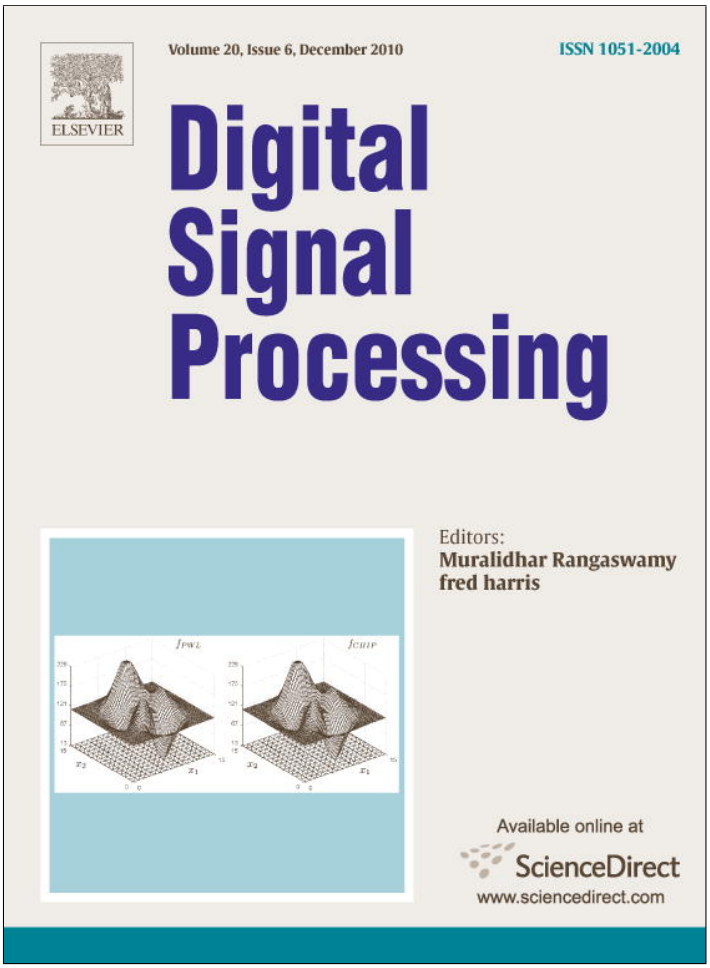

Digital Signal Processing 20 (2010) 1723–1732

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Integrated circuit implementation of multi-dimensional piecewise-linear functions

Martin Di Federico [a], Tomaso Poggi [b], Pedro Julián [a], Marco Storace [b,*]

[a] Universidad Nacional del Sur, Av. Alem 1353, Bahía Blanca, Argentina
[b] University of Genoa, Biophysical and Electronic Engineering Department, Via Opera Pia 11a, I-16145 Genova, Italy

ARTICLE INFO

Article history:
Available online 19 February 2010

Keywords:
Integrated circuit
Nonlinear circuit
Piecewise-linear
Embedded

ABSTRACT

In this paper we present an integrated circuit implementing piecewise-linear (PWL) functions with three inputs, where each input can be either analog or digital. The PWL function to be implemented can be chosen by properly storing a set of coefficients in a 4 kB external memory. Experimental results are shown that demonstrate the circuit working up to 50 MHz with a maximum power consumption of 3.7 mW. Measurements corresponding to both static and time-varying inputs are provided and discussed.

1. Introduction

The availability of dedicated integrated circuit (IC) implementations of n-variate functions is useful for many applications where nonlinearities are needed but not at the expense of a DSP board or a microcontroller. In some applications in sensor networks, low-power small-size units might be used for information compression and estimation in the individual nodes [1]. In other applications, like communications [2], control [3], and circuits mimicking biologically plausible neural networks [4], embedded real-time circuit emulation of nonlinear dynamical systems is desired. A nonlinear calculation processor that can operate at high speeds can be successfully exploited also in applications like signal generation [5,6], pre-distortion in amplifiers and A/D converters [7], compensation of communication channels [8], among others.

In recent papers [9,10] two different (analog and mixed-signal, respectively) architectures related to a PWL approximation technique [11–13] have been proposed. Both architectures realize PWL functions defined over compact domains partitioned into simplices, and each PWL function is linear over any simplex and is expressed as a weighted sum of PWL basis functions. Each architecture is related to a particular basis of functions.

In this paper we show experimental results of a digital circuit integrated in standard CMOS 0.5 μm technology, preliminarily presented in [14]. The circuit is based on the architecture [10] and implements 3-variate PWL functions. The core of the chip and its interface are digital, but a 3-channel 8-bit A/D converter allows the chip to be used with analog input signals, if required. Since the A/D conversion can be viewed as a "boundary" operation with respect to the PWL function evaluation, this paper focuses on the PWL digital block, which is the "core" of the chip. The circuit realization of the PWL function is very compact, and only requires a 4-bit counter, three 8-bit registers, a comparator and a 12-bit adder. Therefore, the proposed circuit can be used at high speeds with relatively low power consumption. The implemented PWL function is also programmable, and the parameters are stored in an external 4 kB memory.

Two sets of measurements are included in this paper. The first series of measurements correspond to static inputs. In this case, the experimental results demonstrate that the circuit represents with no error (except for the quantization error due to the digital implementation) the PWL function stored in the external memory. The second series of measurements correspond

* Corresponding author.
E-mail addresses: pjulian@ieee.org (P. Julián), marco.storace@unige.it (M. Storace).

