

A Modular Programmable CMOS Analog Fuzzy Controller Chip

Angel Rodríguez-Vázquez, *Fellow, IEEE*, Rafael Navas, Manuel Delgado-Restituto, *Member, IEEE*, and Fernando Vidal-Verdú

Abstract—We present a highly modular fuzzy inference analog CMOS chip architecture with on-chip digital programmability. This chip consists of the interconnection of parameterized instances of two different kind of blocks, namely *label blocks* and *rule blocks*. The architecture realizes a lattice partition of the universe of discourse, which at the hardware level means that the fuzzy labels associated to every input (realized by the label blocks) are shared among the rule blocks. This reduces the area and power consumption and is the key point for chip modularity. The proposed architecture is demonstrated through a 16-rule two-input CMOS 1- μm prototype which features an operation speed of 2.5 Mflips (2.5×10^6 fuzzy inferences per second) with 8.6 mW power consumption. Core area occupation of this prototype is of only 1.6 mm² including the digital control and memory circuitry used for programmability. Because of the architecture modularity the number of inputs and rules can be increased with any hardly design effort.

Index Terms—Analog IC design, function approximation, fuzzy hardware.

I. INTRODUCTION

FUZZY controllers are used to map a *multidimensional* input signal $\mathbf{x} = \{x_1, x_2, \dots, x_M\}^T$ onto a scalar output y , in accordance to a well-defined *nonlinear* relationship [1],

$$y = f(\mathbf{x}). \quad (1)$$

In control applications the inputs are usually called *facts*, the output *action*, and the mapping law *surface response*. For instance, a fuzzy controller for a washing machine must univocally set the water level (action) as a nonlinear function (surface response) of the clothes' mass, the water impurity, and the time differential of impurity (facts) [2].

Fuzzy controllers employ the procedure of *fuzzy logic inference* [1] to construct the surface response. Some characteristic features of this procedure are as follows [3], [4].

- The surface response, which is a *global* model predicting the system behavior for any input, is obtained as a composition of *local* functions, each one predicting this behavior only for inputs comprised in a limited region of the input space.

Manuscript received July 31, 1997; revised April 30, 1998. This work was supported in part by the Spanish C.I.C.Y.T under Contract TIC96-1392-C02-02 (SIVA).

A. Rodríguez-Vázquez and M. Delgado-Restituto are with the Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica-C.S.I.C., Edificio CICA-CNM, Avda. Reina Mercedes s/n, 41012-Sevilla, Spain.

R. Navas and F. Vidal-Verdú are with Dto. de Electrónica, Universidad de Málaga, Complejo Tecnológico, Campus de Teatinos, Málaga, Spain.

Publisher Item Identifier S 1057-7130(99)01776-0.

- These local functions represent *insights* on the system operation, and are described through inference *rules* of the type

IF x_1 is A_{1k} AND x_2 is A_{2k} AND \dots x_M is A_{Mk}
THEN *Consequent Action*

where A_{ik} are called *fuzzy labels*, and the consequent action assigns values to y depending on the outcome of the combination of the antecedent clause statements.

- The validity of the statements “IF x_i is A_{ik} ” is continuously graded from 0 to 1; the actual grade of each statement is calculated by evaluating a nonlinear *membership function* $s_{ik}(x_i)$ which is different from zero only inside a subinterval of the whole x_i interval.

Because the statements involved in the fuzzy rules are in natural language, for instance “if the temperature is low,” this modeling technique is very well suited to capture and emulate human expertise. On the other hand, the continuous grading guarantees *generalization* of the local pieces of knowledge and hence, smooth surface responses. Finally, any change which affects only a limited region of the input space can be easily incorporated to the global model by just modifying the affected local functions—*transparency* property [3].

There are many fuzzy controller applications where the inputs and the output are *analog* signals [1], [2]. The hardware required for these applications can be realized in two alternative ways. One employs analog circuitry only at A/D and D/A conversion interfaces, while the fuzzy processing is realized in digital domain by either general-purpose processors or dedicated ASIC's [5]–[8]. The other realizes the fuzzy processing itself in the analog domain, and employs the digital circuitry for *programmability* and reconfigurability [9]. This paper contributes to the latter approach. Generally speaking, this approach is expected to feature larger operation speed, lower power consumption and smaller area occupation than the other [10], [11]. These expectations are confirmed by the techniques presented in this paper, which fully exploit the functional capabilities of the MOS transistor (MOST) to realize the fuzzy operators with very simple circuitry. An inherent disadvantage of analog fuzzy controllers is limited precision. However, it can be overcome through proper modeling of the error sources and the use of sound circuit design techniques [10]–[12].

Circuit blocks and design techniques for CMOS analog fuzzy controllers have been reported elsewhere [13]–[17].

Some of them have been demonstrated through actual *monolithic* circuits, a fraction of which include programmability [15]–[17]. However, although precision is a weak point of the analog approach, most previous contributions do not consider the accuracy issue during the design phase. Their output signal may hence become largely erroneous. These errors can be attenuated by post-fabrication tuning of some critical parameters, guided by learning processes [18]. But the errors must still remain bounded for convergence. Our chip architecture includes design equations to guarantee accurate operation within prescribed error margins. Consequently, it can be programmed in robust and transparent way.

The demonstration chip in this paper implements more rules than previous analog monolithic controllers and features much smaller *Delay* \times *Power* values, namely: 470 ns \times 8.6 mW (with 16 rules) versus 570 ns \times 44 mW (with 9 rules) [16], and 160 ns \times 550 mW (with 13 rules) [17]. Programmability is also a quality of our chip, which incorporates on-chip memories for serial digital programming of the rule consequents, and allows external analog programming of the membership functions. This is advantageous as compared to [16], where the consequents values are learned using software models of the controller and are stored on-chip with no further change possible. Finally, the modular organization around two high level building blocks easily identified from the user and designer point of view, renders our chip architecture feasible for silicon compilation.

II. CHIP ARCHITECTURE

The chip realizes a type of fuzzy inference where the rule consequents are constant values

$$\begin{aligned} \text{IF } (x_1 \text{ is } A_{1k}) \quad \text{AND } (x_2 \text{ is } A_{2k}) \quad \text{AND } \cdots (x_M \text{ is } A_{Mk}) \\ \text{THEN } y = y_k^*, \quad 1 \leq k \leq N \end{aligned} \quad (2)$$

These values y_k^* are called *singletons*. As compared to the general case where the consequents include fuzzy labels [1], this type of fuzzy inference requires much less complex hardware [9], and, thus, less silicon area and less electrical power. Besides, it increases the transparency of the rules and, thus, eases the incorporation of programmability. On the other hand, different studies show that singleton fuzzy controllers are *universal approximators*, i.e., they are capable to approximate any surface response by properly choosing the rules and singletons [3], [4].

The set of membership functions $s_{ik}(x_i)$ constitutes the elementary nonlinearities from which the surface response of a fuzzy controller is built. Fig. 1(a) shows a typical membership function shape [4]—described by three parameters: *width* (2Δ), measured as the length of the interval defined by the *crossover points*; *center* (E_C), the central point of this interval; and *slope* (ζ), the absolute value of the function slope at the crossover points.

For a complete controller description, the surface response formula has to be generated from these elementary nonlinearities. Fig. 1(b) illustrates the building procedure for a one-dimension, four-rules controller. Here, each rule involves only a fuzzy label, “IF x is A_k THEN $y = y_k^*$,” whose validity

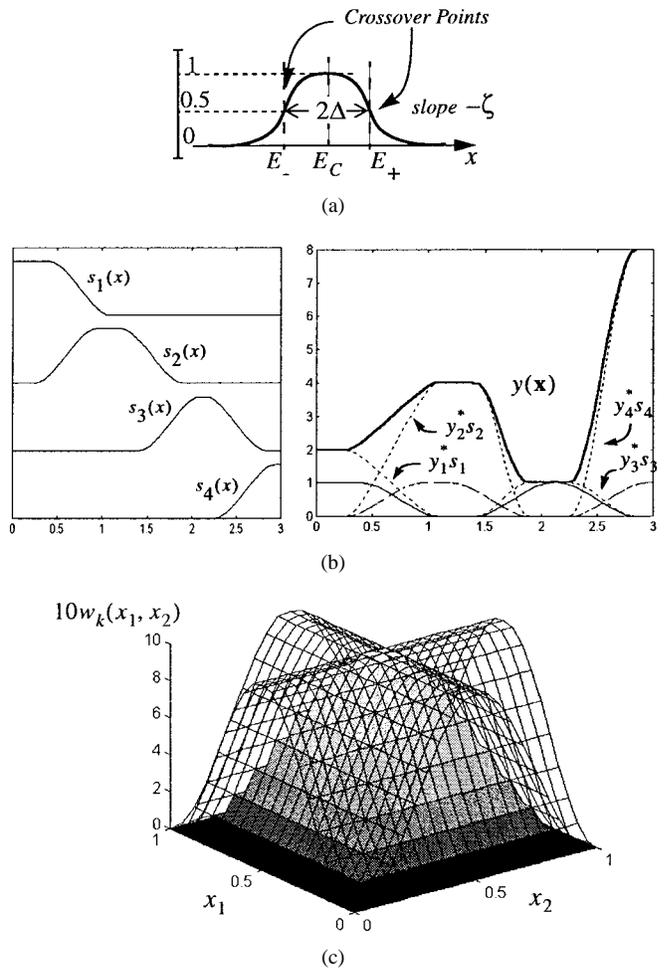


Fig. 1. (a) One-dimensional membership function shape; (b) illustrating function approximation through singleton fuzzy controllers; and (c) two-dimensional membership function.

is evaluated by using the corresponding membership function $s_k(x)$. If the actual input x is at the center of the interval I_k for the k th membership function, then $s_k(x) = 1$ and the output is given by the value of the k th singleton $y = y_k^*$. At any point different from the centers of the membership function intervals, the output does not coincide with any of the singletons but it is interpolated by using the following formula:

$$y = y_1^* s_1^*(x) + y_2^* s_2^*(x) + y_3^* s_3^*(x) + y_4^* s_4^*(x) \quad (3)$$

where $s_k^* = s_k / [\sum_{k=1,4} s_k]$.¹ In this way a global response curve is built from the local data represented by the singletons, as Fig. 1(b) illustrates.

In the general multidimensional case, the surface response is interpolated from the singletons by using multidimensional membership functions $w_k(\mathbf{x})$

$$y = f(\mathbf{x}) = \sum_{k=1, N} y_k^* \frac{w_k(\mathbf{x})}{\sum_{k=1, N} w_k(\mathbf{x})} \quad (4)$$

¹This normalization precludes the output to take a value larger than the largest singleton at any point.

