A Low-Power Current Mode Fuzzy-ART Cell

Teresa Serrano-Gotarredona and Bernabé Linares-Barranco

***Abstract*—This paper presents a very large scale integration (VLSI) implementation of a low-power current-mode fuzzy-adaptive resonance theory (ART) cell. The cell is based on a compact new current source multibit memory cell with online learning capability. A small prototype of the designed cell and its peripheral block has been fabricated in the AustriaMicroSystems (AMS)-0.35-μm technology. The cell occupies a total area of 44 $\times$ 34 μm^2 and consumes a maximum current of 22 nA.**

***Index Terms*—Adaptive resonance theory (ART), hardware implementations, low power.**

I. Introduction

Adaptive resonance theory (ART) is a well-established neural network framework developed by Grossberg *et al.* at the University of Boston, Boston, MA [1]–[3]. The ART algorithms are neural categorizers that share some interesting properties. One of these properties is the online learning, that is, each time a new input exemplar is presented to the system, the system knowledge is updated online to incorporate that knowledge; the system learns while it performs. Another interesting property is that the system maintains a generalization capability which is controlled by a tunable vigilance subsystem. There is a vigilance parameter that tunes the coarness of the established categories. Setting the vigilance parameter to a low value increases the system generalization capability, thus the system tends to form coarser categories. Setting the vigilance parameter to a high value decreases the system generalization, and it tends to form finer categories, thus increasing the number of categories formed for the same set of input data.

In the past, some real-time hardware implementations for the ART-1 [4] and adaptive resonance theory map (ARTMAP) [5] algorithms were reported, that for the first time included their complete functionality.

The fuzzy-ART algorithm is a neural categorizer which self-organizes recognition codes in response to sequences of analog input patterns. In recent years, a wide variety of applications of the fuzzy-ART algorithm has appeared in the literature [6]–[9]. In these applications, the algorithm is implemented in software. However, to include these systems in a portable application, the availability of a dedicated compact low-power consumption hardware would be desirable. Several hardware implementations of fuzzy-ART networks have been reported [10]–[12]. In [10], a very large scale integration (VLSI) implementation of a modified version of the fuzzy-ART algorithm is presented. The hardware is simplified by modifying the definition of similarity between input and exemplars or "choice function." The implementations presented in [10] and [12] store the category weights in analog mode. Thus, weights suffer from decay along time. A refreshing mechanism is implemented in [10]. In [12], the learning rule is modified to compensate for this decay. That modification forces a continuous presentation of input exemplars to prevent the system from forgetting the stored knowledge. The implementation presented in [11] is a fully digital one. Thus, it consumes a large area. To overcome this problem a pipelined multichip architecture is proposed in [11].

Manuscript received March 1, 2006; revised June 16, 2006.

The authors are with the Instituto de Microelectrónica de Sevilla (IMSE), Sevilla 41012, Spain (e-mail: bernabe@imse.cnm.es; terese@imse.cnm.es).

Digital Object Identifier 10.1109/TNN.2006.883725

In this paper, we present a mixed-mode VLSI implementation of the fuzzy-ART cell. The weights are stored as 4-b digital words. However, the computation is done in analog current mode. The cell is able to operate with currents in the order of tenths of nanoamperes, what makes it particularly suitable for applications requiring low power. Also, such low-power consumption per cell, allows to assemble a single-chip fuzzy-ART network with extremely large number of cells while keeping system power consumption at low levels. The fast learning rule of the fuzzy-ART algorithm is implemented inside the cell. This cell is a critical component and first step towards the development of a full fuzzy-ART microchip. In a 1-cm^2 microchip die area, it would be possible to fit an array of 220 $\times$ 280 fuzzy-ART cells consuming a total power below 4.45 mW.

This paper is structured as follows. A brief description of the fuzzy-ART algorithm is provided in Section II. The description of the implemented fuzzy-ART cell is detailed in Section III. Experimental results are provided in Section IV. Finally, in Section V, some conclusions are drawn.

II. Fuzzy-ART Algorithm

The fuzzy-ART neural network [3] is a clustering self organizing neural network for analog input patterns. Fig. 1(a) represents the architecture of a fuzzy-ART network. The network is composed of an attentional subsystem and an orienting or vigilance subsystem. The attentional subsystem is composed of two layers. Layer F_1 is the input layer. Input patterns $\mathbf{b} = (b_1, b_2, \ldots, b_N)$ composed of N analog values are presented to the system. F_2 is the category layer. The system categorizes each input pattern as belonging to one of the $[y_1, y_2, \ldots, y_M]$ categories. The system stores a weight matrix $\{z_{ij}\}$ of analog values that represents the categories learned by the system. Each category y_j is represented by the weight vector $\mathbf{z}_j$ composed of N analog values.

The algorithmic flow diagram of the fuzzy-ART operation is depicted in Fig. 1(b). Initially, all the interconnection weights z_{ij} are set to their maximum analog value "MAX."

When an analog input vector $\mathbf{b} = (b_1, b_2, \ldots, b_N)$ is applied to the system, each F_1 layer cell receives an analog input component $b_i \in [0, \mathrm{MAX}]$. Then, each F_2 category computes its "choice function" T_j, which is a measurement of the similarity between the analog input pattern $\mathbf{b}$ and the analog weight template $\mathbf{z}_j = (z_{1j}, z_{2j}, \ldots, z_{Nj})$ stored in category j

$$T_j = \frac{|\mathbf{b} \wedge \mathbf{z}_j|}{\alpha + |\mathbf{z}_j|} \tag{1}$$

where $\wedge$ is the fuzzy MIN operator defined by $(\boldsymbol{X} \wedge \boldsymbol{Y})_i = \min(X_i, Y_i)$, $|\boldsymbol{X}|$ is the l_1 norm $|\boldsymbol{X}| = \sum_{i=1}^{N} |X_i|$, and α is a positive parameter called "choice parameter."

