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.

Fernando Vidal-Verdú

University of Málaga

Angel Rodríguez-Vázquez

University of Seville

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.[1] 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.[1] In many of these systems, software on conventional microprocessors can produce fuzzy inference, attaining up to 1-Kflips 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.[2]

There are two design approaches to fuzzy inference hardware: ASICs using digital circuits and ASICs using analog circuits, though an exact border between the two technologies is controversial. Digital circuits provide greater accuracy, while analog circuits feature greater speed efficiency for medium- to low-accuracy levels below about 9 bits.[3] (We measure speed efficiency as 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 accuracv requirements range from 10 percent to 1 percent[4]—accuracy even the least expensive VLSI technologies can provide.[5] 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.[6] Thus, the need arises for combining the inference capabilities of fuzzy systems with the learning capabilities of neural networks, as other authors have discussed.[7] 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

0272-1732/95/$04.00 © 1995 IEEE

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.[1] This approach, advantageous for hardware implementation and programming,[4] obtains the output as a weighted linear combination of fuzzy basis functions,

$$y = f(\mathbf{x}) = \sum_{i=1,N} y_i^* w_i^*(\mathbf{x}) \tag{1}$$

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

$$w_i^*(\mathbf{x}) = \frac{\min\left[s_{i1}(x_1), s_{i2}(x_2), \ldots, s_{iM}(x_M)\right]}{\sum_{i=1,N} \min\left[s_{i1}(x_1), s_{i2}(x_2), \ldots, s_{iM}(x_M)\right]} \tag{2}$$

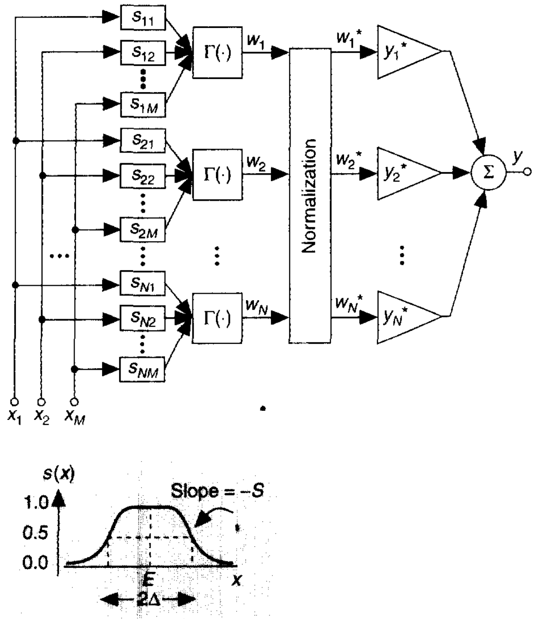


Figure 1. Conceptual architecture of a singleton fuzzy chip. Inset: Bell-like membership function.

Here, $1 \le i \le N$, min is the multidimensional minimum, and $s_{ij}(x_j)$ are membership functions that codify the degrees of matching between each input and its fuzzy labels.[1] For the sake of generality, we have assigned each input in Figure 1 a membership function per inference rule. However, in practical applications, some inputs may have identical membership functions at different rules, thus yielding simpler architectures and circuit implementations.

Figure 1 shows five different types of processing nodes:

- Layer 1. Each node in this layer realizes a nonlinear transformation to evaluate a membership function $s_{ij}(x_j)$, where $1 \le i \le N$, and $1 \le j \le M$.
- Layer 2. Each node here obtains a component of $\mathbf{w} = (w_1, w_2, \ldots, w_N)^T$ as the minimum among the M membership function values associated with the corresponding rule.
- Layer 3. This layer normalizes $\mathbf{w}$ using collective computation to obtain $\mathbf{w}^*$ according to Equation 2.
- Layer 4. Each node in this layer multiplies a component of $\mathbf{w}^*$ by its singleton to obtain $w_i^* y_i^*$.
- Layer 5. This layer contains a single node, which performs the summation in Equation 1.