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2010.02.007

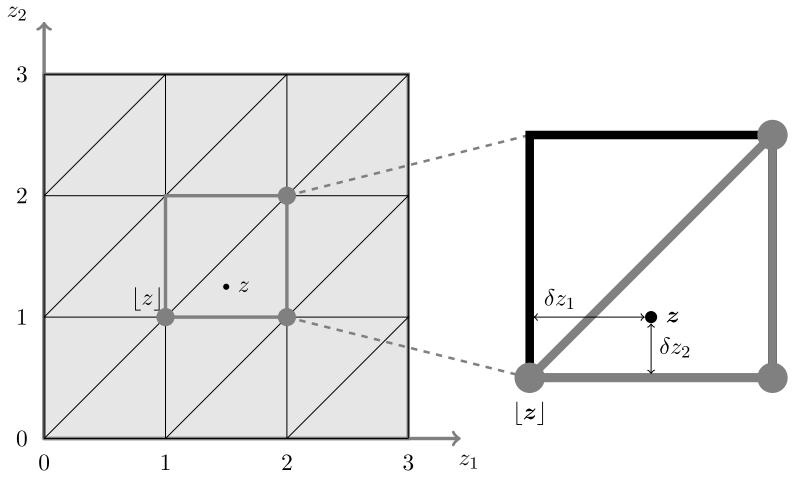

Fig. 1. Example of simplicial partition.

to time-varying inputs in an attempt to evaluate the maximum processing speed. The circuit has been successfully tested up to a clock frequency of 50 MHz showing, for this case, a maximum power consumption of 3.7 mW. Based on this, if analog inputs are used, the maximum frequency content in their spectrum turns out to be limited by about one fortieth of the clock frequency, to ensure a correct behavior. If digital input are used, the bandwidth limit is about one twentieth of the clock frequency.

The paper is organized as follows. The theory foundations of the proposed architecture are summarized in Section 2. The chip architecture is described in Section 3 and the testing results are shown in Section 4.

2. Basic elements

The mixed-signal circuit architecture proposed in [15,16] provides a simple PWL input-output relationship $f(\boldsymbol{x})$ by performing a weighted sum of the so-called α-functions [10]. Each α-function is of a local nature, since it is different from 0 only over a reduced number of simplices of the domain. As a consequence, the value of $f(\boldsymbol{x})$ can be obtained, for any n-dimensional input vector $\boldsymbol{x}$, by combining a limited subset of the basis functions weighted by the corresponding coefficients. Then all basis functions perform basically the same operation and the difference between two basis functions is that they operate over two different regions of the domain. Therefore, $f(\boldsymbol{x})$ can be evaluated by using only one function circuit block and an algorithm to shift the inputs. For every input point $\boldsymbol{x}$, all nonzero basis functions need to be evaluated, weighted and added, but they are at most $n+1$ (that is, linear with respect to the domain dimension), so that the implementation is still remarkably efficient. This principle has been exploited in [17] to derive an electronic implementation for an image processing CNN based on PWL coupled elements. In [10], this implementation has been generalized, by exploiting the technique proposed in [18] to automatically find the simplex containing a given input $\boldsymbol{x}$. The approach has been experimentally tested on an FPGA in [19], using a rank extractor.

In this paper too, the PWL function f is obtained as a linear combination of a set of α-functions. The function is defined over an n-dimensional compact domain $\hat{S} \subset \mathbb{R}^n$, i.e., $f : \hat{S} \to \mathbb{R}$, where $\hat{S}$ is a hyperrectangle (rectangle, if $n = 2$) in the form of:

$$\hat{S} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \colon\ a_i \leqslant x_i \leqslant b_i,\ i = 1, \ldots, n \right\} \tag{1}$$

Since the generic variables x_i have to be coded by electric variables, they need conditioning. In the following, we shall suppose that the input signals are already scaled and transformed into digital values. In the considered architecture, the domain $\hat{S}$ is scaled into (see [10])

$$S = \left\{ \boldsymbol{z} \in \mathbb{R}^n \colon\ 0 \leqslant z_i \leqslant m_i,\ i = 1, \ldots, n \right\} \tag{2}$$

through a linear transformation $\boldsymbol{z} = T(\boldsymbol{x})$.

We will assume that every component z_i of $\boldsymbol{z}$ is represented with $p+q$ bits, p for the integer part, $\lfloor z_i \rfloor$, and q for the decimal part, δz_i. Every dimensional component of the domain S is partitioned into $m = 2^p - 1$ subintervals of unitary length, that divide the whole space in hypercubes. Each hypercube contains $n!$ non-overlapping hypertriangular (triangular, if $n = 2$) regions called simplices and f is linear over each simplex. Summarizing, S is partitioned (*simplicial partition*) into m^n hypercubes (i.e. $n! m^n$ simplices) and contains $N = (m+1)^n = 2^{pn}$ vertices $\boldsymbol{v}_k$. We point out that N is the number of α-functions whose linear combination provides f. Fig. 1 shows an example of partition with $m = 3$.

Once the scaled simplicial domain is defined, the α-basis is directly defined as well. Indeed, the k-th α-function is PWL, holds the value 1 at the vertex $\boldsymbol{v}_k$ and the value 0 at all the other vertices. An example for a two-dimensional domain is shown in Fig. 2. Given the choice of the basis function, the coefficients c_k, coding the shape of a given PWL function f,

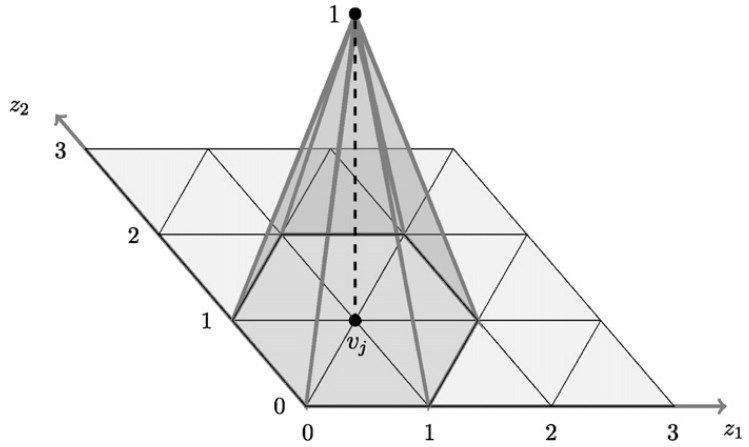

Fig. 2. Example of α-function, centered in the vertex $(1, 1)$, for the domain in Fig. 1.

are the values of f at the vertices $\boldsymbol{v}_k$ of its simplicial domain. They can be obtained, for instance, by applying optimization procedures to a regression set of samples of a function to be approximated [12,13,20]. In this paper we shall assume that the coefficients are already available and we will not focus on approximation problems.

