MIXED-SIGNAL DESIGN OF A FULLY PARALLEL FUZZY PROCESSOR

I. Baturone, A. Barriga, S. Sánchez-Solano and J. L. Huertas.
Instituto de Microelectrónica de Sevilla (IMSE-CNM).
Edificio CICA, Avda Reina Mercedes s/n, 41012 Sevilla, Spain.

Indexing terms: Fuzzy processors, parallel architecture, mixed-signal current-mode design.

ABSTRACT

This paper presents a novel architecture to implement general-purpose fuzzy chips. It allows fully-parallel rule processing employing a reduced number of mixed-signal computing blocks and minimum-sized digital memories. The resulting fuzzy processor can interact directly with continuous sensors and actuators and subsequent digital processing system.

INTRODUCTION

A typical multiple-input single-output fuzzy system contains a set of IF-THEN rules like the following:

rule r: IF x_1 is A_1^r and ... and x_u is A_u^r THEN y is B^r

where x_i (i=1,..., u) are the inputs and y is the output. The antecedents' fuzzy sets, A_i^r (r=1, ..., R), partition each input space into L local fuzzy regions while the consequent B^r describes the behaviour within the joined region (A_1^r, ..., A_u^r). Hence, the total number of possible rules is $R=L^u$.

A fuzzy system is inherently parallel in the u input variables, R rules, and N elements in which the output space is discretized. Regarding hardware, this means a trade-off between high inference speed (parallel processing) and low silicon area (sequential processing). Singleton fuzzy systems that employ singleton values, c_r, to define the consequents, B^r, are usually chosen for hardware realization since they eliminate parallelism in N [1-3]. In the literature, fully-digital approaches have been reported reducing parallelism in R ($=L^u$) by only processing the α^u simultaneously active rules (where α is the overlapping degree of the input membership functions) [1, 4]. To allow parallel processing of the rules, the proposal in [4] is to employ α^u copies of the rule memory and to use multibit computing operators, which are very area consuming. Combining analogue and digital circuitry seems more interesting since digital circuitry eases programmability of the fuzzy processor and compatibility with subsequent digital systems while analogue circuitry offers parallel computing with lower hardware resources. However, digitally-programmable analogue realizations previously reported implement a small number of rules and they do not optimise digital part because the programmable parameters are stored in digital registers and selected by extensive matrixes of switches (multi-port-like digital memories), which occupy a large area [2-3].

The architecture presented in this letter allows optimization of both the analogue and digital part of a fully-parallel fuzzy processor. The analogue core is optimised by using an active-rule driven scheme implemented with current-mode computing blocks. The digital part is also optimised by using an adequate memory organisation that makes possible to retrieve all the required parameters in parallel without a need for replication or multi-port costly memories.

PROPOSED DESIGN

Singleton fuzzy systems carry out the following formula: $y = \Sigma_r h_r \bullet c_r / \Sigma_r c_r$, where h_r is the activation degree of the *r*-th rule [1-3]. The architecture we propose to implement them is illustrated in Figure 1 for the case of two inputs, u=2, and a maximum overlapping degree of two, α=2. The parameters that define the antecedents and consequents are stored in conventional RAMs (X_i-Mem and Y-Mem sets) so that the fuzzy processor can be suitably programmed for a given application.

