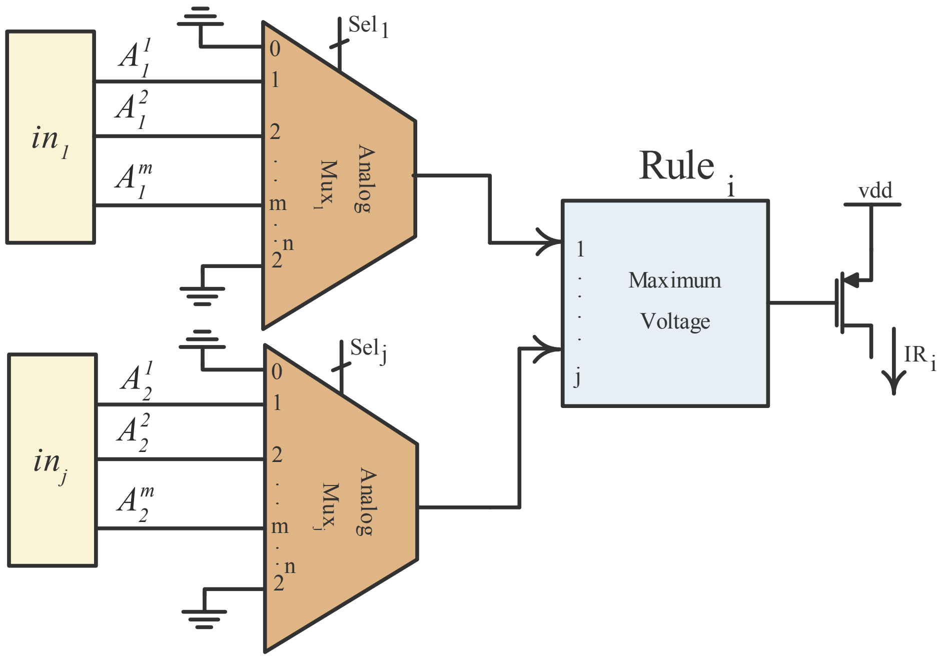


Fig. 9. Inference engine architecture for the *ith* rule defined in the rule base. A maximum voltage calculation circuit with *j* inputs is needed for a controller with *j* input variables. A_k^l is the *lth* membership function defined for *kth* input variable. The activation level of the *ith* rule (IR_i) is reproduced as a current signal by a PMOS transistor.

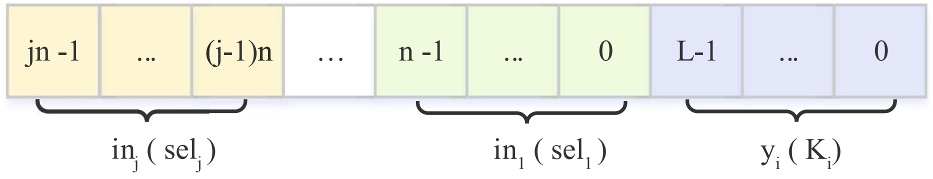


Fig. 10. Bit definition of the rules. Selection in analog multiplexers of the inference engine is done by In_i bits defined for each input variable. The fuzzy singleton is defined in consequent part of the rule and is stored in the first *L* bits of the rule.

because of fuzzy nature, these signals are inappropriate to be applied to a plant which works on crisp domain. Therefore, another transformation is necessary. Defuzzification section generates a crisp output signal based on the action signals produced by rules and their activation level applying a defuzzification method. While several approaches are proposed in the literature [49,50], the center of gravity method is the most popular and generates more accurate results. In this method, the final output is a weighted average over the rule's consequent values.

$$y_o = \frac{\sum_{i=1}^{H} \theta_i K_i}{\sum_{i=1}^{H} \theta_i} \tag{7}$$

where θ_i is the activation level of the *ith* rule, y_o is the final output crisp signal. H is the number of the rules defined in the rule base, and K_i is the constant number defined in the consequent part of the *ith* rule.

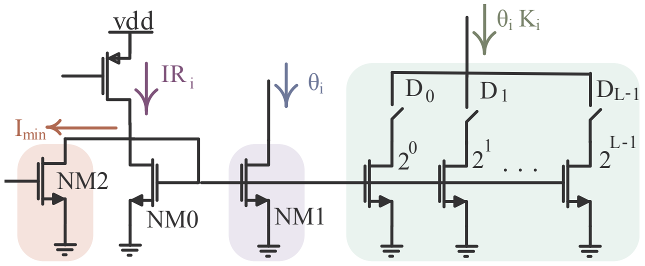


Fig. 11. The structure of the binary weighted current mirror used in the defuzzifier. Such a circuit should be repeated for each rule, but θ_i ($\theta_i K_i$) nodes should be connected together for summing purpose. Prior to the current replication, I_{min} should be subtracted from the activation level, otherwise the final output value would be incorrect. The D_i switches are controlled by the L bits defined in the rule code (as in Fig. 10).

Signal processing is done in the current mode, and the modified binary weighted current mirrors are used for the rule's output generation as depicted in Fig. 11. The numerator of (7) is a weighted sum of the rule's activation level. The scaling factor is specified as a constant (K_i) in the corresponding rule-bit definition. The denominator of (7) is simply the sum of the rule's activation level. Using the current mode techniques in this section of the controller allows saving adder circuits (required in the voltage mode) to calculate the numerator and denominator. Furthermore, summation in the current mode is simply accomplished by leading the current signals to a common low-impedance node.

