A FUZZY CONTROLLER USING SWITCHED-CAPACITOR 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 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'93)
Vol. 1, pp. 516-520, San Francisco - California, March 28 - April 1, 1993.

A FUZZY CONTROLLER USING SWITCHED-CAPACITOR 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, SC networks, CMOS integrated circuits.

Abstract

The use of Switched-Capacitor techniques to build a fuzzy controller is discussed in this contribution. Using a sequential architecture, the required building blocks are introduced and its realization is described. The proposed system can be considered as a starting point for exploring the future capabilities offered by SC networks to the hardware implementations of fuzzy systems.

1. INTRODUCTION

Fuzzy logic, although introduced many years ago [1], has not become until recently a practical alternative to conventional computers for performing inference operations. Many applications claiming the use of fuzzy concepts have recently appeared in the marketplace from japanese companies, but most of these products are based on a sort of software simulation of fuzziness using small wordlength conventional microcontrollers.

Besides these results obtained from purely digital approaches, it has been recognized [2, 3] the need of developing actual circuits performing the basic operations required for a fuzzy system to really take advantage of the full flavor of the fuzzy paradigm. Since a few years ago, attention is being paid to the development of circuit implementations for fuzzy logic, because actual fuzzy circuits might be the only way to extend the applicability of fuzzy logic to more demanding application areas. Then, hardware implementations have become a critical issue for adopting fuzzy solutions at a system level. The expected gains in terms of area reduction, operation speed (both at circuit and system levels), and functional flexibility turn out interesting to explore the possibilities offered by analog techniques, especially in those technologies fully compatible with digital circuits. Analog design techniques seem to be very appealing for the implementation of fuzzy circuits and systems. In particular, there exist a well-founded body of theoretical knowledge and practical experience related to linear (especially filters) and nonlinear switched-capacitor (SC) networks that can be applied to the realization of fuzzy controllers [4,5].

This communication addresses the design and implementation of a sequential microcontroller based on SC circuits. 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 [6] 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 in just one chip. 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 sound analog techniques that can help in the design of the defuzzifier. The basic building blocks for the approach will be discussed as well as their use within the microcontroller. This system is conceived as a chip that can be operating embedded into a standard microprocessor environment.

2. SEQUENTIAL ARCHITECTURE DESCRIPTION

An architecture is proposed to deal with systems handling many rules. The new architecture employs essentially the same basic blocks proposed by Yamakawa, but the number of rules per chip can be significantly increased, and both the fuzzifier and the defuzzifier can be included in the same chip.

An overall view of the new architecture is shown in Figure 1, where its main blocks are detailed. This architecture is an adapted version of the one proposed in [7] for a current-mode fuzzy processor. Essential to this technique is the definition of an operation cycle (defined in terms of N cycles of a fundamental clock, Ck) whose duration will depend mainly on the precision we try to attain and the number of rules we consider. Such operation cycle will impose a limitation to the input signal bandwidth. Each Control Rule is implemented by an analog ROM, some Membership Function Circuits (MFC) and MAX/MIN gates. Started an operation cycle, the analog ROM will provide every clock cycle one value of voltage for truncating the values coming from the MFCs. Hence, the ROM performs as a serial Membership Function Generator (MFG) instead of working in parallel (as proposed by Yamakawa). This means that the M bus lines used by Yamakawa as a fuzzy word are replaced by a single wire that carries M successive samples representing such a word. 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 consists of two iterative summers preparing the numerator and the denominator of a discrete divider. After N clock cycles the divider will give the final output.

Basically, we divide an operation cycle into three phases. Phase 1 is devoted to sampling and holding the input variables as well as to pre-processing them trough the MFCs. Phase 2 is aimed to carry out the inference process by performing MIN-MAX operations on the input variables and the MFG outputs. Finally, in phase 3 the defuzzifying process is performed. For the sake of clarity we will call N_j the number of fundamental clock cycles required for the j-th phase. In order to understand the overall structure, we will give in what follows a functional description of the blocks in Figure 1, detailing their circuit implementation and estimating the value for N_j at every operation phase.