The membership degrees, I_μ, of each input variable are obtained by the transfer functions of α circuits known as MFCs. The MFCs described in [5] (Figure 2), which are based on digitally-programmable current mirrors (D/A), have been selected. They admit analogue input signals and provide trapezoidal functions defined by 4 digital words. Hence, the size of the global memory, X-Mem, that stores the membership functions' parameters of each input is 4•L words. Each X-Mem memory is divided into α parts (conventional RAMs) which store the parameters of the membership functions that are never active simultaneously. This is illustrated in Figure 3 for the simple case of L=4 (in this example, the 4•2 words associated with the membership functions NB and PS are stored in the M_{11} part of the X_1-Mem while the other 4•2 words are stored in the M_{12} part). For a given input x_i, one set of parameters (4 words) of each memory part is addressed by a code of n bits, $\{b_{1xi}, ..., b_{nxi}\}$, n being the integer bigger or equal to $\log_2(L+1-\alpha)$, where L+1-α are the possible combinations of active input fuzzy sets. Hence, calculation of the membership degrees is performed in parallel since the α sets of required parameters per input are retrieved with one access to the X-Mem global memory. Each code $\{b_{1xi}, ..., b_{nxi}\}$ is obtained by comparing the input x_i with the centres of the membership functions that cover the *i*-th input space, as illustrated in Figure 3. This comparison can be done in parallel by using L-α current comparators and a maximum of L-2 programmable current mirrors. Another solution is to opt for a binary-tree comparison scheme. In this case, shown in Figure 1, the operation takes more time (n clock phases governed by the signals $\{R_1, ..., R_n\}$) but no additional current comparators or programmable mirrors are required by exploiting the input stage of the MFCs (shown within a dashed box in Figure 2).

The *i*-th MUX block after the MFCs implements current replications and identifies which MFC output goes to each MIN by using the least significant bits of the set $\{b_{1xi}, ..., b_{nxi}\}$. Computation of the rules' activation degrees is performed in parallel by the α^u multi-input analogue MIN circuits whose structure is described in [5]. The MUX blocks after the MIN circuits identify the corresponding CONS by using the least significant bits of the whole set $\{b_{1x1}, ..., b_{nx1}, ..., b_{1xu}, ..., b_{nxu}\}$.

The CONS blocks are digitally programmable current-mirrors (Figure 2) that weight each rule's activation degree by its corresponding singleton value. The L^u digital words that define all the singleton values are stored in the Y-Mem global memory. This memory is divided into α^u parts (conventional RAMs) where each part stores the consequents' values that are never active simultaneously. For the case illustrated in Figure 3 (α^u =4), each of the 4 parts stores 4 digital words (M_1, for instance, stores c_1, c_3, c_9, and c_{11}). Given an input {x_1, ..., x_u}, one word of each memory part is addressed by the whole code {b_{1x1}, ..., b_{nx1}, ..., b_{1xu}, ..., b_{nxu}} so that the α^u required consequents are retrieved in just one access to the Y-Mem memory. The sums $\Sigma_r h_r \bullet c_r$ and $\Sigma_r h_r$ are simply implemented by wired connection as we are working with current signals. Hence, the whole processing of all the active rules is carried out in parallel.

The block DIV implements the final division $\Sigma_r h_r \bullet c_r$ / $\Sigma_r h_r$ with a successive-approximation technique so that the output is provided in both digital and analogue formats (Figure 2). Division can be performed in parallel by using a flash A/D converter at the cost of silicon area. A good trade-off speed/area is achieved by a divider based on continuous-time algorithmic data converters, like that described in [3]. In this case, the time invested in division increases with output resolution.

From previous designs integrated in 2.4-µm CMOS process [3, 5], we can estimate the following features for a typical two-input fuzzy processor with α=2 implemented with the proposed architecture: Its analogue core occupies a silicon area of about 1mm^2 (considering 8- and 4-bit words to program the antecedents and consequents, respectively, and 5-bit resolution for the output) and it consumes less than about 20mW for a 5-V power supply. Its response time is less than about 2µs. These features slightly change when increasing the total number of rules (for instance implementing 16, if L=4, or 81 rules, if L=9). The area and power consumption of the digital part is also optimised since the number of words stored is the minimum to define the system.

CONCLUSIONS

A novel architecture to implement fuzzy processors has been presented. Area and power consumption is very small because parallel computing is performed in current-mode analogue domain using an active-rule driven scheme. An adequate organisation of the digitally programmable parameters makes possible to retrieve them in parallel from conventional RAM memories. Hence, processing of many rules can be achieved at high inference speed and with very low hardware resources.

REFERENCES

[1] SASAKI, M., UENO, F., and INOUE, T.: “An 8-bit resolution 140 KFLIPS fuzzy microprocessor”, Proc. Fifth IFSA World Congress, Seul 1993, pp. 921-924.