Care should be taken when using the current signal as the membership degree. A linear relationship is defined between the current and the normal membership degree in (3). Using activation level (IR_i) directly in (7) would result in a wrong answer, since the weighted sum of the IR_i is not simply a scaled version of the standard membership degree. This is because of their different origin. While the minimum of IR_i is I_{min}, the membership degree in the standard definition is started from 0. To address this issue, I_{min} should be subtracted from IR_i as it is done by NM2 in Fig. 11. The structure of the defuzzifier is illustrated in Fig. 12. The numerator and the denominator of (7) are produced by connecting the corresponding signals together, while the controller's output signal is generated by an analog current mode divider. Using such a circuit in the last stage of the controller lets our FLC to be directly connected to the plant without requiring any DAC. Furthermore, it can easily be amplified (transformed) to very popular 4–20 mA (0–10 V) industrial signals.

4. Simulation results

To evaluate the performance of the proposed processor, a controller with two inputs, one output and nine rules is simulated. A 6 × 64 crossbar is also considered to store membership functions. This means two input variables with 6-bit resolution, each with three membership functions, are defined in the FLC. The consequent part of the rules is designed to have 16 different values (4-bit resolution). Therefore, rules are defined by (8) and stored as 8 bit codes in the rule base. Table 1 shows the rules defined in the controller.

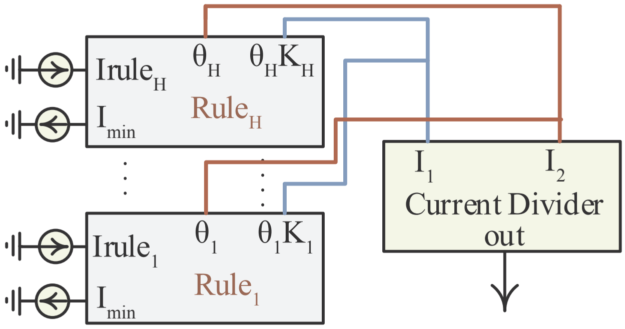


Fig. 12. The structure of the defuzzifier. Each rule produces two signals to be used by the divider circuit. θ_i is the modified activation level of the rule, and $\theta_i K_i$ is the action signal produced by *ith* rule.

Table 1
FLC rule definition; FLC has two inputs (in_1 and in_2), six membership functions ($A_1 \cdots A_3$ for in_1 and $B_1 \cdots B_3$ for in_2) and nine rules ($R_1 \cdots R_9$). The consequent part (K_i) of the rule is a 4-bit constant number of the following values: L = 1, M = 7, H = 15.

Input variable		in_1		
		A_1	A_2	A_3
in_2	B_1	L (R1)	L (R4)	M (R7)
	B_2	L (R2)	M (R5)	H (R8)
	B_3	M (R3)	H (R6)	H (R9)

$$\text{IF } in_1 \text{ is } A_i \text{ and } in_2 \text{ is } B_i \text{ THEN } y = K_i \tag{8}$$

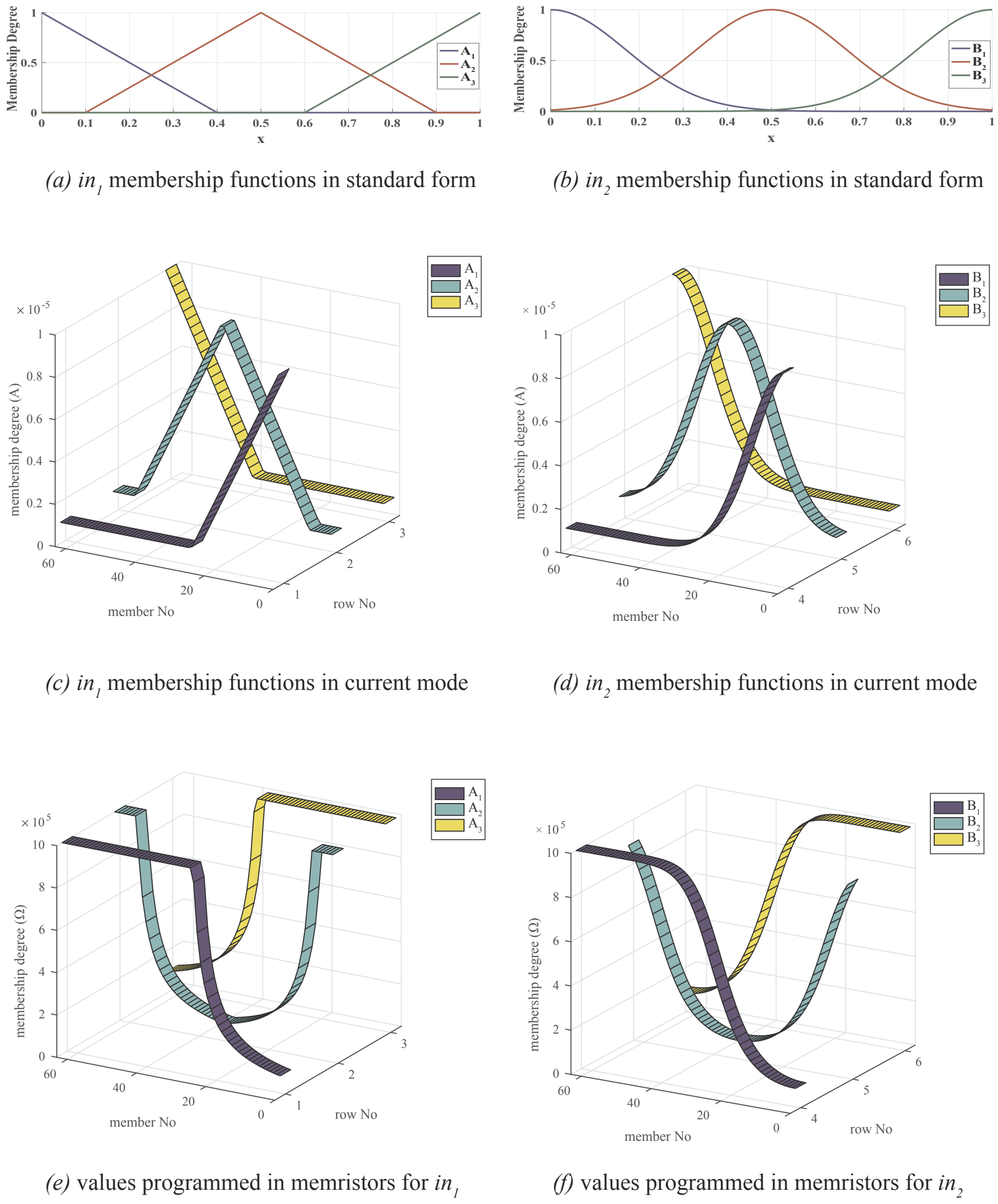
We have used HSPICE to simulate the proposed circuit utilizing 350 nm process. Several methods are presented in the literature to model the memristors. We use the model proposed in [54] for this purpose due to its better convergence and higher precision. Since the signals already produced by the modified binary current mirrors are in the current mode, an analog current mode divider proposed in [55] is used in the proposed circuit. The following values are also considered in the simulation environment:

$$V_w = 3.3\text{ V},\ V_{pw} = 3\text{ V},\ V_{pb} = 1.7\text{ V},\ V_{cm} = 1.5\text{ V},\ V_R = 2.5\text{ V},\ R_{off} = 1\text{ M}\Omega,\ R_{on} = 100\text{ K}\Omega$$

In Fig. 13 the linguistic values defined for each input variable is depicted. We set triangular membership functions for in_1 (Fig. 13a) and Gaussian for in_2 (Fig. 13b). As mentioned earlier, membership functions are stored in each row of the controller, and the crossbar is composed of 64 columns and 6 rows. Hence, 6 different membership functions are defined each with 64 members. The member selection is done by 6-bit input variables, and the controller has two inputs (Fig. 4). Hence, the universe of discourse is mapped into the range of [0, 63] for all inputs. After column selection, in the reading mode, the membership degree information is extracted as a current signal as described in the reading mode. This membership degree is in the range of I_{min} $(=1\mu\text{A})$ and $I_{\max}(=10\mu\text{A})$ (from (2)), which correspond to "no" and "full" membership (based on (3)). In Fig. 13c and Fig. 13d the membership functions defined in each row of the crossbar is depicted in the current mode which is similar to the standard form. However, the memristors should be programmed properly to produce such membership functions based on their bias conditions. Selected memristors are biased by a reading voltage (V_{read}) equal to 1 V in the reading mode, which is smaller than the memristor threshold voltage, thus, memristance will not change. For the abovementioned values the memristance values to be programmed in the memristors are calculated from (4) as depicted in Fig. 13e and f.

In the writing mode, the maximum negative voltage drops on the unselected devices is almost −0.8 V, which is less than the reset voltage of the memristor (−1.5 V). This negative bias breaks the sneak path due to the current rectifying property of the switching element. It cannot disturb the state of the unselected memristors as well [37].

To evaluate the performance of the controller, the in_1 and in_2 are set randomly to 8 and 40, respectively, and the current signals flowing in different parts of the circuit are extracted from the transient simulation. Fig. 14a shows the transient curve related to current signal flowing through the memristors and the read circuit. In this figure, the membership values defined for the aforementioned input variables are depicted for all six membership functions defined in the controller. It should be noted that the information is extracted in different time based on the counter incorporated in the controller. In one cycle the membership valued related to input 1 (member number 8) is extracted and in the next clock cycle information related to the next input (member



