Circuit-Level Considerations for Mixed-Signal Programmable Components

Luigi Carro, Marcelo Negreiros, Gabriel Parmegiani Jahn, Adão Antônio de Souza Jr., and Denis Teixeira Franco
Universidade Federal do Rio Grande do Sul

The use of digital compensation algorithms eliminates the error introduced by switches and the nonlinear behavior of MOS transistors. This approach greatly reduces analog area and permits field-programmable mixed-signal systems built with entirely digital technologies.

IN THE PAST DECADE, researchers have paid significant attention to the design of heterogeneous embedded systems. However, most of their work focused on digital hardware and associated software,[1,2] which remain firmly rooted in an all-digital environment. In contrast, an embedded system must generally interface with the real world. At this level, most signals are continuous time and can assume only analog values. Most sensors (such as those for temperature, pressure, humidity, resistance, and so on) generate small voltages or currents, and systems must preprocess (amplify and filter) these signals before using them.

Currently, many applications use digital signal processing because of its fast design cycle, available tools, and integration with current design methodologies. Designers can use this technique in any problem that does not require a gigahertz frequency bandwidth or extremely low power dissipation. Nevertheless, although digital processors can implement many analog functions, some analog functions must be developed in the analog domain. For example, amplifiers, low-pass antialias filters, and an A/D converter can be developed only in the analog domain; the digital domain has no equivalent. So although most of a mixed-signal system might be digital, such systems still require analog functions.

For linear behavior, a system should in principle use linear components. However, linear capacitors or resistors in MOS technology take up too much area. Moreover, programmability generally comes in the form of MOS switches. These transistors introduce some extra pole-zero pairs, and worse, they have nonlinear voltage-current characteristics.

We will show that, even when using analog technologies that allow components like linear capacitors or resistors, the resulting circuit will display nonlinear behavior because of the programmability requirement. Moreover, this approach will still consume significant circuit area.

For these reasons, designers need a new paradigm for designing analog circuits. Here, we propose the use of nonlinear analog components with digital compensation. Such components can be area effective, allowing designers to easily build important circuits. It is also possible to directly develop this class of circuits in a digital technology. Moreover, the proposed circuit technique is easily programmable using switches and incurs only a slight performance degradation. Some of our results, such as those for amplifiers and integrators, use only a small area and demonstrate a perfect balance between the possibility of programming an analog or a digital device in the same chip. This methodology allows the development of linear or sampled circuits like switched capacitors.

Although some work has addressed analog field-programmable gate arrays,[3-5] most of it focuses on the prototyping of analog functions like filters and amplifiers. However, the embedded-systems market seems to require mixed-signal programmable systems, not

0740-7475/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

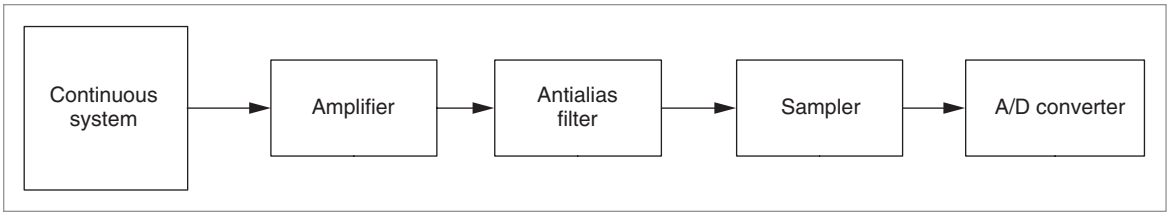

Figure 1. Input path for the target applications of the proposed field-programmable mixed system (FPMS).

only analog circuits. In a mixed-signal context, the most important design parameter might be the trade-off between speed and accuracy, as in the case of Σ-Δ converters. Most of the signal processing should occur in the digital domain, because it is much simpler to design and easier to prototype and to automate than in the analog domain. Also, a mixed-signal chip will include a microprocessor of some sort that a designer could use to develop the signal processing in the digital domain.