Layer F_2 is a winner-takes-all (WTA) competition network [13]. Each jth F_2 cell gives an output y_j which is "1" if that cell is receiving the largest T_j input and "0" otherwise. That is

$$\begin{aligned} y_J &= 1, && \text{if } T_J = \max(T_j) \\ y_{j \neq J} &= 0, && \text{otherwise.} \end{aligned} \tag{2}$$

This way, the F_2 layer selects the category J whose stored pattern $\mathbf{z}_J$ most closely resembles input pattern $\mathbf{b}$ according to the similarity criterion defined by (1). The original fuzzy-ART algorithm [3] states that if more than one T_j is maximal, the category j with the smallest index is chosen. However, as will be discussed in Section III, in the context of the proposed hardware, if more than one T_j is maximal a unique winner

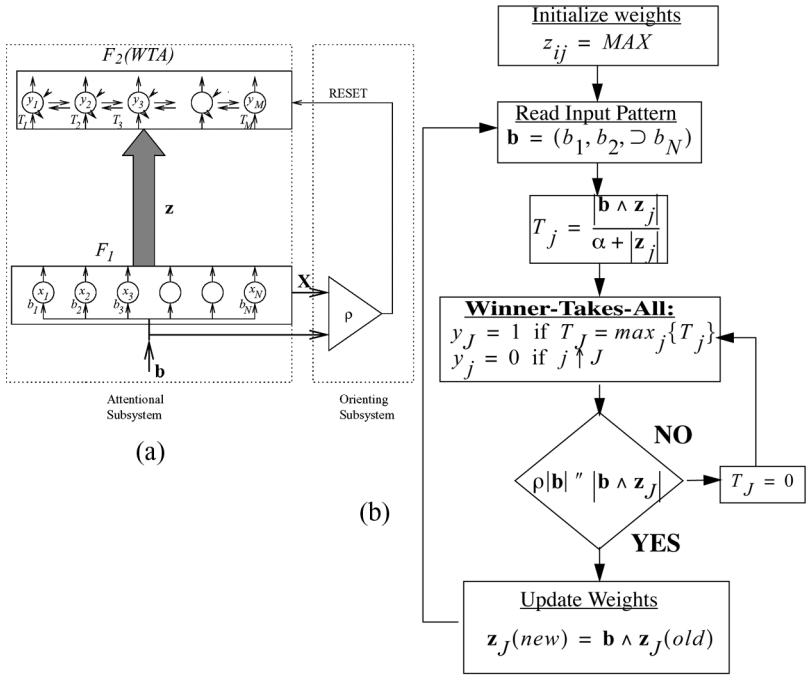

Fig. 1. (a) Topological structure of the fuzzy-ART architecture. (b) Flow diagram of the fuzzy-ART algorithm.

is randomly selected. These different ways of resolving "ties" may result in some cases where the hardware system produce slightly different final categories than the theoretical fuzzy-ART algorithm for the same set of presentations of input patterns. However, this difference does not affect the functional objectives of the neural network categorizer.

For the winning category J, the vigilance subsystem checks the condition $\rho|\mathbf{b}| \leq |\mathbf{b} \wedge \mathbf{z}_J|$, where $\rho \in [0, 1]$ is the so called vigilance parameter. If the condition is not satisfied, category J is disregarded by forcing $T_J = 0$. Layer F_2 will again select the category with maximum T_j, and the vigilance criterion will be checked again. This search continues until a winning category is selected that fulfills the vigilance criterion.

When a category J meeting the vigilance criterion is activated, its weights $\mathbf{z}_J$ are updated according to the learning rule $\mathbf{z}_J(\text{new}) = \mathbf{b} \wedge \mathbf{z}_J(\text{old})$. This learning rule is known as the fast-learning mode of the fuzzy-ART algorithm [3].

III. Fuzzy-ART Cell Description

A fuzzy-ART cell has to perform the following operations:

1) store an analog weight z_{ij}, which must be initially reset to its maximum analog value "MAX;"
2) compute the componentwise fuzzy-min operation between the analog stored value z_{ij} and the analog input component b_i; this analog minimum value will be used in the computation of the choice function T_j and in the evaluation of similarity by the vigilance subsystem;
3) implement the learning rule; when a category J is selected ($y_J = 1$) that fulfills the vigilance criterion, all cells in the selected row must update their stored weight according to the rule $z_{ij}(\text{new}) = \min(b_i, z_{ij}(\text{old}))$.

We have fabricated a cell prototype together with a peripheral cell, to be shared by all cells in the same column. Fig. 2 shows the schematic of the fabricated small test prototype. It includes an isolated fuzzy-ART cell connected to a "bias column cell," or peripheral cell, which in a large system would be common to all the cells in the same column. In hardware cell arrays implementations, it is of primary importance to simplify and compact the array cells as much as possible. Peripheral cells can be more complex since they do not contribute as critically to overall chip area. Peripheral cells will contribute to some of the cell operations, like biasing, data loading, and handling, etc.

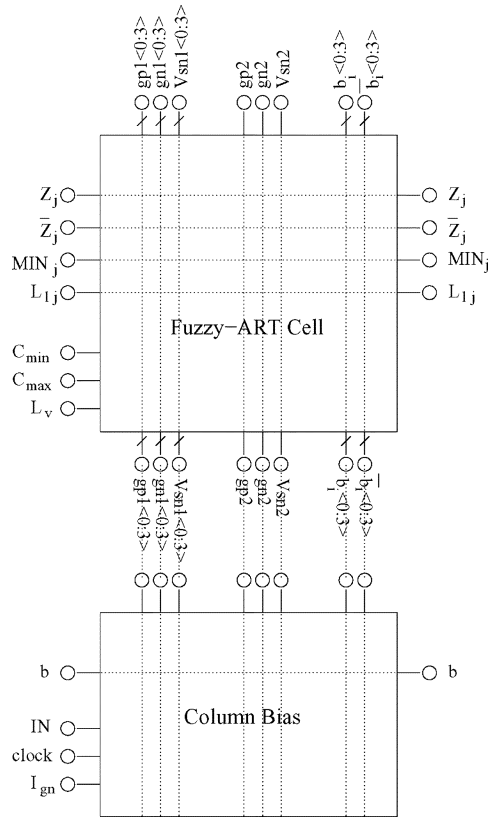

Fig. 2. Block diagram of the connections between the fuzzy-ART cell and the "bias column block."