Consider a given structure determined by the number of membership functions and rules. The transfer function of Figure 1 is parameterized by the vector of singletons $\mathbf{y}^* = (y_1^*, y_2^*, \ldots, y_N^*)^T$ and the vectors of membership functions' centers $\mathbf{E}_i = (E_{i1}, E_{i2}, \ldots, E_{iM})^T$, widths $\Delta_i = (\Delta_{i1}, \Delta_{i2}, \ldots, \Delta_{iM})^T$, and slopes $\mathbf{S}_i = (S_{i1}, S_{i2}, \ldots, S_{iM})^T$. (The inset in Figure 1 shows the shape.) 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. $\beta = \beta_0(W/L)$ is the transconductance factor in the saturation region, β_0 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 $+I_Q$ and $-I_Q$, like those on the left side of Figure 2c. Thus, cross-coupling 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 $2I_Q$. 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_i in practice, so that

$$w_i = \min(s_{i1}, s_{i2}, \ldots, s_{iM}) = \overline{\max}\left(\overline{s_{i1}}, \overline{s_{i2}}, \ldots, \overline{s_{iM}}\right)$$

The overbar denotes complement. Since s_{ij} is a current, we obtain its complement using KCL: $s_{ij} = I_l - \overline{s}_{ij}$, where I_l is the current associated with logical 1. Similarly, after we calculate $\overline{w}_i$ using the maximum operation, we obtain $w_i = I_l - \overline{w}_i$. 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.,[8] where we have shifted all input currents by I_B 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_{DS} 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_G. The largest input current $\overline{s}_{imax}$ sets its steady-state value. V_G drives transistor M_{bj} to draw $\overline{s}_{imax}$, while their externally applied current $\overline{s}_{ij}$ may be smaller than $\overline{s}_{imax}$. Thus, the gate of each top transistor M_{tj} becomes an error-sensitive node that detects differences between $\overline{s}_{ij}$ and $\overline{s}_{imax}$. If $\overline{s}_{ij} < \overline{s}_{imax}$, the error $\overline{s}_{ij} - \overline{s}_{imax}$ is integrated in the gate-to-source capacitor of M_{tj}, causing its gate-source voltage to decrease. Consequently, the drain-source voltage of M_{bj} 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_C) 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_C, V_{ref}. This adjusts V_{ref} to equalize the drain-source voltage of M_O (V_{DO}) 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_{S1} and M_{S2} control the value of V_{ref}. Thus, we achieve matching between V_{Dimax} and V_{DO} 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, fol-

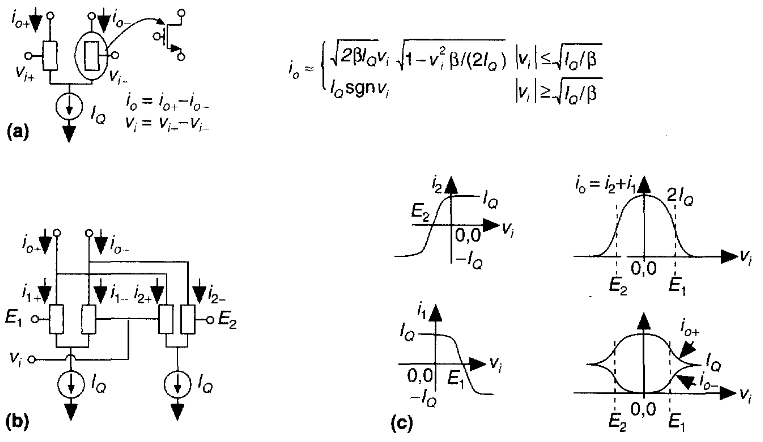


Figure 2. Membership function generation: differential pair as basic cell (a), bell-like membership function circuit (b), and response curves set (c).

