Using Building Blocks to Design Analog Neuro-Fuzzy Controllers

We present a parallel architecture for fuzzy controllers and a methodology for their realization as analog CMOS chips for low- and medium-precision applications. These chips can be made to learn through the adaptation of electrically controllable parameters guided by a dedicated hardware-compatible learning algorithm. Our designs emphasize simplicity at the circuit level—a prerequisite for increasing processor complexity and operation speed. Examples include a three-input, four-rule controller chip in 1.5-μm CMOS, single-poly, double-metal technology.

Fuzzy sets and fuzzy inference enable us to use insights about local features to predict the behavior of a system, even if its exact mathematical description is unknown or ill-defined. For instance, fuzzy inference can stabilize an inverted pole on a moving cart through statements like “if the pole is falling rapidly to the left, then the cart must move rapidly to the left.” For fuzzy inference, as for a human operator, there is no need for exact formulation of the system dynamics.

In recent years, designers have successfully applied fuzzy inference to control problems in vehicles, robots, motors, power systems, home appliances, and so on, as well as to decision-making systems and image processing. In many of these systems, software on conventional microprocessors can produce fuzzy inference, attaining up to 1-KHz inference speed with 8- to 16-bit resolution. However, systems requiring high-speed inference, reduced power consumption, or smaller dimensions have prompted the development of dedicated hardware.

There are two major classes of analog fuzzy chips: fixed function and adaptive. The former are better suited to applications in which the input-output function is already completely defined at the chip design phase and does not change with operation. However, this is not the situation in most practical cases, where designers do not know the exact function a priori, or where the function must adapt to specific environmental characteristics. Thus, the need arises for combining the inference capabilities of fuzzy systems with the learning capabilities of neural networks, as other authors have discussed. Based on these developments, we present a neuro-fuzzy analog chip architecture, circuit blocks for its realization in VLSI CMOS technology, and hardware-oriented algorithms to adapt its parameters through learning. We emphasize the power consumption and area occupation needed for a given speed. Consequently, analog techniques are better suited for applications in which power consumption, system dimensions, or operation speed takes precedence over accuracy. This is actually the case in most fuzzy-system applications, where accuracy requirements range from 10 percent to 1 percent—accuracy even the least expensive VLSI technologies can provide. Another obvious advantage of analog fuzzy circuits is their simple interface with physical sensors and actuators, that requires no data converters.

There are two major classes of analog fuzzy chips: fixed function and adaptive. The former are better suited to applications in which the input-output function is already completely defined at the chip design phase and does not change with operation. However, this is not the situation in most practical cases, where designers do not know the exact function a priori, or where the function must adapt to specific environmental characteristics. Thus, the need arises for combining the inference capabilities of fuzzy systems with the learning capabilities of neural networks, as other authors have discussed. Based on these developments, we present a neuro-fuzzy analog chip architecture, circuit blocks for its realization in VLSI CMOS technology, and hardware-oriented algorithms to adapt its parameters through learning. We emphasize...
the modularity of the circuits used for adaptability; our design methodology is applicable to both fixed- and adaptive-function chips.

**Chip architecture**

Figure 1 shows our chip architecture—an implementation of Takagi's and Sugeno's singleton fuzzy inference rules. This approach, advantageous for hardware implementation and programming, obtains the output as a weighted linear combination of fuzzy basis functions,

\[
y = f(x) = \sum_{i=1}^{N} y_i w_i(x)
\]

where \( x = (x_1, x_2, ..., x_n)^T \) is the input vector in column format, each \( w_i(x) \) corresponds to a rule, and \( y_i \) is the singleton associated with it. We calculate the basis functions from the input as

\[
w_i(x) = \min_{j=1}^{M} \left[ y_i(x_j), s_1(x_1), ..., s_m(x_m) \right]
\]

For the sake of generality, we have assigned each input in Figure 1 of Takagi and Sugeno's singleton fuzz} inference rules. This advantage is useful for hardware implementation and programming, obtaining the output as a weighted linear combination of fuzzy basis functions. For fixed-chip applications, we calculate these parameters off chip and size the circuits accordingly. For applications that require adaptability, the circuits used in layers 1 and 4 must be programmable, and the chips must be made to learn the required transfer function in situ.