(a) in₁ membership functions in standard form

(b) in₂ membership functions in standard form

(c) in₁ membership functions in current mode

(d) in₂ membership functions in current mode

(e) values programmed in memristors for *in₁*

(f) values programmed in memristors for *in₂*

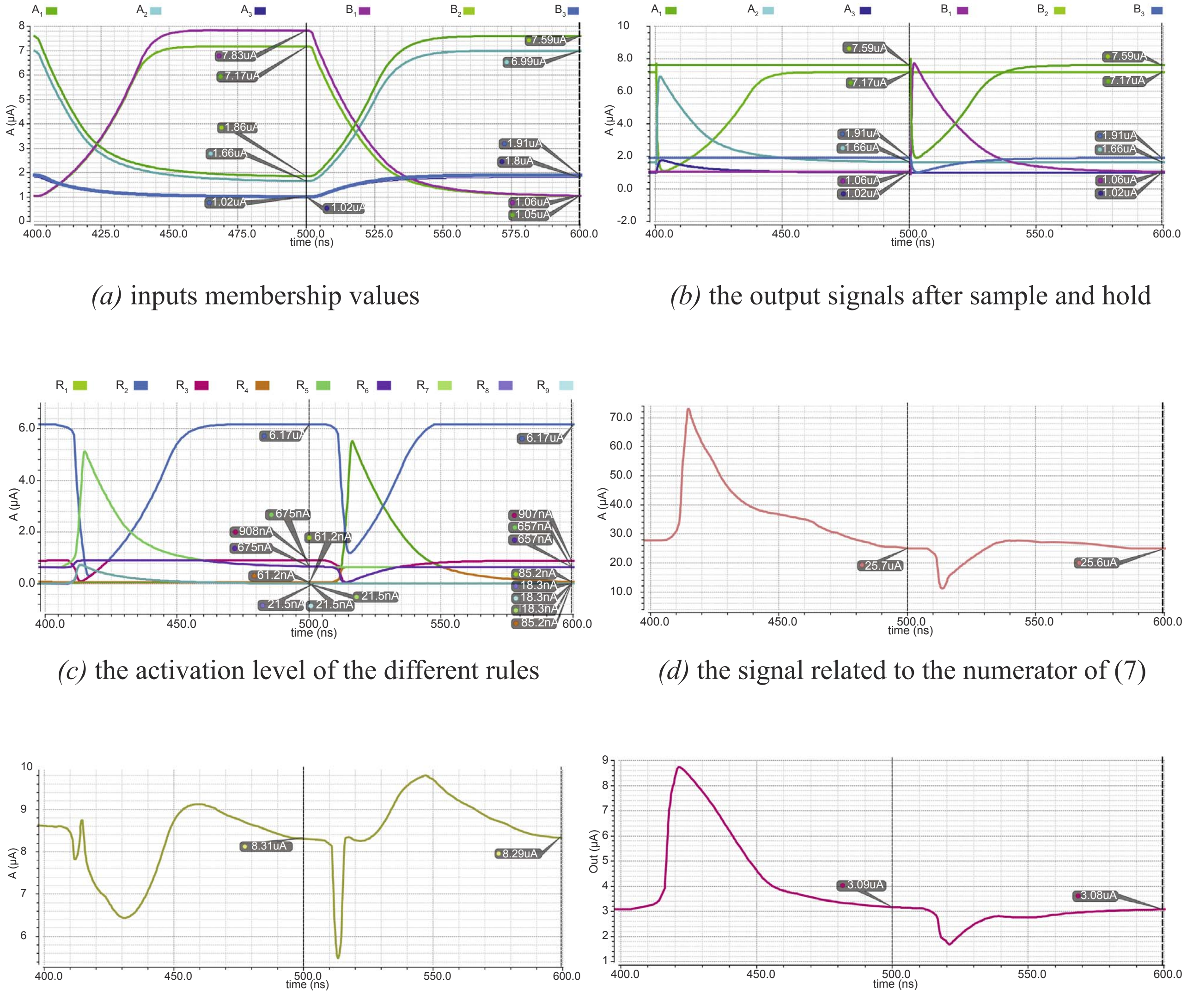
Fig. 13. Membership function defined in the controller.

number 40) is read from the crossbar for all membership functions defined in the controller. However, the signals related to the 8th column which is related to in_1 should be captured in one cycle and the other signals which is corresponds to in_2 and is extracted from 40th column should be sampled in next cycle as depicted in Fig. 14b.

The modified activation level (θ_i) of all nine different rules which are a value within the range of 0 and 9 uA are plotted in Fig. 14c. Smaller values means fewer activation level of the rule and vice versa. In the last section of the controller, the defuzzifier is responsible for generating a crisp output based on the center of gravity method. In Fig. 14d and e, the signals related to the numerator and the denominator of the defuzzifier are plotted. The output signal settles to final value in approximately less than 0.1 μs (Fig. 14f). As a result, it can be inferred that the processor can operate at speeds up to 10 MFLIPS, which is enough for many real time applications.

With two inputs and 64 members each, the simulation should be repeated 4096 times. The control surface resulted from system-level simulation in MATLAB is plotted in Fig. 15a. The universe of discourse of inputs is considered to be in the range of [0,1]. Functional verification of the design is also accomplished by MATLAB and HSPICE co-simulation witch is done 4096 times. Fig. 15b shows the control surface obtained from the co-simulation which shows a good match with the results obtained from the system-level simulation. It should be noted that the inputs are considered to be 6-bit digital signals which is in the range of [0,63]. Resolution of the output signal is also determined by calculating the error due to the limited accuracy of different parts of the controller as depicted in Fig. 15c. As it can be seen, the maximum error generated by the proposed processor is less than 0.2 μA(peak-to-peak) which is equal to 6-bit resolution (output signal full-scale value is 15 μA).