The output function f is conditioned in turn, then also the codomain is scaled in a proper range $[0, f_{MAX}]$. To sum up, we shall focus on PWL functions $f : S \to [0, f_{MAX}]$, expressed as linear combinations (through coefficients c_k) of N α-functions.

As stated before, for a given input $\boldsymbol{z}$, only the $n+1$ α-functions corresponding to the vertices of the simplex containing $\boldsymbol{z}$ are involved in the computation of $f(\boldsymbol{z})$. In other words, the value of $f(\boldsymbol{z})$ can be calculated as a linear interpolation of the f values, i.e. by linearly interpolating a subset of $n+1$ coefficients c_j at the vertices of the simplex containing $\boldsymbol{z}$. This can be expressed by a sum extended to the $n+1$ indices (collected into the set $\Im_z$)

$$f(\boldsymbol{z}) = \frac{\sum_{j\in\Im_z} \mu_j c_j}{\sum_{j\in\Im_z} \mu_j} \tag{3}$$

where the interpolation weights μ_j depend on $\boldsymbol{\delta z}$ [10]. The components of $\boldsymbol{\delta z}$ are first reordered from the larger to the smaller: $\hat{\delta}_1 = \max_i\{\delta_i\} > \hat{\delta}_2 > \cdots > \hat{\delta}_n = \min_i\{\delta_i\}$. Since the decimal parts of $\boldsymbol{z}$ are coded with q bits, the $\hat{\delta}_i$'s range from 0 to $2^q - 1$. Then, the weights μ_j are calculated as simple differences: $\mu_0 = 2^q - \hat{\delta}_1$, $\mu_2 = \hat{\delta}_1 - \hat{\delta}_2, \ldots, \mu_{n-1} = \hat{\delta}_{n-1} - \hat{\delta}_n$, $\mu_n = \hat{\delta}_n$. As a consequence, the μ_j are integers, $0 \leqslant \mu_j \leqslant 2^q$, then $0 \leqslant \frac{\mu_j}{2^q} \leqslant 1$, $\sum_{j=0}^{n} \mu_j = 2^q$ and

$$f(\boldsymbol{z}) = \frac{1}{2^q} \sum_{j\in\Im_z} \left(\sum_{i=1}^{\mu_j} c_j \right) \tag{4}$$

Fig. 1 shows a two-dimensional domain partitioned into $m^2 = 3^2 = 9$ squares. Given a point $\boldsymbol{z}$, one can easily find the square that contains it, since it is unambiguously characterized by $\lfloor \boldsymbol{z} \rfloor$, whereas the point position within the square is coded by $\boldsymbol{\delta z}$. The vertices with an overlapping dot correspond to the set of coefficients $\{c_j\}_{\Im_z}$ needed to compute $f(\boldsymbol{z})$.

Summing up, in the proposed architecture, a circuit realization of a PWL function requires three elements (besides the conditioning block):

1. a method to find, for any given $\boldsymbol{z}$, the set $\Im_z$ of the $n+1$ indices of the significant α-functions and the coefficients μ_j giving $\boldsymbol{z}$ as a weighted sum of the $n+1$ vertices $\boldsymbol{v}_j$ (with $j \in \Im_z$);
2. a memory where the N c_k coefficients are stored;
3. a circuit block performing the sum (4).

3. Chip architecture

3.1. Description

The proposed IC evaluates a PWL function f_{PWL} by performing a weighted sum of the memory values, as explained in [10]. The scaled domain S is partitioned with 16 vertices along each axis, i.e., $m = 15$. The output of the IC is a digital word with 8-bit precision, then $f_{MAX} = 255$. In the present version of the IC, the memory was left outside in order to reduce the silicon occupation.

There are two alternatives to load the input values into the chip. The first alternative is to present three analog values at three input pins. The second alternative is to set the digital values by clocking the internal counter and provide externally the latch signals. In this way, just three pins are used in both cases to load the input values. This strategy was decided based

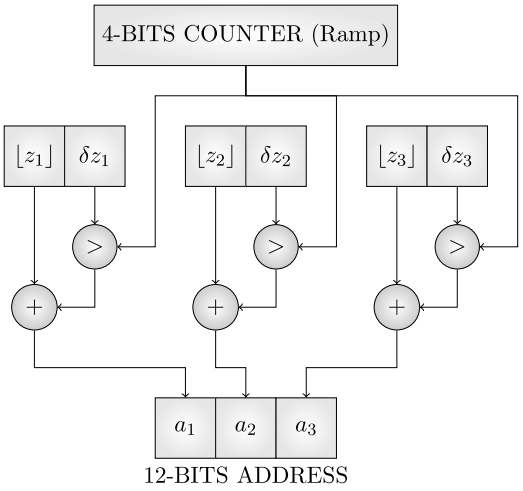

Fig. 3. Address Generator.

on robustness, on the reduced number of available package pins, and on the several testing outputs which were necessary to measure, i.e., mainly for testing purposes. In the following sections, it will become clear that this type of input acquirement is slow with respect to the PWL core, so that other input managing approaches should be used if high-speed processing is needed. The internal counter is also used for the PWL core calculation, therefore, this conversion scheme turns out to be very efficient from the area viewpoint.

Whether the inputs are digital or analog, they are internally stored in 8-bit registers. The p most significant bits of the inputs represent the integer part $\lfloor \boldsymbol{z} \rfloor$ of $\boldsymbol{z}$ and are used to select the simplex the input belongs to; the q less significant bits represent the decimal part $\boldsymbol{\delta z}$ of $\boldsymbol{z}$. In our architecture we have chosen $p = 4$ and $q = 4$, in order to obtain a good trade-off between the number of subdivisions along each dimensional component of the domain (equal to $2^p - 1 = 15$) and the accuracy in the representation of the decimal part. This is a reasonable choice whenever the functions to be implemented are not too wrinkled. The $N = 4096$ weighting coefficients c_k are stored in the 4 kB external memory, which is addressed with a 12-bit word. The whole circuit is timed by a clock signal CK.

