

Mixed-Signal VLSI for Neural and Fuzzy Sequential Processors

J. Madrenas, E. Alarcón, J. Cosp, J.M. Moreno, A. Poveda and J. Cabestany

Department of Electronic Engineering, Universitat Politècnica de Catalunya,
Gran Capità s/n, Mòdul C4 Campus Nord UPC, 08034 Barcelona, Spain
Phone: +34 93 401 67 47, Fax: +34 93 401 67 56, E-mail: madrenas@eel.upc.es

*Proceedings of the 2000 IEEE International Symposium on Circuits and Systems - ISCAS'00,
Gèneve, Switzerland, June 2000.*

©2000 IEEE. Personal use of this material is permitted. However, permission to reprint or republish this material for advertising or promotional purposes or for creating new collecting works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reported without the explicit permission of the copyright holder.

MIXED-SIGNAL VLSI FOR NEURAL AND FUZZY SEQUENTIAL PROCESSORS

J. Madrenas, E. Alarcón, J. Cosp, J.M. Moreno, A. Poveda and J. Cabestany

Department of Electronic Engineering, Universitat Politècnica de Catalunya,
Gran Capità s/n, Mòdul C4 Campus Nord UPC, 08034 Barcelona, Spain
Phone: +34 93 401 67 47, Fax: +34 93 401 67 56, E-mail: madrenas@eel.upc.es

ABSTRACT

A sequentiality study for mixed-signal VLSI implementations of neuro/fuzzy feedforward algorithms is presented. Implications of sequential processing and mixed-signal operation are derived. Basic building blocks for sequential mixed-signal neural and fuzzy computing are proposed, and two sequential example processors are described. Feedback from designed processors and subcircuits allows considering the technology constraints for analysis and extension to different sequentiality degrees.

1. INTRODUCTION

Hardware implementations of neural networks and fuzzy controllers by means of analog processing exhibit compact, fast and low power characteristics in front of digital realizations. This is achieved at the expense of low resolution and an increased sensitivity to noise, temperature and non-idealities [1-3].

Although precision requirements for the learning phase can be high, for the recall phase they are less than 8 equivalent bits in most neural applications [4]. Therefore, proper analog circuits can clearly perform this phase. In addition, learning can be done by means of chip-in-the-loop techniques.

Usually, analog implementations are fully parallel one-to-one mappings of the processing network. This results in high complexity of connections between processing elements as a consequence of the so-called curse of dimensionality. Despite the compact analog implementations, for real-world applications the number of synapses and processing units usually becomes too high to justify a fully parallel implementation, especially when speed can be reduced and still match the application needs.

In contrast, hybrid parallel/sequential architectures can match the analog processor cost, speed and power consumption to the application constraints. Furthermore, in general, sequentialization exhibits flexibility to emulate a different number of network structures and models (e.g. evolutive networks) with much better efficiency than a fully parallel architecture.

A significant weak point in analog implementations is the lack of suitable memory elements. However, in the last years several dynamic and non-volatile analog memory circuits have been developed. This enables the implementation of compact analog architectures.

As far as analog sequential processing is concerned, we can take advantage of several developments on switched-current circuits. These techniques perform current-mode, discrete-time processing [5], using the MOS gate capacitor for temporary charge storage.

Being analog dynamic memories available, sequential processing schemes can be envisaged. A digital block becomes necessary to control the sequential processing, so such systems require mixed-signal design techniques. However, in contrast to most mixed-signal circuits, a distinct feature of the proposed circuits is that the main processing is performed at the analog part.

Although several mixed-signal sequential realizations have been reported, e.g. [6-9], to our knowledge no study has been done up to now on generalization of the area-delay tradeoff. In this paper, we consider two reported sequential realizations that cover feedforward algorithms (MLP, RBF and Takagi-Sugeno –TSK– fuzzy controllers) for the *recall phase*. The common points and sequentialization possibilities are analyzed, and a consistent realization style that allows tuning sequentiality is proposed.

In section 2 we explore the area-delay design space of analog neural and fuzzy processors as a function of sequentiality degree. The most significant basic blocks are described in section 3. The approach feasibility is illustrated in section 4 with two processors that have been implemented in VLSI. Finally, conclusions are discussed.

2. SEQUENTIALITY ANALYSIS

The trivial solution for an analog implementation of a neural or fuzzy system is one-to-one mapping of each synapse and activation function of processing elements onto physical resources. In this case, internal memory is only needed to store the system parameters.

