

**SERIAL ARCHITECTURE FOR FUZZY CONTROLLERS:
HARDWARE IMPLEMENTATION USING ANALOG/ DIGITAL VLSI TECHNIQUES**

J. L. Huertas, S. Sánchez Solano, A. Barriga, I. Baturone

Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Avda. Reina Mercedes s/n, (Edif. CICA)
E-41012, Sevilla, Spain

*Second International Conference on Fuzzy Logic and Neural Networks (IIZUKA'92),
Vol. 1, pp. 535-538, Iizuka, Japan, July 17-22 1992.*

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 reposted without the explicit permission of the copyright holder.

SERIAL ARCHITECTURE FOR FUZZY CONTROLLERS: HARDWARE IMPLEMENTATION USING ANALOG/ DIGITAL VLSI TECHNIQUES

J. L. Huertas, S. Sánchez-Solano, A. Barriga, I. Baturone

Dept. of Design of Analog Circuits,
Centro Nacional de Microelectrónica,
Edificio CICA, Avda. Reina Mercedes s/n, 41012-Sevilla (Spain)

Key words: fuzzy logic, discrete-time techniques, membership function, CMOS integrated circuits.

Abstract

A new architecture is presented for the implementation of fuzzy systems using analog-digital techniques. This architecture is directed towards allowing the implementation of many rules on the same chip, including the fuzzy inference engine and the defuzzifier. This approach is based on a total or partial sequential operation of both the fuzzifier and the defuzzifier. A basic operational cell for a membership function circuit as well as its programmable version are described and used for realizing the proposed architecture in a CMOS technology.

1. INTRODUCTION

Hardware implementations of fuzzy logic are attracting more and more the attention of researchers, the effectiveness of the actual circuits being a critical point for system builders to adopt fuzzy solutions. Besides the results obtained from purely digital approaches, it seems interesting to explore the possibilities offered by analog techniques, especially in terms of using the best of both worlds to come up with efficient circuits.

This communication addresses the development of fuzzy circuits combining analog and digital techniques. This is carried out at two levels: architecture, and cell design. At the first level, the bottlenecks of reported circuits are considered. In particular, we will focus on Yamakawa's (1) since, although his architecture is a valid solution in many practical cases, it seems interesting to look for modifications able to handle the design of systems with many rules. A way to do that may be based on trading speed and interconnection complexity by resorting to the use of a sampled data approach. An additional advantage of this approach is the compatibility with some well-founded analog techniques that can help in the design of the defuzzifier.

2. SERIAL ARCHITECTURE

An alternative architecture is proposed to cope with systems with many rules. The new architecture uses essentially the same basic cells proposed by Yamakawa, but the coefficients are stored in a digital memory, the number of rules per chip is strongly increased, and both the fuzzifier and the defuzzifier may be included in the same chip.

An overall view of the new architecture is shown in Figure 1, where its main blocks are detailed. Essential to this technique is the definition of an operation cycle, whose duration will depend on the precision we try to attain. This operation cycle will be defined in terms of $N+1$ cycles of a fundamental clock. Each Con-

trol Rule is implemented by a digital RAM, a Digital/Analog Current Converter, some Membership Function Circuits (MFC) and MIN gates. Started an operation cycle, the D/A converter will provide every clock cycle one value of current for truncating the values coming from the MFC's. Hence, the converter performs as a serial Membership Function Generator (MFG) instead of working in parallel (as proposed by Yamakawa). In other words, the N bus lines used by Yamakawa as a fuzzy word are changed into N successive samples representing such a word onto a simple wire. The outputs from every Control Rule are processed by a MAX gate and fed the defuzzifier, which implements a center of gravity method. The first stage of this is formed by two iterative analog adders (discrete-time), preparing the numerator and the denominator of a discrete divider. After N clock cycles the divider will give the final output. A ramp generator is required to provide the defuzzifier with both the upper adder coefficients and the internal timing of the divider.

In principle this architecture is slower than a parallel counterpart. However, since the area required for a parallel implementation is enormous (mainly because of the number of bus lines), solutions reported are based on connecting several (or even many) chips instead of a one-chip alternative. Then, the external interconnection delays bring forth a problem associated with higher delay time as compared with the ideal implementation in just one chip. The aim of the proposed serial approach is to establish a trade-off between operational speed and silicon area occupation, but taking into account the value of actual delays when several chips must be connected. Thus, we can sacrifice a part of the internal speed (throughout a sequential operation) to be sure that many more rules can be implemented on-chip, this avoiding external connections other than I/O pins.