The value of f at each scaled input $\boldsymbol{z}$ is the sum (4) of $n+1 = 4$ parameter values. Each coefficient c_j ($j \in \Im_z$) is retrieved from the memory and summed and accumulated for a number of clock cycles equal to the corresponding weight μ_j. As stated before, with this approach it is possible to avoid the integration of multipliers saving area and power consumption. As depicted in Fig. 3, the four addresses to the memory positions where the coefficients c_j ($j \in \Im_z$) are stored are obtained by comparing the values of the digital ramp (counter) with $\boldsymbol{\delta z}$, i.e. with the four less significant bits (LSB) of the 8-bit inputs. This ramp is implemented with the four LSB of the digital counter. Each 12-bit address is obtained by juxtaposing $n = 3$ 4-bit strings a_i. The i-th string is equal to $\lfloor z_i \rfloor$, the four most significant bits (MSB) of the i-th input, if the counter count is greater than the four LSB of the input; otherwise, the i-th string is $\lfloor z_i \rfloor + 1$, the value of the four MSB of the input plus one. The comparison between the counter and each register is done using a digital comparator.

Each address is calculated by a block called Address Generator, and the sum is done with a 12-bit adder. The sum (4) is obtained with no additional effort, since the memory position of c_j is addressed by the Address Generator a number of clock cycles proportional to μ_j. The division by 2^q is performed by shifting the result $q = 4$ bit to the right, i.e. by forwarding to the output only the eight MSB in the accumulator. Accordingly, only the 12-bit adder is necessary to perform the whole sum.

3.2. Architecture

The IC has an analog block and a digital block; both are powered up from different sources to allow them working and being tested separately as shown in Fig. 4. Therefore, a digital signal can be used by latching the values in the registers instead of using the A/D converters. As was mentioned before, this last alternative was used to obtain the experimental results of the IC.

As depicted in Fig. 5, the chip has three different states called NOP, Acquiring and Processing, which are coded with two registers. In the NOP state, the I/O bus works as an output bus and shows the value of the function calculated previously. When the Start Processing (SP) input is "1", the state machine (FSM) goes to the Acquiring state, to store the input in the internal registers. The FSM stays in this state for 256 clock cycles and after that, it goes into the Processing state. In the Processing state the I/O bus works as an input bus connected to the external RAM. In this state the chip performs the 16 additions reading the PWL parameter values from the external memory.

In order to produce the sum, necessary to obtain the value of f_{PWL}, the adder adds successively the sixteen (8-bit) values from the memory and the result is divided by sixteen. In order to add sixteen values of 8 bits, a 12-bit adder is needed; the divide-by-16 operation is easily done by taking only the 8 most significant bits. The 12-bit adder has 8 inputs, so that the

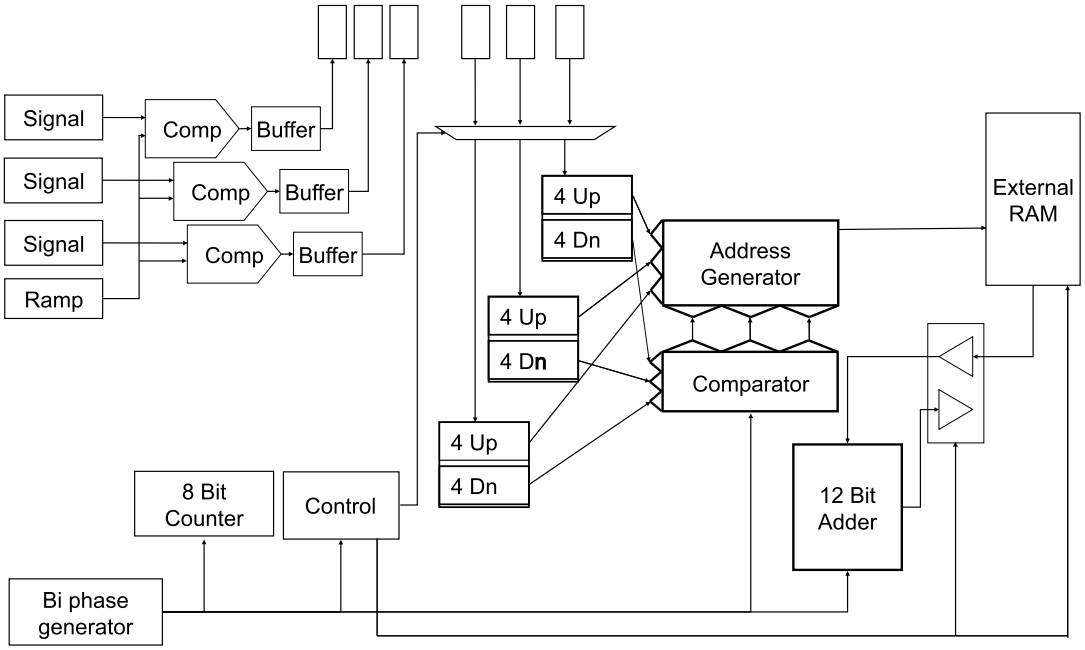

Fig. 4. Chip architecture.

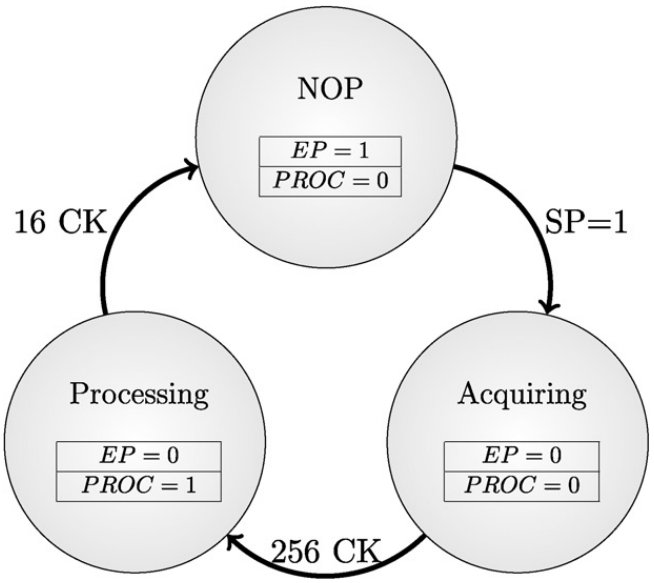

Fig. 5. ASM diagram of the inner finite state machine.