At the opposite side lies a single processor emulating serially synapses and activation functions. In this case, in addition to parameters, memory is required to store temporary intermediate values, as it occurs in single-processor digital implementations. The equivalent mixed-signal elementary processor is shown in Fig. 1, for a) MLP, b) fuzzy controllers and c) Radial Basis Functions (RBF). Aside of the nonlinear activation function and distance/ t -norm calculations, the multiplier-accumulator is an ubiquitous element. It employs a temporary storage element (Analog Transfer Cell, ATC).

The digital controller and parameter storage elements are not shown. Between the extreme solutions, any combination of processors executing in parallel, each one emulating several processing elements, is possible. The following subsections comment on two reported architectures of this kind.

2.1 MLP-like emulation. In Fig. 1d, a synapse-parallel MLP emulating processor is shown [8,12,17]. For this kind of systems, either synapse or neuron parallelism can be applied. In the former, synapse aggregation takes place in space, while it takes place in time for the latter.

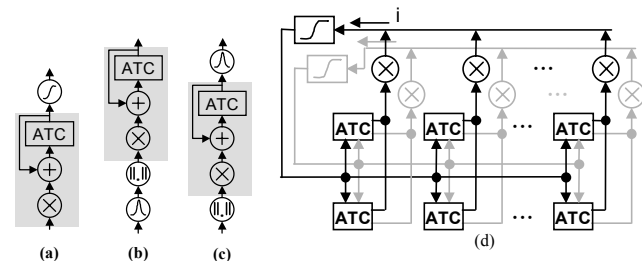


Figure 1. a) MLP slice; b) fuzzy slice; c) RBF slice; d) MLP sequential synapse-parallel realization.

The processor operation principle is based on the sequential emulation of each neuron on a layer-by-layer basis. Let us consider only the black lines. Briefly explained, the input data vector is assumed to be stored in a first row of ATCs. At each clock cycle, all the synapses of one neuron and its activation function are computed. The activation outputs are sequentially stored in the second row of ATCs. After emulating all the neurons on the first neuron layer, the computed outputs can be used as inputs to emulate the next layer. The procedure is repeated until last layer is emulated. Two rows of n ATCs, n synapses and one activation function are enough to emulate a network of an arbitrary number of layers, each one consisting of up to n neurons.

The shadowed drawing in Fig. 1d shows the additional elements needed to add one degree of neuron parallelism to the processor. The area overhead is almost duplicated (only the ATCs stay the same), but also speed since two activation outputs are calculated at each time.

Comparing synapse and neuron parallelism, the former is more area efficient since the number of required activation functions reduces to one, while for each neuron to emulate in parallel an activation function circuit is needed.

Also, layer parallelism can be considered. As in the case of neuron parallelism, for each parallel layer an activation function is required. Furthermore, in this case additional ATCs to store intra-layer data are needed. In Table I, resource figures and execution cycles for different degrees of sequentiality schemes are shown, where s_s , s_n and s_l represent the synapse, neuron and layer sequentialization indexes, the maximum value representing fully sequential and 0 fully parallel. For the sake of simplicity, n stands for the number of synapses per unit and also units per layer. However, extension is straightforward. Index m represents the number of layers. The first row corresponds to a fully sequential system, the shadowed row represents the reported example synapse-parallel processor [12], and the last row corresponds to a fully parallel system. Control overhead is not considered.

s_s	s_n	s_l	# p	# f.	#ATC	T
n	n	m	1	1	$2n$	$m \cdot n^2$
0	n	m	n	1	$2n$	$m \cdot n$
n	0	m	n	n	$2n$	$m \cdot n$
n	n	0	m	m	$2n \cdot m$	n^2
n	0	0	$n \cdot m$	n^2	$2n \cdot m$	n
0	n	0	$n \cdot m$	m	$2n \cdot m$	n
0	0	m	n^2	n	$2n$	m
0	0	0	$m \cdot n^2$	n^2	0	1

Table I. Resources and execution time function of sequentiality.
p: Multipliers; # f: Activation functions; T: Clock cycles.

2.2 Fuzzy/RBF-like emulation. The second sequential example architecture was reported in [9]. It is depicted in Fig. 2a, and, concerning hardware complexity, efficiently maps the Fuzzy Knowledge-Based Controller (FKBC) for the widely accepted TSK first-order model [10].

The motivation for the sequential fuzzy controller is again the capability to exchange speed for area and emulation flexibility. Table II summarizes the hardware complexity of the inference block (Fig. 2b) in terms of the degree of sequentiality. An n -dimensional input space, with a granularity of m membership functions per input dimension, and the associated m^n rules are considered as the general description of the operation of the fuzzy controller.