3. DESIGN SPACE

Before going in depth into actual circuits, we need to discuss the abstract design space where we can make moves. Since speed is essential and current-based processing is normally faster than voltage-based, we decided to handle any information processing by means of currents. The only exception are those circuits requiring to store this information either temporarily or permanently, as well as I/O pads.

Processing using MOS current mirrors is a natural candidate, especially after the results in (2). However, there are a few drawbacks to consider. Channel modulation effects, matching errors and threshold offsets may degrade their performance. Also, previously reported circuits (3) (4) are a kind of piecewise-linear functions, the resulting transfer characteristic being formed by concatenating 2-piece component operators. Adjustment of these characteristics are strongly influenced by dimension errors, the

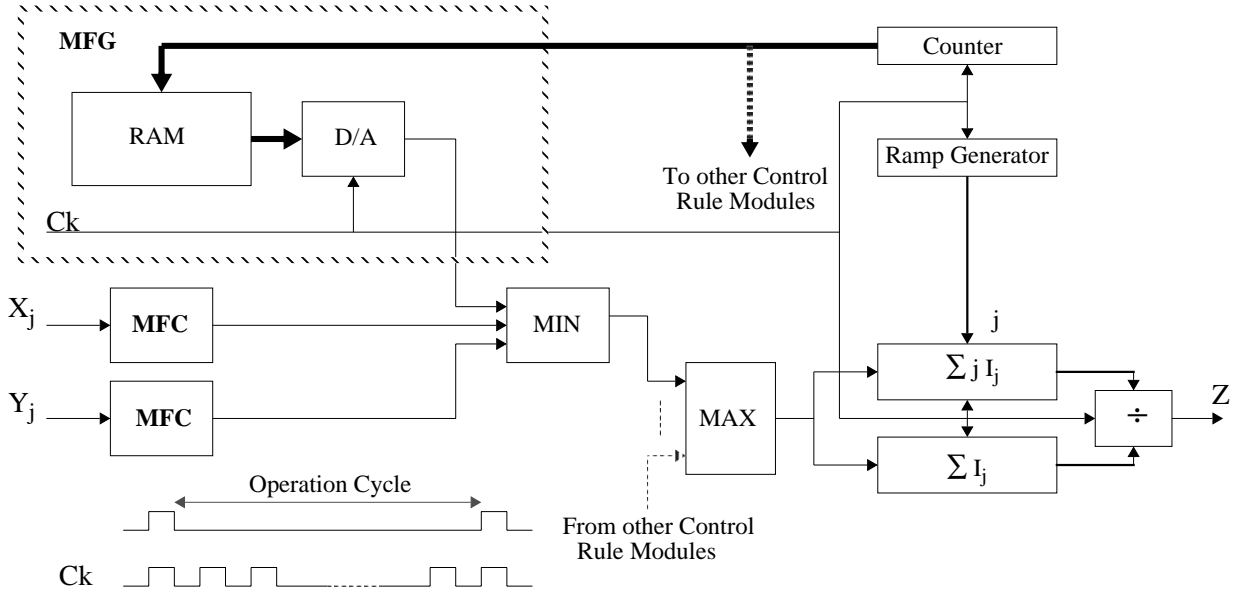


Fig. 1: Proposed architecture for the fuzzy controller.

result being especially significant when precise cancellation or compensation of slopes are required. Then, besides a dimensions reduction, an implementation procedure less sensitive to mismatching must be looked for.

Another consideration is concerning the type of curves encountered when “typical” rules are implemented. Instead of arbitrary transfer characteristics, we usually need symmetrical, equally spaced triangular and/or trapezoidal functions, with programmable slopes and a relationship involving the different parameters to be programmed. With this in mind, a cell providing a symmetrical two-branch input-output characteristic would be preferable to any other implementation.

In what follows the integrated circuit realization of this structure will be discussed. After presenting the serial architecture the next step is to deal with alternative implementations for the basic cells. Our aim is to find an optimal implementation taking advantage of VLSI techniques and following the same principles that have oriented our search for a practical architecture, we focus on practical aspects rather than in generality. The on-going activity has targeted the implementation of the fuzzy inference engine. For the sake of space, only the basic cells of the fuzzifier will be presented herein. In particular, let us concentrate in the implementation of the MFC’s.

4. BASIC CELL

The kernel of our fuzzifier is a MFC able to provide a wide number of membership functions as those referred above as well as allow some kind of reconfigurability throughout electrical programming. Figure 2-a shows a circuit implementation for this basic cell. Before explaining the circuit operation, we need to define the four parameters required to identify any trapezoid representing a fuzzy set. In Figure 2-b, all these parameters are depicted; with the exception of the slope m , these parameters are named I_j because they are associated to actual currents in the circuit of Figure 2-a.