4 most significant bits are connected to "0". The adder circuit is comprised of two modules, one calculates the carry, and another calculates the value of the sum. Appropriately sized buffers were designed to drive the clock and reset lines. Each register is Master–Slave with a two-phase clock, where the Master reads input data with a logic "1" in phase one and locks the data with a logic "0". The slave works in a similar fashion but with the second phase. The 8-bit counter has a modular structure, and is used for two different functions: to acquire the inputs, and also to time the addition of the sixteen sums of the values of the memory (function parameters).

3.3. Comparison with other architectures

Most of the architectures proposed so far by the scientific community [21,22,19] evaluate a PWL function using a scheme that is very similar to the one we propose in this paper, i.e. the output is obtained as a weighted sum of terms contained in a memory. Despite of this fact, the cited works present relevant differences with respect to our circuit: they are based on purely digital and more complicated architectures, which use multipliers to calculate the products in the weighted sum,

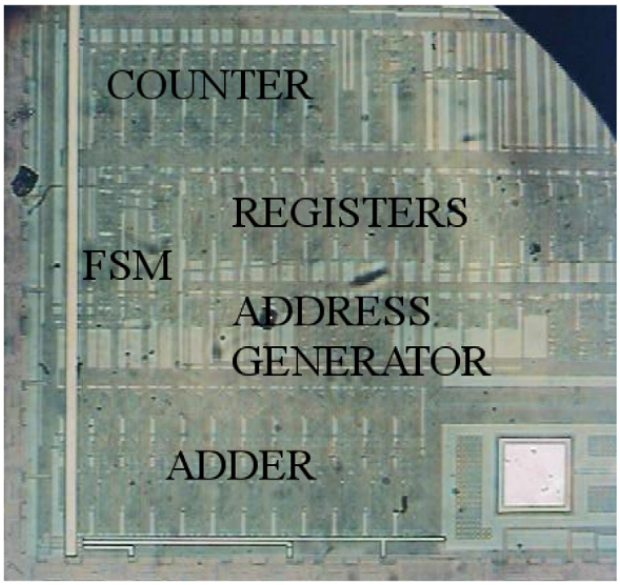

Fig. 6. The IC digital core. The main blocks of the circuit are evidenced.

Table 1
Size of IC parts.

Chip	Area [mm^2]	Percentage
Complete	2.25	100
PADS	1.44	64
Digital	0.43	19
Analog	0.13	5.76

and implement PWL functions using FPGA. As a result, the working frequency and the throughput are increased as well as the circuit complexity and the power consumption.

In [23] the problem of implementing PWL multivariate functions (even discontinuous and defined over compact domains non-regularly partitioned) for control purposes is tackled with an integrated circuit (ASIC). This solution, based on binary search tree exploration, allows the implementation of a larger class of PWL functions but it contains some drawbacks. The number of multiplications and the size of the memory are the highest among every other PWL circuit in the literature (at the best of authors' knowledge); moreover, the circuit has only been simulated and not actually implemented and tested.

Finally, we can state that our solution is more suited for low-power applications whit strictly area constraints (see the Introduction for some examples).

4. Testing results

The IC (shown in Fig. 6) was integrated in an n-well non-silicided 0.5 μm CMOS process through the MOSIS service. This process has 3 metal layers and 2 polylayers. All the digital transistors are minimum size, with PMOS sizes of 3 μm × 0.6 μm and NMOS sizes of 1.8 μm × 0.6 μm. Table 1 shows the sizes of the different parts of the IC.

A board was designed to allocate the chip and allow the application of signals and the collection of data. All signals (clock, inputs, control, etc.) were generated by an FPGA (Xilinx Spartan 3) using VHDL, and were measured using a digital oscilloscope Tektronix TDS 3052 and a logic analyzer HP 1660EP. A voltmeter was used to measure the power consumption.

The testing focuses on the chip's digital block and it consists of three parts:

1. Static check of the chip output data and comparison with expected data;
2. Calculation of power consumption and detection of the maximum working frequency;
3. Dynamical measures with time-varying inputs.

4.1. Static input/output measures

The output values produced by the chip in response to a set of input signals were compared with two models in Matlab. The first model is a PWL version (floating point precision), f_{PWL}, of the continuous Matlab function "peaks"; the second model is a PWL function with 8-bit precision f_{PWL_8} – same as the one implemented on chip.

Fig. 7 shows the results obtained by approximating the properly scaled (over the domain $S = [0, 15]^2$) two-dimensional Matlab® 'peaks' function, with $m = 15$. The samples of f_{CHIP} were measured from the chip by imposing 61×61 static input