Considering a designed testchip [11], which is a 3-input and 2-membership-functions per input FKBC, an architecture exhibiting a certain degree of sequentiality and parallelism is considered (shaded row in Table II). In Fig. 2b, the configuration for the inference block with 3 inputs (x_1, x_2, x_3) and 2 membership functions defined per input is shown. Concerning the consequent block, it consists of a weighted

sum of products and thus it can be analyzed with the guidelines given for MLP implementations. The consequent block implements a 1st-order TSK model (d_i in Fig. 2a), very suitable for control applications [10]. The blocks labelled F_{k,x_i} in the figure implement the k -th membership function for the i -th input, while the blocks labelled $\min()$ provide at their outputs the minimum of their two input signals. The membership functions for the two first inputs are calculated sequentially, while those of the remaining input are calculated in parallel. Thus, at each emulation step two fuzzy rules comprising the 3 inputs are yielded. As a consequence, the 8 rules of the system are calculated in 4 emulation cycles. In a general case, the inference block would provide the complete set of fuzzy rules $M=m^n$ in m^{n-1} cycles, being the number of membership functions required $(n-1)+m$, as Table II shows.

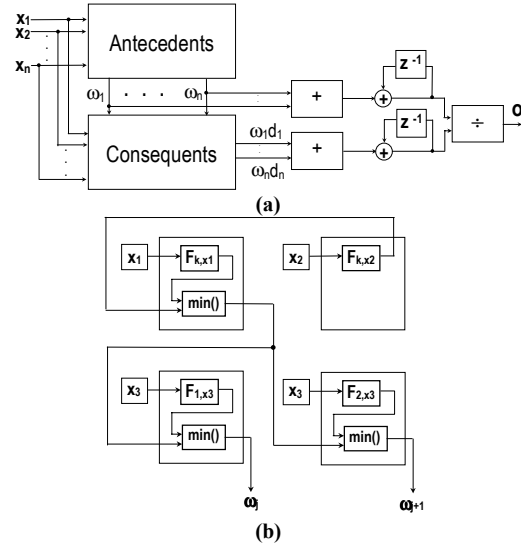


Figure 2. a) Diagram of the FKBC analog sequential architecture example. b) Inference organization.

s	# M.F.	# min.	T	# FR
n	n	$n-1$	m^n	1
$n-1$	$(n-1)+m$	$(n-2)+m$	m^{n-1}	m
...
2	$2+m \cdot (n-2)$	$1 + \sum_{i=1}^{n-2} m^i$	m^2	m^{n-2}
1	$1+m \cdot (n-1)$	$\sum_{i=1}^{n-1} m^i$	m^1	m^{n-1}
0	$m \cdot n$	$\sum_{i=2}^n m^i$	1	m^n

Table II. Architecture complexity of the inference block as a function of the degree of sequentiality s . # M.F: Number of membership functions. # FR: Number of fuzzy rules delivered in each cycle.

3. VLSI CIRCUITS FOR MIXED-SIGNAL SEQUENTIAL IMPLEMENTATIONS

In this section, we introduce several building blocks that have been used to design the two example sequential processors [11,12] described in the previous section. These blocks are suitable to be applied to a broad family of processors with several degrees of sequentiality. Apart from using current-mode processing (inherently suited to addition operations), the main blocks are divided into four

categories: ATCs, multipliers, activation/membership functions and minimum/distance calculations.

3.1 Analog Transfer Cells (ATC). Memory elements for temporary analog storage can be efficiently built using regulated-cascode current copiers (Fig. 3) [13]. The main elements are transistors $T_1..T_3$ and the memory capacitor C_c . This configuration has very high output resistance, so the main storage errors can be considered to be charge injection and clock feedthrough [16]. We have conceived a switching structure that largely reduces these undesired effects by means of a complementary switch and voltage swing tuning of the PMOS transistor. Error in the proposed cancellation scheme has been shown to provide better than 8-bit SNR.