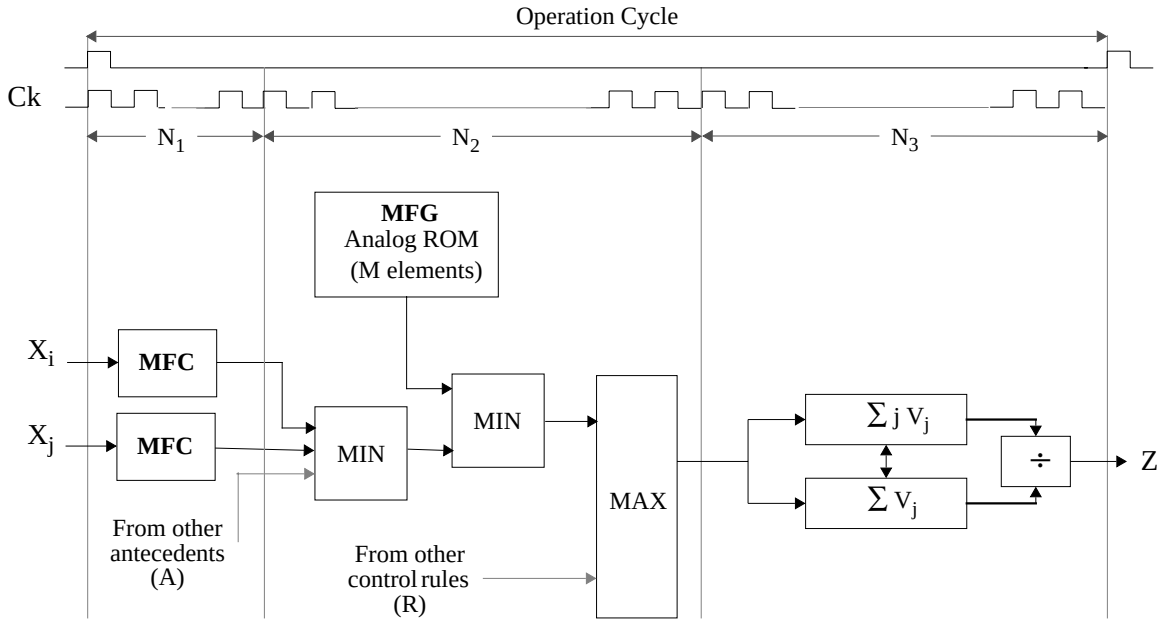


Fig. 1: Proposed architecture for a Switched-Capacitor fuzzy controller.

3. BASIC MFC

The basic element for fuzzy logic is a one-input operator performing a classification of its input variable according to a given membership function. Generally speaking, the functional transformation carried out by this operator is a nonlinear mapping (called a pertenence function), but in most common cases this transformation can be approximated by a symmetric piecewise-linear function of the shape shown in Figure 2-a, where the four parameters required to identify the breakpoints are depicted. To implement any transformation of this form, a possible solution is the circuit in Figure 2-b. The upper part of this circuit performs a piecewise-linear transformation under the control of the lower part. The input variable X is sampled and held sequentially to be compared with four voltage values defining the trapezoid breakpoints. The analog ROM1 in Figure 2-b provides such breakpoints. The results of the comparison are processed by a Finite Sequential Machine (FSM) that controls the switches of the upper circuit. The latter, depending on the result stored by the FSM, transforms the input in accordance with one of the five pieces of its piecewise-linear transformation. The Analog ROM2 supplies the required coefficients for this transformation.

As shown by its operation description, this circuit is active during phase 1 and must hold its output value whilst phases 2 and 3. The total time invested by this circuit to perform the described operation is 4 cycles for carrying out the comparison and 1 cycle for the nonlinear transformation, thus N_1=5.