**CMOS premise circuitry**

The premise part of the architecture includes layers 1 and 2. The circuitry of layer 1 operates in transconductance mode, that is, with voltage inputs and current outputs. The use of voltage inputs simplifies the controller interface. In layer 2, current-mode circuits realize the minimum operator more easily than their voltage-mode counterparts.

**Membership function circuitry**

Let us consider the differential amplifier in Figure 2a. Analysis using a square-law model for the MOS transistor in the saturation region obtains the equation for the large-signal transconductance in Figure 2. \( \beta = \beta_1 \frac{W}{L} \) is the transconductance factor in the saturation region, \( \beta_1 \) its normalized value, and \( W \) and \( L \) the width and length of transistors in the differential pairs. The equation in Figure 2 shows that the large-signal transconductance is a sigmoid with saturations at \( +I_c \) and \( -I_c \) like those on the left side of Figure 2c. Thus, cross-cou-
pling two differential pairs as in Figure 2b obtains the bells depicted on the right side of Figure 2c. Of these, we obtain the one at the top by aggregation of the differential output currents of both pairs using Kirchhoff current law (KCL). It ranges between 0 and 2Ig. Aggregating only the positive or the negative output current components of each pair produces the complementary bell-like characteristics shown at the bottom right of Figure 2c. These latter characteristics prove useful in implementation, as we discuss later.

**Multidimensional minimum circuitry.** We use the Maximum and Complement operation to calculate w in practice, so that

\[ w_i = \min(s_{i1}, s_{i2}, \ldots, s_{in}) = \max(s_{i1}, s_{i2}, \ldots, s_{in}) \]

The overbar denotes complement. Since \( s_i \) is a current, we obtain its complement using KCL: \( s_i' = I_i - s_i \), where \( I_i \) is the current associated with logical 1. Similarly, after we calculate \( w \) using the maximum operation, we obtain \( a_i = I_i - w \). The two design problems that arise at this level are how to realize the maximum operator in current domain and how to interface the membership function circuitry and the maximum circuitry.

**Current-mode maximum.** Figure 3a shows a conceptual CMOS maximum circuit based on the winner-take-all of Lazzaro et al., where we have shifted all input currents by \( I_0 \) for convenience. This circuit exploits the ohmic region of MOS transistors. In particular, it is possible to reduce their current by driving them with small \( V_C \) values—as shown in the shaded area of Figure 3.

Note that all bottom transistors in Figure 3a, including output transistor \( M_o \), have the same gate voltage \( V_C \). The largest input current \( s_{max} \) sets its steady-state value. \( V_C \) drives transistor \( M_o \) to draw \( s_{max} \), while their externally applied current \( s_i \) may be smaller than \( s_{max} \). Thus, the gate of each top transistor \( M_i \) becomes an error-sensitive node that detects differences between \( s_i \) and \( s_{max} \). If \( s_i < s_{max} \), the error \( s_i - s_{max} \) is integrated in the gate-to-source capacitor of \( M_i \), causing its gate-source voltage to decrease. Consequently, the drain-source voltage of \( M_i \) decreases until this transistor enters into the ohmic region and the error-current signal becomes null.

The circuit in Figure 3a requires careful design to reduce errors due to channel length modulation if the drains of the output and input transistors are not equipotential. We reduce these errors by adding cascode transistors (similar to \( M_o \)) in series to the input branches, but this strategy renders poor dynamic response. For better dynamic response, we use adaptive biasing to properly set the gate voltage of \( M_o \), \( V_C \).