The proposed controller is composed of analog (crossbar, inference engine and defuzzifier) and digital circuits (rule base and control circuit). In this design, the analog parts are dominant, therefore, power consumption of the controller is determined by the current flowing through the memristors which represent membership degree values. There is a tradeoff between power consumption and the speed of operation. Using lower currents makes the controller to operate slower while consuming less power. In addition, the level of signals should be properly chosen to be detectable by analog circuits. Altogether, the power consumption of the controller can be controlled by altering the



(a) inputs membership values

(b) the output signals after sample and hold

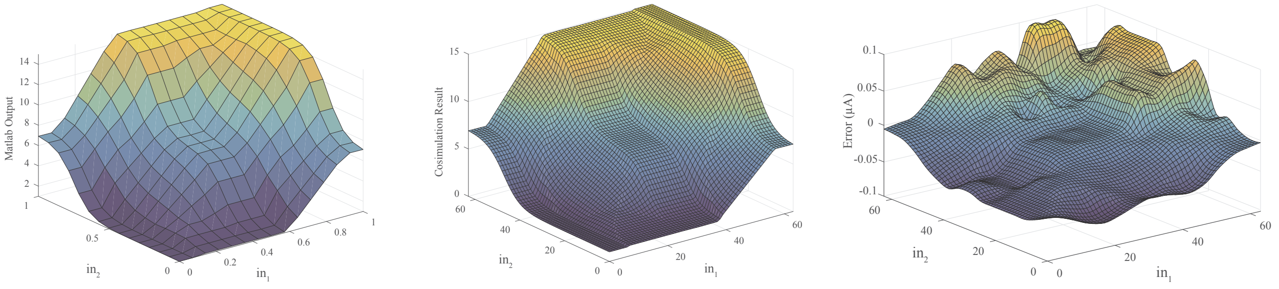
(c) the activation level of the different rules

(d) the signal related to the numerator of (7)

(e) denominator signal of (7)

(f) the crisp final output signal (ideal value=3.11μ A)

Fig. 14. Inputs are set to 8 and 40, respectively, and the resulting signals are depicted. The actual values are marked inside the pictures.



(a) Control surface produced in MATLAB

(b) MATLAB/HSPICE co-simulation result

(c) Error between actual and ideal control surfaces

Fig. 15. The controller simulation results.

memristance of the memristors. With the values chosen in the simulation, the average power consumed by FLC is 3.49 mW.

The layout of the proposed controller is illustrated in Fig. 16. The crossbar section is designed with the dimensions reported in [37]. Most of the area of the fuzzifier section is occupied by the 6-to-64 bit decoder and other digital circuitry. It also comprises six read circuits and sample and holds dedicated to each row. The cross bar, as it was mentioned earlier, is a method for building active Nano-scale devices (memristors) in each cross point of two crossing parallel metalized line. Hence, it is fabricated on top of the CMOS layer and does not increase the chip size. The layout area of the proposed processor is calculated to be 0.614 mm^2 which is very smaller than a similar fully digital approach. Since, CMOS

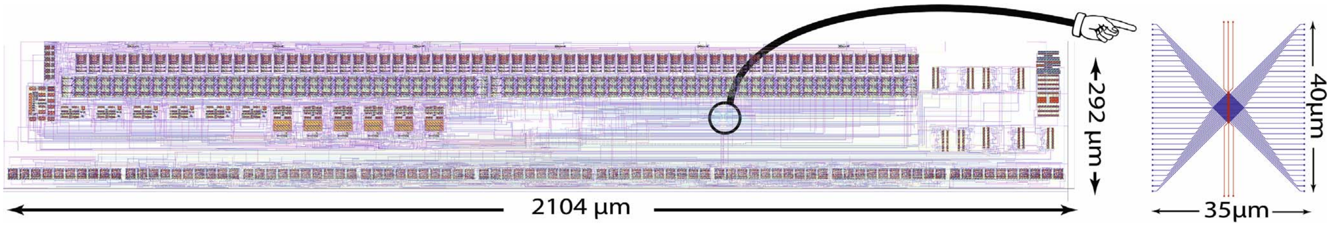


Fig. 16. The layout of the proposed architecture.

Table 2
Comparison of the proposed architecture with other state of the art similar works.

Reference	Technology used (µm)	Design type	Speed (MFLIPS)	Number of inputs	MF's allowed per input	Number of outputs	MF's allowed per output	Number of rules	Bits per rule	Overlapped MF's	Output resolution	Area (mm^2)	Power (mW)
[57]	CMOS (0.35)	Mixed Signal	–	3	5	1	5	125	–	2	5	3.567	–
[58]	CMOS (0.35)	Mixed Signal	0.5	2	3	1	–	9	–	2	–	6.8	–
[59]	CMOS (2.5)	Digital	0.08	2	31	1	31	16	124	2	6	10.4	–
[60]	–	Josephson technology	790	2	–	1	–	4	–	2	5	4	0.89
[61]	CMOS (1.2)	Mixed Signal	10	2	3	1	3	9	–	–	–	0.4	10
[62]	CMOS (2.4)	Mixed Signal	2	2	3	1	3	9	12	–	6	0.945	–
[63]	CMOS (1.2)	Analog	6.25	2	3	1	5	9	–	2	–	0.67	16.3
[64]	CMOS (0.35)	Analog	–	2	4	1	–	16	–	2	–	0.1113	8.6
[65]	CMOS (0.35)	Digital	6.34	2	8	1	128	64	9	2	7	3	200
This work	CMOS (0.35)	Mixed Signal	10	2	3	1	16	9	8	2	6	0.614	3.49

transistor should be utilized to form a digital storage for membership functions, and therefore the chip size should be increase inevitably. Note that, we did not perform any the post-layout simulations since, till date, there exists no commercial design kit for memristor as well as design rules related to the crossbar structure.