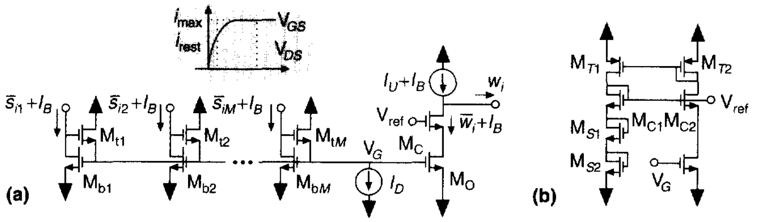


Figure 3. CMOS current-mode maximum/propagate circuit: basic schematic (a) and bias circuit (b).

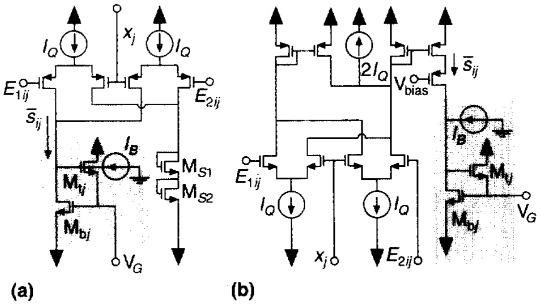


Figure 4. Interface between membership function and minimum circuits (shaded). Alternatives offer smaller silicon area (a) or greater input range (b).

lowing analog layout and sizing guidelines based on the formulation by Pelgrom[5] minimizes mismatching errors due to random variations of technological parameters.

Interface to membership function circuitry. Bear in mind that output i_{o-} of the membership function circuit of Figure 2b already has the shape of a complemented bell. However, using PMOS instead of NMOS transistors in this circuit produces a current that leaves rather than enters the output node. Thus, interface to a minimum block built with NMOS transistors is direct, as shown in Figure 4a. The devices in the shaded area belong to the minimum circuitry, where $I_U = I_Q$.

Figure 4b shows an alternative that also uses NMOS transistors in the membership function differential pairs. Although its cost in silicon area is larger, it features almost 30 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 $I_U = 2I_Q$, also reduces area and parasitic penalties associated with obtaining large slope values at the crossovers. Figure 4b also enables us to produce replicas of $\overline{s}_{ij}$ by using multioutput current mirrors and consequently reduces the circuit complexity for cases where different rules share fuzzy membership functions.

CMOS consequent circuitry

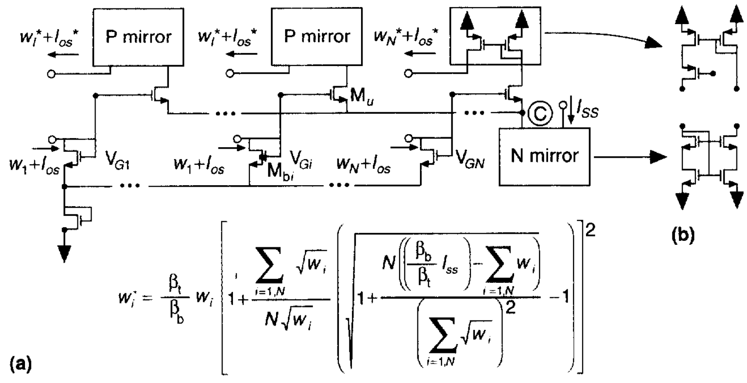
The consequent part of Figure 1 includes layers 3, 4, and 5. The circuitry at these three layers operates in current domain, which enables us to fully exploit the functional features of MOS transistors and, consequently, yields extremely simple circuit realizations. The output signal is a current, which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

Normalization circuit. Figure 5a shows a simple CMOS schematic based on Gilbert's bipolar junction transistor normalization circuit.[9] For convenience, we have shifted all input currents by I_{OS}. Each transistor M_{bi} in the bottom array senses a component w_i of **w** and realizes a nonlinear I-to-V transformation to obtain voltage V_{Gi}. This voltage is applied to the top transistor M_{ti} and transformed into current w_i^*, which follows the equation in Figure 5. β_t and β_b are the transconductance factors of top and bottom transistors.