Considerations for mixed-signal programmable systems

Many electronic devices, such as current system-on-a-chip (SoC) designs, contain components that have increasing levels of integration and fit into a single chip. However, when it comes to field configurability, just as for field-programmable gate arrays (FPGAs), the analog interface to the world—which includes all conditioning and acquisition circuits—remains outside the IC. Maintaining this separation requires external analog devices and increases the probability of connection-related malfunctions. Such a situation is typical for the analog front end in systems that perform tasks such as data acquisition in control and measurement applications.

One goal of the proposed programmable analog circuit is to cover such needs in a SoC environment. For this environment, we would like to provide a front-end analog subsystem that permits full development, prototyping, and testing of typical embedded applications. The entire field-programmable mixed system (FPMS) is usable in the same way as purely digital FPGAs. Designers could then fully develop an acquisition device composed of a sensor, its conditioning circuitry, and the digital data manipulation and storage block on the FPMS device. They could also avoid testing delays by eliminating the need for designing a board and assembling all the necessary ICs and analog components.

Figure 1 shows the typical front-end topology for a desired target application. A real-world, continuous-time process generally needs some amplification, because sensors rarely give the required voltage or current levels. Because of present-day technology and supporting tool sets, most signal processing occurs in the digital domain, so an antialias filter is absolutely required. Finally, depending on the required converter's speed and resolution, an application might need a sampler to maintain the resolution while working with fast signals.

Assuming that Figure 1 covers most applications that we want to address, several parameters must be subject to programmability. The signal from the continuous system can be either a current or a voltage signal, which means that the amplification stage should be able to change its input impedance and gain. In addition to this requirement, the signal's frequency response must be treatable in a second stage that can implement filters. This, in turn, means that we must be able to place poles and zeros at certain frequencies.

Finally, analog sampling should be one of the programmable features. You should be able to choose whether A/D and D/A converters are multiplexed between channels and whether the system should trade speed for resolution in any of the A/D or D/A converters.

Analog front-end topology

Looking at Figure 1, we can divide the analog block into three main parts. The first part receives and amplifies the analog signal; the second performs the antialias filtering on the signal; and the final block converts the signal to the digital domain. In a design, we must consider which topology to use in each of these stages to achieve an optimal combination of programmability and overall performance.

Our initial efforts to design and test the building blocks for the future FPMS' logic cell focus on continuous-time circuits. Other groups have already used switched capacitor and switched current as the basic technology for the programmable analog part.[3-5] In contrast, we chose the continuous-time implementation to ensure that all the circuitry before the antialias filter would not introduce spurious frequencies.

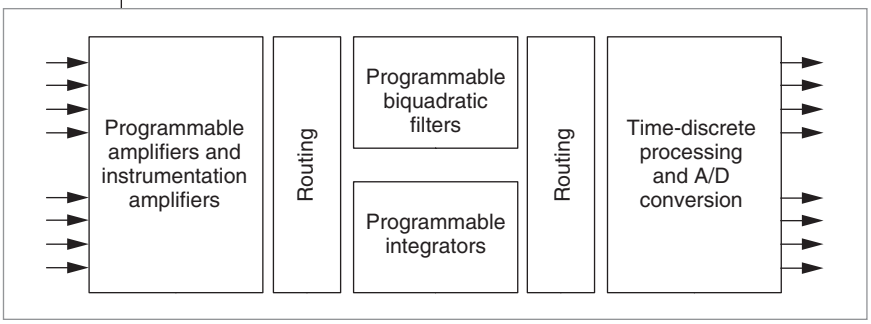

Figure 2. FPMS macroarchitecture.

First stage: Analog-signal amplification

To amplify the input signal, we must provide a block that can implement voltage and current gain. We started with the well-known differential amplifier with three operation amplifiers (op amps) and also provided shortcuts to allow one input to use only the final inverting stage. Thus, every two analog inputs are associated with three op amps, and are usable as differential voltage inputs. The first input of each pair can also bypass the input pair of op amps and use the third op amp as an inverting amplifier; the second input becomes unavailable.

