# Digital Implementation of SISC Fuzzy Controllers

C. J. Jiménez, A. Barriga, S. Sánchez Solano

Dept. of Analog Design. Centro Nacional de Microelectrónica
Avda. Reina Mercedes S/N. 41012-Sevilla, SPAIN

***Abstract.-- A classification of inference systems based on approximate reasoning techniques is proposed. An alternative realization method is described for the particular SISC case, which enables reducing the silicon area and increasing the operation speed, making it especially appropriate for real time control applications.***

## I. Introduction

Fuzzy logic provides a conceptual and mathematical frame for those problems where the imprecise definition of variables and vague resolution strategies applied require the use of approximate reasoning techniques. This has led to the development of new processing structures which allow hardware implementations of fuzzy inference mechanisms [1].

Distinct taxonomies can be established for fuzzy inference systems according to different criteria. However, in order to compare their hardware realizations, it is preferable to introduce a classification depending on the type of information they handle at the system input and on the description of the rules' consequents. In this sense, the realizations reported in literature are found in one of the three categories shown in Fig. 1.

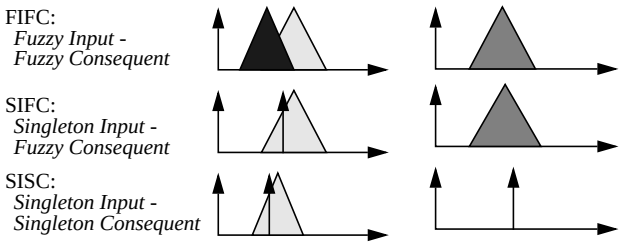


Fig. 1.- Taxonomy of fuzzy inference systems.

In conventional fuzzy inference systems the rule base is described by linguistic labels represented by fuzzy sets. In decision-making applications and complex control problems, input to the fuzzy inference system are described as a distribution of possibilities represented also by a fuzzy set (FIFC). On the contrary, in most control applications the input comes from sensors which give crisp values that can be associated to fuzzy singletons (SIFC). In this case, calculation of each rule's activation level is reduced to combining each antecedent membership degree by means of a connective operator. Finally, the controllers must give a concrete output value, which imposes introducing of a defuzzifying stage. To reduce the processing time of this stage, the designer usually recurs to the use of simplified methods which allow handling the predefuzzified information. From an implementation point of view, the most efficient of these approaches is that which substitutes the fuzzy information of the rules' consequents with a *fuzzy singleton* located in the point of maximum membership degree in the original fuzzy set (SISC).

## II. Hardware implementation of fuzzy systems

The first digital realizations of FIFC [2] and SIFC [3] systems use an architecture in which a data-path is supplied for each rule. The size and characteristics of the required memory elements to store the knowledge base are the main limitations of this realization strategy.

Calculation of the activation degrees in SIFC systems is a scalar operation, easily performed in a single clock cycle. This enables a new implementation strategy of fuzzy systems, where the different rules are processed sequentially. Its main drawback is the difficulty encountered to increase the number of control rules without degrading the system response time.

Different authors have proposed mechanisms to minimize both the calculation time and the area consumption of fuzzy inference systems. Among the most interesting contributions, the following deserve mention: 1) introduction of the active rule concept [4] and the use of singleton consequents [5] [6], which reduce the inference time; and 2) the *a priori* limitation of the degree of overlapping of the membership functions [7] [8], which considerably reduces the system size.

Adequately combining these concepts leads to the architecture of the SIFC fuzzy controller proposed in the next section, which generates efficient microelectronic realization.

## III. Active rule realization of SISC systems

In a SISC inference system, each rule proposes a specific conclusion ($B^r$), whose "strength" is determined by its corresponding activation level ($\alpha^r$). Calculation of the combined action of many rules requires a mechanism to evaluate the average of the different conclusions. This is made by taking the activation levels as weights:

$$\bar{y} = \sum_r \alpha^r \cdot B^r / \sum_r \alpha^r \qquad (1)$$

The use of fuzzy singletons to describe the conse-

quents of the rules makes the minimum number of cycles needed to evaluate (1) comparable with those needed for the division. For this reason, the inference should be driven by active rules, for which a scheme of associative memory is sought where only the active rules lead to the conclusion.