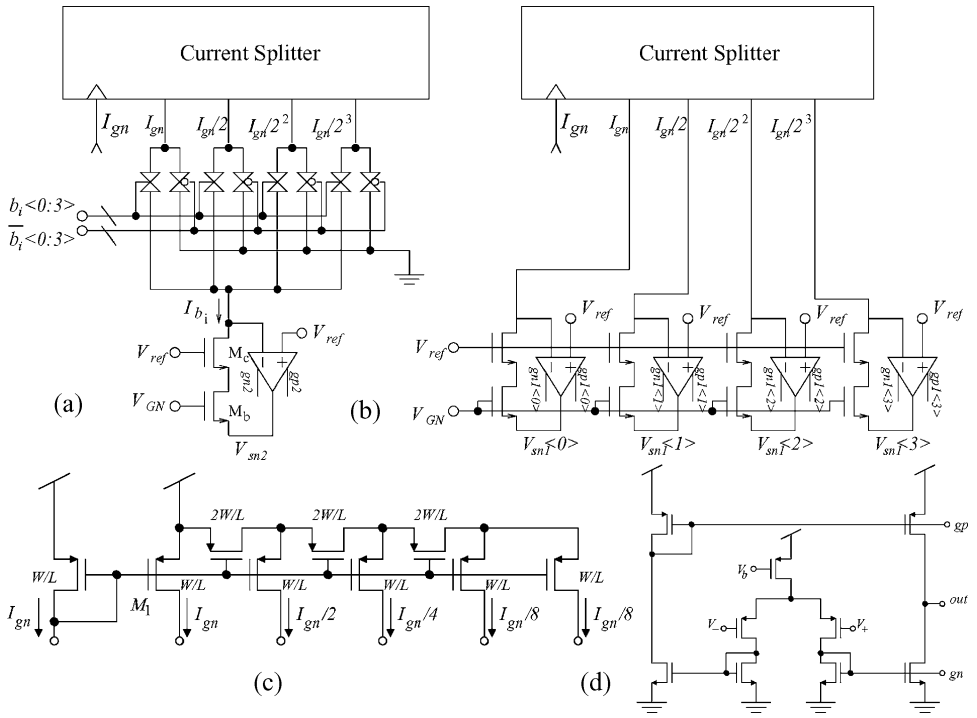

Fig. 3. Schematic of the "bias column" block. (a) Current splitter generating the input current I_{b_i} to be reproduced in all the cells located in the same column. (b) Current splitter generating five binary weighted currents to be reproduced in all the cells located in the same column. (c) Schematic of the PMOS current splitter. (d) Schematic of the amplifiers.

A system implementing the fuzzy-ART operation would contain a two-dimensional (2-D) matrix of fuzzy-ART cells. Each column of the fuzzy-ART cells array would correspond to a component b_i of the input vector $\mathbf{b} = (b_1, b_2, \ldots, b_N)$. Each row of the fuzzy-ART cells array would correspond to one F_2 category node. Each row delivers two output currents: One current representing the analog l_1-norm of the weight vector $|\mathbf{z}_j|$, and a second current representing the l_1-norm of the fuzzy-min vector $|\mathbf{b} \wedge \mathbf{z}_j|$. These currents would be delivered to a "peripheral row cell" for the computation of the T_j values that would compete in a WTA circuit. The peripheral row cells and the WTA network have not been implemented in the present prototype. However, several circuit topologies for these circuits have already been proposed in the literature [13], [18]–[20]. The use of an analog WTA circuit [13] to select the maximum T_j current guarantees that if its gain is sufficiently high a unique winner is always selected. Even in the case that several T_j currents are nominally identical (which is, for example, the case whenever a new uncommitted node is selected as a winner) the mismatches between the T_j currents would bias the WTA network to select a unique winner.

A. Bias Column Peripheral Cell

The bias column peripheral cell is formed by the blocks shown in Fig. 3(a) and (b). It consists basically of two p-type channel metal–oxide–semiconductor (PMOS) current splitters. Each PMOS current splitter [see Fig. 3(c)] [14], [15] receives a bias current equal to I_{gn} and delivers four output currents. The first output current is just a replica of the bias current I_{gn}. The output currents of the next branches are just the result of dividing by two the current of the immediately preceding branch [15].

The output currents of the first current splitter [see Fig. 3(a)] are digitally combined using the 4-b digital word $b_i\langle 0:3\rangle$. Thus, a weighted current $I_{b_i} = I_{gn} \sum_{k=0}^{3} b_i(k)/2^k$ is generated. This current goes to the input of a source-driven active current mirror [16]. The input branch of the current mirror is formed by transistors M_b and M_c, and the feedback amplifier. Transistor M_c acts as a cascode transistor. The gate of the mirroring transistor M_b is set to a fixed bias voltage V_{GN}, while its source is driven by the feedback amplifier. The input of this current mirror presents low input impedance (as input voltage is kept clamped to V_{ref}). For low currents, it has better stability properties than the traditional active input current mirror. Delivering appropriately voltage V_{sn2} to all the fuzzy-ART array cells in the same column, the weighted current $I_{b_i} = b_i I_{gn}$ can be replicated in all the fuzzy-ART array cells located in the same column. In fact, the output branch of the amplifier [see Fig. 3(d)] is replicated in each fuzzy-ART array cell of the same column to scale the output current of the amplifier with the number of cells.

The second current splitter [Fig. 3(b)] generates four binary weighted currents $[I_{gn}, I_{gn}/2, I_{gn}/4, I_{gn}/8]$, that go, respectively, to the input of four source-driven active current mirrors, generating the bias voltages $V_{sn1}\langle 0:3\rangle$. Delivering appropriately these four voltages to all the cells in the same column, the binary weighted currents can be replicated in every cell. In this case, the four output branches of the amplifiers are also replicated in each fuzzy-ART array cell of the same column.

The "bias column" cell also delivers the digital control word $b_i\langle 0:3\rangle$ to all the fuzzy-ART array cells located in the same column. The input values $b_i\langle 0:3\rangle$ are loaded serially through a shift register, not shown here, located in the "bias column cells."

B. Fuzzy-ART Array Cell

Fig. 4(a) shows the schematic of the fuzzy-ART array cell. Each cell contains four cells named "bitcell" [whose schematic is shown in Fig. 4(c)], a "learn_ctr" cell [with detailed schematic shown in Fig. 4(b)], and some circuitry for the computation of the minimum.