Transistors T_1 and T_2 in Figure 2-a together with the current mirror M_1 perform as a current rectifier for the signal difference

$I_{in} - I_{aux}$ (5). I_{in} is the MFC input current and I_{aux} is a current value to set up the symmetry axis for the membership function. When this difference is positive, transistor T_2 is off, transistor T_1 is conducting and forces a current out of the mirror M_1 , this current is a copy of $-(I_{in} - I_{aux})$ and drives the mirror M_2 . On the contrary, when $I_{in} - I_{aux}$ is negative, transistor T_1 is off, transistor T_2 conducts, and the mirror M_2 is driven by a copy of the input current difference.

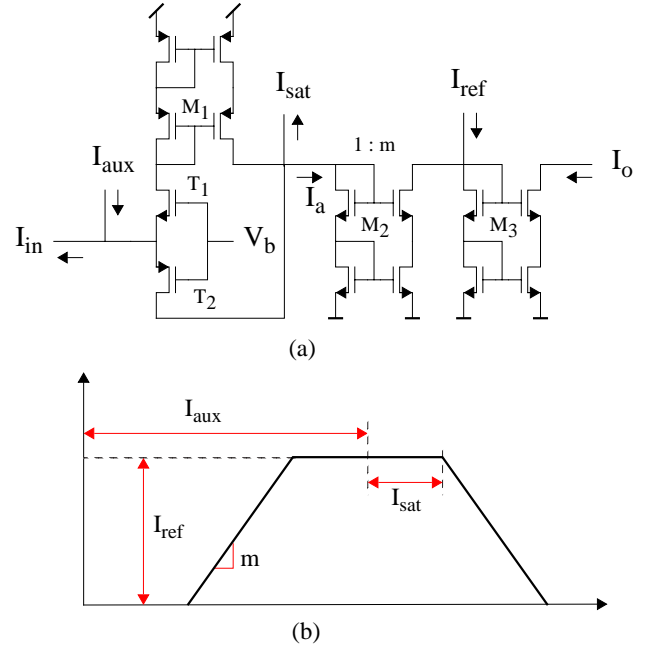


Fig. 2: (a) Circuit schematic for the MFC basic cell, (b) parameters used to identify a symmetrical trapezoid.