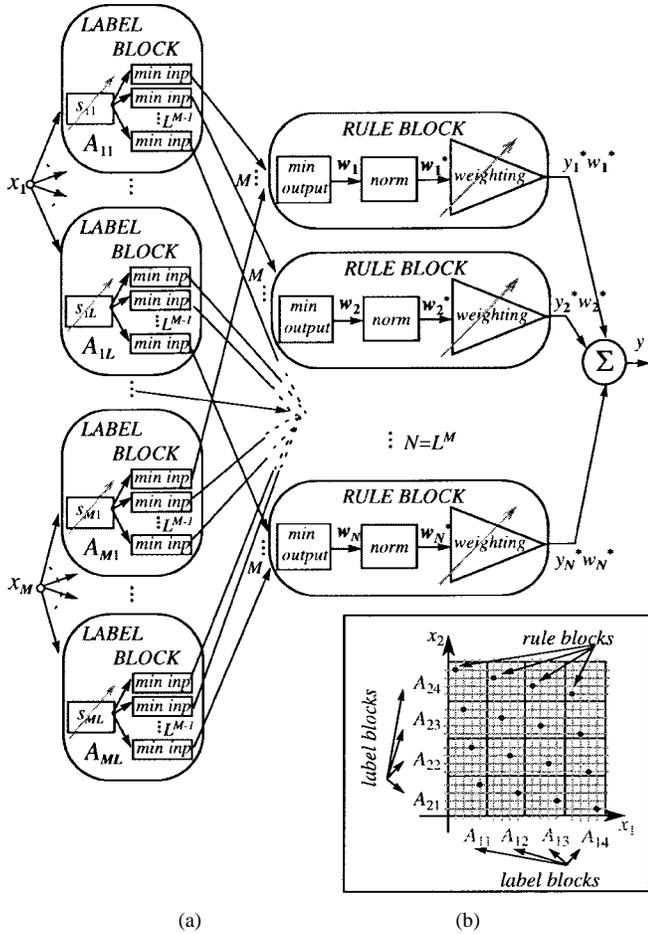


Fig. 2. (a) Controller chip architecture and (b) interconnection of label and rule blocks in the 1 μm CMOS prototype.

where the function $w_k(\mathbf{x})$ is evaluated by choosing the *minimum*² among the values of the unidimensional membership functions $s_{ik}(x_i)$ associated to the k th rule

$$w_k(\mathbf{x}) = \min\{s_{1k}(x_1), s_{2k}(x_2), \dots, s_{Mk}(x_M)\}. \quad (5)$$

Fig. 1(c) illustrates the build-up procedure and final shape of a two-dimensional membership function.

The fuzzy controller chip architecture of Fig. 2 realizes (4) for a system with M inputs, L fuzzy labels per input and $N = L^M$ rules. The architecture is composed of the interconnection of blocks of two different types, namely: *label* and *rule*. Each fuzzy label, say A_{ij} (the j th fuzzy label of the i th input), has an associated label block which evaluates the corresponding membership function $s_{ij}(x_i)$ and generates L^{M-1} replicas of the result. These replicas are processed in the “*min inp*” sub-blocks of the label blocks to make a first step toward the realization of the minimum. Each rule blocks combines M inputs coming from the label blocks to: first, realize the second step of the minimum operation; second, evaluate the function $w_k^*(\mathbf{x})$; and, third, multiply this function by its associated singleton to obtain $y_k^* w_k^*(\mathbf{x})$. The final aggregation leading to (4) is performed at the output node.

²This is the AND operator used in our chip. Other operators could be used as well [1]–[4].

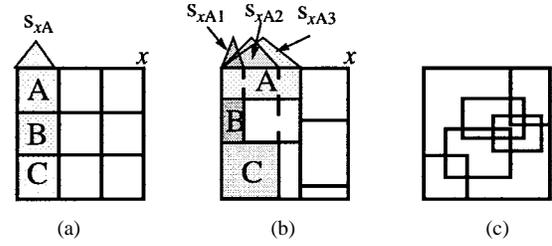


Fig. 3. Examples of different types of input space partitions.

Fig. 2(b) illustrates the interconnection of label and rule blocks for a system with two inputs and four fuzzy labels per input, as it is the case for the CMOS prototype presented in Section VI of this paper. Each box in the grid corresponds to a rule, has an associated singleton value, and is defined by two labels, one per input. Because of this membership function sharing, the architecture of Fig. 2(a) can only generate *lattice* partitions [see Fig. 3(a)]; *tree* [Fig. 3(b)] and *scatter* [Fig. 3(c)] partitions [4] are not allowed.

Generally speaking lattice partitions have the potential disadvantages of *course of dimensionality* (the number of rules needed to perform a good approximation may become prohibitively large for large number of inputs) and *inappropriate generalization* (the partition granularity needed to approximate the function in a region of the input space may be inappropriate in other region). However, these potential disadvantages are not really significant for the type of problems which analog fuzzy controllers are intended for (medium-to-low complexity problems with low number of inputs and low number of rules). In this scenario, the architecture of Fig. 2 features significant pros for hardware implementation, namely:

- Area and power consumption required for the implementation of the rules antecedents are smaller than in the case of scatter and/or tree partitions. This is because the replication operation is much less area- and power-demanding than the membership function evaluation itself.
- The whole architecture is highly modular and can be made to grow in very simple manner. Consequently, it is very well suited for design automation.³
- Programmability can be easily incorporated.

Inputs to the chip are voltages for easier interfacing. On the other hand, the minimum and the normalization operations are realized in current domain because this requires much simpler circuitry that their voltage domain counterparts [19]. Thus, the inputs to the membership function circuits are voltages, while their outputs are currents. However, as already mentioned, the label blocks do not directly deliver the membership function currents to the rule blocks; these currents are nonlinearly preprocessed to produce intermediate output voltages. This simplifies the realization of the minimum operation in the rule block. Besides, transmitting these voltages (instead of the original currents) from the label to the rule blocks largely sim-

³Highly complex controllers with prohibitive aggregated errors may result however for large input and label count, due to the course of dimensionality of lattice partitions [3].

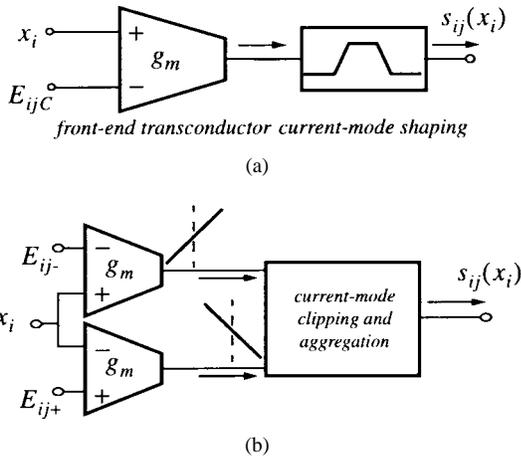


Fig. 4. Concepts for the realization of a transconductance membership function by current shaping: (a) global shapping in current-mode [20] and (b) partial shapping in current-mode [23].

plifies the interblock routing as these latter blocks have only one input node (instead of M if currents were transmitted).

III. LABEL BLOCK

Each label block is driven by a component x_i of the input voltage vector to, first, obtain a membership function current $s_{ij}(x_i)$ and, second, generate L^{M-1} replicas of a voltage V_{Gij} which is a nonlinear function of this current—a preprocessing step for the realization of the minimum operator in the rule blocks. This section describes first the membership function circuitry, then the complete minimum circuitry and, finally, outlines some major design considerations to reduce systematic errors in these circuits.

A. Membership Function Circuitry

A few alternative realizations of the pseudo-trapezoidal function shape of Fig. 1(a) have been reported in literature [15], [21]–[23]. One, see Fig. 4(a), consists of a cascade of a linearized transconductor, to convert the input voltage into a current, and a current-mode nonlinear block to realize the pseudo-trapezoidal shape [20]; this latter block can be realized by using the techniques proposed in [15], [21], [22]. A drawback of this implementation is the extra area occupation and power consumption of the linearization circuitry. Also, because the transconductor cannot be linearized in the whole input range, some of this range is wasted.

Fig. 4(b) employs a slightly different strategy [23]. It uses two quasi-linear transconductance amplifiers to, at a first step, obtain monotone increasing and decreasing, respectively, currents around the crossover points; then, at a second step, these currents are first clipped and then aggregated in current domain. This strategy shares the drawbacks associated to linearization. However, as compared to Fig. 4(a), it has the advantage that the centers and widths of the membership functions are controlled through voltages applied to high-input impedance nodes, which requires a simpler control circuitry and yields smaller loading errors in the application of the control signal.

The membership function circuit used in our chip (see the shaded region at the left in Fig. 5) approximates the shape of Fig. 1(a) by using the nonlinear dc characteristics of a CMOS differential pair. This strategy is based on the work by Fattaruso and Meyer on CMOS function approximation [24], and was proposed for analog fuzzy design in [25]. Analysis of this circuit assuming equal differential pairs and using the square-law MOS transistor characteristics [26] obtains

$$s_{ij} = \begin{cases} 0, & \tilde{x}_{ij-} < -E_Q \\ \left(1 + \frac{\sqrt{2}}{E_Q} \tilde{x}_{ij-}\right) I_Q \sqrt{1 - \frac{\tilde{x}_{ij-}^2}{2E_Q^2}}, & -E_Q < \tilde{x}_{ij-} < E_Q \\ 2I_Q, & \text{otherwise} \\ \left(1 - \frac{\sqrt{2}}{E_Q} \tilde{x}_{ij+}\right) I_Q \sqrt{1 - \frac{\tilde{x}_{ij+}^2}{2E_Q^2}}, & -E_Q < \tilde{x}_{ij+} < E_Q \\ 0, & \tilde{x}_{ij+} > E_Q \end{cases} \quad (6)$$

where $\tilde{x}_{ij-} = (x_i - E_{ij-})$, $\tilde{x}_{ij+} = (x_i - E_{ij+})$, $E_Q = \sqrt{I_Q/\beta_{N1}}$, β_{N1} is the large signal transconductance factor of the transistors in the differential pairs,⁴ and we assume that the membership function width is large enough to allow the output current reaching the logic unit value (I_Q) at the center.

This membership function circuit shares the advantages of Fig. 4(b) regarding control of the centers and widths through voltages applied to high input impedance terminals

$$2\Delta_{ij} = E_{ij+} - E_{ij-} \quad 2E_{ijc} = E_{ij+} + E_{ij-}. \quad (7)$$

On the other hand, the slope at the crossover points ζ_{ij} is controlled by the large signal transconductance of the MOS transistor⁵

$$\zeta_{ij} = \sqrt{2I_Q\beta_{N1}}. \quad (8)$$

The main advantage of this membership function circuit is that it does not require any linearization circuitry—why to linearize if the whole behavior is nonlinear? Thus, it features minimum area occupation and power consumption, and full usage of the transconductor input dynamic range. On the other hand, it has been shown that the shape in (6) can actually realize the universal approximation feature, even when parasitics (systematic, as well as random) are taken into account [18].

Considerations about the main nonidealities that influence the membership function circuitry, and the design strategies adopted to reduce their influence, are presented in subsequent sections. However, because they are influenced by the preprocessing circuitry used for the minimum operation, we will describe this circuitry first.

B. Minimum Circuitry

As mentioned in Section II, the minimum operation is realized in three steps: two in the label blocks and other in the

⁴We assume that the positive and negative input transistors are equal.

⁵Using the bias current to control the slope is not convenient because the bias current set the logical value “1.”