KCL at node © of the circuit in Figure 5a forces the sum of components of vector **w*** to remain constant and equal to I_{SS}, as required for normalization. Proper design obtains quasi-linear transformation of w_i into w_i^*. However, linearity is not strictly necessary in a neuro-fuzzy system, where nonlinearities are tolerated or corrected through adaptation.

The design in Figure 5a improves the dynamic response by a factor of about four over other strategies.[10] The level-shifting current I_{OS} helps maintain this advantage by decreasing the impedance of input nodes. Proper sizing of the top and bottom transistors eliminates the offset at the output. Nonetheless, speed considerations dictate the use of similar gains for both transistor arrays, producing offset I_{OS}^* also at the output.

Main error sources of the design in Figure 5a are channel length modulation and common mode rejection. We minimize errors due to channel length modulation at the p mirrors by inserting cascode transistors. Similarly, cas-



$$w_i^* = \frac{\beta_t}{\beta_b} w_i \left[1 + \frac{\sum_{i=1,N} \sqrt{w_i}}{N\sqrt{w_i}} \left(\sqrt{1 + \frac{N\left(\left(\frac{\beta_b}{\beta_t} I_{ss}\right) - \sum_{i=1,N} w_i\right)}{\left(\sum_{i=1,N} \sqrt{w_i}\right)^2}} - 1 \right) \right]^2$$

Figure 5. Open-loop CMOS normalization circuit: basic schematic (a) and cascode current mirrors (b).

coding the n mirror that replicates I_{ss} reduces common mode rejection errors. In both cases, we avoid stacked cascode 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_i^*. Figure 6a depicts a current mirror with generic transconductors. We can use different transconductor implementations depending on design requirements.[9] 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.

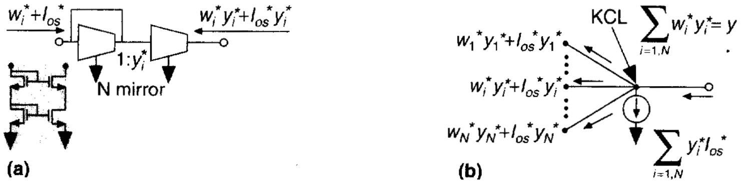

$w_i^*+I_{os}^*$
$w_i^*y_i^*+I_{os}^*y_i^*$
$1:y_i^*$
N mirror
(a)
KCL
$\sum_{i=1,N} w_i^*y_i^*= y$
$w_1^*y_1^*+I_{os}^*y_1^*$
$w_i^*y_i^*+I_{os}^*y_i^*$
$w_N^*y_N^*+I_{os}^*y_N^*$
$\sum_{i=1,N} y_i^*I_{os}^*$
(b)

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

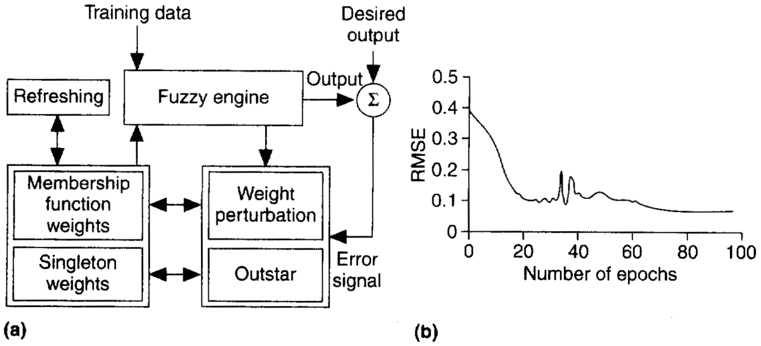


Figure 7. Concept of supervised, learnable fuzzy engine (a), and performance of the learning algorithm (b). RMSE signifies root mean square error.

