A CURRENT-MODE MEMORY AND COMPUTING BLOCK FOR ADVANCED FUZZY CHIPS

S. Sánchez Solano, I. Baturone, A. Barriga, J. L. Huertas

Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Avda. Reina Mercedes s/n, (Edif. CICA)
E-41012, Sevilla, Spain

Sixth International Fuzzy Systems Association World Congress (IFSA’95),
A CURRENT-MODE MEMORY AND COMPUTING BLOCK FOR ADVANCED FUZZY CHIPS

Santiago SÁNCHEZ-SOLANO, Iluminada BATURONE, Angel BARRIGA, José Luis HUERTAS

Centro Nacional de Microelecetrónica. Avda. Reina Mercedes s/n, 41012-Sevilla (Spain)

Abstract. This paper presents a versatile current-mode circuit which combines the advantages of analog and digital techniques to perform long-term memory and computing functions. It is based on current-mode algorithm data converters and standard semistatic latches. Hspice simulations are included to illustrate its behavior. The design of the A/D subcell has been improved by using current injection techniques and enhanced current comparators. The cell can be employed as a building block to implement programmable fuzzy chips, introduce learning in so-called adaptive neuro-fuzzy chips, or develop fuzzy sequential processors as an extension of ordinary binary computers.

1. Introduction

Practical applications of fuzzy logic to control processes demand hardware implementations with real-time capability and small-area requirements. Analog circuits are well-suited for fuzzy controllers; in particular current-mode analog circuits are very efficient for implementing basic operations in a fuzzy system, such as sum, bounded difference, or minimum/maximum. The low number of transistors employed and the reduced voltage swings on nodes make these circuits very small and capable of operating at low power and high speed [1-5].

To date, most analog fuzzy chips proposed in literature [5-7] implement single-stage inference mechanisms where input and output fuzzy sets, as well as the rule base, do not change. They are fixed prior to fabrication or via programming. On the other hand, a memory element is indispensable to implement advanced fuzzy chips capable of dynamical changes. Programming ease and efficient storage mechanism are then essential to implement programmable or advanced fuzzy chips. While these requirements are difficult to attain with analog circuits, the problem is easily solved through digital circuitry.

In this paper we present a versatile building block which combines the advantages of analog and digital techniques. It is based on current-mode data converters and standard semistatic latches. Hspice simulations are included to illustrate its behavior. The design of the A/D subcell has been improved by using current injection techniques and enhanced current comparators. The cell can be employed as a building block to implement programmable fuzzy chips, introduce learning in so-called adaptive neuro-fuzzy chips, or develop fuzzy sequential processors as an extension of ordinary binary computers.

2. Functionality of The Proposed Cell

The proposed circuit consists of a combination of a latch register and data converters, as depicted in Fig. 1. A control signal (not shown in the figure for simplicity) enables loading the internal storage element either with data coming from the digital input $w_i$ or with the result of converting the analog input $I_i$ through the A/D converter. The cell also provides two different output: $w_o$ is the binary word stored in the register and $I_o$ corresponds to its quantized version generated by the D/A converter. Finally, the reference currents of the A/D and D/A data conversion processes, $I_a$ and $I_d$ respectively, can also be used as analog input.

When $I_i$ and $I_a$ are used as terminals and the two reference currents $I_a$ and $I_d$ are equal, the quantized value of the analog input ($I_o=\text{quant}(I_i)$) can be restored, so that the circuit performs as a long-term current-mode analog memory element.

On the other hand, if $I_i$ and $I_a$ are used as analog input, the output digital word of the cell, $w_o$, represents an analog division:

$$w_o = \sum_{i=1}^{n} b_i 2^{-i} = \frac{I_i}{I_a} \tag{1}$$

where $b_i$ are the bits obtained by the A/D converter.

If the digital word, $w_i$, is given as an input, and the signal $I_d$ is another input, the circuit can be used to implement a programmable multiplier, resulting in the following analog output signal:

Fig. 1: Scheme of the proposed circuit.
Finally, combining (1) and (2) obtains a current-mode multiplier/divider:

\[ I_o = I_d \left( \sum_{i=1}^{n} b_i 2^{-i} \right) = I_d w_i \quad \text{ (2)} \]

3. Circuit Description and Operation

Unlike their voltage-mode counterparts, current-mode data converters, enable very efficient implementation of the proposed circuit. The design can be further improved by employing A/D conversion methods which do not require external control signals (namely clock signals). Among these, the most commonly employed are flash or linear conversion, successive approximation, and algorithmic type. We have chosen algorithmic conversion because it requires neither as many comparators as the flash type nor as many reference currents as the successive approximation type. Moreover, its modularity (the input signal flows through a cascade of cells) is a very appealing feature since enable very simple layout [8-10]. Although most reported algorithmic A/D converters implement the multiplying algorithmic technique, the dividing technique illustrated in Fig. 2 is more convenient for our application.