A way to generate the voltages representing the breakpoints is shown in Figure 3-a [8]. Since current flowing out of the circuit to the comparator is neglectible, an almost ideal operation can be assumed. The form ratios of the different transistors will fix the comparison voltages. Since the output voltages are decreasingly ordered from top to bottom, a switching scheme successively addressing the different output voltages in increasing order is used. Typically, a claimed drawback for the circuit arrangement in Figure 3-a is its dependence on the power supply; however, since in our case the discourse universe is fixed by the bias voltages, this is not a problem any more.

Another solution to implement a membership function can be obtained from the former circuit just replacing the lower part

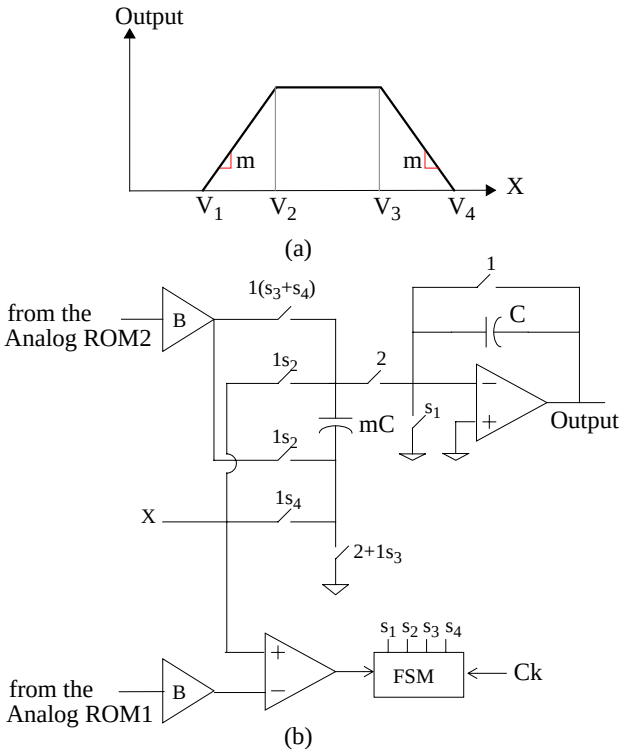


Fig. 2: (a) Symmetric trapezoid representing a membership function. (b) Circuit schematic of a MFC.

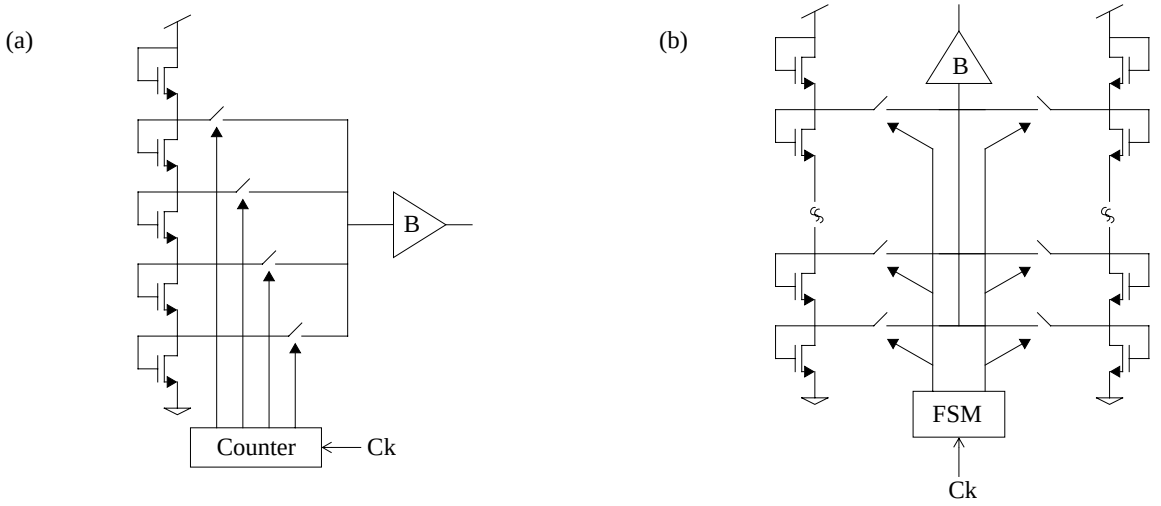


Fig. 3: (a) Analog ROM for the MFCs. (b) Analog ROM for the MFGs.