This adjusts \( V_C \) to equalize the drain-source voltage of \( M_o \), \( V_{ds} \), and that of the input transistor that drives the maximum current. We achieve this through the design in Figure 3b. In this design, the large signal transconductances of transistors \( M_o \) and \( M_i \) control the value of \( V_C \). Thus, we achieve matching between \( V_{ds} \) and \( V_C \) by properly sizing these transistors. We obtain systematic errors below 0.3 percent for input currents of up to 20 μA. In this circuit, as in the others,
percent greater input range. Through optimum design, it may provide higher speed due to the larger mobility of electrons as compared to holes. This alternative, where $I_i = 2I_o$, also reduces area and parasitic penalties associated with obtaining large slope values at the crossovers. Figure 4b also enables us to produce replicas of $x_i$ by using multoutput current mirrors and consequently reduces the circuit complexity for cases where different rules share fuzzy membership functions.

**CMOS consequent circuitry**

The consequent part of Figure 1 includes layers 3, 4, and 5. The circuitry at these three layers operates in current domain, which enables us to fully exploit the functional features of MOS transistors and, consequently, yields extremely simple circuit realizations. The output signal is a current, which we can transform off chip into a voltage through a linear resistor.

The consequent part of Figure 1 includes layers 3, 4, and 5. The circuitry at these three layers operates in current domain, which enables us to fully exploit the functional features of MOS transistors and, consequently, yields extremely simple circuit realizations. The output signal is a current, which we can transform off chip into a voltage through a linear resistor.

**Normalization circuit.** Figure 5a shows a simple CMOS schematic based on Gilbert's bipolar function transistor normalization circuit.
coding the n mirror that replicates \( I_n \) reduces common mode rejection errors. In both cases, we avoid stacked cascade mirrors to preserve range. Instead, we use a cascode transistor at the output branch for p mirrors and a cascode structure with high output voltage swing for the bottom n mirror (Figure 5b).

**Singleton weighting and aggregation.** We achieve singleton weighting using current mirrors with scale factors \( y \). Figure 6a depicts a current mirror with generic transconductors. We can use different transconductor implementations depending on design requirements.

Since interface with the normalization circuit does not impose severe limitations in voltage range, stacked cascode mirrors (shaded area of Figure 6a) offer good DC matching and output resistance. Besides cascoding, splitting the output transistor into multiples of the input transistor reduces channel length modulation errors due to mirror asymmetry. We accomplish aggregation, in current mode by KCL, simply by wiring all rule outputs (Figure 6b). We must also provide the output node with a bias current to eliminate offset created by the normalization circuit, in case it is not eliminated there.

**Hardware-compatible learning**

Figure 7a shows the concept of supervised learning applied to the management of parameter adaptation in a fuzzy engine. We must choose the algorithms used to adapt the parameters of membership functions and the singleton values to guarantee hardware compatibility.

Our choices take advantage of the many similarities between the chip architecture of Figure 1 and the architectures of neural networks. To highlight these similarities, we recast Figure 1 into the two-layer architecture of Figure 8a. Here, each neuron in the input layer has a multidimen-

---

**Figure 6.** Singleton weighting (a), stacked cascade current mirror (inset), and aggregation (b).

**Figure 7.** Concept of supervised, learnable fuzzy engine (a), and performance of the learning algorithm (b). RMSE signifies root mean square error.

**Figure 8.** Two-layer fuzzy architecture (a); one-dimensional projection of input layer nodes for fuzzy and RBFNN systems (b); one-dimensional projection of input layer nodes (Kohonen's layer) for counterpropagation network (c); and measured two-dimensional surface response for a 1.5-\( \mu \)m CMOS analog fuzzy chip (d).
Figure 9. Transconductance as a function of bias current and $B$ for single and compound MOS transistors. Bias current is $I/2$ for all cases (a-d); Implementation of $B$ with transistors (e).
Circuit strategies for adaptability

The circuits we have presented by themselves realize fuzzy controllers with fixed function. However, simple modular modifications enable us to use them for controllable function as well. This is based on the replacement of some MOS transistors with compound transistor structures with electrically controllable characteristics.

**Compound transistors.** A characteristic of the MOS transistor of primary importance for analog design is its operation as a voltage-controlled current source—modeled by transconductance gain $g_m$. We achieve programmability by exercising electrical control on $g_m$. A simple transistor can achieve programmability, as Figure 9a illustrates for n-channel, where we assume operation in the saturation region within strong inversion. The formula included in Figure 9a shows that biasing current $I_c$ controls $g_m$. However, this is inconvenient for fuzzy membership function blocks, where any change of the bias current modifies the electrical value of logical 1.