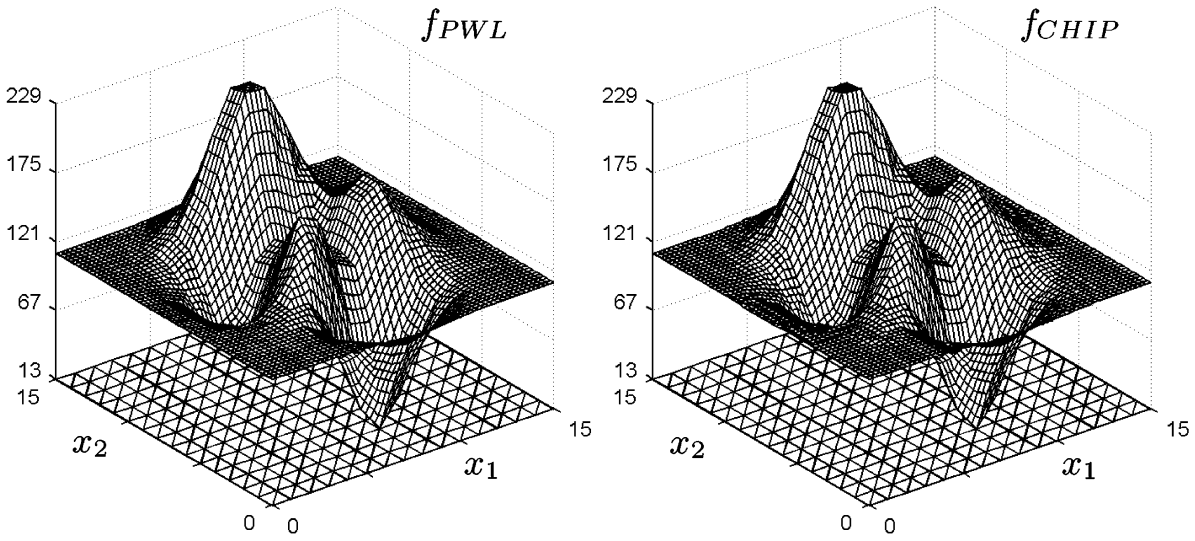

Fig. 7. Approximation results: left, PWL approximation f_{PWL} of the original function; right, measured samples of the implemented PWL function f_{CHIP}. The red lines evidence the simplicial partition of the domain S.

Table 2

f_{CK} [kHz]	Current [mA]	Power consumption [mW]
0.5–100	0.007	0.0231
500	0.013	0.0429
1000	0.025	0.0825
1500	0.036	0.1188
5×10^3	0.115	0.3795
10^4	0.234	0.7722
15×10^3	0.347	1.1451
20×10^3	0.454	1.4982
50×10^3	1.096	3.6168

pairs (z_1, z_2) regularly distributed over S. The first test was to check that f_{PWL_8} was equal to f_{CHIP} at all points. This test was also checked with several functions generated randomly, and confirmed the correct computation of the chip.

The maximum and the mean error $|f_{PWL} - f_{CHIP}|$ of the model implementation are 1.2262 and 0.3494, respectively. These results evidenced the good accuracy of the obtained implementation, since f_{PWL} ranges in the interval $[13.47, 228.98]$. The differences between f_{PWL} and f_{CHIP} are due only to the 8-bit integer precision of the chip (in the representation of both the function f_{CHIP} values and the coefficients c_k), versus floating point accuracy of f_{PWL} produced by Matlab.

Despite the use of a two-dimensional example (chosen to handle data easier to display), the result is significant since it shows that the circuit runs exactly the algorithm described in [10].

4.2. Power consumption and maximum working frequency

The measurements of the power consumption at different clock frequencies f_{CK} with a supply voltage of 3.3 V produced the results summarized in Table 2.

4.3. Dynamic input/output measures

The aim of this testing phase is to establish the maximum clock speed of the IC, and the equivalent processing speed of the PWL core, whether analog or digital inputs are intended to be used.

In order to test the maximum clock speed, the function of previous section was used and the clock frequency was increased looking for a wrong chip output. The input vectors, generated by an FPGA, were sent in sequence to the chip. Furthermore they were taken over an $11 \times 11 = 121$ points grid and "swept" completely over the function domain. As shown in Fig. 8, the chip input signals z_1 and z_2 can be viewed as a pair of periodic ramps of period 11 T_d and 121 T_d, respectively, while the third input signal z_3 is held constant.

Fig. 9 illustrates the results obtained at $f_{CK} = 10$ MHz. The upper panel shows a PWL approximation of the 'peaks' function calculated with 8-bit finite arithmetic and the lower panel shows the function calculated by the chip, measured with the logic analyzer. The error is zero over all points of the domain.

Several tests were run and the results were correct up to the maximum FPGA clock speed of 50 MHz. Operation beyond this frequency continues, but could not be tested with the available laboratory instruments.

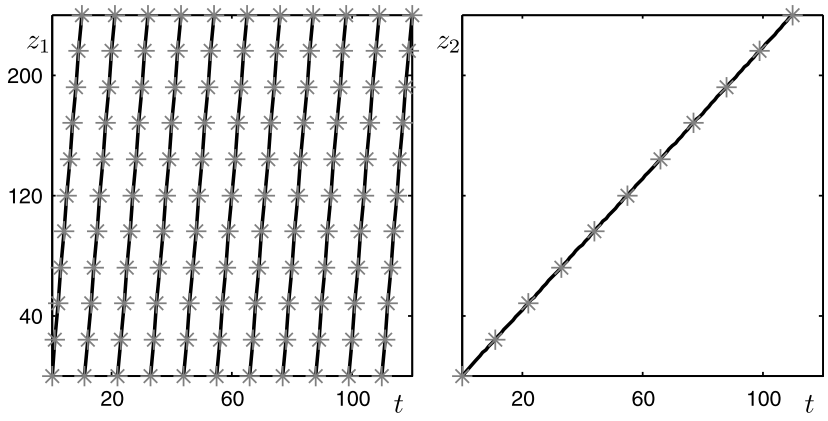

Fig. 8. Chip input signals $z_1(t)$ and $z_2(t)$ used to test the IC.