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-

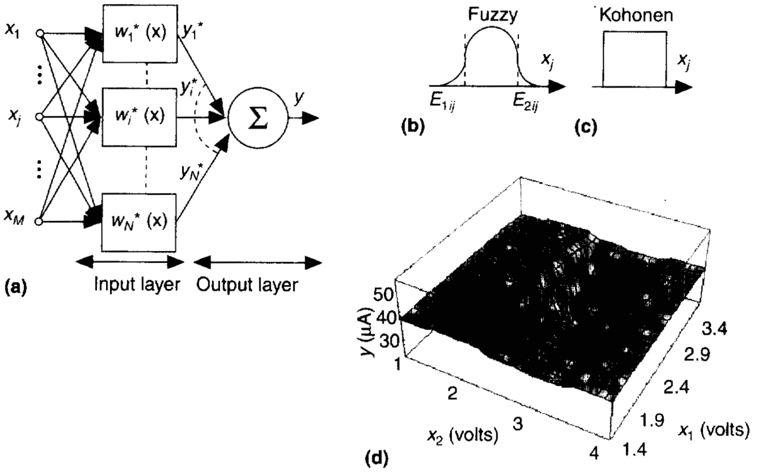


Figure 8. Two-layer fuzzy architecture (a); one-dimensional projection of input layer nodes for fuzzy and RBFNN systems (b); one-dimensional projection of input layer nodes (Kohonen's layer) for counterpropagation network (c); and measured two-dimensional surface response for a 1.5-μm CMOS analog fuzzy chip (d).