To overcome this problem, we replace the transistor in Figure 9a with one of the compound transistors in Figures 9b–d. A digital word controls the $g_m$ value of the transistor in Figure 9b. We achieve digital control by switching elementary devices on and off to the signal path under control of digital word $B = (b_0, b_1, \ldots, b_n)$. The sizes of these elementary devices are most typically binary-weighted, giving a quadratic relationship between $g_m$ and the decimal number coded in the digital word $B$. The shape of $g_m$ versus $B$ shown in Figure 9b illustrates the relationship obtained in this situation.

The compound transistors of Figures 9c and 9d provide continuous control of $g_m$. Figure 9c is a series configuration where the bottom transistor cannot operate in the saturation region due to the biasing voltage $B$. Thus, assuming that the top transistor operates in the saturation region, we obtain the equation for $g_m$ included with the figure. The shape of $g_m$ versus $B$ (Figure 9c) illustrates this function. Showing a minimum for $B = 0$, and monotonic growth for positive values of $B$. The exact shape depends on the values of $\beta$ and $\beta_0$. As $\beta_0$ and/or $\beta$ increase, the change rate of $g_m$ with $B$ increases as well.

Now consider the parallel configuration (Figure 9d), with transistors operating in the saturation region. The shape of the transconductance expression is an ellipse in the $g_m$ versus $B$ plane. Actual devices cover only a portion of this ellipse, which includes the point of maximum transconductance at $B = 0$ and exhibits saturation regions for large negative and positive values of $B$. The heavy line in the graph illustrates this, where the exact shape depends again on $\beta_0$ and $\beta$. The saturation value for $B < 0$ is larger than that for $B > 0$ if $\beta_0 > \beta$, and smaller otherwise.

**Membership function programmability.** As we mentioned, the cell in Figure 2b exhibits two characteristics which qualify for practical use: the $i_c$ curve and the $i_e$ curve (see Figure 2c). Both have the same width and center, which are separately controlled by $E_c$ and $E_e$. $2\Delta = E_c - E_e, \ 2E = E_c + E_e$ within the common-mode range of the differential pairs, and with a constraint on minimum width $\Delta_m = (E_c - B_c)^{1/2}$ imposed by the operation of the differential pairs.

The other tunable parameter, the slope at the crossover points, is given by the formula in Figure 9a for the $i_c$ curve, with $S = g_m$. Note that we can modify $S$ on chip by changing $I_c$. However, this forces us to include an additional clamping stage to maintain equality of logic 1 for all fuzzy labels, in spite of the actual value of the bias current for each corresponding differential pair. Consequently, the membership function shapes will be less smooth. Even more important, the correlation between slope and width increases. For simpler design and easier on-chip tuning, all membership functions should have the same bias current. We then control their slope by using compound transistors in the differential pairs. Figure 10 (next page) shows different $i_c = i_e$ shapes produced by the cell in Figure 2b for different compound transistor configurations and different values of $B$. Expressions of the slope as a function of $B$ for the curve $i_c$ coincide with those given in Figure 9.

**Singleton programmability.** As with membership function circuits, using compound transistors obtains a current mirror for which parameter $B$ controls the input-to-output characteristics. Figures 10d–f depict parametric families for three compound transistor configurations. The observed nonlinearities are not problematic if the error signals that guide the learning procedure are measured on the chip.

**Programmability strategies.** The three compound transistors of Figure 9 have the common feature of controlling $g_m$ without changing the bias current. The advantages of a digitally controlled configuration are an easier interface to conventional equipment, lower sensitivity to technological parameters, and simpler design. The disadvantages are larger area and power consumption. The other configurations have less control. Apart from these considerations, we base comparative evaluation of the different strategies for programmability on the following criteria:

- variation range of the adaptive parameter,
- variation range of the control parameter,
- influence of the controlled circuit on common-mode input range, and
- smoothness of the relationship between control parameter and adaptive parameter.