The complete circuit consists of a cascade of N identical alternating cells (named even and odd bit cells, as in [10]), being N the resolution obtained. Each bit cell contains an A/D subcell, a standard semistatic LATCH subcell for local information storage or external digital programmability, and a D/A subcell. Even and odd bit cells are depicted in Fig. 3.

Circuit performance is governed by current mirrors and current comparators operation. Regarding accuracy, the limit is imposed by mismatching in the current mirrors due to systematic and random errors. For an N-bit circuit the absolute current error in the A/D converter should be at or below \( I_d/2^{n+1} \), and at or below \( I_d/2^{n+1} \) in the D/A converter. This means that a wide dynamic range will permit higher errors. As discussed in [8], random errors are caused mainly by threshold voltage variations between ideally identical transistors, and can be reduced by employing large devices. Systematic errors appear to be the most influential. On one hand they are caused by a difference between the input and output voltages of the current mirror (\( V_o - V_{in} \)), and on the other, by its finite output resistance. Among the great variety of techniques to attenuate this systematic error we have opted for current injection, as proposed in [10], to reduce the difference (\( V_o - V_{in} \)). The required modification of the dividing algorithm is simpler than the multiplier one. A bias current (\( I_{bias} \)) is added to the input current \( I_i \) and only one transistor is added to each cell to change the reference level \( I_{ref} \) to \( (I_{ref}/2 + I_{bias}) \). Resolutions of up to 6 bits are achieved with simple current mirrors of moderated long channels (L=6.4 µm in a 2.4 µm CMOS technology).

Regarding operation speed, the performance depends on the transient response of the current mirrors (which is improved via the current injection technique) and on the settling time of the current comparator. Previous designs [8,10] resort to the use of a resistive input comparator [11]. We have employed the enhanced capacitive input comparator proposed in [12], which exhibits shorter propagation delay.
delays.
Fig. 4 shows Hspice simulations of a complete circuit achieving a resolution of 4 bits. Device ratios are 12.8\mu m/6.4\mu m and 19.2\mu m/6.4\mu m in the transistors of n-type and p-type current mirrors, respectively. The two reference currents have the same value, 16\mu A, and the injection current I_{bias} is 8\mu A. Fig. 4a shows the circuit performance as a quantizer for an analog input current (represented by the dashed line) which continuously varies from 0 to 15.5\mu A. Fig. 4b shows the digital word obtained. The average power consumption for a voltage supply of 5v is 1.45mw. The maximum delay, which appears in the transition 0111 to 1000, has been measured for a pulse in I_i changing from 7.5\mu A to 8.5\mu A. It is less than 50ns to obtain the digital word, and less than 100ns for the analog output.

4. Architecture of Advanced Fuzzy Chips
The storage and processing features of the proposed cell are presented in this section through its use in fuzzy chips whose parameters can be easily programmed or dynamically modified.

Fig. 5 shows the block diagram of a fuzzy chip implementing a singleton or simplified inference mechanism. For the sake of simplicity, symmetrical triangle membership functions are chosen to describe the antecedent fuzzy sets. Hence, the parameters defining the k-th rule are: c_k (singleton consequent), I_{ck} (position of the antecedent fuzzy set), and m_k (slope of the triangle membership function). Each antecedent membership function circuit (MFC) follows the scheme proposed in [13], adding an inverter to improve the behavior of the switch current [14]. One of the proposed cells fixes I_{ck}, the other fixes m_k and also performs the required multiplication. A multi-input max circuit [4] computes the activation degree (h_k) of each rule. Multiplication and definition of c_k is achieved by another of the proposed cell. Finally, the
cell is also used to calculate the global output \( \frac{\sum h_k c_k}{\sum h_k} \). The output is provided in digital and analog formats, which improves the capability of the fuzzy chip to interact with other circuits.

We distinguish three steps of operation in these advanced fuzzy chips: initial programming, inference, and parameter update. Programming is attained via a standard digital bus, enabling easy parameter definition \( \{ c_k, I_k, m_k \} \) for each rule. In the inference step, the D/A subcells of the proposed building block are employed to provide the parameters and perform multiplications, and the A/D subcell of the final block is employed as a divider. In the update step, the D/A subcells are employed to store the new parameter values (with long-term memory capability).

References