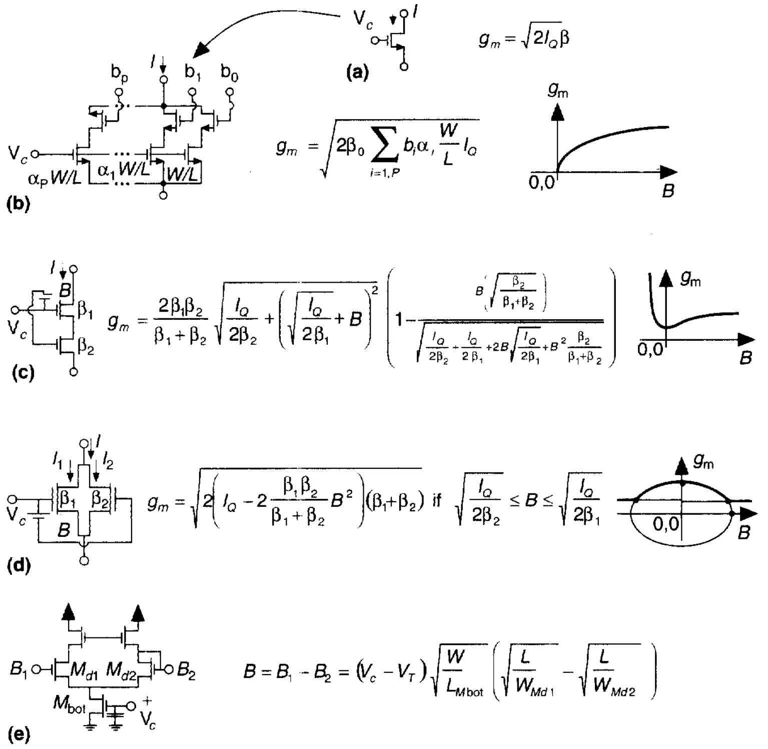

V_c
I
(a)
$g_m = \sqrt{2I_Q\beta}$
b_p
I
b_1
b_0
V_c
$\alpha_P W/L$
$\alpha_1 W/L$
W/L
(b)
$g_m = \sqrt{2\beta_0 \sum_{i=1,P} b_i\alpha_i \frac{W}{L} I_Q}$
g_m
0,0
B
I
B
β_1
V_c
β_2
(c)
$g_m = \frac{2\beta_1\beta_2}{\beta_1+\beta_2}\sqrt{\frac{I_Q}{2\beta_2}+\left(\sqrt{\frac{I_Q}{2\beta_1}}+B\right)^2}\left(1-\frac{B\left(\sqrt{\frac{\beta_2}{\beta_1+\beta_2}}\right)}{\sqrt{\frac{I_Q}{2\beta_2}-\frac{I_Q}{2\beta_1}+2B\sqrt{\frac{I_Q}{2\beta_1}}+B^2\frac{\beta_2}{\beta_1+\beta_2}}}\right)$
g_m
0,0
B
I
I_1
I_2
β_1
β_2
V_c
B
(d)
$g_m = \sqrt{2\left(I_Q - 2\frac{\beta_1\beta_2}{\beta_1+\beta_2}B^2\right)(\beta_1+\beta_2)}$ if $\sqrt{\frac{I_Q}{2\beta_2}} \le B \le \sqrt{\frac{I_Q}{2\beta_1}}$
g_m
0,0
B
B_1
M_{d1}
M_{d2}
B_2
$M_{\rm bot}$
+
V_c
(e)
$B = B_1 - B_2 = (V_c - V_T)\sqrt{\frac{W}{L_{M\rm bot}}}\left(\sqrt{\frac{L}{W_{Md1}}} - \sqrt{\frac{L}{W_{Md2}}}\right)$

Figure 9. Transconductance as a function of bias current and *B* for single and compound MOS transistors. Bias current is $I_Q/2$ for all cases (a-d); Implementation of *B* with transistors (e).

sional, nonlinear transfer function $w_i^*(\mathbf{x})$, and the activation function of the neuron in the output layer is unity.

Geometrically, $w_i^*(\mathbf{x})$ is a multidimensional membership function whose one-dimensional projections are bell-shaped (Figure 8b). They divide the input universe into clusters, shown in Figure 8d. This graph shows measurements taken from a silicon prototype of a three-input, four-rule analog fuzzy controller fabricated in 1.5-μm CMOS, single-poly, n-well technology. The chip uses our design methodology and features 5-Mflips operation with 1-percent accuracy. The figure depicts a two-dimensional projection of the surface response and shows four different clusters, one for each inference rule.

The clustering performed by the fuzzy inference procedure is similar to the role played by basis functions in radial basis function neural networks[11] (RBFNNs), although radial basis functions are not commonly normalized. This leads us to explore learning strategies borrowed from RBFNNs: a clustering algorithm to determine membership functions and an error-correction algorithm for the weights in output layers. This has already been considered at the algorithmic level,[7] using a back-propagation algorithm for the antecedents (layer 1) and least mean squares for the consequents (layer 4). However, back propagation is hard to implement in hardware. Instead, we consider weight perturbation,[12] where we replace derivatives with finite differences and avoid feedback paths by calculating the influence of each parameter on the global error. If ω is the learning parameter, and ζ the global error at output, a change in the value of ω is given by

$$\Delta\omega = \frac{-\eta\left[\zeta(\omega) - \zeta(\omega + \text{pert})\right]}{\text{pert}} \quad \textbf{(3)}$$

where pert is a small perturbation, η is the learning rate, and both are constant. Note that weight update hardware evaluates the error with perturbed and unperturbed weight and then multiplies by a constant.