Each "bitcell" cell stores a bit $z_{ij}(k)$ of the digitally stored weight $z_{ij}\langle 0:3\rangle$ in a current source flip-flop. Fig. 5(a) illustrates the basic

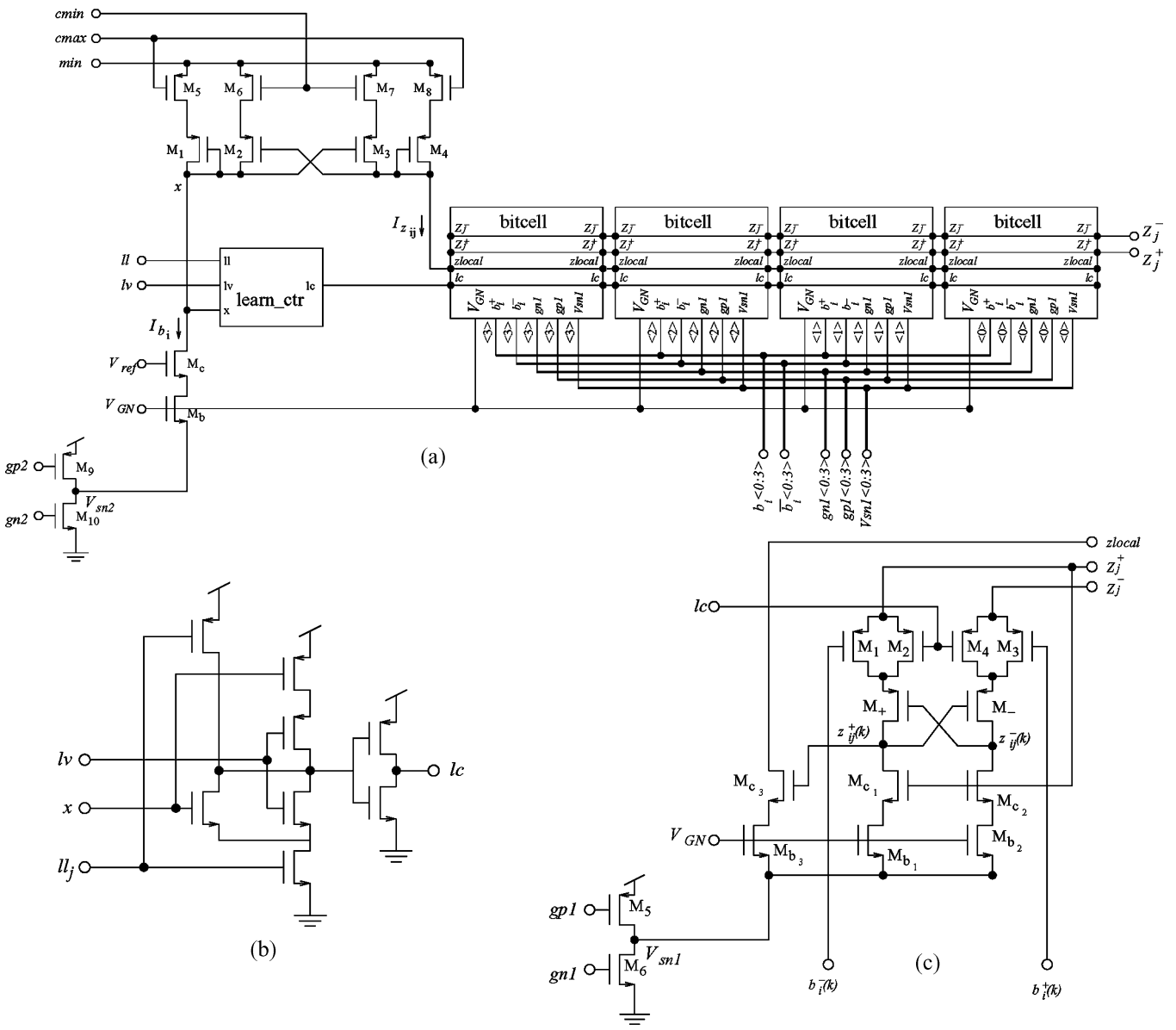

Fig. 4. Schematic of the fuzzy-ART cell. (a) Complete schematic. (b) Schematic of the box labeled learn_ctr in (a). (c) Schematic of the box labeled bitcell in (a).

concept of the current source flip-flop circuit which is based on a full MOS version of a MOS/bipolar circuit reported previously [17]. Two current sources of equal value I_{gn} are connected with two PMOS transistors M_+ and M_- in a flip-flop configuration. The sources of the two PMOS transistors are connected to two lines tied to a fixed voltage V_s by peripheral row-wise voltage sources. The currents injected into these lines change depending on the state of the flip-flop. If $z_{ij}^+ = 1 (z_{ij}^- = 0)$, transistor M_+ is ON, M_- is OFF, and the circuit injects a current I_{gn} into row line Z_j^+, while no current is injected into row line Z_j^-. Conversely, if $z_{ij}^+ = 0\,(z_{ij}^- = 1)$, transistor M_+ is OFF, M_- is ON, and the circuit injects current I_{gn} into line Z_j^-, while no current is injected into line Z_j^+. This current source flip-flop is inspired by the flip-flop reported in [18], but with added biasing current sources. As will be shown later, the addition of the current sources allows appropriate control and weighting of the delivered currents.

Fig. 5(b) shows a group of n current source flip-flops biased with binary weighted currents. This circuit forms a current source multibit memory cell. In this case, the current injected into lines Z_j^+ and Z_j^- would be

$$I\left(Z_j^+\right) = I_{gn} \sum_{k=0}^{n-1} z_{ij}^+(k) \frac{1}{2^k} = I_{gn} z_{ij}^+$$

$$I\left(Z_j^-\right) = I_{gn} \sum_{k=0}^{n-1} z_{ij}^-(k) \frac{1}{2^k} = I_{gn} z_{ij}^- \tag{3}$$

where $z_{ij}^+ \in [0, 2 - (1/2^{n-1})]$ and $z_{ij}^- \in [0, 2 - (1/2^{n-1})]$, with the restriction $z_{ij}^+ + z_{ij}^- = 2 - (1/2^{n-1})$.