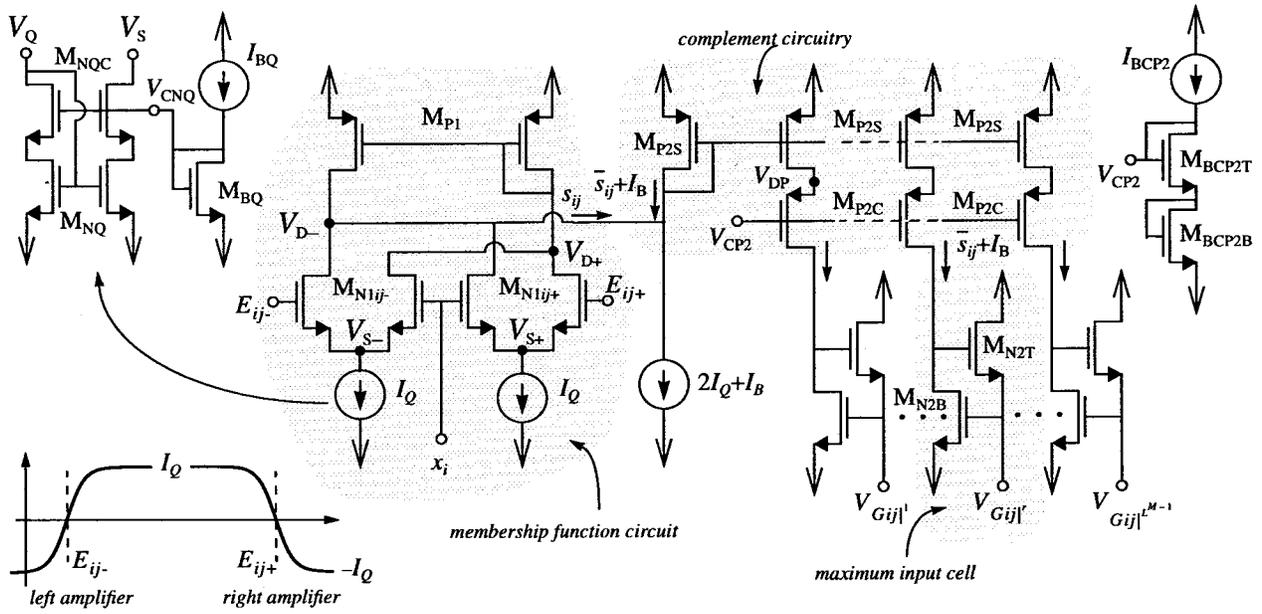


Fig. 5. The label block.

rule block. However, for clarity, these three steps are described as a whole in this section.

The whole operation of the minimum circuit is to select-and-propagate the minimum among a set of M input currents $s_{ik}(x_i)$. However, for convenience, we do not directly select the minimum among the input currents, but the maximum among their *fuzzy complements*

$$\overline{s_{ik}(x_i)} \equiv 2I_Q - s_{ik}(x_i) \quad (9)$$

where the current level $2I_Q$ corresponds to the logic “1.” This is based on the De Morgan’s law [39]

$$w_k(\mathbf{x}) = \frac{\min\{s_{1k}(x_1), s_{2k}(x_2), \dots, s_{Mk}(x_M)\}}{\max\{\overline{s_{1k}(x_1)}, \overline{s_{2k}(x_2)}, \dots, \overline{s_{Mk}(x_M)}\}} \quad (10)$$

and takes advantage of the larger simplicity of the current-mode maximum circuitry [27].

Fig. 6(b) shows conceptual circuits to evaluate the fuzzy complements by KCL, for positive (entering to) and negative (leaving from) currents. Regarding the maximum circuit itself, several alternatives appear which have to be evaluated bearing in mind the following major architectural features:

- Neither constraints nor penalties should be imposed to the number of inputs since it coincides with the number of controller inputs.
- The inter-block routing should be the smallest possible for increased modularity

These considerations lead us to discard realizations with $O(n^2)$ complexity [28]. Realizations based on sequential binary selection trees [29] are also discarded because, although they have $O(n)$ complexity, their implementation requires $\log_2(n)$ circuit layers, and causes the errors and delays to be accumulated proportionally to the number of inputs. The maximum circuit used in our chip [see Fig. 6(a)] is based on

the winner-take-all circuit by Lazzaro [30] and was proposed in [25]. Its steady-state circuit operation is simple: the bottom transistor driving the maximum current will force the common voltage V_G by means of its associated top transistor, while the remaining bottom transistors are driven into ohmic region to comply with their input currents and, consequently, their associated top transistors are cutoff. Then, provided the output transistor works in saturation region, its current coincides with the maximum one. When the maximum current is switched from one input terminal to another, a transient takes place where the difference between the new and the old maximum current is integrated in the latter terminal, thus driving this transistor into a conducting state and, eventually, changing the value of the common voltage V_G .

This circuit exhibits the architectural features mentioned above: 1) it has $O(n)$ complexity; 2) the different inputs share only the node V_G . This latter feature allows us to partition the circuit as Fig. 6(a) shows, so that the rule block has only one input. Another current-mode maximum circuit based also on Lazzaro’s was proposed in [31] and used in [14]. It connects the output transistor as a diode, removes the current source I_G , and connects the drains of the top transistors M_{N2Tk} to a common node which is the output node. Thus, the inputs share two nodes instead of one. Besides, the removal of the current I_G makes the resolution of this circuit dependent of the output current level and, specifically, small for large currents [27]. Finally, because the output node load increases with the input count, this circuit performs poorer than Fig. 6(a) when the number of inputs increases.

Let us now describe the realization of the two first steps for the minimum in the label block. The first (complementation) is realized by KCL at the input node of the right-top current mirror in Fig. 5. Its input current is $\overline{s_{ij}} + I_B$ where the current I_B is added to preclude the transistors entering in subthreshold, where the operation speed would become significantly degraded. Note, on the other hand, that this current mirror has

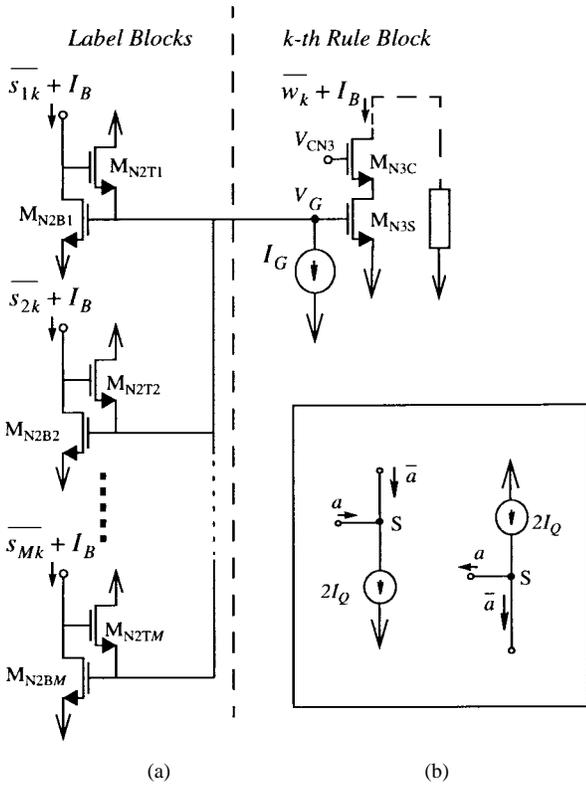


Fig. 6. Circuitry for the minimum computation: (a) maximum circuit and (b) complement implementation.

L^{M-1} output branches to generate the membership function output replicas s_{ij}^r ($1 \leq r \leq L^{M-1}$) for the different rules. The second step is realized also in the label block and consists of the generation of a set of intermediate voltages $V_{G_{ij}^r}$ as nonlinear functions of the currents $\overline{s_{ij}^r} + I_B$. Each of these voltages is generated in the right-bottom shaded area of Fig. 5 by a two transistor circuit (see also Fig. 6(a)); for proper operation of this two transistor circuit, some artifact must be added to discharge the node V_G —provided by the current source I_G included in the rule block.

The next step for the minimum operation is realized in the rule block (bear in mind, Fig. 2(a), that this block has one input and one output). To that purpose the set of voltages $V_{G_{ik}}$ for the M membership function values associated to the k th rule are routed and tied together at the input node of the rule block [see the left-hand part of Fig. 6(a)]. Thus, a collective computation is performed at this common node such that the maximum among the set of voltages prevails. From this maximum voltage the corresponding maximum current $\overline{s_{ik}(x_i)}|_{\max} + I_B$ is generated by the transistor M_{N3S} in Fig. 6(a). According to (10) this corresponds to the fuzzy complement $w_k(x)$ of the multidimensional membership value shifted by I_B .

C. Design Considerations in the Label Block

A thorough analysis of the static (systematic and random) and dynamic errors of Fig. 5, Fig. 6(a) and other label block circuits is found in [27]. This section summarizes some main results regarding systematic errors due to the finite output

resistance of the MOST's⁶ which are relevant to design purposes. Random errors are covered for the whole controller in Section V.

1) *Membership Function Circuit*: A first consideration refers to the common-mode input range of the differential amplifiers of Fig. 5. It is calculated by constraining the transistors to be ON and operate in saturation region

$$V_{C\Omega} + V_{TN1} + \sqrt{\frac{I_Q}{2\beta_{N1}}} \leq x_i \leq V_{DD} - \sqrt{\frac{2I_Q + I_B}{\beta_{P2S}}} + V_{TN1} - V_{TP2S} \quad (11)$$

where $V_{C\Omega}$ is the limiting voltage of the current I_Q , β_{N1} is the large signal transconductance of the input nMOST's, V_{TN1} is the corresponding threshold voltage, β_{P2S} is the transconductance of the top pMOST's, and V_{TP2S} is the corresponding threshold voltage. A strategy to improve the common-mode range is using bias current circuits with the smallest possible value of $V_{C\Omega}$, such as that attached to Fig. 5 where

$$V_{C\Omega} = \sqrt{\frac{I_Q}{\beta_{NQ}}} + \sqrt{\frac{I_Q}{\beta_{NQC}}} \quad (12)$$

Biasing of the current mirror that generates I_Q is then carried out by the circuit at the left of Fig. 5, where I_{BQ} is a reference current and the geometry of M_{BQ} is obtained from

$$S_{BQ} \equiv \frac{\beta_{BQ}}{\beta_{0N}^*} = \frac{I_{BQ}}{\beta_{0N}^*(V_{TNQC} + V_{C\Omega} - V_{T0N})^2} \quad (13)$$

where β_{0N}^* is the large-signal transconductance density of the nMOST and V_{T0N} is its zero-bias threshold voltage. Typical input range values are around 3.25 V by following this approach with a 1- μm CMOS standard technology and 5-V supply voltage.

Another error source is dc voltage mismatching between the drains of the input transistors (nodes V_{D+} and V_{D-} in Fig. 5) which might cause offset and distortion of the membership output current for finite MOST Early voltages. However, because these two nodes are both of low-impedance type, the voltage excursions are largely attenuated by the transconductance of the pMOST's and the error is, hence, negligible.

