

Design Space Tradeoff in VLSI Implementations of Mixed-Signal Neuro-Fuzzy Processors.

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

Department of Electronic Engineering, Technical University of Catalunya,
Jordi Girona 1 i 3, 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 Fifth International Symposium on Artificial Life and Robotics - AROB'00,
Oita, JAPAN, January 26-28, 2000.*

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.

DESIGN SPACE TRADEOFF IN VLSI IMPLEMENTATIONS OF MIXED-SIGNAL NEURO-FUZZY PROCESSORS

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

Department of Electronic Engineering, Technical University of Catalunya,
Jordi Girona 1 i 3, 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

Key words: Sequential processing Mixed-signal, Neural networks, Fuzzy controllers, Design space, VLSI

ABSTRACT

Mixed-signal VLSI design provides a tradeoff between the fast and compact but fixed analog implementations of neuro/fuzzy feedforward algorithms and the programmable but area and power consuming digital counterparts. In this paper, a sequentiality study of such systems is performed. Basic blocks for sequential mixed-signal neural and fuzzy computing are proposed, and two sequential example processors are described. Feedback from the designed processors and subcircuits allowed considering the technology constraints for analysis and extension to different sequentiality degrees.

1. INTRODUCTION

Efficiency in terms of silicon area occupancy versus computation performance (speed) is fundamental in autonomous intelligent systems. For perception tasks, such systems take advantage of several proposed neural and fuzzy models. In real-time operation, hardware implementation of the required models can be most suitable. However, for a given application, the hardware resources could be adapted to the specific performance requirements in order to reduce silicon and power cost. In this paper we analyze mixed signal sequential architectures that enable this tradeoff.

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, sensitivity to noise, temperature and non-idealities [1-3].

An important 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.

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 architecture 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 (e.g. evolutive networks) and models with much better efficiency than a fully parallel architecture.

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 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 Sugeno-Takagi 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 VLSI implementation of two test processors is summarized in section 3. An example of cost-performance tradeoff is described in section 4. 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 happens in single-processor digital implementations. The equivalent mixed-signal elementary processor is shown in

Fig. 1, for a) Multi-Layer Perceptron (MLP), b) Fuzzy Knowledge-Based Controllers (FKBC) and c) Radial Basis Functions (RBF). Aside of the nonlinear activation function and distance/minimum calculations, the multiplier-accumulator is a ubiquitous element. It uses a temporary storage element (Analog Transfer Cell, ATC). The digital controller and parameter storage elements are not shown.

In 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 [8,10,11]. 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.

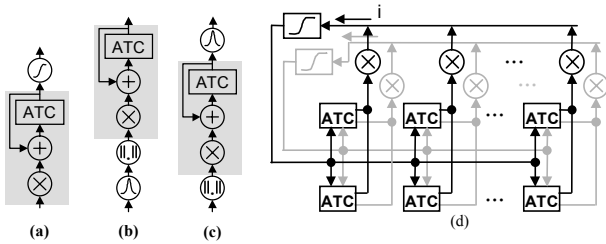


Fig. 1. a) MLP slice; b) fuzzy slice; c) RBF slice; d) MLP sequential synapse-parallel realization.

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 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 [10], 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
1	n	n	m	1	1	2n	$m \cdot n^2$
2	0	n	m	n	1	2n	$m \cdot n$
3	n	0	m	n	n	2n	$m \cdot n$
4	n	n	0	m	m	2n·m	n^2
5	n	0	0	n·m	n·m	2n·m	n
6	0	n	0	n·m	m	2n·m	n
7	0	0	m	n^2	n	2n	m
8	0	0	0	$n^2 \cdot m$	$n^2 \cdot m$	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 Takagi-Sugeno first-order model [12].

Again, the point of the sequential fuzzy controller is 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.

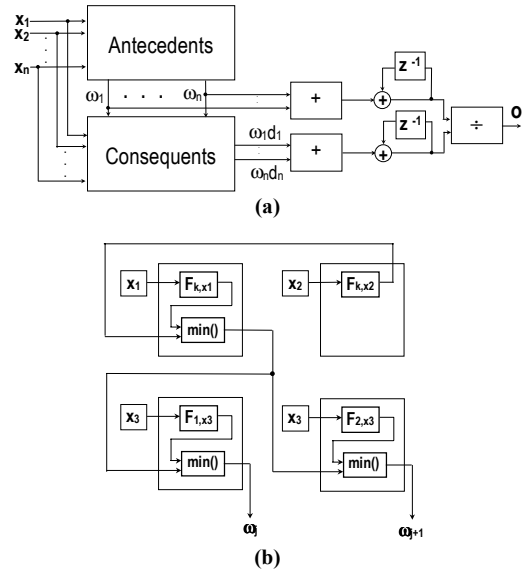


Fig 2. a) Diagram of the FKBC analog sequential architecture example. b) Inference organization.