Connecting N fuzzy-ART current source memory cells in a row sharing lines Z_j^+ and Z_j^- enables the generation of currents

$$I\left(Z_j^+\right) = I_{gn} \sum_{i=1}^{N} z_{ij}^+ = I_{gn} |\mathbf{z}_j|$$

$$I\left(Z_j^-\right) = I_{gn} \sum_{i=1}^{N} z_{ij}^- = I_{gn} \left(N \left(2 - \frac{1}{2^{n-1}} \right) - |\mathbf{z}_j| \right). \tag{4}$$

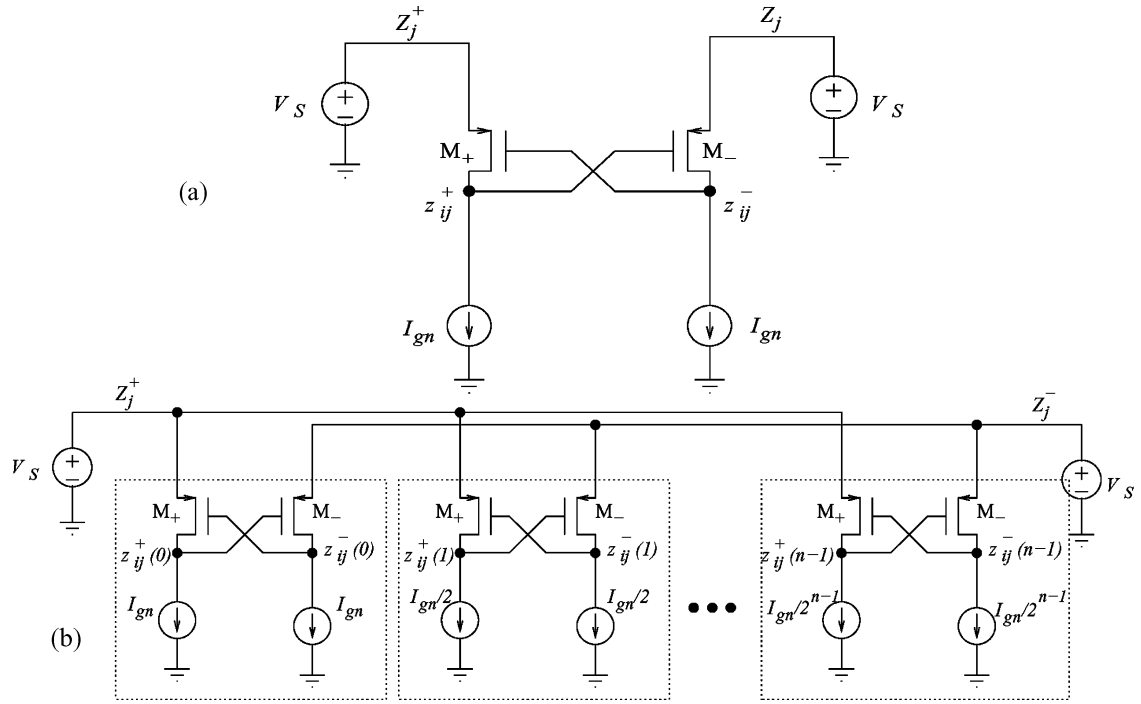

Fig. 5. (a) Conceptual block diagram of the current source flip-flop. (b) N-b current source memory cell.

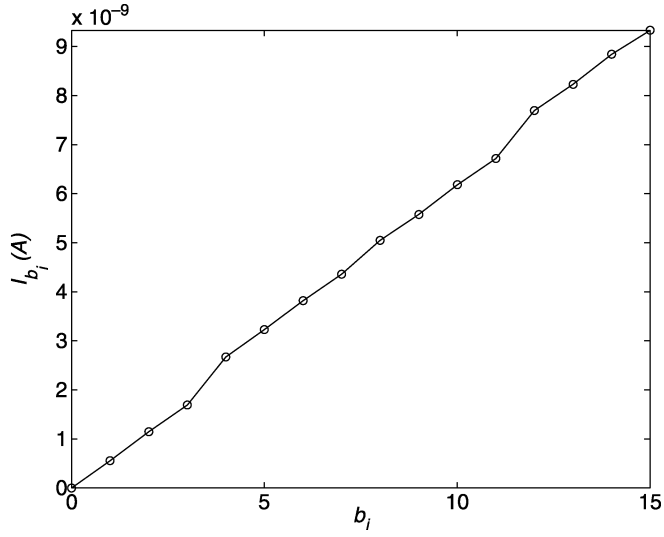

Fig. 6. Measured current I_{b_i} as a function of the 4-b input word b_i.

This way, the $|\mathbf{z}_j|$ currents required for the computation of the choice functions T_j in (1) are generated for each row.

Fig. 4(a) illustrates the implementation of our 4-b current source memory cell. Four bitcells are connected inside each fuzzy-ART array cell. Each bitcell contains a current source flip-flop composed of transistors M_+ and M_- biased by current sources M_{b_1} and M_{b_2} [see Fig. 4(c)]. The four current source flip-flops are connected sharing nodes Z_j^+ and Z_j^-, which are also to be shared by all the fuzzy-ART cells in the same row.

Transistors M_{b_1}, M_{b_2}, and M_{b_3} in Fig. 4(c) replicate the binary weighted current $I_{gn}/2^k$ generated in the "column bias" cells [see Fig. 3(b)], as they form an output branch of the source driven current mirror with their gate voltage clamped to the fixed global voltage V_{GN}

TABLE I
PERFORMANCE COMPARISON OF REPORTED FUZZY-ART IMPLEMENTATIONS

	cell area	current per cell	long-term storage	weight bits
This work	125λx125λ	~10nA	YES	4
[11]	1600λx1600λ	not reported	YES	11
[10]	35λx35λ	~1μA	NO	not applicable
[12]	83λx38λ	~10nA	NO	not applicable

and their sources are connected to nodes $V_{sn1}\langle k\rangle$. Transistors M_5 and M_6 in Fig. 4(c) form a replica of the output branch of the amplifier in Fig. 3(d).

As can be observed in Fig. 4(c), the current flowing through branch $z_{ij}^+(k)$ is also replicated by transistors M_{b3} and M_{c3} flowing out through pin $z\mathrm{local}$. Currents flowing through $z\mathrm{local}$ are summed in a common node inside each fuzzy-ART cell [see Fig. 4(a)]. Thus, a current $I_{z_{ij}} = I_{gn}\sum_{k=0}^{3} z_{ij}^+(k)(1/2^k) = I_{gn}\, z_{ij}$ is generated locally. This current is going to be used in the computation of the componentwise fuzzy-min operation that will be described later.