Second stage: Antialias filtering

Just after amplification, the signal must pass through an antialias filter before it undergoes A/D conversion. The antialias filter's underlying complexity strongly relates to the A/D conversion process and the sampling ratio. Therefore, if the application requires a high sampling rate, it is more likely to have an A/D conversion process that cannot be strongly oversampled, hence the need for an effective analog antialias filter. On the other hand, if a high oversampling ratio is allowable, even a single-pole integrator can serve as an antialias filter.

We can now picture two distinct blocks, one with a complete, programmable antialias-filter design and the other with single integrators. Each of these groups can connect to the first row of amplifying cells through a connection network.

Another project possibility is to design the entire block as a set of integrator or gain cells that designers can rearrange to provide higher-order antialias filters. Researchers have used this type of design with arrays in some switched-capacitor work;[3-5] these designs are also an option for further analysis. In our initial architecture, we chose to have both integrators and second-order antialias filters, and provided internal paths, allowing designers to cascade these cells to form higher-order filters.

We can use two different topologies to actually implement the programmable analog part. The first topology could be one presented earlier, which offers an array of amplifiers, switches, and capacitors.[3-5] This work uses only linear devices, a distinct disadvantage in terms of area usage. The programmable topology proposed by Pavan, Tsividis, and Nagaraj would be another possibility.[6] This work addresses the problem of filter programmability; its implementation uses only MOS transistors and gate-to-channel capacitors. Although this implementation uses nonlinear capacitors, the capacitors are biased so that they operate in their linear region. Our approach is quite different than all previous work because it does not cancel or mask nonlinearities in the analog domain, but rather works within the digital domain.

Final stage: Discrete-time and signal digitalization

We will separate this stage from the filtering row by using a connection network to assign each A/D conversion input to a filter output. You can use the same network to perform time multiplexing of the A/D converters. Using a Σ-Δ converter automatically lets the designer customize the A/D conversion, balancing sampling rate and data resolution. The entire system is based on oversampling and consists of an integrator, a one-bit A/D converter, and a one-bit D/A converter on the analog side, plus a digital filter.

Once the input signal reaches this last row of cells in the FPMS, it has already passed the antialias filter. Thus, an implementation might use discrete-time techniques, such as switched capacitors or currents, to perform additional signal processing, before converting the signal to the digital domain. The implementation can apply a more selective filter to narrow the band of allowed frequencies or to eliminate environmental noise. This last set of cells is definable as a standard, reconfigurable switched-capacitor filter directly connected to the input of each conversion cell. Alternatively, you can use these cells as a new row of interconnectable cells between filtering and conversion stages.[3-5]

Figure 2 presents the proposed FPMS macroarchitecture. For each group of eight inputs, four programmable biquadratic filters are available to work as antialias filters, as well as four programmable integrators. As mentioned earlier, you can cascade these filters to produce more complex filters when oversampling is not possible.

After the antialias filters, further amplification is possible, but now would occur in the sampled data domain of switched capacitors. The next block consists of programmable components targeted to implement A/D and D/A converters, followed by regular digital processing.

Analog programmability and nonlinear behavior

In discussing analog programmability in nonlinear behavior, it is necessary to discuss sources of nonlinearity in switches, as well as externally linear, internally nonlinear circuits. We also discuss adaptive filters, which generally work with linear systems.

Sources of nonlinearity in switches

The concept of programmability in the analog domain introduces several problems. Let us take a simple programmable integrator as an example. Imagine that both R and C are linear. However, in a programmable device, you would expect the value of R and C to be programmable, at least. This requires placement of a set of switches that permit the connection of capacitors and resistors in series-parallel arrays. The important point is that a switch will always introduce charge sharing and extra resistivity in the signal path. A second level of programming related to the connection of different analog blocks (such as filters, A/D converters, or amplifiers) could aggravate these problems.