Considering a designed testchip [13], which is a 3-input and 2-membership-functions per input FKBC, an architecture exhibiting a certain degree of sequentiality (complexity spread over time) 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 are defined for each input. Concerning the consequent block, it consists of a weighted sum of products. Therefore it can be analyzed with the guidelines given for MLP implementations. The consequent block implements a 1st-order Sugeno model, very suitable for control applications [12]. The blocks labelled $F_{k,xi}$ 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 of a fuzzy system with n inputs and m membership functions defined per input, the inference block would provide the complete set of fuzzy rules $M=m^n$ in m^{n-1} cycles. The number of membership functions required is $(n-1)+m$, as Table II shows.

s	# M.F.	# min.	T	# O
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. # O: Number of fuzzy rules delivered in one cycle.

3. VLSI IMPLEMENTATION OF MIXED-SIGNAL SEQUENTIAL PROCESSORS

Two mixed-signal sequential test processors emulating the described MLP and FKBC models have been implemented in a full-custom testchip, using a $0.8\mu\text{m}$ CMOS technology. A photograph of the manufactured chip is shown in Fig. 3. The chip is divided in four parts: (1) 8-bit 3-synapse MLP processor, (2) 8-rule/3-input FKBC processor, (3) Standard-cell digital control common for both processors and (4) Analog delay circuit for digital control signals to reduce switching noise during analog storage.

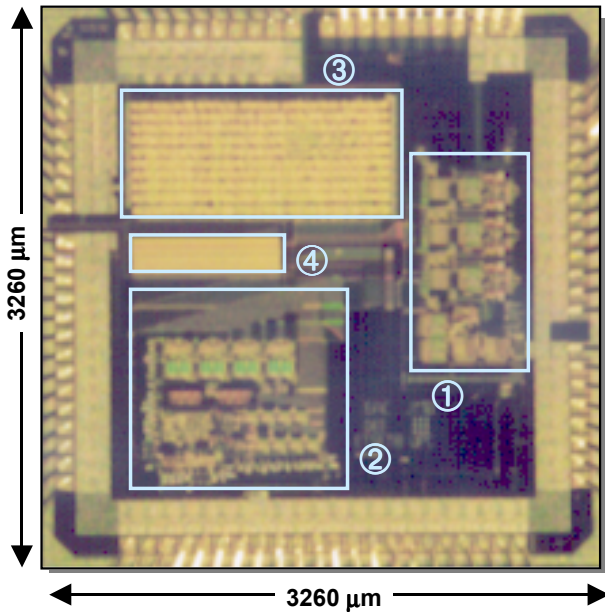


Fig. 3. Testchip including MLP and FKBC sequential processors.

Additional logic and resources were provided to have access to the basic elements of the processors, in order to be able to validate each one individually, so area could be reduced in a final design. Also, the flexible digital controller enables checking all the processor capabilities. It can also be significantly reduced.

Fig. 4 shows an enlarged photograph of the MLP processor. Its area occupancy is 0.79 mm^2 , and it includes three synapse slices (number 1 in the picture), a sigmoid function (2) and an 8-bit scaling circuit (3). The synapse slice contains two ATCs (4) [14,15] and an 8-bit mixed digital-analog multiplier (5) [16-18], as well as control logic. The synapse-slice scheme allows a straightforward change of the number of synapses.

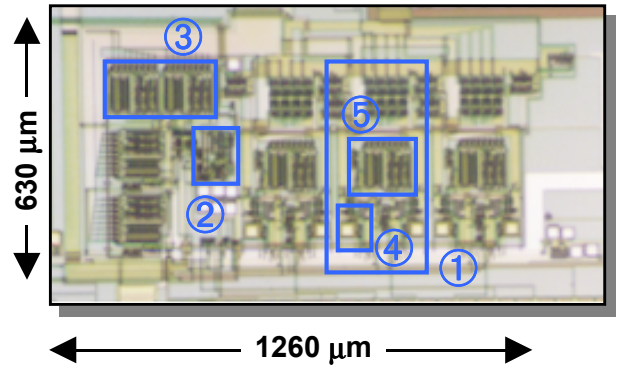


Fig. 4. MLP mixed-signal sequential test processor.

The FKBC controller features about 0.85 mm^2 (block 2 in Fig. 3) and is shown in more detail in Fig. 5. The main composing elements are: Four membership functions (1), three 2-input minimum circuits (2) [19], eight 6-bit mixed digital-analog multipliers (3), two analog translinear multipliers (4), two accumulating ATCs (5) and a pulse-width modulator (PWM) [20] used as a divider and output circuit (6), since the processor is targeted to control applications