Table 2 compares the performance of the proposed architecture with a few other similar works. The speed of the controller is high enough to be considered as a fast controller as compared with other works while it consumes less power. Moreover, it is possible to reduce the power consumption of the controller to the range of micro-watts by ultra-low current memristors. These types of memristors can operate within the Nano-ampere range which makes the passive crossbars to be ultra-low power as proposed in [56]. On the other hand, the speed of the processor can be increased by using independent and separate crossbar for each input variable. This way, fuzzification is done in a parallel manner, however, power consumption and the size of the circuit also increase due to additional decoders needed for proper operation of each crossbar.

In digital circuits, membership functions are stored in transistor based memory, which is larger and consumes more power compared to their memristor based counterparts. Our proposed method has better area and power consumption merits in comparison with other works. The larger area compared to analog implementations is due to the existence of the digital parts which is necessary for proper operation of the processor.

The inference engine of most works reported throughout the literature has a fixed structure in the antecedent part. Only some programmability related to the consequent part is embedded in the architecture. This way, any type of the rules especially those which are not dependent to some inputs cannot be correctly defined in the controller. In our proposed method, the full programmability on the rule base is supported and any sort of rules can be processed.

5. Conclusion

A new fuzzy logic controller architecture was presented in this paper. This controller utilizes a novel memristor based analog memory structure and can accommodate membership functions with any desired shape and any overlapping ratio. The controller can operate up to 10MFLIPS while consuming 3.49 mW. The low area size of 0.614 mm^2 and full programmability in both the rule base and the membership function generation are some advantages of the proposed architecture.

Designing a fuzzifier with dedicated crossbar to each input causes the processor to operate at higher speed. The processor can also be designed to function in very low power budgets, however, signal levels should be high enough to be detectable by the analog circuits.

The number of the input variables and the membership functions defined in the controller are limited by the size of the crossbar. Larger crossbars can accommodate more membership functions. Most hardware realizations (especially analog and mixed signals) can produce only trapezoid and triangular membership functions. Furthermore, the overlapping ratio of the linguistic values are limited to 2. This means only two adjacent membership functions can overlap. In our proposed method, membership functions with any shape and overlapping ratio can be defined in the processor.

References

[1] Habib M, Hussain A, Rasheed S, Ali M. Adaptive fuzzy inference system based directional median filter for impulse noise removal. AEU - Int J Electron Commun 2016;70:689–97. http://dx.doi.org/10.1016/j.aeue.2016.02.005.

[2] Jadidoleslamy H, Aref MR, Bahramgiri H. A fuzzy fully distributed trust management system in wireless sensor networks. AEU - Int J Electron Commun 2016;70:40–9. http://dx.doi.org/10.1016/j.aeue.2015.09.017.

[3] Maity HK, Maity SP. PEE based RW using fuzzy conditional entropy for image partitioning. AEU - Int J Electron Commun 2016;70:211–24. http://dx.doi.org/10.1016/j.aeue.2015.11.010.

[4] Jesus IS, Barbosa RS. Smith-fuzzy fractional control of systems with time delay. AEU