It is possible to implement an integrator connecting two capacitors C and single resistor R. Consider an ideal operational amplifier, but include resistance R_d of the switch and the drain and source capacitance, C_d. The frequency response of the model in Figure 1 is

$$\frac{v_o}{v_i} = -\frac{1}{2}\left(\frac{sCR_d + sC_dR_d + 1}{s^2R_dCC_d + sC}\right)\left(\frac{sR_dC_d}{sRR_dC_d + R_d + R}\right)$$

This equation shows a clear frequency response modification because of the introduction of extra poles and zeros, and assumes linear devices. The switch, implemented with MOS devices, actually has a nonlinear resistance (dependent on the voltage across it), as well as parasitic capacitors, which are junction capacitors, and hence also voltage dependent.

Externally linear, internally nonlinear circuits

The simple mathematical study developed in the previous section shows that any programmable analog connection developed with MOS devices would require a dedicated and complex compensation circuit to restore the desired analog behavior. Moreover, such MOS-based devices severely compromise simple operations like providing gain because any switch would introduce a frequency-dependent behavior, requiring further compensation for stability or an extremely limited passband.

Any approach using MOS devices as switches will introduce nonlinearities in the signal path, necessitating a correction circuit. However, this correction circuit would introduce an area overhead, which would add to the large area for the linear array of capacitors. Moreover, as shown before, programmability implies a certain nonlinearity, which the circuit must compensate for in one way or another, even when it uses linear components. For example, the area for the compensation circuit of a linear MOSFET-C filter is roughly 23% of the analog part, and the linear capacitor area adds another 23% to the total circuit area.[7]

Researchers have discussed the advantage of using gate-to-channel capacitance.[6,8,9] More recent work examined nonlinear devices manufactured in a typical 0.25-micron process with five metal layers. For these devices, the capacitance-to-area ratio of the gate-to-channel capacitor could be seven times higher than the ratio for the poly-poly capacitor and 35 times the capacitance available in a sandwich of all metal layers and poly.[9] This work shows that the use of nonlinear circuits in the analog part of mixed-signal systems produces huge area advantages. Of course, the voltage dependence of such capacitances also causes nonlinearities in the signal-processing circuit's output.

To cope with these problems of area and parasitic nonlinearity, we suggest using a new class of circuits, with externally linear, internally nonlinear behavior. Example uses of these systems are available, such as the MOSFET-C filter, which uses the MOS transistor as a linear resistor.[8,9] By employing the same nonlinear voltage-current characteristic of the MOS transistor, other researchers have devised a linear current division technique.[10] Tsividis presents a revision of nonlinear techniques applied to linear circuit design.[11] Every nonlinear-behavior compensation scheme has a certain cost, which can come in the form of area, speed, or design time (because a complex circuit requires more design time).

Externally linear, internally nonlinear analog circuits, although still not as characterized as linear ones, do have some advantages. You can

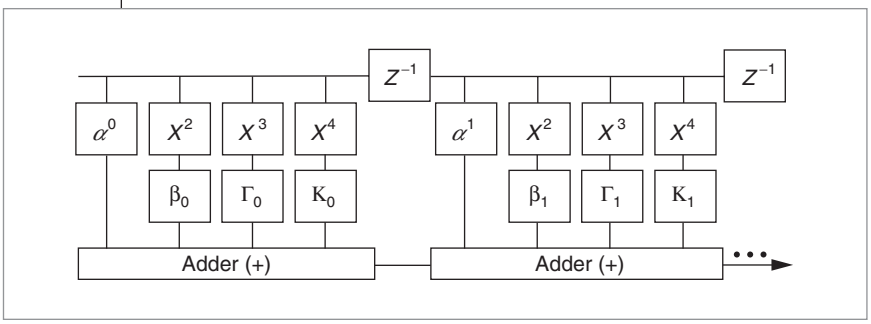

Figure 3. Adaptive nonlinear filter.