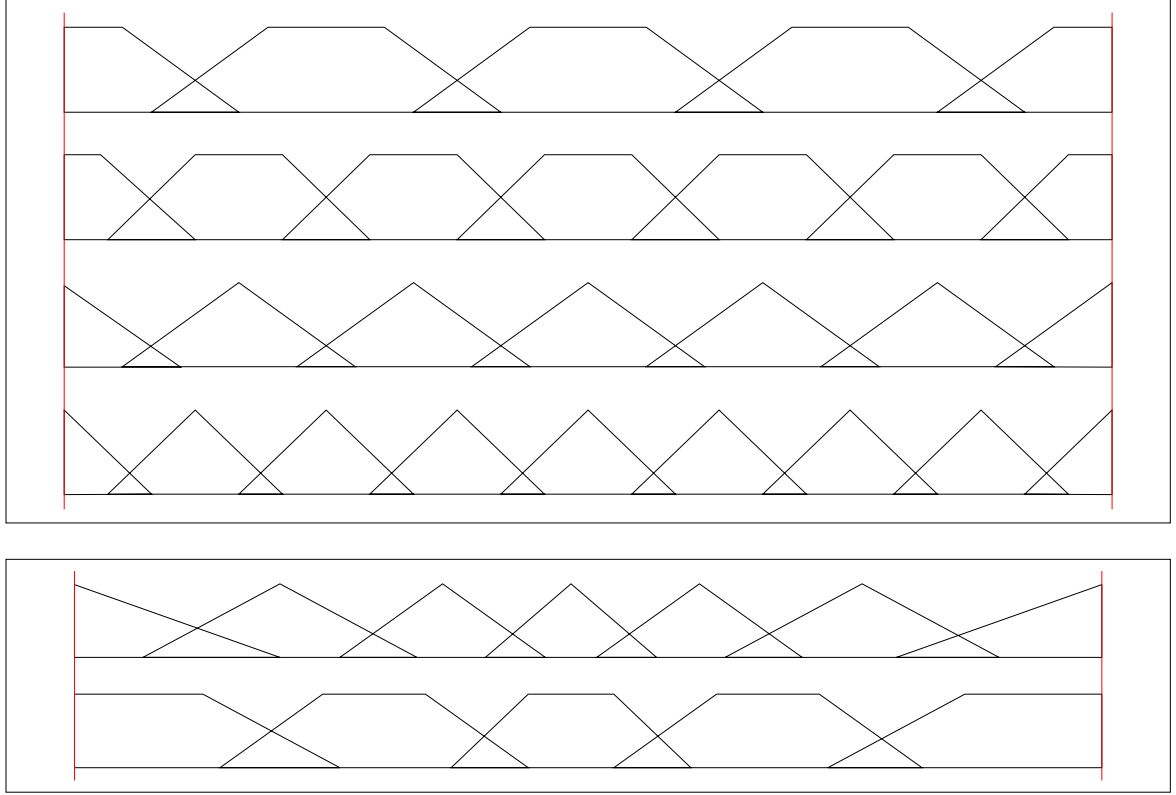


Fig. 3: Simulation results of the MFC. (a) Four sets of functions with the same width, (b) two sets of functions with different width.

In both situations described above, a constant current I_{sat} is subtracted at the input of the mirror M_2 . This mirror multiplies its input current by a factor m , to provide some control on the membership function slope. The value m can be fixed by selecting the aspect ratios of transistors forming this current mirror. Finally, I_{ref} is added before the final stage (mirror M_3). Then the output current can be expressed by:

$$I_o = \begin{cases} I_{ref} - mI_a & \text{if } I_{ref} > mI_a \\ 0 & \text{otherwise} \end{cases}$$

where

$$I_a = \begin{cases} |I_{in} - I_{aux}| - I_{sat} & \text{if } |I_{in} - I_{aux}| > I_{sat} \\ 0 & \text{otherwise} \end{cases}$$

These expressions lead to a geometrical form of the input-output relationship of the shape in Figure 2-b.

A few simulation results can be seen in Figure 3. Four different sets with all their functions of the same width are depicted in Figure 3-a. Cases where the width is different might be found in practice, like the two sets represented in Figure 3-b.

5. PROGRAMMABLE CELL

The circuit in Figure 2-a does not only give an input-output characteristic as is required for a membership function but it paves the way for adding programmability in a rather straightforward manner. Since there are three parameters in Figure 2-b that corre-

spond to currents, they can be actually changed by changing the bias current supplied to the involved node. On the other hand, for the fourth parameter to be changed, the area ratio of transistors in mirror M_2 is made programmable by splitting these transistors into several ones, and connecting and disconnecting the extra transistors depending on the particular ratio required.

For the sake of programming simplicity, we have developed a circuit structure that uses 8 bits to fix the actual function implemented by the circuit in Figure 4. This structure can generate a family of 5, 7, 9 or 11 membership functions, depending on the value of bits C_0 and C_1 . Within each family we can select either triangles or trapezoids with a slope controlled by D_0 and D_1 . Once the number of labels and the function type have been chosen, bits A_0 to A_3 allow to select the particular function to be implemented. Figure 5 is intended for representing the different sets of membership functions that can be implemented by means of the circuit in Figure 4. Since this is a preliminary circuit, there is still room for dealing with a few more sets using the same number of bits to re-program the system.

In the actual circuit we have carried out a trade-off relating dynamic range, area occupation, power and speed. For the available technology we selected $I_{ref} = 15 \mu A$, and a maximum dynamic range for I_{in} of $6 I_{ref}$.

Since this circuit must handle a variable number of labels on a fixed interval, there are a few relationships constraining the parameters. If we fix an overlap of 25% between two adjacent functions, and we call $2l_b$ the width of lower base of the trapezoid and E the number of labels:

$$I_b = \frac{4 \cdot I_{ref}}{E - 1}$$

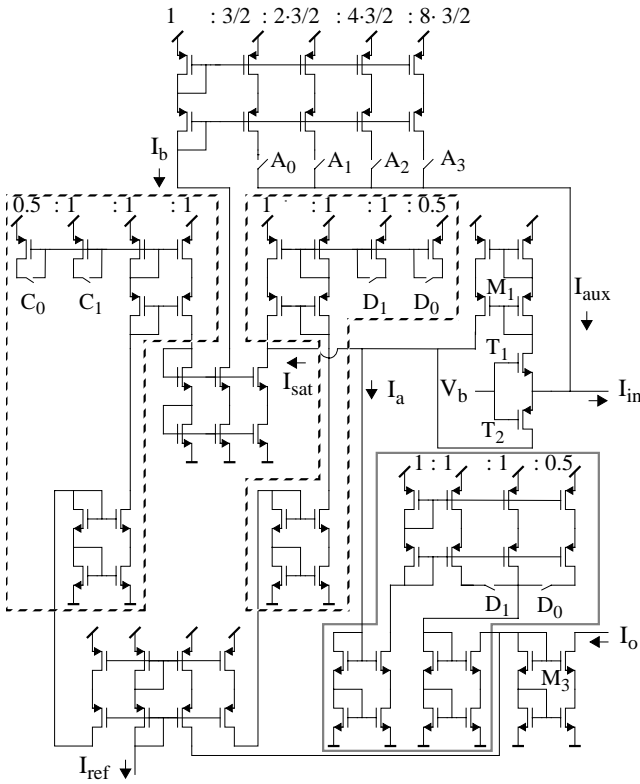


Fig. 4: Circuit schematic for the programmable MFC.

The current I_b drives the four-output mirror at the top of Figure 4 to generate the parameter I_{aux} with an expression given by:

$$I_{aux} \approx \frac{3}{2} \cdot I_b$$

On the other hand, the parameter I_{sat} is obtained from I_b , I_{ref} and m according to the expression:

$$I_{sat} = I_b - \frac{I_{ref}}{m}$$

With the constraints chosen in the implementation of the programmable MFCs, the scaling of I_{ref} to obtain I_b and I_{sat} is performed by the two identical circuits surrounded by dashed lines in Figure 4. Finally, the block surrounded by a dotted line in Figure 4 correspond to the programmable version of the mirror M_2 in the basic cell of Figure 2-a.

6. CONCLUDING REMARKS

We have reported a new architecture for a fuzzy controller. Our approach takes the advantages offered by analog and digital techniques. A sampled data implementation has been chosen for the fuzzy inference engine and the defuzzifier. We have also proposed and implemented a CMOS basic cell for the MFC and its programmable version, able to generate different sets of membership functions selected by an eight-bit digital register.

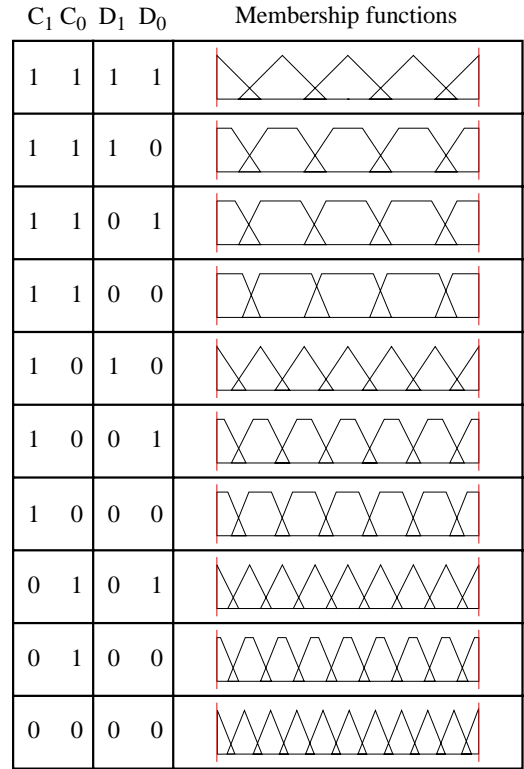


Fig. 5: Function sets implemented by the programmable MFC.

REFERENCES

- [1] T. Yamakawa, "High-Speed Fuzzy Controller Hardware System: The Mega FIPS Machine", Elsevier Science Pub. Comp. Inc., 1988.
- [2] T. Yamakawa, T. Miki and F. Ueno, "The design and fabrication of the current mode fuzzy logic semi-custom IC in the standard CMOS IC technology", Proc. 15th IEEE Int. Symp. Multiple-Valued Logic, pp. 76-82, May 1985.
- [3] T. Yamakawa and H. Kabuo, "A Programmable Fuzzifier Integrated Circuit: Synthesis, Design and Fabrication", Elsevier Science Pub. Comp. Inc., 1988.
- [4] T. Inoue, F. Ueno and T. Motomura, "Analysis and Design of Analog CMOS Building Blocks for Integrated Fuzzy Inference Circuits", ISCAS, 1991.
- [5] Z. Wang, "Novel Pseudo RMS Current Converter for Sinusoidal Signals Using a CMOS Precision Current rectifier", IEEE trans. Instrumentation and Measurement, Vol. 39, N. 4, August 1990.