Transistors $M_1 - M_4$ in Fig. 4(c) act as switches and implement the cell learning. During normal operation mode digital signal lc is low, so that transistors M_2 and M_4 are ON, and the two flip-flop branches are connected to supply voltages at nodes Z_j^+ and Z_j^- as explained previously. When learning is activated for that cell, signal lc goes high, so that transistors M_2 and M_4 turn OFF. Then, if input bit $b_i(k)$ is high ($b_i^+(k) = 1$, $b_i^-(k) = 0$), transistor M_1 is ON and transistor M_3 is OFF. Therefore, the positive branch remains connected to a positive supply voltage through node Z_j^+, while no current flows through the negative branch. This way, the bitcell memory is updated, becoming $z_{ij}^+(k) = 1$ and $z_{ij}^-(k) = 0$. On the contrary, when cell learning is activated ($lc = 1$), if input bit $b_i(k)$ is low ($b_i^+(k) = 0$, $b_i^-(k) = 1$), the bitcell memory is updated to $z_{ij}^+(k) = 0$ and $z_{ij}^-(k) = 1$. As a result, whenever learn signal lc is activated, the stored bit $z_{ij}(k)$ is updated to the input bit $b_i(k)$.

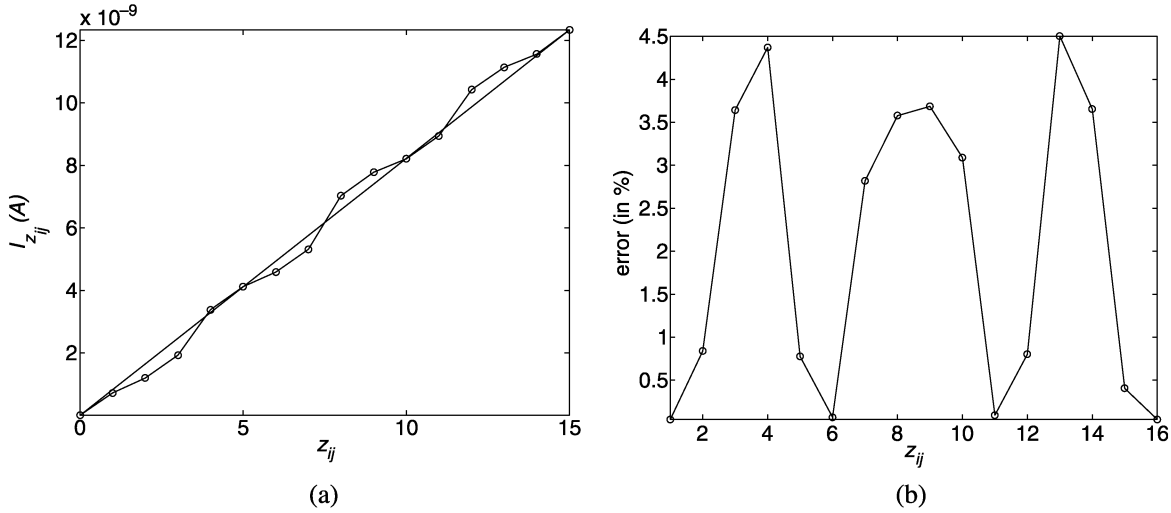

Fig. 7. (a) Measured analog current $I_{z_{ij}}$ generated in the cell versus the digital stored value z_{ij}. (b) Error in percentage.

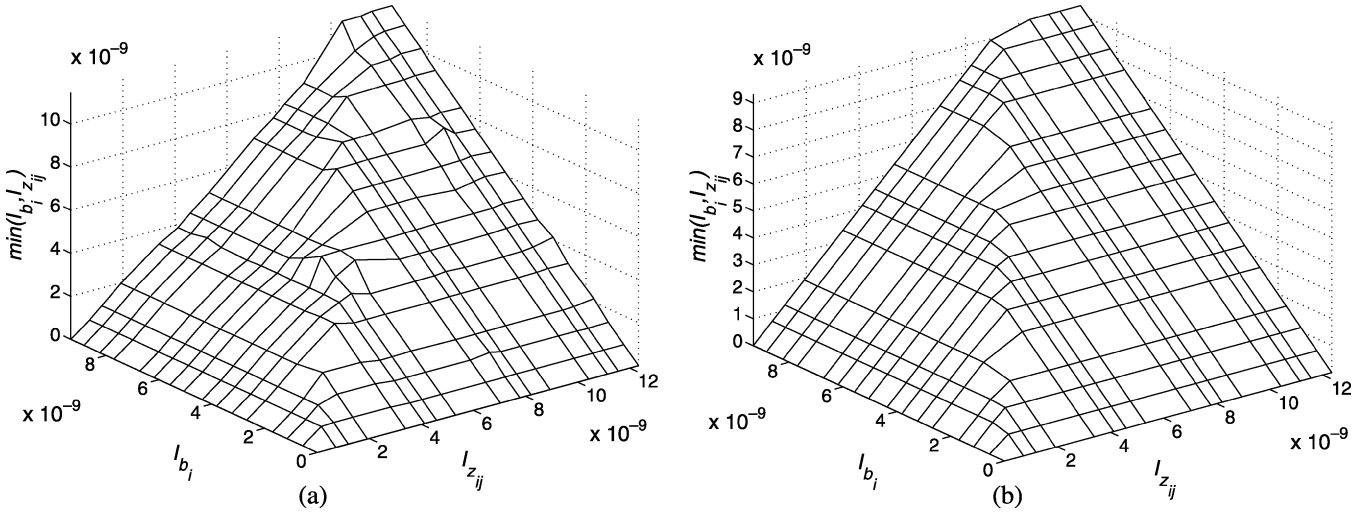

Fig. 8. (a) Measured minimum output current as a function of currents I_{b_i} and $I_{z_{ij}}$. (b) Computed minimum output current.

Transistors M_b and M_c in Fig. 4(a) are a replica of the corresponding ones in the bias column block in Fig. 3(a). They deliver a copy of current $I_{b_i} = I_{gn} \sum_{k=0}^{3} b_i(k)/2^k$. Transistors M_9 and M_{10} are a replica of the output branch of the feedback amplifier in Fig. 3(a).

Transistors $M_1 - M_8$ in Fig. 4(a) form a current-mode circuit for the computation of the minimum [19]. It receives two input currents $I_{b_i} = I_{gn} \sum_{k=0}^{3} b_i(k)/2^k$ and $I_{z_{ij}} = I_{gn} \sum_{k=0}^{3} z_{ij}(k)/2^k$. The comparison is controlled by digital signals $c\min$ and $c\max$, which must be properly sequenced [19]. These global signals are provided by an external off-chip controller after the loading of each input pattern. After comparison, the minimum current $\min(I_{z_{ij}}, I_{b_i})$ flows through terminal $\min$. If the result of the comparison is $I_{b_i} < I_{z_{ij}}$, current I_{b_i} flows through terminal $\min$ and voltage at node x goes high after comparison. If $I_{z_{ij}} < I_{b_i}$, current $I_{z_{ij}}$ flows through terminal $\min$ after comparison and voltage at node x goes low. Furthermore, current flowing through terminal $\min$ is summed along the same row to compute the term $|b_i \wedge z_{ij}|$ to be used in the computation of the choice function T_j and in the evaluation of similarity by the vigilance subsystem.