- Int J Electron Commun 2017;78:54–63. http://dx.doi.org/10.1016/j.aeue.2017.05.014.
[5] Kumar A, Kumar V. Hybridized ABC-GA optimized fractional order fuzzy pre-compensated FOPID control design for 2-DOF robot manipulator. AEU - Int J Electron Commun 2017;79:219–33. http://dx.doi.org/10.1016/j.aeue.2017.06.008.
[6] Guo S, Peters L, Surmann H. Design and application of an analog fuzzy logic controller. IEEE Trans Fuzzy Syst 1996;4:429–38. http://dx.doi.org/10.1109/91.544303.
[7] Marshall GF, Collins S. Fuzzy logic architecture using subthreshold analogue floating-gate devices. IEEE Trans Fuzzy Syst 1997;5:32–43. http://dx.doi.org/10.1109/91.554445.
[8] Miki T, Yamakawa T. Fuzzy inference on an analog fuzzy chip. IEEE Micro 1995;15:8–18. http://dx.doi.org/10.1109/40.400638.
[9] Peters L, Guo S, Camposano R. A novel analog fuzzy controller for intelligent sensors. Fuzzy Sets Syst 1995;70:235–47. http://dx.doi.org/10.1016/0165-0114(94)00221-R.
[10] Yamakawa T. A fuzzy inference engine in nonlinear analog mode and its application to a fuzzy logic control. IEEE Trans Neural Networks 1993;4:496–522. http://dx.doi.org/10.1109/72.217192.
[11] Cilingiroglu U, Pamir B, Gunay ZS, Dulger F. Sampled-analog implementation of application-specific fuzzy controllers. IEEE Trans Fuzzy Syst 1997;5:431–42. http://dx.doi.org/10.1109/91.618278.
[12] Fattaruso JW, Mahant-Shetti SS, Barton JB. A fuzzy logic inference processor. IEEE J Solid-State Circuits 1994;29:397–402. http://dx.doi.org/10.1109/4.280687.
[13] Huertas JL, Sanchez-Solano S, Barriga A, Baturone I. A fuzzy controller using switched-capacitor techniques. In: Fuzzy Syst. 1993., Second IEEE Int. Conf., IEEE; 1993, p. 516–20.
[14] Landolt O. Low-power analog fuzzy rule implementation based on a linear MOS transistor network. Microelectron. Neural Networks, 1996., Proc. Fifth Int. Conf., IEEE; 1996, p. 86–93.
[15] Tsukano K, Inoue T. Synthesis of operational transconductance amplifier-based analog fuzzy functional blocks and its application. IEEE Trans Fuzzy Syst 1995;3:61–8. http://dx.doi.org/10.1109/91.366571.
[16] Liao K-H, Lai J-L, Wang W-C, Chen R-J. Design a current-mode max/min circuit in CMOS technology. In: Adv. Infocomm Technol. (ICAIT), 2013 6th Int. Conf., IEEE; 2013, p. 57–8.
[17] Amirkhanzadeh R, Khoei A, Hadidi K. A mixed-signal current-mode fuzzy logic controller. AEU - Int J Electron Commun 2005;59:177–84. http://dx.doi.org/10.1016/j.aeue.2004.11.019.
[18] Chowdhury SR, Chakrabarti D, Saha H. FPGA realization of a smart processing system for clinical diagnostic applications using pipelined datapath architectures. Microprocess Microsyst 2008;32:107–20. http://dx.doi.org/10.1016/j.micpro.2007.12.001.
[19] Di Stefano A, Giaconia C. An FPGA-based adaptive fuzzy coprocessor. In: Int. Work. Artif. Neural Networks, Springer; 2005, p. 590-7.
[20] Dixit AM, Singh H, Meitzler T. Soft computing approach to crack detection and FPGA implementation. Mater Eval 2010;68:1263–72.
[21] Feng H-M, Chen C-Y, Horng J-H. Intelligent omni-directional vision-based mobile robot fuzzy systems design and implementation. Expert Syst Appl 2010;37:4009–19. http://dx.doi.org/10.1016/j.eswa.2009.11.030.
[22] Hamzeh M, Mahdiani HR, Saghafi A, Fakhraie SM, Lucas C. Computationally efficient active rule detection method: algorithm and architecture. Fuzzy Sets Syst 2009;160:554–68. http://dx.doi.org/10.1016/j.fss.2008.05.009.
[23] Javadi MHS, Mahdiani HR, Kh EZ. A hardware oriented fuzzification algorithm and its VLSI implementation. Soft Comput 2013;17:683–90.
[24] Kung Y-S, Quynh NV, Hieu NT, Huang C-C, Huang L-C. Simulink/modelsim co-simulation and FPGA realization of speed control IC for PMSM drive. Procedia Eng 2011;23:718–27. http://dx.doi.org/10.1016/j.proeng.2011.11.2571.
[25] Messai A, Mellit A, Massi Pavan A, Guessoum A, Mekki H. FPGA-based implementation of a fuzzy controller (MPPT) for photovoltaic module. Energy Convers Manage 2011;52:2695–704. http://dx.doi.org/10.1016/j.enconman.2011.01.021.
[26] Patyra MJ, Grantner JL, Koster K. Digital fuzzy logic controller: design and implementation. IEEE Trans Fuzzy Syst 1996;4:439–59. http://dx.doi.org/10.1109/91.544304.
[27] Aminifar S, Khoei A, Haidi K, Yosefi G. A digital CMOS fuzzy logic controller chip using new fuzzifier and max circuit. AEU-International J Electron Commun 2006;60:557–66.
[28] Baturone I, Sanchez-Solano S, Barriga A, Huertas JL. Implementation of CMOS fuzzy controllers as mixed-signal integrated circuits. IEEE Trans Fuzzy Syst 1997;5:1–19. http://dx.doi.org/10.1109/91.554443.
[29] Bouras S, Kotronakis M, Suyama K, Tsividis Y. Mixed analog-digital fuzzy logic controller with continuous-amplitude fuzzy inferences and defuzzification. IEEE Trans Fuzzy Syst 1998;6:205–15.
[30] Loan SA, Murshid AM. A novel fuzzy inference processor using trapezoidal-shaped membership function. In: Open Syst. (ICOS), 2011 IEEE Conf., IEEE; 2011, p. 220–4.
[31] Yosefi G, Aminifar S, Neda S, Daneshwar MA. Design of a mixed-signal digital CMOS fuzzy logic controller (FLC) chip using new current mode circuits. AEU-Int J Electron Commun 2011;65:173–81.
[32] Strukov DB, Likharev KK. Defect-tolerant architectures for nanoelectronic crossbar memories. J Nanosci Nanotechnol 2007;7:151–67.