The last error is due to dc voltage mismatching between the input and output nodes of the pMOS current mirror driving the minimum input cell,

$$\varepsilon \approx 1 + \frac{V_{DP} - V_{D-}}{V_{AP}} \quad (14)$$

where V_{AP} is the equivalent Early voltage of the pMOST's. This error can be attenuated by proper setting of the bias voltage V_{CP2} of the cascode transistor M_{P2C} . For optimum attenuation, this voltage should be different for different input values. However, system-level considerations [27] show that it suffices to obtain the largest possible error attenuation at the

⁶They will be modeled through an equivalent Early voltage V_A which is a quasi-linear function of the channel length [26].

crossover points. The corresponding voltage is calculated to annul the following expression of the absolute current error:

$$\varepsilon^* \cong \left(\frac{3}{V_{AP}} + \frac{1}{V_{AN1}} \right) \frac{I_Q}{2\sqrt{\beta_P}} \left(\sqrt{I_Q + I_B} - \sqrt{\frac{3I_Q}{2}} \right) + \frac{V_R}{V_{AP}} (I_Q + I_B) \quad (15)$$

where

$$V_R = V_{DD} - 2V_{T0P} - \gamma_{P2C} \left[\sqrt{V_{DP} + \phi_B} - \sqrt{\phi_B} \right] - \sqrt{\frac{I_Q + I_B}{\beta_{P2C}}} - \sqrt{\frac{I_Q + I_B}{\beta_P}} - V_{CP2}$$

and we assume (as it happens in practice) that all pMOS signal transistors in Fig. 5 are equal ($\beta_{P1} = \beta_{P2S} \equiv \beta_P$, $V_{AP1} = V_{AP2S} \equiv V_{AP}$); V_{T0P} is the zero-bias threshold voltage of the pMOST and γ_P and ϕ_B are technological parameters [26] which account for the influence of the substrate on the threshold voltage. This optimum voltage can be generated by the circuit at the right in Fig. 5, where we assume that the two diode-connected transistors have the same aspect ratio,

$$S_{BCP2} \equiv \frac{\beta_{BCP2}}{\beta_{0N}^*} = \frac{4I_{BCP2}}{\beta_{0N}^* (V_{CP2} - V_{TBCP2T} - V_{TBCP2B})^2} \quad (16)$$

and I_{BCP2} is a reference current. This choice reduces the relative error in (14) to around 0.5%—negligible at the system level.

2) *Static and Dynamic Errors in the Maximum Circuit Operation*: Two major features related to the dc operation are the discrimination (the circuit ability to distinguish two close input values), and the error due to dc voltage mismatching between the input node sinking the maximum current and the drain of the output transistor. The discrimination of Fig. 6(a) is calculated as [27]

$$\frac{\Delta I}{I} \approx \frac{1}{V_{AN2B}} \sqrt{\frac{I_G}{\beta_{N2T}}} \quad (17)$$

where ΔI is the minimum current increment that can be detected by the circuit, and V_{AN2B} is the equivalent Early voltage of the bottom MOST. This equation shows that the discrimination improves for I_G decreasing, β_{N2T} increasing and V_{AN2B} increasing. The 1- μm CMOS controller demonstrator in this paper obtains ΔI values as small as 8 nA, for input currents around 10 μA , with $I_G = 0.5 \mu\text{A}$, and transistors sizes $W = 10 \mu\text{m}$ and $L = 5 \mu\text{m}$.

On the other hand, the current gain error due to input–output dc voltage mismatching is given by

$$\varepsilon \approx 1 + \frac{V_{D|M_{N2B \max}} - V_{D|M_{N3S}}}{V_{AN}} \quad (18)$$

where we have assumed equal Early voltages for the input and output transistors. Calculation of this error for Fig. 6(a) and a

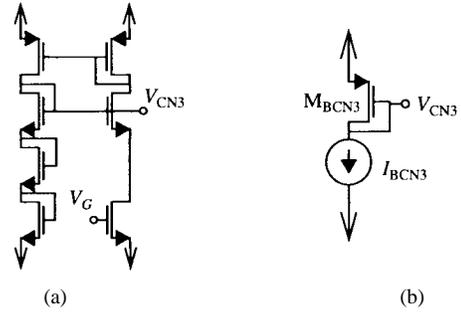


Fig. 7. Circuitry for the minimum computation: (a) adaptive bias circuit and (b) fixed bias circuit.

maximum current level I obtains

$$\varepsilon \approx \frac{\frac{[V_{T0N} + V_{TN2T} + V_{TN3C} + V_1 - V_{CN3}]}{V_{AN}}}{1 + \frac{[V_{T0N} + V_{TN2T} + V_2]}{V_{AN}}} \quad (19)$$

where

$$V_1 = \sqrt{\frac{I}{\beta_{N3C}}} + \sqrt{\frac{I}{\beta_N}} + \sqrt{\frac{I_G}{\beta_{N2T}}}$$

and

$$V_2 = \sqrt{\frac{I}{\beta_N}} + \sqrt{\frac{I_G}{\beta_{N2T}}}$$

and we assume $\beta_{N2B} = \beta_{N3S} \equiv \beta_N$. This expression shows that V_{CN3} can be chosen to annul the error for a given current level. Because the compensation value depends on the current, the adaptive biasing stage of Fig. 7(a) [27] can be used to obtain V_{CN3} varying with the current level. In the 1- μm CMOS technology used in the paper's prototype, this adaptive biasing obtains errors as low as 0.3% for input currents up to 20 μA —a precision larger than needed for most practical fuzzy logic applications. In practice a simpler biasing stage [see Fig. 7(b)] providing a constant V_{CN3} value is enough. This voltage V_{CN3} can be obtained by making $\varepsilon = 0$ in (19) for I corresponding to the middle of the range. The size of M_{BCN3} in Fig. 7(b) is then determined by

$$S_{BCN3} \equiv \frac{\beta_{BCN3}}{\beta_{0P}^*} = \frac{I_{BCN3}}{\beta_{0P}^* (V_{DD} - V_{CN3} - V_{TBCN3})^2} \quad (20)$$

where β_{0P}^* is the large-signal transconductance density of the pMOST.

Another strategy to attenuate this error is by adding cascode transistors [see Fig. 8(b)] to equalize both drain voltages in (18). However, this slows down the transient following an interterminal switching of the maximum input current. This transient has two phases: during the first the voltage V_G remains quasi-constant while the voltage at the new winning input terminal is builtup (henceforth called *switching transient*); during the second phase the voltage V_G is updated to conform to the new current (*propagation transient*). Differences between Fig. 8(a) and (b) arises mostly at the switching transient and can be assessed by comparing the time constants

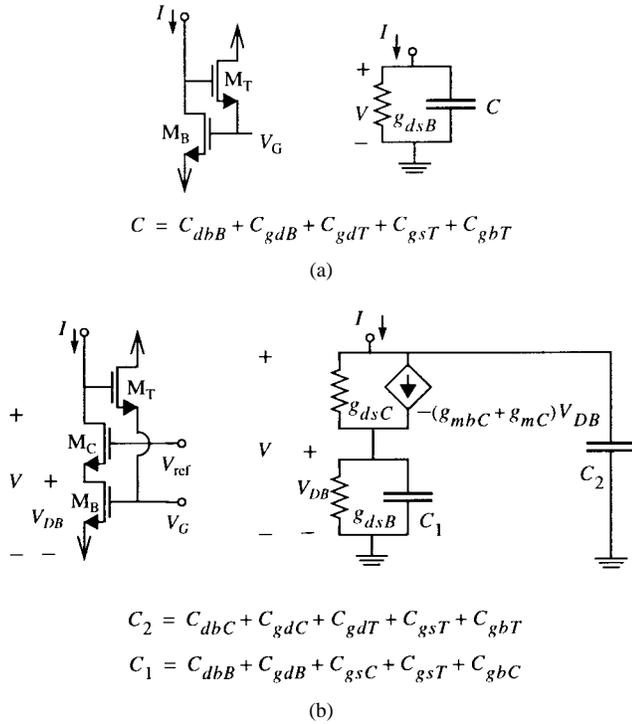


Fig. 8. Circuitry for the minimum computation—small signal models of input unit cells: (a) noncascode and (b) cascode.

of the first-order models attached to the figures,

$$\tau|_{\text{simple}} = \frac{C}{g_{dsB}} \quad \tau|_{\text{cascode}} = \frac{C_2}{g_{dsB}} \times \frac{g_{mC} + g_{mBC}}{g_{dsC}}. \quad (21)$$

Assuming equal transistor sizes so that $C = C_2$, and because $g_m \gg g_{ds}$, we obtain $\tau|_{\text{cascode}} \gg \tau|_{\text{simple}}$ —the reason leading us to discard cascode input transistors.

Besides of dynamic aspects involved in the switching process, we have to take into account that the dynamic response of these implementations depend on the number of inputs, since the parasitic capacitance at the common gate increases. Possible solutions for a high number of inputs are using bias currents and trees with complemented pMOS and nMOS circuits [32].

IV. RULE BLOCK

The k th rule block is intended to: 1) calculate the current w_k and 2) generate an output current given by

$$y_k = y_k^* \frac{w_k}{\sum_{k=1, N} w_k} \quad (22)$$

and these currents are then routed to a common node to implement (4) through KCL.

There are three main approaches for the analog implementation of (22): 1) using an extension of Mead [33] follower-aggregation circuit with weighting capability [16], [37]; 2) using weighting-plus-division circuits [14], [22], [35] [36]; and 3) using normalization-plus-weighting circuits [9], [25], [28]. The first uses an elegant circuit concept, see Fig. 9(a), to implement a nonlinear version of (4) with voltage output. However, because of the feedback, its transient response is not

optimum; also, because a large signal current w_k is applied at the transconductance amplifier bias terminals, the linear operation range and the transient response, are largely nonhomogeneous over the universe of discourse; finally, additional MDAC's are required to incorporate digital programmability of the singletons. Fig. 9(b) and (c) shows the concepts of the other two approaches. Both permit transparent digital programmability of the singletons. However, different reasons lead us to using the normalization-plus-weighting approach. First, the weighting-plus-division approach requires replication of the input currents and wide-range linear current-mode dividers, while the normalization can be realized through a collective computation circuit with only two transistors per input; the chosen approach results, hence, in simpler circuits. Second, because the transmission path for the numerator and the denominator of (4) are not the same in Fig. 9(b) this approach is more sensitive to mismatching. Third, the transient response of Fig. 9(b) is largely-dependent on the signal level. Fourth, there is no simple way to compensate for the errors in the divider—the only way is using very accurate dividers.

Fig. 10 shows the schematics of the rule block where four different operations are realized: first, the current $\overline{w}_k + I_B$ is generated as explained in Section III-B; second, this current is complemented and shifted to obtain $w_{ks} = w_k + I_{OS}$; third, a collective computation is carried out by all the rule blocks (they share the global nodes A_{NOR} and B_{NOR}) to realize the normalization operation; fourth, the resulting current is weighted by a digitally controlled current mirror to obtain the shifted version of the k th rule output current.

A. Normalization Circuitry

Fig. 11 shows the CMOS normalizer circuit used in our chip based on a translinear BJT circuit by Gilbert [34]. As a difference to the normalizers used in [9], [28], Fig. 11 does not involve any global feedback loop and, hence, features much faster dynamic response. Note that Fig. 11 can be split into N cells, one per each input-output pair, plus a little common circuitry consisting of the transistor M_{NA} and the current source I_{SS} . Fig. 10 exploits this modularity by incorporating one of these cells at each rule block.