The architecture that we propose is shown in Fig. 2. The membership function circuits provide for each input value as many pairs (label, activation level), as degree of overlapping has been fixed in the system. Since a fixed degree of overlapping infers restricting the maximum number of active rules, the next step is to sequentially process each of these rules; a counter-controller multiplexer array is used for this. In each counter cycle, the membership degrees are combined through the MIN operator to calculate the activation level of the rule, while the antecedent labels address the memory position which contains their corresponding consequent. Finally, the arithmetic unit (defuzzifier in the figure) performs the multiplication, sum, and division operations of equation (1).

The active rules processing mechanism allows increasing the inference speed of the system, since rather than go through the whole rule memory, it only considers those rules which contribute to the final solution. The number of active rules depends on the overlapping degree of input membership functions and will be much less than that of the possible rules. Introducing *pipeline* stages, the number of required clock cycles to produce a control output is limited by the maximum between the number of active rules and the number of bits of the denominator.

Increasing the number of controller input increases the number of potentially active rules. However, all potentially active rules are not usually truly active. To eliminate these inactive rules, a counter with inhibiting input can be used that selects only those combinations of antecedents whose membership degrees are not zero. This allows reducing the number of clock cycles needed to evaluate the rules and improves the inference speed when the number of rules is the limiting factor of the operation speed.

## IV. Practical results

Based on the herein described architecture, a fuzzy controller prototype has been designed with three input and an output which obtains 8 bit resolution and uses 8 different membership functions with an overlapping degree of 2 for each input. With these characteristics, the controller can perform an inference every 9 clock cycles which, considering the frequency of a typical clock as 20 MHz, infers an operation speed over 2 MFLIPS.

## References


[1] Chuen Chien Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller" Parts I and II, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20 N. 2 (1990) pp. 404-432.

[2] M. Togai and S. Chiu, "A Fuzzy Logic Chip and Fuzzy Inference Accelerator for Real-Time Approximate Reasoning", *Proc. of 17th. International Symposium of Multiple Valued Logic*, May 1987, pp. 25-29.

[3] H. Watanabe, W. Dettloff and K. Young, "A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture", *IEEE J. Solid State Circuits*, Vol. 25 No 2 Apr 1990 pp 376-382.

[4] H. Ikeda, N. Kisu, Y. Hiramoto and S. Nakamura, "A Fuzzy Inference Coprocessor Using a Flexible Active-Rule-Driven Architecture", *Proc. IEEE ICFS'92*, San Diego, pp. 537-544.

[5] M. Sasaki, F. Ueno and T. Inoue, "An 8-bit Resolution 140 kFLIPS Fuzzy Microprocessor", *Proc. fifth IFSA World Congress 1993*, Seoul, pp. 921-924.

[6] T. Katashiro, "A Fuzzy Microprocessor for Real-time Control Applications", *Proc. fifth IFSA World Congress 1993*, Seul, pp. 1394-1397.

[7] Tzi-cker Chiueh, "Optimization of Fuzzy Logic Inference Architecture", *IEEE Computer*, Vol. 25 N. 5 (1992), pp. 67-71.

[8] Herbert Eichfeld, Michael Löhner and Michael Müller, "Architecture of a CMOS Fuzzy Logic Controller with optimized memory organization and operator design" *Proc. IEEE ICFS'92*, San Diego, pp 1317-1323.


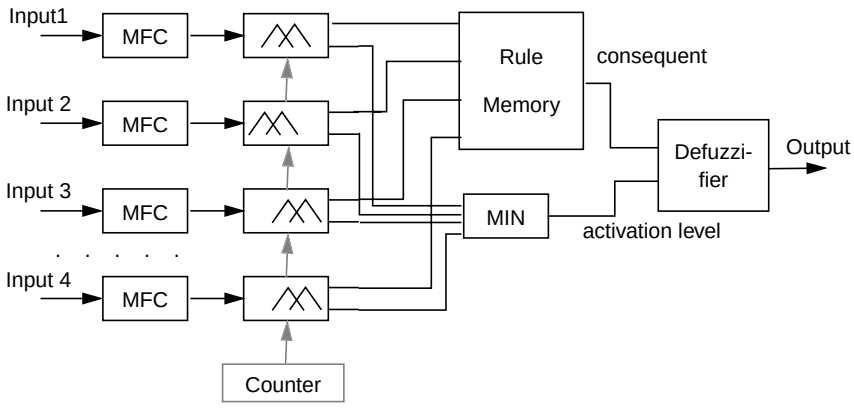


Fig. 2.- Block diagram of the proposed architecture.