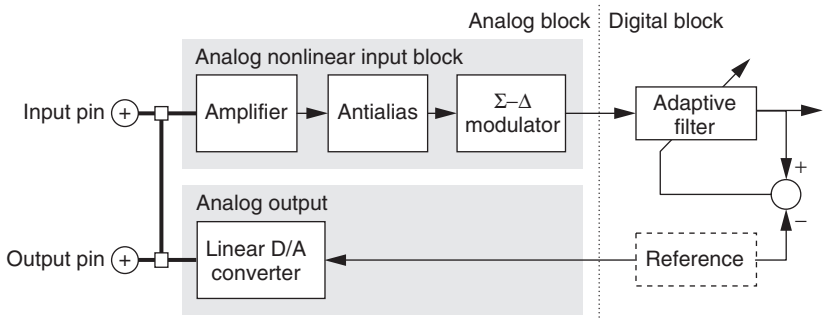

Figure 4. Circuit to train the filter and for operation in steady state.

- develop the circuits in less area, because the MOS gate capacitor is well controlled and has a high capacitance per area;
- use MOS channels as resistors; and
- use switched-capacitor circuits that depend on nonlinear capacitances.

As an example, consider a simple integrator implemented using nonlinear devices. Imagine a nonlinear capacitor, which has a capacitance that varies linearly with the voltage across it: $C = f(v_C) = C_0 + K_1 \times v_C$. In this case, assuming that the op amp gain is large enough to put the negative input at or near virtual ground, capacitance C is a function of the output signal. Considering a linear resistor, solving the differential equation yields the following equation, in which the output has a term dependent on the square of the output.

$$v_o = -\frac{1}{RC_o}\left(\int v_i \, dt + RK_1 \frac{{V_o}^2}{2}\right)$$

Alternatively, you can consider the resistor as $R = g(v_R) = R_0 + K_2 \times v_R$, and a linear capacitor, and the result will show a cross product between the input and the output:

$$v_o = -\frac{1}{R_o C}\left(\int v_i \, dt + CK_2 v_i v_o\right)$$

When both components are nonlinear, the output will have considerable harmonic distortion, even for such simple nonlinear characteristics. Although the preceding analysis assumes imaginary components (MOS transistors have more complex resistance-voltage and capacitance-voltage curves), the overall effect of nonlinearity is to generate harmonics at the output. In addition to these harmonics, other problems arise from the effect of limited op amp gain and bandwidth, and the static characteristics such as offset. Digital compensation's goal is to restore the frequency behavior of the output signal without harmonic distortion.

Nonlinear compensation: Use of adaptive digital filters

In this work, we use the nonlinear adaptive filter shown in Figure 3 to perform digital compensation. Adaptive filters are generally designed to work with linear systems. Most practical applications, however, include some nonlinear behavior. Some designers choose to ignore the nonlinearity or to limit the signal's dynamic range so that the nonlinear behavior is not meaningful.

Here, we have used a nonlinear, adaptive least-mean-square filter, modified from an example by Widrow and Walach.[12] Figure 3 shows the modified filter. The main modification concerns the filter topology, where the nonlinear behavior comes from the exponential operations applied to input signal x in each filter tap.

An important point regarding adaptive filters is their need for training before operational deployment. Figure 4 shows the overall structures necessary to support the training phase of digital adaptive filters. After establishing all the connections in the analog block, designers use white noise to excite the input of each analog processing path. They then compare the resulting acquired data with the expected signal, derived from the internal reference programmed with the behavior of the implemented analog function. The resulting error signal becomes the basis for adjusting the filter coefficients.

The implementation must have a D/A converter to excite the filter and the analog circuit (which needs a known excitation in the training phase). This converter must also have enough passband to guarantee that it excites the filter circuit with all frequencies of interest. The requirement for a D/A converter is usually not a problem because mixed-signal programmable systems typically include one.

The training signal must be rich in frequencies to excite all the circuit poles and zeros that require compensation. In operational mode, all the necessary characteristics have been trained into the filter, which compensates for the nonlinear behavior developed within the circuit's nonlinear devices. The circuit doesn't need the feedback from the error signal anymore; it also no longer needs the D/A converter for compensation (though it might perform other functions in a mixed-signal system).