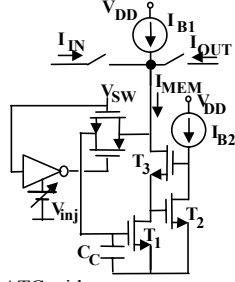


Figure 3. ATC with storage error compensation

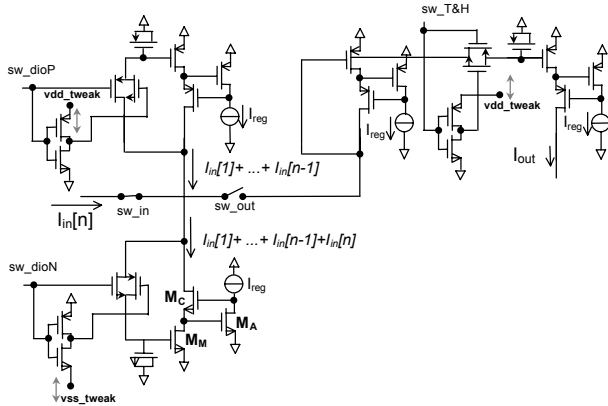


Figure 4. Switched-current discrete-time storing/integrating ATC.

An extension of the previous circuit results in a discrete-time storing and accumulating ATC, which performs discrete-time integration. The designed circuit, shown in Fig. 4, uses complementary-type current copiers and proper switching control signals. For each sequential accumulating cycle, an internal cycle is included to transfer the partial current sum stored at the main NMOS type current-copier to the PMOS type current copier. Thus, the accumulated signal at the k -th instant is:

$$i[k]_{NMOS} = i_{in}[k] + \sum_{j=1}^{k-1} i[j]_{PMOS} = \sum_{j=1}^k i[j]_{NMOS} \quad (1)$$

When all the accumulation steps are done, the final values of stored current are steered to current-mode sample-and-hold circuits, which drive succeeding stages.

It is interesting to observe that a discrete-time accumulator can be replaced by a continuous-time integrator with proper reset pulses. This reverse step to continuous-time operation could simplify the accumulator ATC design and is currently under research.

3.2 Multipliers. Assuming that the system parameters are stored in digital form, digital-to-analog (DAC) multipliers are used for signal weighting. These cells are bulkier than purely analog multipliers, but digital storage is very convenient for network parameters and fits appropriately with the mixed-signal sequential approach.

Furthermore, because of sequentiality operation, the multiplier count is reduced, so that an increase in area overhead can be afforded.

Fig. 5 DAC multiplier is based on a steering-current D/A converter using semi-algebraic techniques [14,18,19]. The use of a two-step algorithmic approach after

$$I_{OUT} = I_{IN} \left(\left(\frac{b_{n-1}}{2} + \dots + \frac{b_{n+1}}{2^2} \right) + \frac{1}{2^2} \left(\frac{b_{n-1}}{2} + \dots + \frac{b_0}{2^2} \right) \right) \quad (2)$$

results in a significant area reduction. In the figure, a 6-bit low-voltage current-steering DAC multiplier is depicted, but an 8-bit version is also available.

To perform the product of the consequent by the rule activity in FKBCs 1st-order TSK models, pure analog product is necessary [11]. A compact, 2-quadrant analog multiplier is shown in Fig 6. For this current multiplier, a four bipolar transistor translinear cell has been designed. This cell naturally obtains current-domain products by virtue of voltage summations around the so-called translinear loop. The shaded PNP bipolar transistors in Fig. 6 stand for parasitic lateral bipolar transistors available in standard CMOS technology.

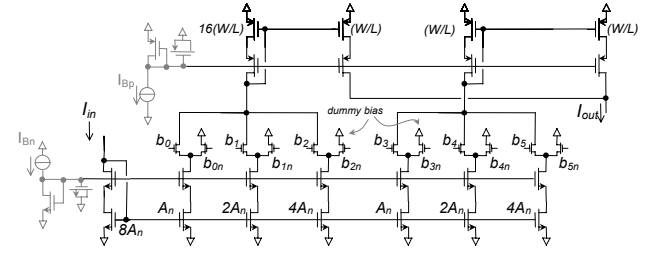


Figure 5. Mixed-signal multiplier based on high-speed current-mode digital-to-analog converter.

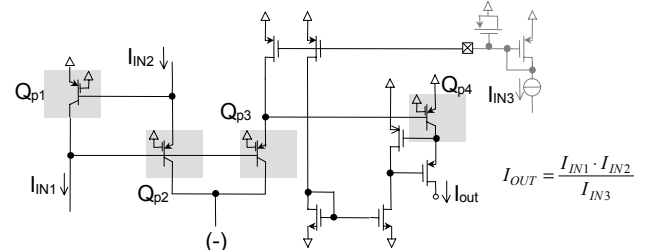


Figure 6. Parasitic-bipolar translinear current-multiplier with regulated cascode current output.