[2] MANARESI, N., FRANCHI, E., GUERRIERI, R., and BACCARANI G.: “A field programmable analog fuzzy processor with enhanced temperature performance”, Proc. ESSCIRC’96, Neuchatel 1996, pp. 152-155.

[3] BATURONE, I., SÁNCHEZ-SOLANO, S., BARRIGA, A., and HUERTAS, J. L.: "Flexible fuzzy controllers using mixed-signal current-mode techniques", Proc. FUZZ-IEEE’97, Barcelona 1997, pp. 875-880.

[4] CHIUEH, T.-C.: “Optimization of fuzzy logic inference architecture”, IEEE Computer, 1992, 24, (5), pp. 67-71.

[5] BATURONE, I., SÁNCHEZ-SOLANO, S., BARRIGA, A., and HUERTAS, J. L.: “Implementation of CMOS fuzzy controllers as mixed-signal IC’s”, IEEE Trans. on Fuzzy Systems, 1997, 5, (1), pp. 1-19.

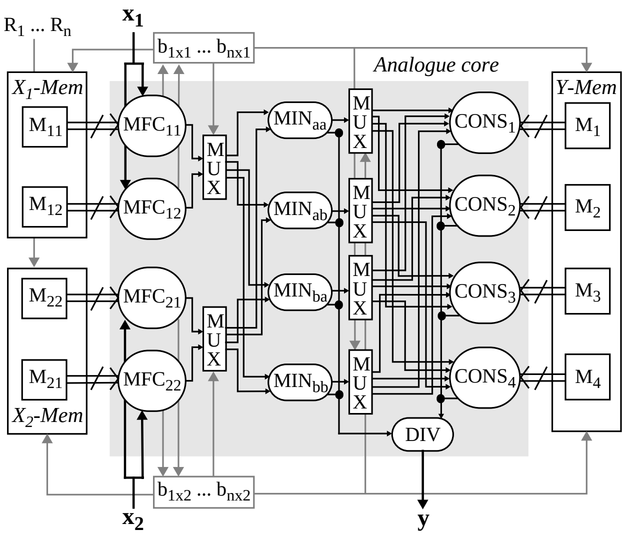


Figure 1: Proposed architecture for the case of two input variables and a maximum overlapping degree of two.

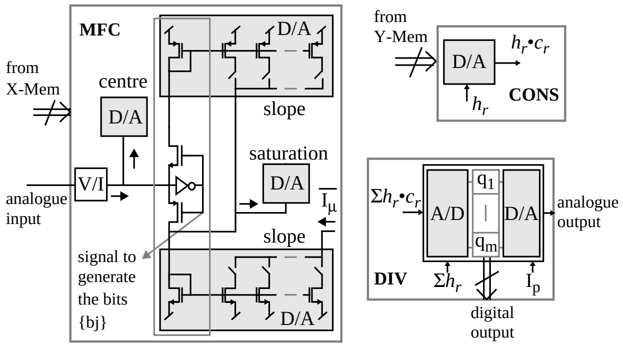
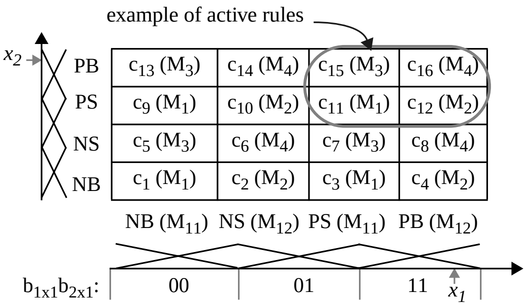


Figure 2: Schematics of the computing blocks [3, 5].



	NB (M_{11})	NS (M_{12})	PS (M_{11})	PB (M_{12})
PB	c_{13} (M_3)	c_{14} (M_4)	c_{15} (M_3)	c_{16} (M_4)
PS	c_9 (M_1)	c_{10} (M_2)	c_{11} (M_1)	c_{12} (M_2)
NS	c_5 (M_3)	c_6 (M_4)	c_7 (M_3)	c_8 (M_4)
NB	c_1 (M_1)	c_2 (M_2)	c_3 (M_1)	c_4 (M_2)

Figure 3: Rule table that illustrates the partitions of the antecedents' and consequents' memories.