Practical results

We assembled a strongly nonlinear circuit and devices for its compensation from discrete components. Using this prototype, we measured real-time data using a digital filter implemented as a software routine running on a DSP. Figure 5 shows this nonlinear circuit; because of the diode, this circuit demonstrates a strong nonlinearity, which ranges from the second- to higher-order harmonics, as Figure 6a shows. The system performs training and operation to a single sine waveform, such as when you should excite and monitor the output of a strain gauge, for example. Figure 6 shows the fast Fourier transform (FFT) of the output signal and the signal after linearization. As you can see in Figure 6b, the nonlinear filter cancels out all harmonics inserted by the nonlinear amplifier.

We implemented 75% of the adaptive filter in an Altera 10K10 FPGA (8 equivalent bits, running at 8 kHz). We chose these components because of the available memories to store filter data and coefficients. We developed another version for audio passband with 48 words of code for a DSP running at 40 MHz. It is interesting to notice the very small number of nonlinear coefficients: They are only one-third the number of the linear ones.

Practical results with nonlinearity in the signal's passband

An important question concerns the compensation for nonlinearities whenever the harmonic components fall inside the design's band of interest. So we assembled a new set of experiments to show our compensation method's robustness. We modified the circuit in Figure 5 to change its pole to approximately 100 Hz. A 33-Hz sine wave served as the input signal to the system. To evaluate the distortion introduced by the circuit, we computed the total harmonic distortion (*THD*) using

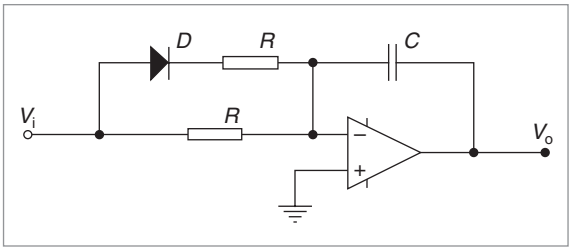

Figure 5. Nonlinear integrator used for validation. In this implementation, R = 10.2 kΩ (± 1%), and C = 680 nF (± 10%). It used a 1N4148 diode and LM741CN op amp with a ± 15-V supply.

$$THD = \sqrt{\frac{\Sigma A_h^2}{A_f^2}} \times 100$$

where A_f is the amplitude of the fundamental frequency and A_h is the amplitude of the harmonics. We obtained the signal components using an FFT of length 1,024 points. We normalized the fundamental frequency to 0 dB; the detection threshold was –80 dB. Our analysis considered frequencies up to 2 kHz and used an integrator developed with linear components as a reference circuit. Measured *THD* for the linear reference circuit was 0.02%.

Table 1 shows the signal's components after the signal passes through the nonlinear integrator. The resulting *THD* is 12.72% within the band of interest (dc to 100 Hz) and 12.83% within the band from dc to 2 kHz. Table 2 presents the measured values after nonlinear digital compensation for each specific harmonic component. In this situation, *THD* is 0.37% from dc to 100 Hz, and 0.39% from dc to 2 kHz. This is still 10 times worse when compared to the linear signal. However, the *THD* of the digital signal used as the reference during train-

Table 1. Signal components after nonlinear circuit (above –80 dB).

Frequency (Hz)	Amplitude (dB)
32.95	0.00
65.91	–18.78
99.12	–25.27
132.08	–35.90
165.03	–60.35
197.99	–47.11
230.95	–52.19
263.91	–75.29
296.87	–62.69