by four comparators. A simple combinatorial logic controls the switches so that the membership value is calculated in only one cycle and the ROM2 would not be needed. The four breakpoints are given in parallel by an analog ROM similar to that described, but the counter is not necessary now.

4. RULE EVALUATION CIRCUITRY

As was pointed out above, a good solution to avoid a large silicon area when many rules are considered, is achieved by the use of a serial MFG. Like MFCs, MFGs usually exhibit some symmetry, since they generate membership functions of fuzzy sets. Therefore, the number of membership values to be provided can be reduced so that the area occupation is still smaller.

The circuit in Figure 3-b has been used as the MFGs. The only difference with the circuit of Figure 3-a is due to the switching scheme. For the former a FSM must store the order in which the switches have to be closed. There is a practical limit to the number of transistors to be stacked up, roughly speaking this number is given by VDD/VT. In practice, it is preferable to derive the different voltage levels from several transistor poles. This is more flexible and avoids difficult trade-offs. Since the MFGs exhibit some symmetry, thus reducing the number of different voltage levels, the area occupation is not large.

Besides the MFG, MAX and MIN multi-input operators are required. A typical 4-input MIN gate is shown in Figure 4. In this figure, the four inputs are sequentially compared with the voltage previously stored by capacitor C. When the stored voltage is higher than an input voltage, the latter replaces the former since the corresponding switch is ON. Otherwise, the stored voltage does not change. Then, after four clock pulses we have the minimum of the four input signals. Either increasing the number of inputs or obtaining the MAX function is straightforward.

Taking into account its timing, the first level of MIN (in general MAX-MIN) operators will take a number of clock cycles that depends on the number of antecedents within every rule. Then, we must wait for a time equal to the slowest operation, which is equivalent to say a number of cycles equal to the highest number of antecedents in any rule. For the MFG operation, a two-input MIN is required, which means only a comparison and can be performed in just one cycle. Finally, for the last MAX stage, the time duration is equal to the number of rules times the clock cycle. Since these two-input MIN and multiple-input MAX operations have to be done for every element of the MFG:

$$N_2 = \max(A_j) + M\,(1 + R)$$

where A_j applies for the number of antecedents within the j-th rule, M is the number of elements in the MFG, and R is the number of rules.

5. DEFUZZIFIER

The final stage of the fuzzy controller is implemented by two summers followed by a SC divider, as is illustrated in the block diagram of Figure 5-a. The circuit in Figure 5-b shows a circuit realization with a reduced number of elements; every time a