The digital logic controlling the activation of the learn signal is shown in Fig. 4(b). The circuit performs the logic computation $lc = ll_j(x + lv)$. Signal ll_j is generated for each row from the periphery. It is controlled by the category selection circuit and the vigilance subsystem. Signal lv is for global reset and is set to "0" during normal operation. If category j is selected by the system as the winning category and the vigilance criterion is fulfilled, signal ll_j is activated for that row. Then, for every cell on that row, if signal x is low, meaning that $I_{z_{ij}} \wedge I_{bi} = I_{z_{ij}}$, signal lc remains low, thus, no learning takes place for that (i, j) cell, and the same weight z_{ij} remains stored. On the contrary, if signal x is high, meaning that $I_{z_{ij}} \wedge I_{b_i} = I_{b_i}$, signal lc goes high, and learning takes place for that cell, so that the values stored in the flip-flops are updated to $z_{ij}\langle 0:3\rangle = b_i\langle 0:3\rangle$. This way, after learning takes place for a row, the weights z_{ij} are updated to the minimum (z_{ij}, b_i), as required by the learning rule presented in Section II.

Observe in Fig. 4(b) that we have added a global control signal lv. When signal lv is activated, for the selected row (ll_j), signal lc goes high, and the memory cells on that row, are forced to store the input values applied at that moment. At the initial stage, an all 1's input vector $\mathbf{b}$ is applied to the system while signal lv is activated, and signal ll_j is activated in a row-by-row basis. Thus, initial reset of the weights z_{ij} is performed. This feature is also useful for forcing any initial arbitrary value at $\mathbf{z}_j$.

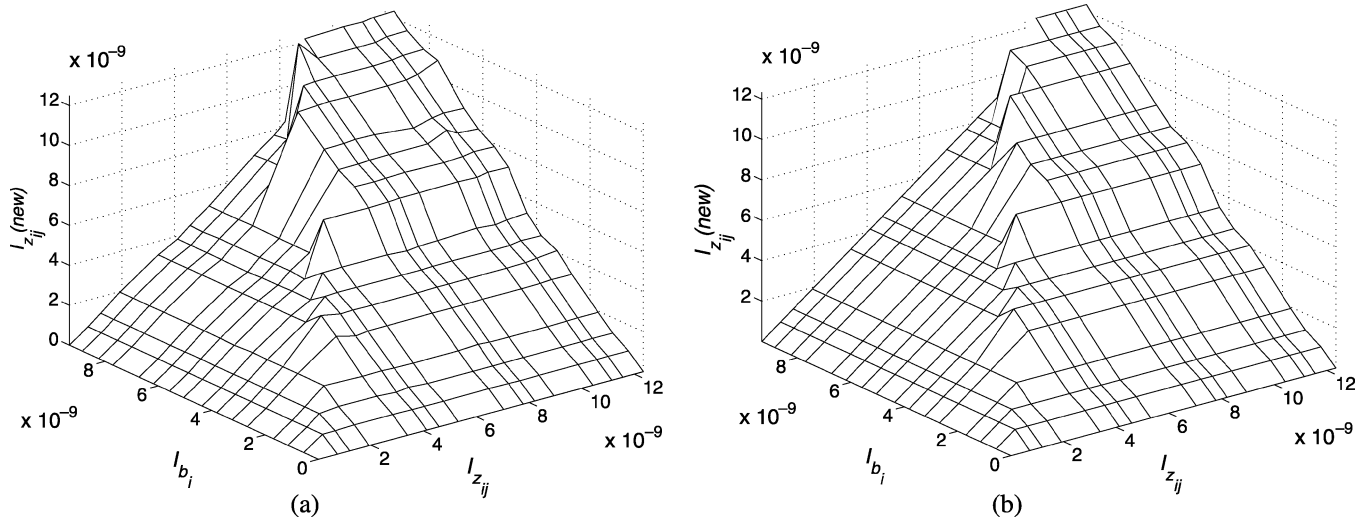

Fig. 9. (a) Measured $I_{z_{ij}}$ current after learning. (b) Computed learning rule.

IV. Experimental Results

The described prototype has been fabricated in the AMS 0.35-μm double-poly triple-metal complementary MOS (CMOS) technology. The area of the complete cell is 44 $\times$ 34 μm^2. Thus, a 128 $\times$ 128 array could be assembled in a chip with a total area of 16 $\times$ 4.8 mm^2.

In this section, we present experimental results of the "bias column" cells and the fuzzy-ART array cell. The currents used in these experiments are in the order of 10 nA what makes the system clearly outperform the system reported in [10] in terms of current consumption, as these currents are 50–100 times lower. This system is comparable to the system in [12] in terms of current consumption and precision; however, it achieves long-term weight storage. Compared to the fully digital system in [11], it consumes both less area and power, although it is less precise. Table I compares the performance of this implementation compared with previously reported ones [10]–[12]. As the compared designs have been fabricated in different technologies, in order to compare the cell area, we have expressed the cell areas in λ units. λ is the minimum gate length allowed in the corresponding fabrication technology.

A. Measurement of "Column Bias" Cells

Fig. 6 depicts the input current I_{b_i} generated in the "bias column" block (see Fig. 3) versus the digital input word $b_i\langle 0:3\rangle$. The currents I_{b_i} are going to be summed along the row of "bias column" cells, thus generating a total current $I_b = \sum_{i=1}^{N} I_{b_i}$ to be used by the vigilance subsystem to check the similarity criterion between the input and the different weight templates.

B. Measurement of "Weight Currents"

Fig. 7(a) shows the measured analog current $I_{z_{ij}}$ generated in the cell versus the digital stored weight $z_{ij}\langle 0:3\rangle$. Superimposed in Fig. 7(a), we have represented the best linear fit. In Fig. 7(b), the error in percentage between the true value and the linear fit has been represented. The maximum error corresponds to 4.48%.

C. Cell Operation

The cell operation is tested in the following way.