Assuming that the transistors operate in strong inversion, where the BJT translinear principle does not hold, the circuit is found to realize the following nonlinear transformation:

$$w_{ks}^* = \frac{\beta_{N4t}}{\beta_{N4b}} w_{ks} \left[1 + \frac{\eta(\mathbf{w}_s)}{\sqrt{w_{ks}}} \right]^2 \quad (23)$$

where the function $\eta(\mathbf{w}_s)$ is

$$\eta(\mathbf{w}_s) = \frac{\sum_s \sqrt{w_{ks}}}{N} \left(\sqrt{1 + \frac{N(I'_{SS} - \sum_k w_{ks})}{\left(\sum_k \sqrt{w_{ks}^2}\right)}} - 1 \right),$$

$$I'_{SS} = \frac{\beta_{N4b}}{\beta_{N4t}} I_{SS} \quad (24)$$

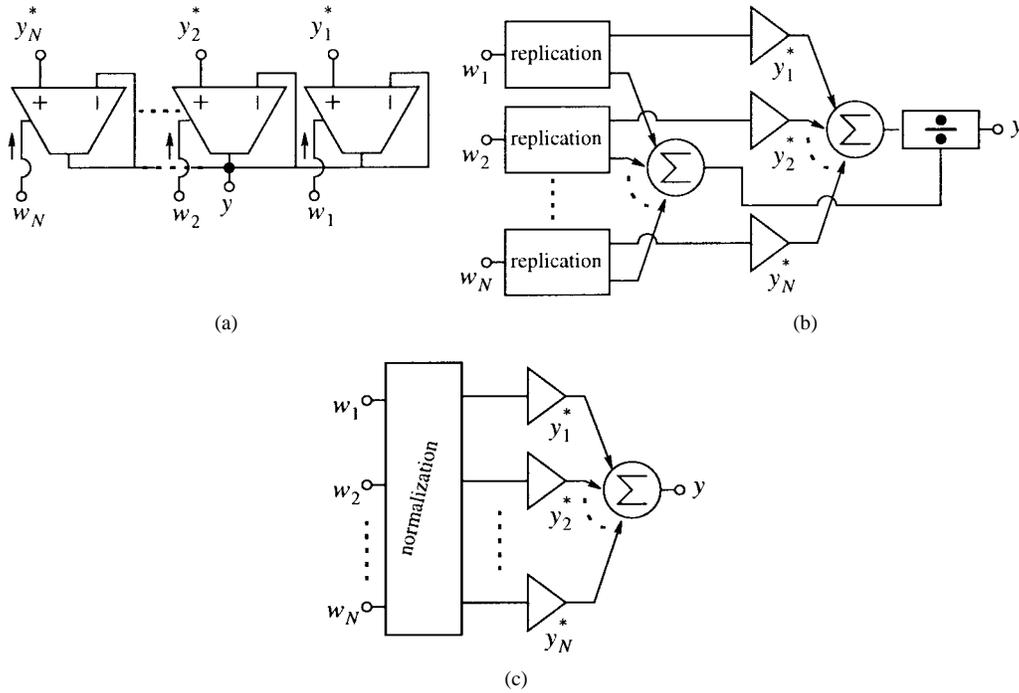


Fig. 9. Singleton defuzzification strategies: (a) follower-aggregation; (b) weighting-plus-division; and (c) normalization-plus-weighting.

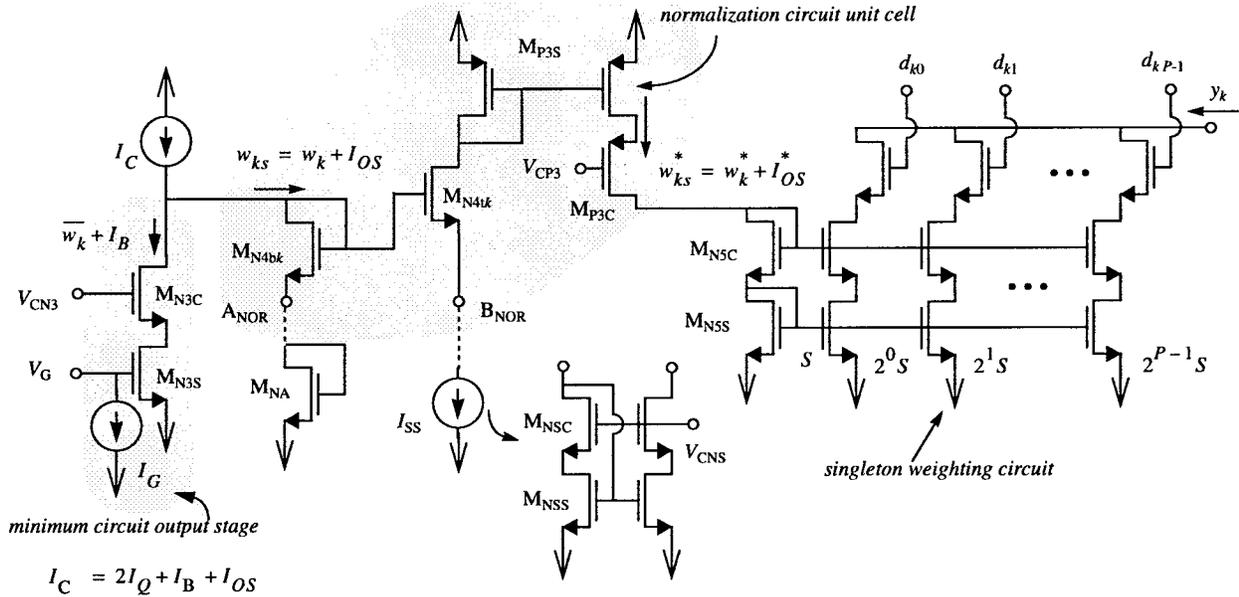


Fig. 10. Rule block.

and

$$w_k^* = w_{ks}^* - I_{OS}^*; \quad I_{OS}^* = w_{ks}^*(I_{OS}). \quad (25)$$

The offset current I_{OS} is added to improve the dynamic behavior. Note from Fig. 10 that it is related to the bias currents in the rule antecedent by $I_{OS} = I_C - 2I_Q - I_B$. Thus, it can be introduced by just increasing the current I_C without additional area cost, although it will be preserved in figures and equations to gain clarity.

The circuit in Fig. 11 exhibits the following features: 1) the sum of all output currents is constant and equal to I_{SS} ; 2) for each input, the input-output transformation is

a soft monotonic one, i.e., the higher an input current, the higher the corresponding output current. Thus, the relative strengths of the different rule antecedents are preserved at the outputs—as required for defuzzification [1]–[4]. Hence, although this circuit does not realize the ideal normalization operation, it keeps the essential features needed for defuzzification; nonlinearity is not problematic because the whole controller chip is highly nonlinear. Actually, system-level analysis shows that, despite this nonlinearity, the normalization-plus-weighting defuzzification approach features smaller deviations from the linear interpolation than the ideal weighting-plus-division structure [27].

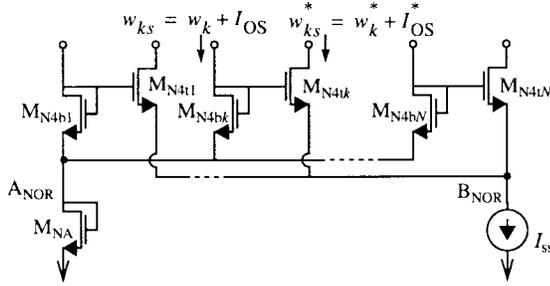


Fig. 11. Normalization circuit schematics.

B. Design Considerations in the Normalization Circuit

A first consideration refers to the input range of the normalization circuit when embedded into Fig. 10. Consider first the common-mode range, where all input currents are equal. If they increase, transistors M_{N4tk} evolve toward the ohmic region; on the other hand, if they decrease, the transistors used in the current source I_{SS} evolve toward the ohmic region. Thus, the common-mode input range is given by

$$I_{Bl} - I_{OS} \leq w_k \leq I_{Bh} - I_{OS},$$

$$I_{Bl} = \left[\frac{V_{Bnorm\Omega} - V_{T0N} + \sqrt{\frac{I_{SS}}{N\beta_{N4t}}}}{\sqrt{\frac{1}{\beta_{N4b}} + \sqrt{\frac{N}{\beta_{NA}}}}} \right]^2$$

$$I_{Bh} = \left[\frac{V_{DD} - V_{T0P} - V_{T0N} - \sqrt{\frac{I_{SS}}{N\beta_{P3S}}}}{\sqrt{\frac{1}{\beta_{N4b}} + \sqrt{\frac{N}{\beta_{NA}}}}} \right]^2 \quad (26)$$

where $V_{Bnorm\Omega}$ is the limiting voltage for the current source I_{SS} and we have assumed that the threshold voltages of top (M_{N4tk}) and bottom (M_{N4bk}) transistors are approximately equal, because their sources are at similar voltage. The bottom limit in (26) is valid whenever $V_{Bnorm\Omega} \geq V_{T0N} - \sqrt{I_{SS}/(N\beta_{N4t})}$, otherwise the real condition limit is zero. The wide range cascode current mirror enclosed in Fig. 10 allows us to obtain a good common mode range (given by $V_{Bnorm\Omega} = \sqrt{I_{SS}/\beta_{NSS}} + \sqrt{I_{SS}/\beta_{NSC}}$) as well as good precision.

Consider now the differential range; if one input current increases while the others are kept constants, the top transistor for the changing current will eventually drive all the current I_{SS} , and the other top transistors will be cut-off. The differential range is given by,

$$0 \leq w_k \leq \beta_{N4b} \left(\sqrt{\frac{I_{OS}}{\beta_{N4b}}} + \sqrt{\frac{I_{SS}}{\beta_{N4t}}} \right)^2 - I_{OS} \quad (27)$$

where we have considered that the set of fuzzy rules is consistent [39], i.e., when an input is maximum the remaining are zero.

There are three main sources of systematic errors in Fig. 10: the finite impedance of I_{SS} , the dc voltage mismatching among output nodes of the circuit core (transistors M_{N4tk} and M_{N4bk} in Fig. 11), and the dc voltage mismatching between input and output nodes of the output pMOS mirrors. The

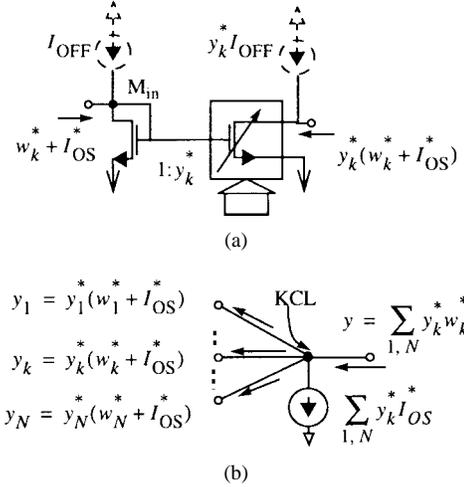


Fig. 12. (a) Singleton weighting concept and (b) controller output node.

adopted cascode realization of I_{SS} makes the first negligible. On the other hand, because the top transistors are connected to low-impedance nodes, the second error is largely attenuated by the transconductances of the pMOST's used at these nodes. Concerning the third error source, it can be minimized by inserting cascode transistors, as Fig. 10 shows. The error is then given by

$$\varepsilon(w_{ks}^*) = \frac{w_{ks}^* \left(V_{DD} - 2V_{T0P} - \sqrt{\frac{w_{ks}^*}{\beta_{P3S}}} - \sqrt{\frac{w_{ks}^*}{\beta_{P3C}}} - V_{CP3} \right)}{V_{AP3S}} \quad (28)$$

which is minimized by proper choice of V_{CP3} . Again, a particular signal value has to be selected to guide the choice of V_{CP3} . Because most output branches drive a current value I_{OS}^* , such current level defines a good choice. Thus, V_{CP3} is obtained from (28) for $\varepsilon = 0$ and $w_{ks}^* = I_{OS}^*$, and it is generated in similar as already explained for Fig. 7(b).