Each compound transistor exhibits pros and cons when contemplated in light of these criteria. The series configuration features large control range and good input range, since the global cut-in voltage equals a simple threshold voltage, $V_T$. On the downside, it displays a low range of adaptive parameters—a negative consequence of the low incremental change of the transconductance with $B$.

On the other hand, the parallel configuration features bet-
Figure 10. Tuning the slope of the membership functions and singletons through compound transistors: parallel transistor (a-d), digitally controlled transistor (b-e), and series transistor (c-f).

The range of adaptive parameter, but worse input range, since the cut-in voltage of the global transistor depends on the control parameter. Its control range is also smaller, and its nonlinearity larger than for series configuration.

Finally, the digital configuration has input range similar to that of the series, thus greater than the parallel configuration. It is also the most flexible implementation in terms of control and adaptive ranges. However, its linearity is smaller. Analog implementation of learning with the adaptive parameters stored in capacitors is also more suitable for previous configurations, which offer an analog interface.

ANALOG FUZZY CONTROLLERS save silicon area and power as compared to their digital counterparts. This is because they exploit the MOS transistor fully to realize the linear and nonlinear operators used for fuzzy inference. A major drawback of analog circuits is limited precision. However, careful modeling of mismatches and the use of sound circuit strategies and design techniques obtain great enough accuracy for many practical applications.

A major issue for successful analog VLSI fuzzy chips is adaptability. This encompasses the interrelated problems of developing proper circuitry and feasible adaptation rules. Our modular solution to the first problem produces simple circuits and, thus, helps keep the intrinsic analog area and power advantages. It is also simple to comprehend for system-level designers. As a counterpart, its linearity is far from perfect and imposes that adaptation parameters be adjusted in situ using error feedback schemes. To that purpose, adaptation rules capable of coping with the parasitics and nonlinearities of the hardware must be developed. This is one of our major current research activities, and runs parallel to the development of behavioral models of the proposed hardware. On the one hand, these models allow us to develop...
and refine adaptation algorithms in a computer simulation environment, without actual chip implementations. On the other hand, it enables us to evaluate the designs prior to chip fabrication.

A major handicap of analog fuzzy techniques is that they are much more difficult to design and far less flexible than their digital counterparts. Consequently, they are not very attractive for system-level designers, or whenever there is a need for rapid prototyping. To alleviate these problems, we are currently extending the techniques presented in this article to the design of a new generation of mixed-signal chips that combine the area and power advantages of analog with the flexibility of digital. The basic objective is to give system-level designers the possibility to cover a large variety of problems through programming, instead of designing a different chip for each application.

References

Fernando Vidal-Verdú is a teaching assistant in the Department of Electronics of the University of Málaga, Spain. His research interests include analog and mixed-signal IC design and neural and fuzzy networks.

Vidal received the Licenciado en Física degree from the University of Seville. He is currently working toward a PhD in hardware implementation of neuro-fuzzy controllers.

Angel Rodríguez-Vázquez is professor of electronics in the Department of Electronics and Electromagnetism at the University of Seville. His research interests are analog VLSI, including neural, fuzzy, and chaotic circuits; mixed-signal IC design; and computer-aided design and modeling of analog integrated circuits.

Rodriguez-Vázquez received the Doctor en Ciencias Físicas degree from the University of Seville, Spain. He has published some 50 journal papers, 90 conference papers, and several book chapters. One of his papers was awarded the 1995 Guillemin-Cauer award of the IEEE Circuits and Systems Society. He is an associate editor of the IEEE Transactions on Circuits and Systems.

Direct questions concerning this article to Angel Rodríguez-Vázquez, Dept. Analog and Mixed-Signal Circuit Design, Centro Nacional de Microelectronica, Universidad de Sevilla, Edificio CIGA, C/Tarfia s/n, 41012-Sevilla, Spain; angel@cnni.us.es.

Reader Interest Survey
Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 165  Medium 166  High 167