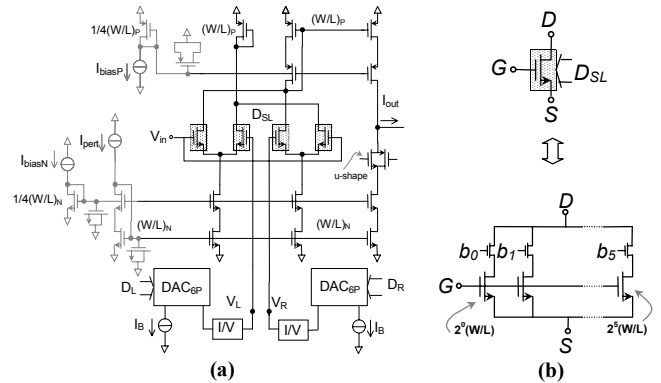


Figure 7. a) Local function. b) Transistor array for digital slope adjustment.

3.3 Activation/membership functions. For sigmoid functions, the most natural circuit is the differential pair. However, this voltage-

current block requires a previous I/V converter to obtain a current-input current-output sigmoid circuit. For low-level currents, this converter needs to have high resistance, so the most suitable device is a transistor operating as an active load [12]. Local functions (*i.e.* fuzzy membership functions or RBFs) are implemented with the circuit in Fig. 7. As the DAC multiplier, it is digitally programmable. In this case, position and slope are configured with 6-bit resolution, being this enough for controller applications. Voltage offsets are applied at the differential pairs to adjust the position of the membership transition region over the input voltage. These voltage levels are obtained by means of current-steering D/A converters, as those shown in Fig. 5, biased with constant currents.

3.4 Minimum/distance calculation. The adopted minimum circuit is described in [15]. The circuit uses parasitic PNP transistors. Concerning distance calculation, Manhattan distance is straightforward using current comparators, while Euclidean distance can be performed by means of squarer circuits [20].

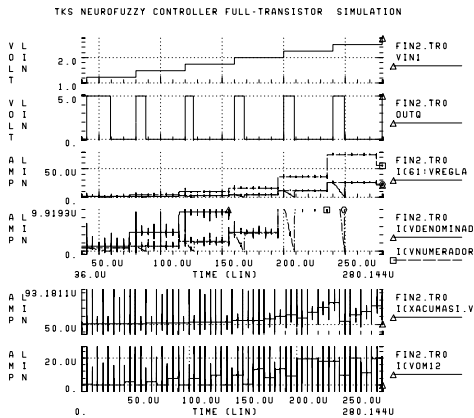


Figure 8. Discrete-time integration and output waveforms.

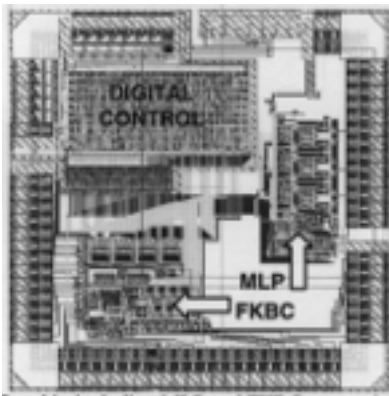


Figure 9. Testchip including MLP and FKBC sequential processors.

4. MIXED-SIGNAL SEQUENTIAL PROCESSOR DESIGN

Sequential MLP and FKBC processors have been designed using a 0.8 μ m CMOS analog technology. Simulations show proper operation. As an example, Fig. 8 shows a post-layout simulation of the FKBC processor operation. It illustrates the current-sample accumulation. As an output divider circuit a pulse-width modulator (PWM) circuit is used [21], since the processor is targeted to control applications. Fig. 9 shows the final layout. The area of an 8-bit 3-synapse Multi-Layer Perceptron is 1mm². The analog part of the FKBC controller occupies 0.85mm² for the eight-rule/three-input prototype testchip. A flexible

digital controller has been included to check all the processor capabilities. In a final design it could be significantly reduced.

5. CONCLUSIONS

Issues on sequential design of neural and fuzzy processors have been analyzed. Proper selection of the sequentiality allows an application-specific efficient selection of the point in the power, speed and area design space. Furthermore, mixed-signal sequential processing exhibits high flexibility to emulate different system structures. Two sequential processors which show the feasibility of the approach have been designed and validated.

A relevant conclusion is that sequentialization considerations are not technology independent, but take into account technology and subcircuit constraints. A consistent design style has been proposed.