With regard to the dynamic response, analysis recommends to scale the width of M_{NA} as well as the value of I_{SS} proportionally to the number of normalizer inputs, i.e., rules in the controller, in order to preserve the dynamic response as the complexity increases.

C. Singleton Weighting and Output Layer

Fig. 10 employs a digitally-controlled current-mirror [represented at the conceptual level in Fig. 12(a)] to implement a programmable singleton value y_k^* . As compared to analog-programmed current mirrors [38], [40], the digital approach is preferred because it is more robust and accurate, compatible with standard memory circuits and directly controllable through conventional computers.

Regarding the mirror circuitry itself, and because the normalization circuit output stage does not impose major range limitations, a stacked (self-biased) cascode structure is used to minimize errors due to dc mismatching. On the other hand, parallel-connected unit transistors are used to realize the binary weighting and, thus, reduce systematic errors caused by the

lack of symmetry. The bias current depicted with dashed lines in Fig. 12(a) is added to reduce speed degradations due to the increase of the parasitic capacitance for large singleton values.

After singleton weighting the rule block outputs are wired up to the output node where a current $-\sum_{k=1, N} y_k^* I_{OS}^*$ is added to remove the offset and, thus, obtain (4).

V. GLOBAL CONSIDERATIONS

A. Dependence on Temperature

Changes of the controller transfer function with temperature are basically due to the temperature dependence of the large-signal transconductance densities $\beta_{0P, N}^*$ and the zero bias threshold voltages $V_{T0N, P}$. Thus, those building blocks whose input-output relation is not affected by these parameters do not contribute to output changes when the temperature varies. This is the case of the minimum and the singleton-weighting circuits, where temperature-dependence cancellation is based on the same principles as for current mirrors. The normalization circuit does not contribute either because β_{0N}^* appears at the numerator and the denominator in (23) and (24). The membership function circuit is the only whose transfer function [given by (6)] depends on temperature. However, the electrical values of the logical zero and one are not affected, provided the current reference is temperature-independent, because these values are associated to logical states of the transistors in the differential pairs. On the other hand, the width and center defined in (7) do neither depend on $\beta_{0P, N}^*$ nor $V_{T0N, P}$. The only parameter which is affected by temperature changes is the membership function slope ζ . Its dependence can be expressed through parameter $\rho_T = (1/\zeta)(\partial\zeta/\partial T)$ whose maximum value is given by

$$\rho_T|_{x_i=E_{ij}} = -\frac{3}{4T} \quad (29)$$

at the crossover points. From a global point of view, this means that the slope of the generated function between interpolation points changes with temperature, as Fig. 13(a) illustrates for a controller with four rules. Thus, the interpolation smoothness changes with temperature, but the interpolation points are not affected if membership functions are wide enough to saturate in the whole temperature range. Fig. 13(b) shows the difference of the values provided by Fig. 13(a) for every input value. Note that the difference is minimum in the interpolation points (for input values 1.75, 2.25, 2.75, and 3.25 V). Note also that such difference is always below 0.75% of the full output range for the temperature changing from 0–100°C.

B. Power Estimation

Let us consider a controller with M inputs, N rules, and $L = N^{1/M}$ fuzzy labels whose maximum singleton value in the associated rule base is $y_{k \max}^*$. The maximum static power consumption is calculated as:

$$\begin{aligned} P_w &= I_{st} V_{DD} + V_{load} I_{SS} y_{k \max}^*, \quad \text{for } I_{st} \text{ being} \\ I_{st} &= M \times [N \times (I_U + I_B) + L \times (2L_U + I_B)] \\ &\quad + N \times (I_C + I_G) + 2I_{SS} \end{aligned} \quad (30)$$

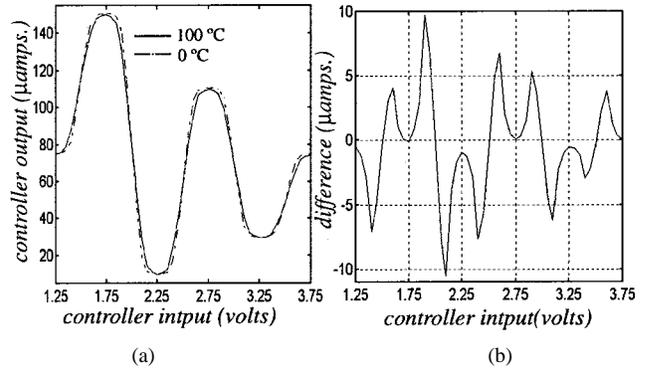


Fig. 13. Illustrating dependence on temperature.

where $I_U = 2I_Q$ and the currents I_B , I_G , and I_C are defined in Figs. 5, 6, and 10.

C. Mismatching Errors

Random variations of the transistor parameters V_{T0} , β , and γ can be modeled as normal distributions whose mean values are the nominal parameter values. For close and small enough transistors the variances depend mostly on the device area [12]

$$\begin{aligned} \sigma^2(V_{T0}) &= \frac{A_{V_{T0}}^2/2}{WL}, \\ \sigma^2(\gamma) &= \frac{A_\gamma^2/2}{WL}, \\ \frac{\sigma^2(\beta)}{\beta^2} &= \frac{A_\beta^2/2}{WL} \end{aligned} \quad (31)$$

where W and L are the transistor channel width and length, and $A_{V_{T0}}^2$, A_γ^2 , and A_β^2 are technology-dependent.

Based on (31), we can obtain expressions for the errors in the fuzzy controller blocks. The detailed explanation of these errors is beyond the scope of this paper; thus, only those resulting in important design equations will be outlined.

Consider the membership function circuit first. Analysis shows that the most significant error corresponds to the case where the rule output is maximum [27]. The variance of the complement of the membership function current (its mean value is I_B) is given by,

$$\begin{aligned} \sigma^2(\tilde{s}_{ij}) &= 4I_Q^2 \frac{\sigma^2(\beta_{NQ})}{\beta_{NQ}^2} + 16\beta_{NQ} I_Q \sigma^2(V_{T0NQ}) \\ &\quad + 8I_Q^2 \left(\frac{\sigma^2(\beta_P)}{\beta_P^2} + \frac{4\beta_P \sigma^2(V_{T0P})}{2I_Q} \right) \\ &\quad + I_B^2 \frac{\sigma^2(\beta_{NB})}{\beta_{NB}^2} + 4\beta_{NB} I_B \sigma^2(V_{T0NB}) \\ &\quad + 2I_B^2 \left(\frac{\sigma^2(\beta_P)}{\beta_P^2} + \frac{4\beta_P \sigma^2(V_{T0P})}{I_B} \right) \end{aligned} \quad (32)$$

where we assume that the pMOST's $MP1$ and $MP2$'s are equal, and cascode transistors mismatching is not computed because their influence is negligible as compared to signal transistors. This expression includes the errors due to the

nMOS transistors (parameters β_{NB} and V_{TONB}) of the current mirror used to provide I_B in Fig. 5.⁷

The error at the rule output is calculated by adding the error caused by the minimum circuit to the previous one. The variance for the worst case (only one antecedent active in the rule and nonsharing of the membership function circuits) is

$$\begin{aligned} \sigma^2(w_{ks}) = & 2I_B^2 \left(\frac{\sigma^2(\beta_{N3S})}{\beta_{N3S}^2} + \frac{4\beta_{N3S}\sigma^2(V_{TON3S})}{I_B} \right) \\ & + I_C^2 \frac{\sigma^2(\beta_{PC})}{\beta_{PC}^2} + 4\beta_{PC}I_C\sigma^2(V_{TOPC}) + \sigma^2(\tilde{s}_{ij}) \end{aligned} \quad (33)$$

and the corresponding mean value is $I_C - I_B$. The bracketed terms correspond to the maximum circuit; the others to the complement and membership function circuits. The mismatch is smaller for any other case, although the expression of the variance is difficult to obtain because of correlations between variables. Parameters β_{PC} and V_{TOPC} in (33) correspond to the large signal transconductance value and the zero bias threshold voltage respectively, of the noncascade output pMOS transistor in a current mirror that provides I_C in Fig. 5.

The errors due to the normalization circuit are characterized by the following approximate variance expression

$$\begin{aligned} \sigma^2(w_{ks}^*) \approx & 4\beta_{N4t}w_{ks}^*(1-\Gamma)^2 \cdot \left[\sigma^2(V_{TON4b}) \right. \\ & + \sigma^2(\gamma_{N4b}) \left(\sqrt{V_{ANOR} + \phi_B} - \sqrt{\phi_B} \right)^2 \left. \right] \\ & + 4\beta_{N4t}w_{ks}^*(1-\Gamma)^T \cdot \left[\sigma^2(V_{TON4t}) \right. \\ & + \sigma^2(\gamma_{N4t}) \left(\sqrt{V_{BNOR} + \phi_B} - \sqrt{\phi_B} \right)^2 \left. \right] \\ & + (w_{ks}^*)^2(1-\Gamma)^2 \frac{\sigma^2(\beta_{N4t})}{\beta_{N4t}^2} \\ & + \frac{\beta_{N4t}}{\beta_{N4b}} w_{ks}^* w_{ks} (1-\Gamma)^2 \frac{\sigma^2(\beta_{N4b})}{\beta_{N4b}^2} \\ & + 2(w_{ks}^*)^2 \left(\frac{\sigma^2(\beta_{P3S})}{\beta_{P3S}^2} + \frac{4\beta_{P3S}\sigma^2(V_{TOP3S})}{w_{ks}^*} \right) \\ & + \left(\frac{N-1}{N} \frac{\beta_{N4t}}{\beta_{N4b}} \sqrt{\frac{I'_{SS}}{Nw_{ks}}} \right)^2 \sigma^2(w_{ks}) \end{aligned} \quad (34)$$

where the mean value of w_{ks}^* is given by (23)

$$\Gamma = \frac{\sqrt{w_{ks}^*}}{\sum_{k=1, N} \sqrt{w_{ks}^*}} \quad (35)$$

and $\sigma^2(w_{ks})$ is given by (33) for a maximum rule antecedent output current.

The approximation used to calculate (34) consists of neglecting the mismatching in those normalizer inputs others than the k th. These terms contribute only around 3% of the variance for the 16 rules CMOS prototype in this paper,

⁷This mirror was omitted there in behalf of clarity and because its design is not critical for other performance parameters.