1) A weight value z_{ij} is stored in the cell memory.
2) An input word b_i is loaded.
3) The fuzzy minimum $b_i \wedge z_{ij}$ is computed through proper sequencing of signals $c\min$ and $c\max$. The resulting minimum current $I_{b_i} \wedge I_{z_{ij}}$ is measured through pin $\min$.
4) Learning signal ll_j is activated.
5) The updated z_{ij} is measured.

Fig. 8(a) depicts the current measured through pin $\min$ after the computation of the minimum as a function of 16 generated currents I_{b_i} and of 16 generated weight currents $I_{z_{ij}}$. Current I_{b_i} is swept between 0–9.5 nA, while current $I_{z_{ij}}$ changes between 0–12.2 nA. In Fig. 8(b), we show for comparison the ideal result of the minimum operation $\min(I_{b_i}, I_{z_{ij}})$ as a function of the same currents. The maximum error between measured and ideally computed results is 18%, which occurs for $I_{b_i} = 9.3$ nA and $I_{z_{ij}} = 12.3$ nA. At this point the theoretical minimum current is $\min(I_{b_i}, I_{z_{ij}}) = 9.3$ nA; however, the measured minimum current is 11.4 nA.

Fig. 9(a) plots the $I_{z_{ij}}$ current measured after the updating of the weights $I_{z_{ij}}(\text{new})$. In Fig. 9(b), we also show for comparison the computed ideal $I_{z_{ij}}(\text{new})$ value, as a function of the initial input currents I_{b_i} and $I_{z_{ij}}$. Maximum error between measured and ideal results is 34%, obtained for $I_{b_i} = 8.8$ nA and $I_{z_{ij}} = 8.2$ nA.

V. Conclusion

A low-power fuzzy-ART cell has been designed, fabricated, and tested. The cell is designed to operate in weak inversion with currents as low as one nanoampere. Maximum cell current consumption is 22 nA. This low current consumption makes this cell specially suitable to implement fuzzy-ART systems in portable applications. The cell has been implemented in a CMOS 0.35-μm technology and occupies an area of 44 $\times$ 34 μm^2. The cell was designed to store weights with 4-b precision. This cell is intended to be used in a complete array for a fully functional fuzzy-ART microchip system which is now under development.

References

[1] S. Grossberg, "Adaptive patterns classification and universal recoding I: Parallel development and coding of neural feature detectors," *Biol. Cybern.*, no. 23, pp. 121–134, 1976.
[2] ——, "How does the brain build a cognitive code?," *Psychol. Rev.*, no. 87, pp. 1–51, 1980.
[3] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy-ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Netw.*, no. 4, pp. 759–771, 1991.

[4] T. Serrano-Gotarredona and B. Linares-Barranco, "A real-time clustering microchip neural engine," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 4, no. 2, pp. 195–209, Jun. 1996.

[5] ——, "An ART1 microchip and its use in multi-ART1 systems," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1184–1194, Sep. 1997.

[6] C. S. Lin, T. Srikanthan, K. V. Asari, and S. K. Lam, "Fuzzy-ART based image compression for hardware implementation," in *Proc. Asia–Pacific Conf. Circuits Syst.*, Oct. 2002, vol. 2, pp. 147–150.

[7] R. Araujo, G. Gouveia, and N. Santos, "Mobile robot localization using a fuzzy-ART world model," in *Proc. Annu. Conf. Ind. Electron. Soc.*, Nov. 2002, vol. 3, pp. 2409–2414.

[8] R. Kondadadi and R. Kozma, "A modified fuzzy-ART model for soft document clustering," in *Proc. Int. Joint Conf. Neural Netw.*, May 2002, vol. 3, pp. 2545–2549.

[9] L. Cingue, G. L. Foresti, A. Gumma, and S. Levialdi, "A modified fuzzy-ART model for image segmentation," in *Proc. Int. Conf. Image Anal. Process.*, Sep. 2002, pp. 102–107.

[10] J. Lubkin and G. Cauwenberghs, "VLSI implementation of fuzzy adaptive resonance and learning vector quantization," *Analog Integr. Circuits Signal Process.*, vol. 30, no. 2, pp. 149–157, Feb. 2002.

[11] E. Grange, Y. Blaquiere, Y. Savaria, M. A. Cantin, and P. Lavoie, "A VLSI architecture for fast clustering with fuzzy ART neural networks," *J. Microelectron. Syst. Integr.*, vol. 5, no. 1, pp. 3–18, 1997.

[12] M. Cohen, P. Abshire, and G. Cauwenberghs, "Mixed-mode VLSI implementation of fuzzy ART," in *Proc. IEEE Int. Symp. Circuits Syst.*, Monterey, CA, 1998, vol. 3, pp. 251–254.

[13] J. Lazaro, R. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-takes-all networks of O(N) complexity," *Adv. Neural Inf. Process. Syst.*, vol. 1, pp. 703–711, 1989.

[14] K. Bult and J. G. M. Geelen, "An inherently linear and compact MOST-only current division technique," *IEEE J. Solid State Circuits*, vol. 27, no. 12, pp. 1730–1735, Dec. 1992.

[15] C. C. Enz and E. A. Vittoz, "CMOS low-power analog circuit design," *Tutorials IEEE Int. Symp. Circuits Syst. (ISCAS'96)*, pp. 79–133, 1996, chap. 1.2.

[16] T. Serrano-Gotarredona, B. Linares-Barranco, and A. G. Andreou, "Very wide range tunable CMOS/bipolar current mirror with voltage clamped input," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 11, pp. 1398–1407, Nov. 1999.

[17] ——, "Bipolar/CMOS current-source flip-flop for application in neuro-fuzzy systems," *Electron. Lett.*, vol. 35, no. 16, pp. 1326–1328, Aug. 1999.

[18] P. O. Pouliquen, A. G. Andreou, and K. Strohbehn, "Winner-takes-all associative memory: A hamming distance vector quantizer," *Analog Integr. Circuits Signal Process.*, vol. 13, pp. 211–222, 1997.

[19] A. Demosthenous, S. Smedley, and J. Taylor, "A CMOS analog winner-take-all network for large-scale applications," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 45, no. 3, pp. 300–304, Mar. 1998.

[20] A. G. Andreou and K. Boahen, "Translinear circuits in subthreshold MOS," *Analog Integr. Circuits Signal Process.*, vol. 9, pp. 141–166, Sep. 1996.