We use this strategy for the membership functions. We exploit the similarities of singleton fuzzy inference with the counterpropagation network. The similarities become evident when we use "crisp" rather than fuzzy sets. In this case, Figure 8c depicts the one-dimensional projections of the membership functions, which are similar to a trained counterpropagation network with Kohonen input nodes and Grossberg output node. Based on this, our learning algorithm uses the outstar rule,

$$y^*_{i_{\rm new}} = y^*_{i_{\rm old}} + \mu\left[T - y(\mathbf{x})\right] \quad \textbf{(4)}$$

where *T* is the target output, μ is the learning rate, and y_i^* is the singleton whose rule antecedent is maximum, that is, $w_i^*(\mathbf{x}) = \max[w_1^*(\mathbf{x}), w_2^*(\mathbf{x}), \ldots, w_N^*(\mathbf{x})]$. Figure 7b illustrated the performance of our learning algorithm. We teach the multidimensional function $y = 2 + \sin(\pi x)\sin(\pi y)$ to a nine-rule controller by showing it 36 input-output data pairs in the interval [0, 1] × [0, 1]. We initialize the system with membership functions uniformly distributed along the universe of discourse, with all singletons equal to 2. Figure 7b shows the root mean square error for our learning rule, with pert = 0.05, η = 0.005 (see Equation 3), and μ = 0.01 (see Equation 4).

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 composite 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_Q 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 $B = (b_0, b_1, \ldots, b_p)$. 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 β_1 and β_2. As β_1 and/or β_2 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 β_1 and β_2. The saturation value for $B < 0$ is larger than that for $B > 0$ if $\beta_2 > \beta_1$, and smaller otherwise.

Membership function programmability. As we mentioned, the cell in Figure 2b exhibits two characteristics which qualify for practical use: the i_o curve and the i_{o-} curve (see Figure 2c). Both have the same width and center, which are separately controlled by E_1 and E_2, $2\Delta = E_2 - E_1$, $2E = E_2 + E_1$ within the common-mode range of the differential pairs, and with a constraint on minimum width $\Delta_{\min} = (I_Q/\beta)^{1/2}$ 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_o curve, with $S = g_m$. Note that we can modify S on chip by changing I_Q. However, this forces us to include an additional clamping stage to maintain equality of logical 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 $i_{o+} - I_Q$ 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_o 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-