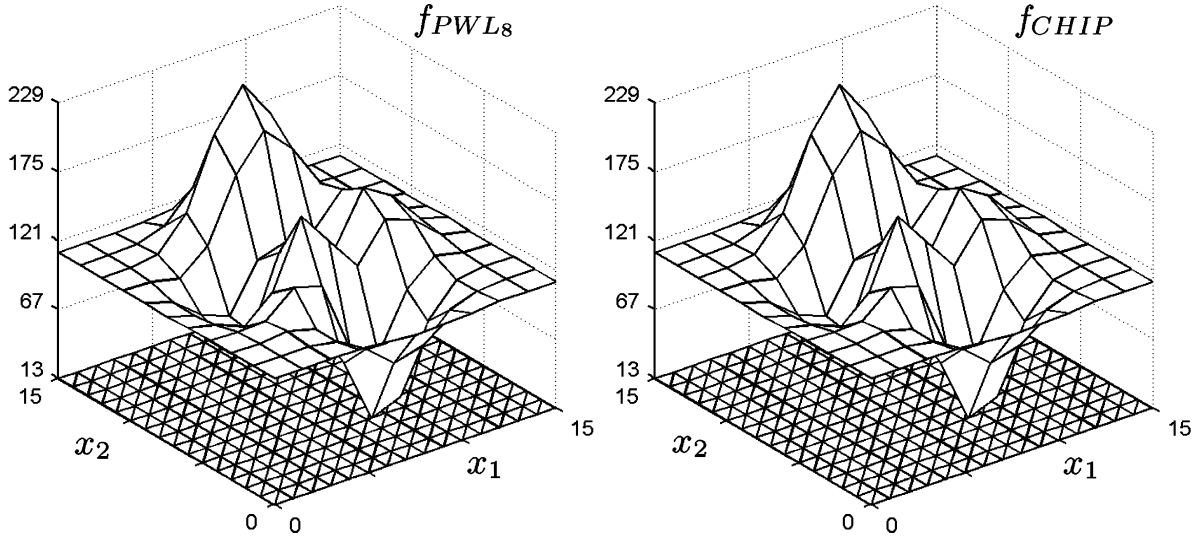

Fig. 9. PWL functions: f_{PWL_8} (left) is calculated with 8-bit finite arithmetic and f_{CHIP} (right) is calculated by the chip.

Assuming now that the input signals are analog, the chip A/D converters (or other A/D converters) must be used. The A/D must comply with Nyquist sampling theorem in order to take into account the input signal spectrum, providing a given output sample rate at a given clock frequency. From then on, we have a digital system that only needs to process each input sample in real time, i.e., before the next sample conversion is completed.

Once the input signal has been converted into the digital domain by the A/D, the speed limitation of the PWL core resides primarily on the input/output memory access and also on the adder realization. The use of an internal cache memory can increase the operation speed significantly. Especially, if the chip is implementing a dynamical system that follows a continuous trajectory, because in that case, the possible regions where the system will go can be anticipated and the vertices values can be pre-stored in the cache. In addition, the use of pipeline techniques can also lead to an increase of the system speed.

According to the Nyquist theorem, the relationship between the bandwidth B_s and the time T_d the chip needs to process an input vector (i.e., the sampling time) is ruled by:

$$2B_s \leqslant \frac{1}{T_d} \tag{5}$$

T_d is linked to the chip clock frequency f_{CK} and can be calculated considering the number of clock cycles required to perform every single operation. There are sixteen (16) cycles required to complete the PWL calculation, and two (2) more overhead cycles of the state machine. At this point, note that the input acquisition time is not considered, as only the PWL core processing time is of interest. It is clear that the user might select an acquisition method of his choice, considering trade offs of speed, power consumption and area.

Based on these considerations, a total number of eighteen (18) clock cycles is required to completely process an input. This means that:

$$T_d = \frac{18}{f_{CK}} \tag{6}$$

By substituting in Eq. (5), we obtain the relationship

$$B_s \leqslant \frac{f_{CK}}{36} \tag{7}$$

that expresses the maximum input bandwidth in terms of the clock frequency. The maximum clock frequency tested was 50 MHz, which internally is converted into a 25 MHz bi-phase clock. This would imply, according to (7), a maximum bandwidth of 694 KHz for analog operation. If digital inputs are used, the core can operate at most at 1.38 MHz.

As a final remark, we notice that if we want to work with input and output analog signals, the real maximum working frequency depends on application. Indeed, the upper limit estimated through the Nyquist theorem is valid only for the input signal.

If the digital output of the chip has to be converted back to analog, the real maximum clock frequency must be estimated by taking into account both input and output signals spectra.

The frequency spectrum of the output signal cannot be directly inferred from that of the input signal, due to nonlinearity. One can estimate this spectrum either *a priori* by simulations not involving the circuit or *a posteriori* by measuring the circuit output and increasing the clock frequency until the output spectrum does not change anymore.

5. Concluding remarks

An IC has been designed to provide a platform for the realization of generic simplicial PWL functions with 8-bit precision. The IC has been successfully tested and can be used to approximate with good accuracy many nonlinear functions, also defining dynamical systems. The architecture is quite fast, and produces a valid result after 18 clock cycles. The simplicity of the architecture has another beneficial side effect, which is the low power consumption. Another important advantage of the architecture is the possibility of increasing the number n of inputs by adding relatively few elements to the chip, owing to the high degree of parallelism in the architecture of Fig. 4.

Acknowledgments

The authors would thank Prof. Mauro Parodi for useful comments and discussions. This work was partially supported by the European Community through the MOBY-DIC project (FP7-IST-248858).