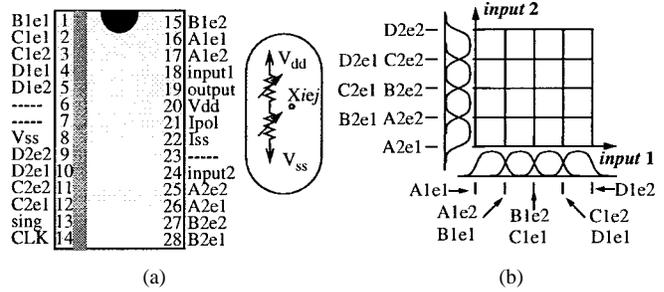


Fig. 14. Illustration of programmability: (a) chip pin-out and (b) example of a uniform lattice partition programming.

and their contribution decreases as the rule count increases. This highlights an interesting feature of Fig. 11 which is not shared by other approaches to the normalization operation; namely, the mismatching errors of the different rules are nearly independent. Thus, they are not mixed in the output node and manifest as offsets (easy to correct) at the points where the rule outputs are maximum (the most significant to design purposes).

The global error at the rule block output includes also the influence of the weighting circuit

$$\begin{aligned} \sigma^2 y_k = & (y_k^*)^2 \sigma^2(w_{ks}^*) \\ & + \left[y_k^2 \frac{\sigma^2(\beta_{in})}{\beta_{in}^2} + 4\beta_{in}y_k\sigma^2(V_{TOin}) \right] \left(1 + \frac{1}{y_k^*} \right) \end{aligned} \quad (36)$$

where $\sigma^2(w_{ks}^*)$ is given by (34), and β_{in} and V_{TOin} refers to the noncascade input transistor in the weighting circuit [see Fig. 12(a)]. The first term at the right in (35) corresponds to the error transmitted by the weighting circuit from previous stages, while the second term corresponds to the error introduced by itself. Note that the latter decreases when the singleton value grows.

While residual systematic errors may be filtered out by the normalizer [27], the only way to attenuate the random errors is solving the design equations (31)–(36) to obtain proper transistor sizes, which is more conveniently performed with the help of an iterative optimizer.

D. Programmability

Fig. 14(a) shows the pin-out of the prototype presented in this paper. Pins $Xiej$ (for $X = A, B, C, D$; $i = 1, 2$ and $j = 1, 2$) as well as “sing” and “CLK” are dedicated to programming. The desired lattice partition is programmed by means of analog voltages at $Xiej$ inputs, as illustrated in Fig. 14(b). Note that they are easily identified once the partition is decided [an uniform partition is shown in Fig. 14(b) for the sake of clarity, but any other lattice partition can be generated]. These voltages coincide with those at the crossover points in (7) that define the center and width of the membership functions. Such voltages are generated externally in the prototype of the paper using variable resistors, as the inset in Fig. 14 illustrates. On the other hand, the controller output at the interpolation points (core of the fuzzy sets in the partition) is serially programmed by digital signals at inputs “sing” and “CLK.” Singletons associated to each fuzzy set in the partition are the digital words $\{d_{k0}, d_{k1}, \dots, d_{kP-1}\}$ in

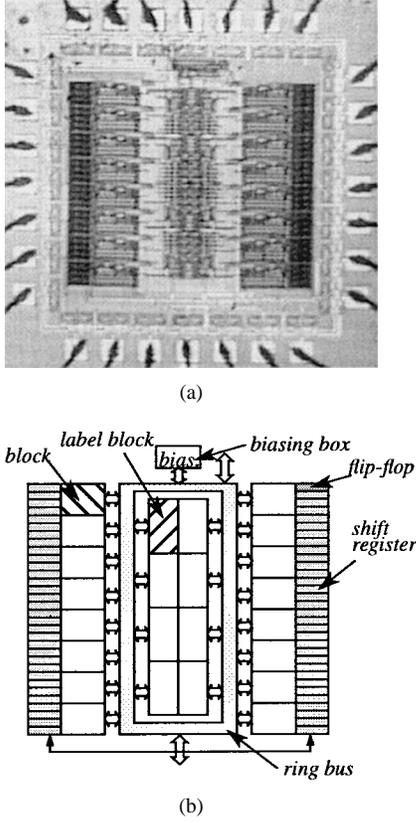


Fig. 15. (a) Chip microphotograph and (b) internal architecture.

TABLE I
TRANSISTOR SIZES (W/L) IN $\mu\text{m}/\mu\text{m}$ IN THE PROTOTYPE

| M_{N1ij} | M_{N1ij+} | M_{P1} | M_{P2S} | M_{P2C} | M_{N2B} | M_{N2T} | M_{Nq} | M_{NQC} |
|------------|-------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 40/10 | 40/10 | 50/5 | 50/5 | 50/2 | 20/10 | 20/10 | 10/10 | 10/1 |
| M_{N3S} | M_{N3C} | M_{N4w} | M_{N4l} | M_{P3S} | M_{P3C} | M_{N5S} | M_{N5C} | |
| 20/10 | 20/1 | 30/5 | 50/5 | 50/5 | 50/1 | 20/10 | 20/5 | |

the singleton weighting circuit inside the rule block of Fig. 10. In the prototype of the paper, the singletons are encoded in digital words of 4 bits and stored on-chip in a static RAM.

VI. EXPERIMENTAL RESULTS

Fig. 15(a) shows the microphotograph of a chip that performs the processing tasks involved in (4) and Fig. 2(a). It is a lattice controller with two inputs and four labels per input [see Fig. 2(b)]. Thus, eight label blocks, four per chip input, are needed, as well as sixteen rule blocks. The label blocks outputs are connected to inputs of rule blocks through a “ring bus.” Bias circuitry, as well as one diode connected transistor and one current mirror, which complete the normalization circuit in Fig. 11, are implemented in the “biasing box.” Table I show the most relevant transistor sizes in this chip.

Digital values to program the output current mirror and hence the singleton values are stored in a “shift register” which is the chip internal memory element and is serially programmed through two pads. Apart from digital programmability of the singleton values, width and location of membership

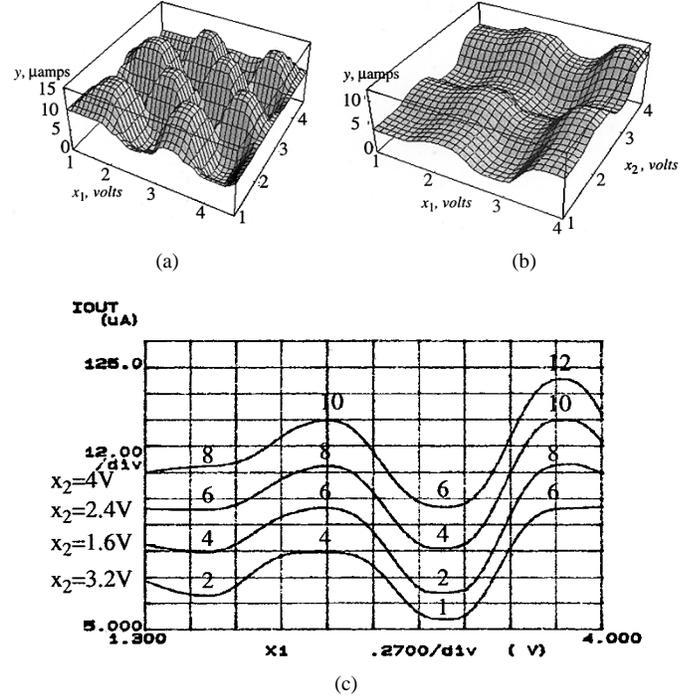


Fig. 16. (a) and (b) Controller output for two different sets of singleton values: (c) and sections from (b) at maximum local points.

functions are also analogically programmable by setting the voltages E_{ij+} and E_{ij-} [see (7) and Fig. 5].

Fig. 16(a) and (b) shows two output surfaces generated by the chip. The bias signals are $V_{DD} = 5\text{ V}$, $V_{SS} = 0\text{ V}$, $I_Q = 7.5\ \mu\text{A}$, $I_B = 10\ \mu\text{A}$, $I_G = 0.5\ \mu\text{A}$, $I_C = 35\ \mu\text{A}$, and $I_{SS} = 37\ \mu\text{A}$, while the voltages E_{ij} are fixed to obtain a uniform lattice partition of the input space. The circuit was loaded with a constant voltage source of 2.5 V and a current source to remove the offset introduced in the normalization circuit. Singletons are set to decimal values 1 and 15 in Fig. 16(a), which highlights the locality of the fuzzy basis functions, while Fig. 16(b) illustrates an exemplary surface obtained with different singleton values. Finally, Fig. 16(c) depicts a set of sections from Fig. 16(b) which show the output when it reaches their local maximum values, thus the singleton values.

Maximum circuit delay is 471 ns (90% of the full scale output current) for a step input. For this test, all the singletons in the controller had the decimal value 1, except one of them which was set to the maximum value 15. Under these conditions, one input was forced to remain constant, while the other input changes following a pulse. As a consequence, the output changes from the minimum value to the maximum one and vice-versa. The maximum power consumption measured in the previous experiment was 8.6 mW. With respect to the resolution, it is around 6.5%. The latter was obtained through Monte Carlo simulations (30 iterations) which take into account parameter mismatching among transistors, with 3σ ($\pm 1.5\sigma$) as error figure. Finally, input voltage range is over 3.25 V and the area of the chip without pads is 1.6 mm². It is possible to achieve faster designs by introducing bias currents at input and output branches of the current mirror

TABLE II
CMOS ANALOG IMPLEMENTATIONS OF FUZZY CONTROLLERS

| CMOS | Manaresi [16] | Guo [17] | Proposed |
|----------------------------|---------------------------|----------------------------|----------------------------|
| Complexity | 9rules@2input @2output | 13rules@3input @1output | 16rules@2input @1output |
| Technology | 0.7 μ m CMOS | 2.4 μ m CMOS | 1 μ m CMOS |
| Power Consumption | 44mW@5V | 550mW@10V | 8.6mW@5V |
| Input to Output Delay | 570 ns | 160 ns | 471 ns |
| Precision | No data | No data | 6.5% (3 σ) |
| Interface (inputs@outputs) | voltages@ voltages | voltages@ voltages | voltages@ currents |
| Area | 1.9 mm ² | 16.2 mm ² | 1.6 mm ² |

that replicates membership function output, and in the output mirror that implements singleton weighting. It is also possible to achieve a higher precision by inserting the chip in a learning loop with a computer and using the hardware-compatible learning algorithms presented in [18].

For comparison purposes, Table II shows data from three continuous-time analog CMOS monolithic controllers that implement similar algorithm as the paper's prototype. Other reported implementations are difficult to compare because they are not monolithic and/or differ from the presented prototype in some important aspect. On the other hand, although [16] and [17] report examples of the controllers in a wider context, they provide data enough to make the comparisons. Because the prototypes are realized using different technologies no absolute conclusions can be drawn from the comparison. However, and bearing this in mind, Table II shows that the proposed prototype features much smaller $Delay \times Power$, while their remaining features are competitive. Finally, it is the only one that reports results of the resolution under random fluctuations of the technological parameters—an issue which affects the other controller features because it defines many tradeoffs during the design cycle.