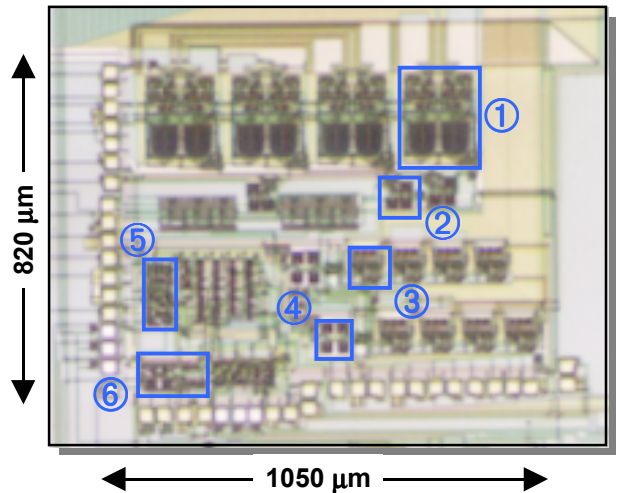


Fig. 5. FKBC mixed-signal sequential test processor.

4. COST-PERFORMANCE TRADEOFF

In this section we explore the design space for the MLP circuit as a design example that can be adapted to the application requirements of cost and performance. As a cost parameter we use an area occupancy index, whereas speed is used as the performance index, measured as the latency in clock periods needed by the processor to generate the output vector from an input vector.

Table III displays the measured area of the three key elements of the MLP processor. Let us assume a real application requirement of three layers of neurons, each layer including 10 neurons, and 10 synapses per neuron. This represents for Table I, $n_n = n_s = 10$ and $n_l = 3$. Using the expressions of that table with these figures, and calculating the area from Table III, we can plot each table row in an area-delay plot (Fig. 6). Of course, the number of possible combinations is much larger, however an interesting set of designs is already obtained.

8-bit A-D multiplier	sigmoid	ATC	
0.025	0.009	0.0135	mm ²

Table III. MLP key elements area.

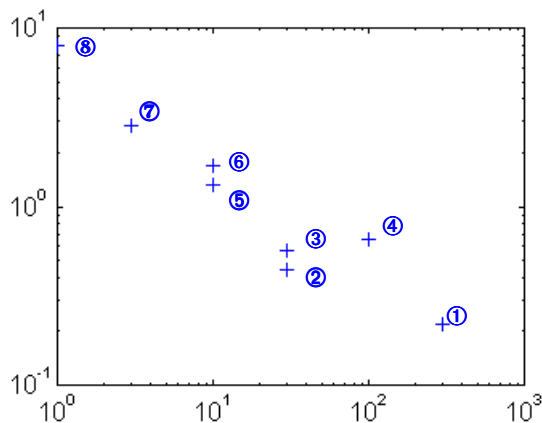


Fig. 6. MLP area-delay plot.

Although the plotted area figures are not directly the processor area, because glue logic and routing area is not accounted, the global area should be, with some deviation, proportional with the key element area. It is very interesting to see that, although some solutions are sub-optimal, there is a set of designs that trade off cost for performance. A hyperbolic curve of fairly constant area-delay product can be deduced as a straight line in the log-log plot of Fig 6. Hence, the most suitable option can be selected for a given application constraints.

5. CONCLUSIONS

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

An important point is that sequentialization considerations are not technology independent, but take into account

technology and subcircuit constraints. A consistent design style has been done.

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

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

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. 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] Madrenas J., et al., "VLSI Design of a Flexible-Structure Sequential Mixed-Signal Neural Processor", MIXDES'99, pp. 259-264, Krakow (Poland).
- [11] 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.
- [12] 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.
- [13] E. Alarcón et al., "Implementation of an Application-Specific Fuzzy Controller by Means of a Mixed-Signal Sequential Architecture", Proc. IEEE MWSCAS'98, Notre-Dame, Indiana, Aug. 98.
- [14] Toumazou C., et al., Eds. "Switched-Currents: An analogue technique for digital technology", IEE Peter Peregrinus, London, 1994.
- [15] Espejo S. et al., "Tunable feedthrough cancellation in switched-current circuits", Electron. Lett., Vol. 30, No. 23, pp. 1912-1914, Nov. 94.
- [16] Paulino N. and Franca J.E., "A CMOS digitally programmable current multiplier", Proc. IEEE ISCAS'96, May 1996.
- [17] K.L. Fong and A.T. Salama, "A 10 bit semi-algorithmic current mode DAC", Proc. IEEE ISCAS'93, pp. 978.
- [18] Bastiaansen C.A.A., et al., "A 10-b 40-MHz 0.8µm CMOS current-output D/A converter", IEEE Jour. of Solid-State Circ., Vol. 26, N°7, July 91, pp. 917.
- [19] Ramirez-Angulo J., et al., "Current-Mode and Voltage-Mode VLSI Fuzzy Processor architecture", Proc. IEEE ISCAS'95, pp. 1156.
- [20] Alarcón E., et al., "Novel Pulse-Width-Modulated Current-Mode Analog Defuzzifier for the Fuzzy Control of Switching DC-DC Converters", 42nd MWSCAS. NMSU, Aug. 1999.