References

[1] Special Issue on Sensor Networks and Applications, IEEE Proc. 91 (8) (Aug. 2003).
[2] G.S.E.M.P. Kennedy, R. Rovatti (Eds.), Chaotic Electronics in Telecommunications, CRC Press, Boca Raton, 2000.
[3] A. Bemporad, M. Morari, V. Dua, E. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica 38 (1) (2002) 320.
[4] M. Storace, F. Bizzarri, Towards accurate PWL approximations of parameter-dependent nonlinear dynamical systems with equilibria and limit cycles, IEEE Trans. Circuits Syst. I 54 (3) (March 2007) 620–631.
[5] Z. Zhou, G.L. Rue, A 12-bit nonlinear DAC for direct digital frequency synthesis, IEEE Trans. Circuits Syst. I 55 (9) (Oct. 2008) 2459–2468.
[6] J. Langlois, D. Al-Khalili, Novel approach to the design of direct digital frequency synthesizers based on linear interpolation, IEEE Trans. Circuits Syst. II 50 (9) (Sept. 2003) 567–578.
[7] O. Hammi, F. Ghannouchi, S. Boumaiza, B. Vassilakis, A data-based nested LUT model for RF power amplifiers exhibiting memory effects, IEEE Microwave Wireless Comp. Lett. 17 (10) (Oct. 2007) 712–714.
[8] M. Vidal, D. Massicotte, A VLSI parallel architecture of a piecewise linear neural network for nonlinear channel equalization, in: Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference, vol. 3, 1999, pp. 1629–1634.
[9] M. Storace, M. Parodi, Towards analog implementations of PWL two-dimensional non-linear functions, Int. J. Circuit Theory Appl. 33 (2) (Mar.–Apr. 2005) 147–160.
[10] M. Parodi, M. Storace, P. Julián, Synthesis of multiport resistors with piecewise-linear characteristics: a mixed-signal architecture, Int. J. Circuit Theory Appl. 33 (4) (Jul.–Aug. 2005) 307–319.
[11] P. Julián, A. Desages, O. Agamennoni, High-level canonical piecewise linear representation using a simplicial partition, IEEE Trans. Circuits Syst. I 46 (4) (Apr. 1999) 463–480.
[12] P. Julián, A. Desages, B. D'Amico, Orthonormal high level canonical PWL functions with applications to model reduction, IEEE Trans. Circuits Syst. I 47 (5) (May 2000) 702–712.
[13] M. Storace, L. Repetto, M. Parodi, A method for the approximate synthesis of cellular non-linear networks – Part 1: Circuit definition, Int. J. Circuit Theory Appl. 31 (3) (May–Jun. 2003) 277–297.
[14] M. Di Federico, P. Julián, T. Poggi, M. Storace, A simplicial PWL integrated circuit realization, in: Proceedings of the 2007 IEEE International Symposium on Circuits and Systems (ISCAS'07), New Orleans, USA, May 2007, pp. 685–688.
[15] P. Julián, R. Dogaru, L. Chua, A piecewise-linear simplicial coupling cell for CNN gray-level image processing, IEEE Trans. Circuits Syst. I 49 (2002) 904–913.
[16] R. Dogaru, P. Julián, L. Chua, M. Glesner, The simplicial neural cell and its mixed-signal circuit implementation: An efficient neural-network architecture for intelligent signal processing in portable multimedia applications, IEEE Trans. Neural Networks 13 (2002) 995–1008.
[17] P. Mandolesi, P. Julian, A. Andreou, A scalable and programmable simplicial CNN digital pixel processor architecture, IEEE Trans. Circuits Syst. I 51 (5) (May 2004) 988–996.
[18] M. Chien, E. Kuh, Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision, IEEE Trans. Circuits Syst. 24 (1977) 305–317.
[19] A. Boggiano, S. Delfitto, T. Poggi, M. Storace, FPGA implementation of a new scheme for the circuit realization of PWL functions, in: Proceedings of the European Conference on Circuit Theory and Design (ECCTD'07), Aug. 2007, pp. 874–877.
[20] M. Storace, O. De Feo, Piecewise-linear approximation of nonlinear dynamical systems, IEEE Trans. Circuits Syst. I 51 (4) (Apr. 2004) 830–842.

[21] R. Rovatti, C. Fantuzzi, S. Simani, High-speed DSP-based implementation of piecewise-affine and piecewise-quadratic fuzzy systems, Signal Process. 80 (2000) 951–963.
[22] P. Echevarria, M. Martínez, J. Echanobe, I. del Campo, J. Tarela, Digital hardware implementation of high dimensional fuzzy systems, in: Applications of Fuzzy Sets Theory, in: Lecture Notes in Comput. Sci., Springer, Berlin, 2007, pp. 245–252.
[23] T. Johansen, W. Jackson, R. Schreiber, P. Tøndel, Hardware synthesis of explicit model predictive controllers, IEEE Trans. Control Syst. Technol. 15 (1) (Jan. 2007) 191–197.

Martin Di Federico received his Bachelor degree in Electronic Engineering in 2006 from the Universidad Nacional del Sur in Argentina. In 2006, he started his doctorate studies at the same university. He is currently a teacher assistant in Analog Circuits Design. He is the GOLD Representative of Region 9 and a member of the IEEE CASS Board of Governors for 2009–2011. Currently at Universidad Nacional del Sur he is working in the development of CNN 3D pixel processors. His current field of research is the applications of VLSI circuits for image sensing and processing. His research interests include digital image processing, CNNs, neuromorphic circuits and bio-electronic implants.

Tomaso Poggi was born in Varazze, Italy, in 1982. He received the Laurea degree (Summa Cum Laude) in electronic engineering from the University of Genoa, in November 2006. He is currently a Ph.D. student at the Department of Biophysical and Electronic Engineering at the University of Genoa, Italy. His main research interests are in the area of embedded systems, with emphasis on circuit implementation of piecewise-linear functions and piecewise-linear approximation/identification techniques.

Pedro Julián was born in Bahía Blanca, Argentina, in 1970. He received the Ingeniero Electrónico degree in 1994, and the Ph.D. degree in Systems Control in 1999, both from the Universidad Nacional del Sur, Bahía Blanca, Argentina. He is currently an Associate Professor in the Microelectronics Area of the Departamento de Ingeniería Eléctrica y Computadoras of the Universidad Nacional del Sur, Bahía Blanca, Argentina and Independent Researcher of the National Scientific and Technological Research Council (CONICET) of Argentina. He was a visiting Associate Professor in the ECE Dept. at Johns Hopkins University and a visiting scholar in Johns Hopkins University (2002–2003) and in the University of California at Berkeley (2000–2001). His research interests include systems and circuit theory and design, practical and theoretical aspects of computation structures, including parallel and sensory systems with a special emphasis on the design and analysis of low-power VLSI systems.

Marco Storace was born in Genoa, Italy, in 1969. He received the "Laurea" (M.Sc.) five-year degree (Summa Cum Laude) in Electronic Engineering in March 1994 and the Ph.D. degree in Electrical Engineering in April 1998, both from the University of Genoa, Italy. Since 2004 he has been an Associate Professor in the Department of Biophysical and Electronic Engineering, University of Genoa. He was a visitor in EPFL, Lausanne, Switzerland, in 1998 and 2002. His main research interests are in the area of nonlinear circuit theory and applications, with emphasis on circuit models of nonlinear systems (e.g., hysteresis, biological neurons), methods for the piecewise-linear approximation of nonlinear systems and for the consequent circuit synthesis, and nonlinear dynamics. He is the author or coauthor of about ninety scientific papers, more than a half of which have been published in international journals. Dr. Storace served as Associate Editor of the IEEE Transactions on Circuits and Systems – II (2008–2009) and is a member of the IEEE Technical Committee on Nonlinear Circuits and Systems (TC-NCAS). He is coordinator of the MOBY-DIC European project (FP7-IST-248858, www.mobydic-project.eu).