VII. CONCLUSIONS

A highly modular fuzzy controller chip has been proposed. The design methodology is based on two high-level building blocks, the *label* and the *rule* blocks, which are respectively identified with the antecedent labels and the consequent of fuzzy inference rules. These blocks are prepared to be readily connected for the realization of lattice fuzzy partitions. We propose circuit implementation for these high-level blocks and present detailed discussions concerning their practical design. The results from a prototype of 16 rules, 2 inputs, and 1 output are shown in Table II and compared with other proposals. They demonstrate that the approach provides very good results for medium-to-low complexity controllers, as those usually implemented with analog techniques. Further improvements should face the error aggregation at collective computation nodes as well as the degradation of the $Delay \times Power$ when the complexity increases. Current work of the authors takes

advantage from the building blocks of this paper to build controllers of higher complexity using mixed-signal circuits. Automatic tuning or learning rules are also used to increase the complexity while maintaining the $Delay \times Power$ and the precision.

REFERENCES

- [1] J. M. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, pp. 345–377, Mar. 1995.
- [2] H. Takagi, "Applications of neural networks and fuzzy logic to consumer products," in *Fuzzy Logic Technologies and Applications*. New York: IEEE Press, 1994, pp. 8–12.
- [3] M. Brown and C. Harris, *Neuro-Fuzzy Adaptive Modeling and Control*. Englewood Cliffs: Prentice Hall, 1994.
- [4] J. S. R. Jang and C. T. Sun, "Neuro-fuzzy modeling and control," *Proc. IEEE*, vol. 83, pp. 378–406, Mar. 1995.
- [5] H. Watanabe, W. D. Dettloff, and K. E. Yount, "A VLSI fuzzy logic controller with reconfigurable, cascadable architecture," *IEEE J. Solid-State Circuits*, vol. 25, pp. 376–382, 1990.
- [6] K. Nakamura, N. Sakashita, Y. Nitta, K. Shimomura, and T. Tokuda, "Fuzzy inference and fuzzy inference processor," *IEEE Micro*, vol. 13, pp. 37–48, Oct. 1993.
- [7] H. Eichfeld, M. Klimke, M. Menke, J. Nolles, and T. Künemund, "A general-purpose fuzzy inference processor," *IEEE Micro*, vol. 15, pp. 12–17, June 1995.
- [8] A. Costa, A. de Gloria, P. Faraboschi, A. Pagni, and G. Rizzotto, "Hardware solutions for fuzzy control," *Proc. IEEE*, vol. 83, pp. 422–434, Mar. 1995.
- [9] T. Yamakawa, "A fuzzy inference engine in nonlinear analog mode and its application to a fuzzy logic control," *IEEE Trans. Neural Networks*, vol. 4, pp. 496–522, May 1993.
- [10] E. A. Vittoz, "The future of analog in the VLSI environment," in *Proc. 1990 IEEE Int. Symp. on Circuits and Systems*, 1990, pp. 1372–1375.
- [11] K. A. Nishimura, "Optimum partitioning of analog and digital circuitry in mixed-signal circuits for signal processing," Ph.D. dissertation, U.C. Berkeley, 1993.
- [12] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, pp. 1433–1440, Oct. 1989.
- [13] J. W. Fattaruso, S. S. Mahant-Shetti, and J. B. Barton, "A fuzzy logic inference processor," *IEEE J. Solid-State Circuits*, vol. 29, pp. 397–402, Apr. 1994.
- [14] J. L. Huertas, S. Sánchez-Solano, I. Baturone, and A. Barriga, "Integrated circuit implementation of fuzzy controllers," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1051–1058, July 1996.
- [15] L. Lemaitre, M. J. Patyra, and D. Mlynek, "Analysis and design of CMOS fuzzy logic controller in current mode," *IEEE J. Solid-State Circuits*, vol. 29, pp. 317–322, Mar. 1994.
- [16] N. Manaresi, R. Rovatti, E. Franchi, R. Guerrieri, and G. Baccarani, "A silicon compiler of analog fuzzy controllers: From behavioral specifications to layout," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 418–428, Nov. 1996.
- [17] S. Guo, L. Peters, and H. Surmann, "Design and application of an analog fuzzy logic controller," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 429–438, Nov. 1996.
- [18] F. Vidal-Verdú and A. Rodríguez-Vázquez, "Learning under hardware restrictions in CMOS fuzzy controllers able to extract rules from examples," in *Proc. IFSA'95*, Sao Paulo, Brazil, July 1995, pp. 189–192.
- [19] A. Rodríguez-Vázquez, M. Delgado-Restituto, and F. Vidal, "Synthesis and design of nonlinear circuits," in *The Circuits and Filters Handbook W.-K. Chen, Ed.* Boca Raton, FL: CRC, 1996, pp. 935–972.
- [20] A. Rodríguez-Vázquez and M. Delgado-Restituto, "CMOS design of chaotic oscillators using state variables: A monolithic Chua's circuit," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 596–613, Oct. 1993.
- [21] ———, "Generation of chaotic signals using current-mode techniques," *J. Intelligent Fuzzy Syst.*, vol. 2, pp. 15–37, 1994.
- [22] T. Kettner, C. Heite, and K. Schumacher, "Analog CMOS realization of fuzzy logic membership functions," *IEEE J. Solid-State Circuits*, vol. 28, pp. 857–861, July 1993.
- [23] M. Sasaki, N. Ishikawa, F. Ueno, and T. Inoue, "Current-mode analog fuzzy hardware with voltage input interface and normalization locked loop," *IEICE Trans. Fundamentals*, vol. E75-A, pp. 650–654, June 1992.

- [24] J. W. Fattaruso and R. G. Meyer, "MOS analog function synthesis," *IEEE J. Solid-State Circuits*, vol. 22, pp. 1059–1063, Dec. 1987.
- [25] A. Rodríguez-Vázquez and F. Vidal, "Analog CMOS design of singleton fuzzy controllers," in *3rd Int. Conf. Industrial Fuzzy Control Intelligent Systems*, Dec. 1993.
- [26] Y. Tsividis, *Mixed Analog-Digital VLSI Devices and Technology*. New York: McGraw-Hill, 1996.
- [27] F. Vidal, "Design of mixed-signal CMOS neuro-fuzzy controllers," Ph.D. dissertation, Univ. of Málaga, 1996.
- [28] M. Sasaki, T. Inoue, Y. Shirai, and F. Ueno, "Fuzzy multiple-input maximum and minimum circuits in current mode and their analyzes using bounded difference equations," *IEEE Trans. Computers*, vol. 39, pp. 768–774, June 1990.
- [29] T. Yamakawa and T. Miki, "The current mode fuzzy logic integrated circuits fabricated by the standard CMOS process," *IEEE Trans. Computers*, vol. C-35, pp. 161–167, Feb. 1986.
- [30] J. Lazzaro, R. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-take-all networks of $O(n)$ complexity," in *Advances in Neural Information Processing Systems*, vol. 1, D. S. Touretzky, Ed. Los Altos, CA: Morgan Kaufmann, 1989.
- [31] C. Y. Huang and B. D. Liu, "Current-mode multiple-input maximum circuit for fuzzy logic controllers," *Electron. Lett.*, vol. 30, pp. 1924–1925, 1994.
- [32] K. D. Peterson and R. L. Geiger, "Area/bandwidth tradeoffs for CMOS current mirrors," *IEEE Trans. Circuits Syst.*, vol. CAS-33, no. 7, pp. 667–669, July 1986.
- [33] C. Mead, *Analog VLSI and Neural Systems*. Addison Wesley, 1989.
- [34] B. Gilbert, "Current-mode circuits from a translinear view point: A tutorial," in *Analogue IC Design: The Current-Mode Approach*, C. Toumazou, F. J. Lidgley, and D. G. Haigh, Eds. London, UK: Peter Peregrinus Ltd., 1990.
- [35] T. Miki, H. Matsumoto, K. Ohto, and T. Yamakawa, "Silicon implementation for a novel high-speed fuzzy inference engine: Mega-flips analog fuzzy processor," *J. Intell. Fuzzy Syst.*, vol. 1, no. 1, pp. 27–42, 1993.
- [36] V. Catania, A. Puliafito, and L. Vita, "A VLSI fuzzy inference processor based on a discrete analog approach," *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 2, pp. 93–106, May 1994.
- [37] K. Tsukano and T. Inoue, "Synthesis of operational transconductance amplifier-based analog fuzzy functional blocks and its application," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 61–68, Feb. 1995.
- [38] M. Sasaki and F. Ueno, "A VLSI implementation of fuzzy logic controller using current mode CMOS circuits," in *3rd Int. Conf. Industrial Fuzzy Control Intelligent Systems*, Dec. 1993, pp. 215–220.
- [39] L.-X. Wang, *A Course in Fuzzy Systems and Control*. Englewood Cliffs, NJ: Prentice Hall, 1997.
- [40] A. Rodríguez-Vázquez, S. Espejo, R. Domínguez-Castro, and J. L. Huertas, "Current mode techniques for the implementation of continuous and discrete-time cellular neural networks," *IEEE Trans. Circuits Syst. II*, vol. 40, pp. 132–146, Mar. 1993.

Angel Rodríguez-Vázquez (M'80–SM'95–F'96), for photograph and biography, see this issue, p. 230.

Rafael Navas-Gonzalez received the degree of Licenciado en Ciencias Físicas (Branch of Electronics) from the University of Granada, Spain, in 1987. From 1988–1993, he worked as a Design Engineer in the microelectronics group of the R&D department of Fujitsu España S.A. During this period, he participated in several ASIC's specification and design projects. In 1993 he joined the University of Málaga, Spain, where he is currently teaching and making research toward the Ph.D. degree. His research activity focuses in specification and design of integrated circuits to implement fuzzy and neuro-fuzzy controllers using analog and mixed signal techniques and their applications.



Manuel Delgado-Restituto (M'96) received the Bachelor's degree in physics-electronics in 1988, and the Ph.D. degree in physics-microelectronics in 1996, both from the University of Seville, Spain.

Since 1990, he has been working at the Institute of Microelectronics of Seville—Centro Nacional de Microelectrónica (IMSE-CNM), where he is currently a member of the research staff. His research interests include analog and mixed-signal integrated circuit design for nonlinear signal processing using neuro-fuzzy controllers and chaotic circuits. He is

also interested in the design of integrated wireless transceivers.

Fernando Vidal-Verdú received the degree of Licenciado en Ciencias Físicas (Branch of Electronics) from the University of Seville, Spain, in 1988. From September 1988 to April 1991, he worked at the Department of Electronics and Electromagnetism of the University of Seville, in a project supported by Fujitsu-España S.A. In May of 1991 he joined the University of Málaga as a Profesor Ayudante and got the Ph.D. degree in March 1996 with a dissertation on the design of VLSI fuzzy and neuro-fuzzy controllers using analog and mixed signal techniques. His current research activity focuses on looking for applications and building experimental demonstrators for analog or mixed-signal based neuro-fuzzy controllers as well as the design of these controllers. He is also interested in adaptive algorithms for hardware implementations and in the design of low-power analog circuits.