[33] Strukov DB, Snider GS, Stewart DR, Williams RS. The missing memristor found. Nature 2008;453:80–3.
[34] Sarwar SS, Saqueb SAN, Quaiyum F. Memristor-based nonvolatile random access memory: hybrid architecture for low power compact memory design. Access IEEE 2013;1:29–34.
[35] Merrikh-Bayat F, Shouraki SB, Merrikh-Bayat F. Memristor crossbar-based hardware implementation of fuzzy membership functions. In: Proc. - 2011 8th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2011, vol. 1, 2011, p. 645–9. http://dx.doi.org/10.1109/FSKD.2011.6019541.
[36] Afrakoti IEP, Shouraki SB, Bayat FM, Gholami M. Using a memristor crossbar structure to implement a novel adaptive real-time fuzzy modeling algorithm. Fuzzy Sets Syst 2016.
[37] Kim K-H, Gaba S, Wheeler D, Cruz-Albrecht JM, Hussain T, Srinivasa N, et al. A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. Nano Lett 2011;12:389–95. http://dx.doi.org/10.1021/nl203687n.
[38] Zavala AH, Nieto OC. Fuzzy hardware: a retrospective and analysis. IEEE Trans Fuzzy Syst 2012;20:623–35. http://dx.doi.org/10.1109/TFUZZ.2011.2181179.
[39] Waser R, Aono M. Nanoionics-based resistive switching memories. Nat Mater 2007;6:833–40.
[40] Lee JS, Lee S, Noh TW. Resistive switching phenomena: a review of statistical physics approaches. Appl Phys Rev 2015;2:31303.
[41] Chua LO. Memristor-the missing circuit element. Circuit Theory IEEE Trans 1971;18:507–19.
[42] Chua LO, Kang SM. Memristive devices and systems. Proc IEEE 1976;64:209–23.
[43] Kang BS, Ahn S, Lee M, Stefanovich G, Kim KH, Xianyu WX, et al. High-current-density $CuO_x/InZnO_x$ thin-film diodes for cross-point memory applications. Adv Mater 2008;20:3066–9.
[44] Alibart F, Gao L, Hoskins BD, Strukov DB. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. Nanotechnology 2012;23:75201.
[45] Pershin YV, Di Ventra M. Practical approach to programmable analog circuits with memristors. Circuits Syst I Regul Pap IEEE Trans 2010;57:1857–64.
[46] Qureshi MS, Pickett M, Miao F, Strachan JP. CMOS interface circuits for reading and writing memristor crossbar array. In: 2011 IEEE Int. Symp. Circuits Syst., IEEE; 2011, p. 2954–7. http://dx.doi.org/10.1109/ISCAS.2011.5938211.
[47] Vontobel PO, Robinett W, Kuekes PJ, Stewart DR, Straznicky J, Williams RS. Writing to and reading from a nano-scale crossbar memory based on memristors. Nanotechnology 2009;20:425204.
[48] Yi W, Perner F, Qureshi MS, Abdalla H, Pickett MD, Yang JJ, et al. Feedback write scheme for memristive switching devices. Appl Phys A 2011;102:973–82.
[49] Kandel A, Langholz G. Fuzzy hardware: architectures and applications. Springer Science & Business Media; 2012.
[50] Wang L-X. A course in fuzzy systems. USA: Prentice-Hall press; 1999.
[51] Ramirez-Angulo J, Ducoudray-Acevedo G, Carvajal RG, Lopez-Martin A. Low-voltage high-performance voltage-mode and current-mode WTA circuits based on flipped voltage followers. Circuits Syst II Express Briefs IEEE Trans 2005;52:420–3.
[52] Prommee P, Chattrakun K. CMOS WTA maximum and minimum circuits with their applications to analog switch and rectifiers. Microelectron J 2011;42:52–62.
[53] Padash M, Khoei A, Hadidi K, Ghasemiyan H. A high precision high frequency VLSI multi-input min-max circuit based on WTA-LTA cells. In: Electron. Devices, Syst. Appl. (ICEDSA), 2011 Int. Conf., IEEE; 2011, p. 152–6.
[54] Sheridan P, Kim K-H, Gaba S, Chang T, Chen L, Lu W. Device and SPICE modeling of RRAM devices. Nanoscale 2011;3:3833–40.
[55] Popa C. Improved accuracy current-mode multiplier circuits with applications in analog signal processing. IEEE Trans Very Large Scale Integr Syst 2014;22:443–7.
[56] Gaba S, Cai F, Zhou J, Lu WD. Ultralow sub-1-nA operating current resistive memory with intrinsic non-linear characteristics. IEEE Electron Device Lett 2014;35:1239–41.
[57] Orcioni S, Biagetti G, Conti M. A mixed signal fuzzy controller using current mode circuits. Analog Integr Circuits Signal Process 2004;38:215–31.
[58] Chen C-Y, Hsieh Y-T, Liu B-D. Circuit implementation of linguistic-hedge fuzzy logic controller in current-mode approach. IEEE Trans Fuzzy Syst 2003;11:624–46.
[59] Togai M, Watanabe H. Expert system on a chip: an engine for real-time approximate reasoning. In: Proc. ACM SIGART Int. Symp. Methodol. Intell. Syst., ACM; 1986, p. 147–54.
[60] Morisue M, Kogure Y. A superconducting fuzzy processor. In: Fuzzy Syst. 1992., IEEE Int. Conf., 1992, p. 443–50. http://dx.doi.org/10.1109/FUZZY.1992.258655.
[61] Lemaitre L, Patyra MJ, Mlynek D. Analysis and design of CMOS fuzzy logic controller in current mode. IEEE J Solid-State Circuits 1994;29:317–22.
[62] Huertas J, Sánchez-Solano S. Integrated circuit implementation of fuzzy controllers. IEEE J Solid; 1996.
[63] Peyravi H, Khoei A, Hadidi K. Design of an analog CMOS fuzzy logic controller chip. Fuzzy Sets Syst 2002;132:245–60.
[64] Mokarram M, Khoei A, Hadidi K. CMOS fuzzy logic controller supporting fractional polynomial membership functions. Fuzzy Sets Syst 2015;263:112–26.
[65] Falchieri D, Gabrielli A, Gandolfi E. Very fast rate 2-input fuzzy processor for high energy physics. Fuzzy Sets Syst 2002;132:261–72. http://dx.doi.org/10.1016/S0165-0114(02)00122-7.