The processors have been manufactured, and are currently being fully characterized. A quantitative comparison in terms of speed, area and power consumption between the several solutions will be performed.

Acknowledgements. Work supported by CICYT Project No. TIC96-1195. Jordi Cosp holds a CIRIT Research fellowship.

6. REFERENCES

- [1] Glessner M., Pöschmüller W., Neurocomputers, Chapman and Hall, 1994.
- [2] Del Corso D., "Hardware Implementations of Artificial Neural Networks", *Lecture Notes in Computer Science*, J. Mira, F. Sandoval (Eds.), pp. 405-419, Springer-Verlag, 1995.
- [3] Madrenas J., et al. "Analog and Mixed-Signal Neural VLSI Processors: Taxonomy, Comparison and Performance Evaluation", *Proc. MIXDES'96*, Lodz (Poland), 1996, pp. 389-400.
- [4] Schalkoff R.J., Artificial Neural Networks, McGraw-Hill, 1997.
- [5] C. Toumazou, F.J. Lidgley and D. Haigh, Editors "Analogue IC design: The current-mode approach", IEE Peter Peregrinus, London, 1991.
- [6] Yazdi N. et al., "Pipelined Analog Multi-Layer Feedforward Neural Networks", *Proc. IEEE ISCAS'93*, pp. 2768-2771.
- [7] Lee JC et al., "A Mixed-Signal VLSI Neuroprocessor for Image Restoration", *IEEE Trans. on CAS*, Vol 2, No.3, Sept. 29, pp. 319-324.
- [8] Moreno J.M. et al., "An Analog Systolic Neural Processor Architecture", *IEEE MICRO*, Vol.14, No.3, Jun. 94, pp.51-59.
- [9] Moreno J.M. et al. "Analog Sequential Architecture for Neuro-Fuzzy Models VLSI Implementation", *Proc. ICANN'97*, Lausanne, Oct 1997, pp.1199-1204.
- [10] Takagi T., Sugeno M., "Fuzzy Identification of Systems and its Applications to Modelling and Control", *IEEE Trans. on Systems, Man and Cybernetics*, Vol 15, n°1. Jan. 1985.
- [11] E. Alarcón et al., "Implementation of an Application-Specific Fuzzy Controller by Means of a Mixed-Signal Sequential Architecture", *Proc. IEEE MWSCAS'98*, Univ. of Notre Dame, Indiana, Aug. 98.
- [12] Madrenas J., et al., "VLSI Design of a Flexible-Structure Sequential Mixed-Signal Neural Processor", *MIXDES'99*, pp. 259-264, Krakow (Poland).
- [13] Toumazou C., et al., Eds. "Switched-Currents: An analogue technique for digital technology", IEE Peter Peregrinus, London, 1994.
- [14] Paulino N. and Franca J.E., "A CMOS digitally programmable current multiplier", *Proc. IEEE ISCAS'96*, pp. 254-257, May 1996.
- [15] Ramirez-Angulo J., et al., "Current-Mode and Voltage-Mode VLSI Fuzzy Processor architecture", *Proc. IEEE ISCAS'95*, pp. 1156.
- [16] Espejo S. et al., "Tunable feedthrough cancellation in switched-current circuits", *IEE Electronics Letters*, Vol. 30, No. 23, pp. 1912-1914, Nov. 94.
- [17] Moreno J.M. et al., "Systolic Modular VLSI Architecture for Multi-Model Neural Network Implementation", *Proc. of the 4th Int. Conf. on Microelectronics for Neural Networks and Fuzzy Syst.*, pp. 118-124, 1994.
- [18] K.L. Fong and A.T. Salama, "A 10 bit semi-algorithmic current mode DAC", *Proc. IEEE ISCAS'93*, pp. 978.
- [19] Bastiaansen C.A.A., et al., "A 10-b 40-MHz 0.8 μ m CMOS current-output D/A converter", *IEEE Jour. of Solid-State Circ.*, Vol. 26, N°7, July 91, pp. 917.
- [20] Bult K., Wallinga H., "A class of analog CMOS circuits based on the square-law characteristic of an MOS transistor in saturation", *IEEE Journal of Solid-State Circuits*, Vol. 22, N°3, June 1987, pp. 357.
- [21] Alarcón E., et al., "Novel Pulse-Width-Modulated Current-Mode Analog Defuzzifier for the Fuzzy Control of Switching DC-DC Converters", *Proc. IEEE MWSCAS 99*, NMSU, Aug. 1999.