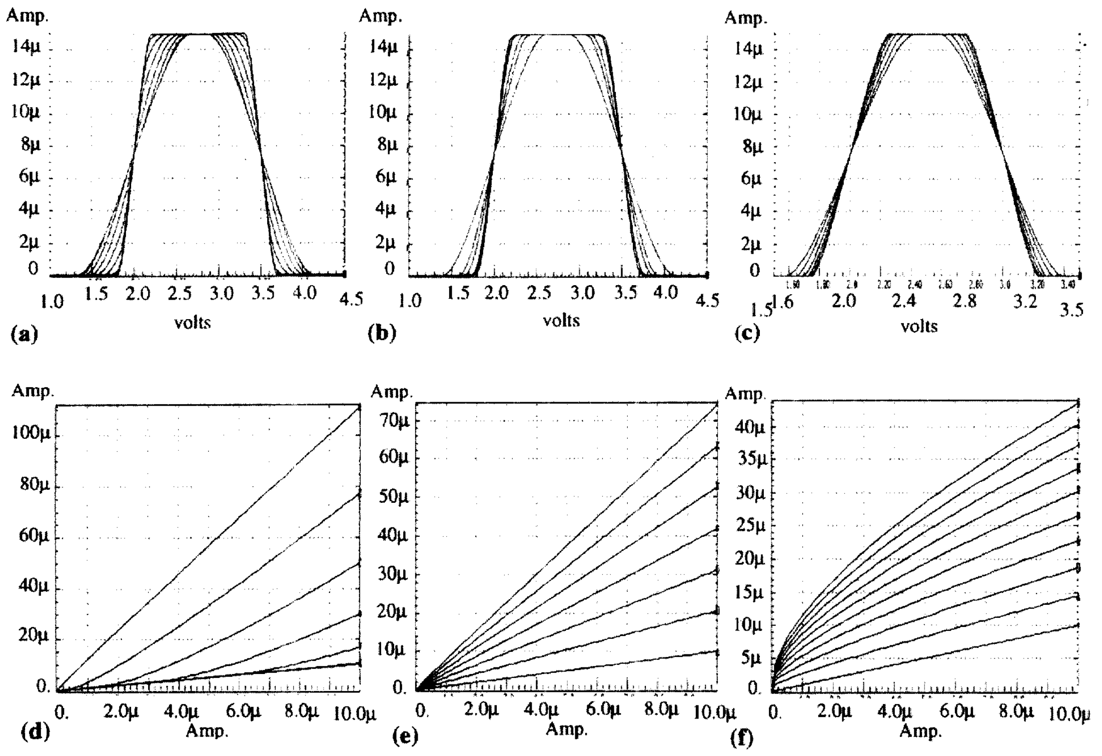


Figure 10. Tuning the slope of the membership functions and singletons through compound transistors: parallel transistor (a-d), digitally controlled transistor (b-e), and series transistor (c-f).

ter range of adaptive parameter, but worse input range, since the cut-in voltage of the global transconductor depends on control parameter *B*. 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.

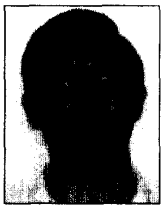
ANALOG FUZZY CONTROLLERS save silicon area and power as compared to their digital counterparts. This is because they exploit the MOS transistor fully to realize the linear and nonlinear operators used for fuzzy inference. A major drawback of analog circuits is limited precision. However, careful modeling 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, 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.

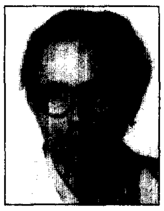
References

1. R.J. Marks, ed., *Fuzzy Logic Technologies and Applications,* IEEE, Piscataway, N.J., 1994.
2. K. Namakura et al., "Fuzzy Inference and Fuzzy Inference Processor," *IEEE Micro,* Vol. 13, No. 5, Oct. 1993, pp. 37-48.
3. E.A. Vittoz, "Future of Analog in the VLSI Environment," *Proc. IEEE Int'l Symp. Circuits and Systems,* IEEE, Piscataway, N.J., 1990, pp. 1372-1375.
4. T. Yamakawa, "A Fuzzy Inference Engine in Nonlinear Analog Mode and Its Application to a Fuzzy Logic Control," *IEEE Trans. Neural Networks,* Vol. 4, No. 5, May 1993, pp. 496-522.
5. M.J.M. Pelgrom et al., "Matching Properties of MOS Transistors," *IEEE J. Solid-State Circuits,* Vol. 24, No. 5, Oct. 1989, pp. 1433-1440.
6. H. Takagi, "Applications of Neural Networks and Fuzzy Logic to Consumer Products," in *Fuzzy Logic Technologies and Applications,* IEEE, Piscataway, N.J., 1994, pp. 8-12.
7. J.S.R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Trans. Systems, Man, and Cybernetics,* Vol. 23, No. 3, May 1993, pp. 665-685.
8. J. Lazzaro et al., "Winner-Take-All Networks of O(n) Complexity," in *Advances in Neural Information Processing Systems,* Vol. 1, D.S. Touretzky, ed., Morgan Kaufmann, San Francisco, 1989, pp. 703-711.
9. C. Toumazou et al., eds., *Analog IC Design: The Current Mode Approach,* Peter Peregrinus, London, 1990.
10. C. Mead, *Analog VLSI and Neural Systems,* Addison-Wesley, Reading, Mass., 1989.
11. J.S.R. Jang and C.T. Sun, "Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems," *IEEE Trans. Neural Networks,* Vol. 4, No. 1, Jan. 1993, pp. 156-159.
12. M. Jabri and B. Flower, "Weight Perturbation: An Optimal Architecture and Learning Technique for Analog VLSI Feedforward and Recurrent Multilayer Networks," *IEEE Trans. Neural Networks,* Vol. 3, No. 1, Jan. 1992, pp. 154-157.



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.

Rodríguez-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 CICA, C/Tarfia s/n, 41012-Sevilla, Spain; angel@cnm.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