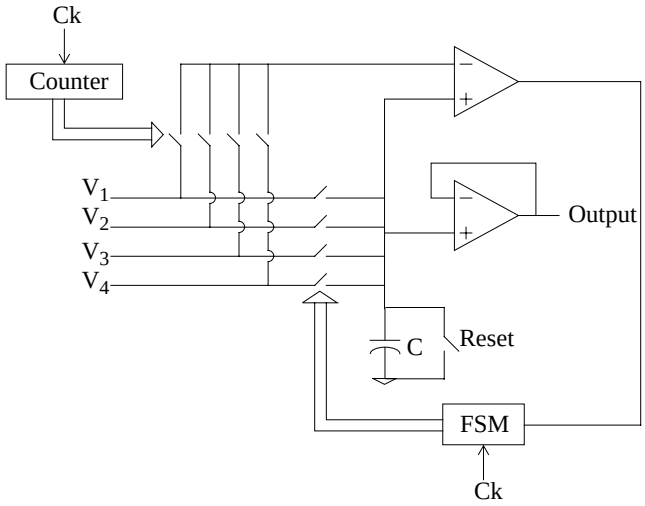
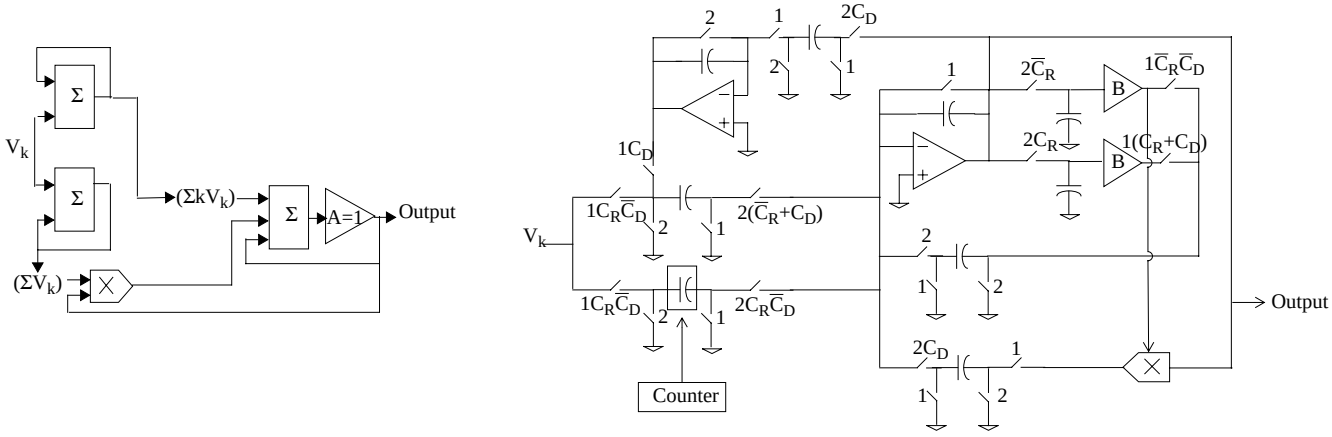


Fig. 4: Four-input MIN operator.



(a) (b)

Fig. 5: (a) Block diagram of a defuzzifier. (b) Circuit schematic.

MAX output (V_k) is validated, two partial sums (ΣV_k and ΣkV_k) are stored in capacitors C1 and C2, respectively. In these capacitors, we are incrementally adding the voltages until the last value of a MFG is generated and processed. Then, a discrete-time divider (as can be seen in Figure 5) provides the final system output.

The counter in Figure 5-b is devoted to controlling a capacitor array which gives the weight k for the sum ΣkV_k. This type of arrays are frequently used in SC circuits but we can eliminate it and save on area by using two cascaded summers as is shown in the block diagram of Figure 6-a. Figure 6-b illustrates the resulting circuit.

In any case, the time required for the summer operation is 2 clock cycles for every MFG value, but since it can be done while the previous stage is processing, this value only accounts for the last step. The divider requires D pulses to operate, which gives for N_3 a value of $N_3 = D+2$.

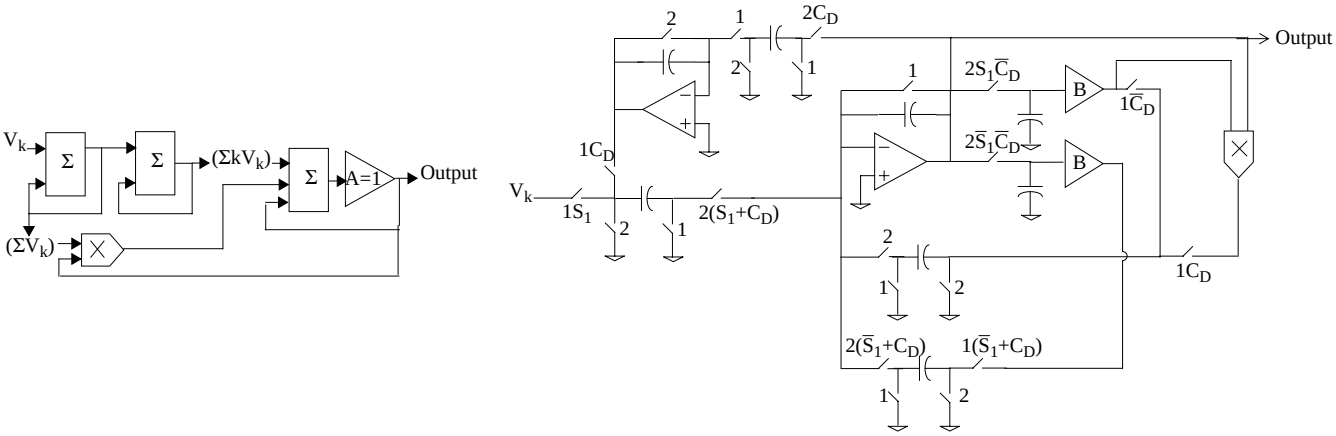
6. PIPELINING

Because of the way we are implementing the different blocks, once the last value of a MFG cycle has been produced, we can start a new operation cycle (overlapping phase 3 with phase 1 and phase 2). It means a reduction on the overall operation cycle of D+2, the operation cycle can be expressed as:

$$N = b + 1 + \max(A_j) + M\,(R+1),$$

where b is the number of breakpoints in the MFC (normally 3 or 4).

The main limitation in this architecture is due to the sequential operation of the MAX circuit, which introduces the factor MR in the expression above. This is a consequence of having used only one MAX block in the microcontroller in order to minimize its area. However, if q MAX blocks are used the pipelining can be increased, dividing the factor MR by q. Hence, depending on the



(a) (b)

Fig. 6: Alternative defuzzifier with cascaded summers: (a) Block diagram. (b) Circuit schematic.

applications, a compromise must be done between the number of MAX blocks (area) and the logic inference speed.

7. CONCLUDING REMARKS

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 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.

On the other hand, taking into account that SC circuits can operate at high frecuencies, the proposed microcontroller can be really competitive compared with digital implementations. In this sense, the main advantage of our system is its ability to deal with analog signals making possible the direct processing of membership values (what results in the elimination of A/D, D/A converters).

REFERENCES

[1] L. A. Zadeh. *"Fuzzy Sets"*, Inform. and Control, 1965.

[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 Programable Fuzzifier Integrated Circuit: Synthesis, Design and Fabrication*", Elsevier Science Pub. Comp. Inc., 1988.

[4] J. L. Huertas, A. Rodríguez-Vázquez, A. Rueda and L. O. Chua, "*Nonlinear Switched-Capacitors Networks: Basic Principles and Piecewise-Linear Design*", IEEE Trans. Circuits and Systems, V. CAS-32, pp 305-319, 1985.

[5] A. Rodríguez-Vázquez, J. L. Huertas, A. Rueda, B. Pérez-Verdú and L. O. Chua, "*Chaos in Switched-Capacitors: Discrete Maps"* Proc. of the IEEE Vol. 75, pp 1090-1107, 1987.

[6] T. Yamakawa, "*High-Speed Fuzzy Controller Hardware System*: *The Mega FIPS Machine*", Elsevier Science Pub. Comp. Inc., 1988.

[7] J. L. Huertas, S. Sánchez-Solano, A. Barriga, I. Baturone, *"Serial Architecture For Fuzzy Controllers: Hardware Implementation Using Analog/ Digital VLSI Techniques"*, 2nd. International Conference on Fuzzy Logic and Neural Networks, Iizuka, 1992.

[8] R. Gregorian and G. C. Temes, *"Analog MOS Integrated Circuits for Signal Processing"*, Wiley-Interscience Pub, 1986.