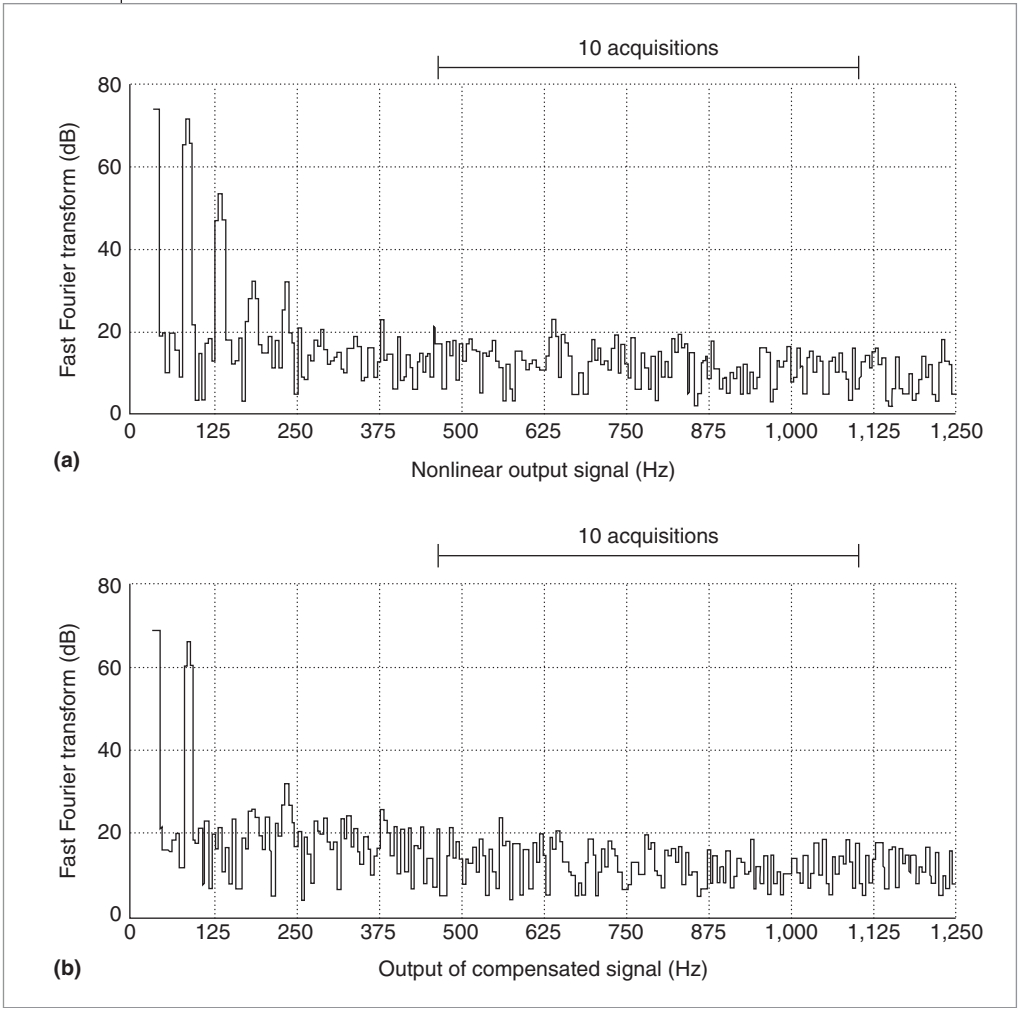

Figure 6. FFT of the nonlinear output (a) and the signal after compensation (b). Notice that compensation cancels out the second and third harmonics.

Table 2. Signal components of compensated circuit (above –80 dB).

Frequency (Hz)	Amplitude (dB)
32.95	0.00
65.91	–56.46
98.87	–49.46
132.08	–58.62
165.03	–68.31
197.99	–66.26
230.95	–68.82
263.91	–79.69
329.83	–77.51

ing was 0.15% within the passband, and 0.15% within the full range (from dc to 2 kHz). This means that the compensation method worked almost as well as the reference signal. A better reference signal could lead to an even lower *THD*.

For all these tests, we used a nonlinear filter with 100 linear taps, and 20 second- and 20 third-order taps. We developed the program in a DSP; it takes only 928 program words, including those for communication with a PC, start-up, and so on. The actual compensation takes 298 instructions per sample, and using an 8-kHz sampler, requires a total of 8,000 × 298 = 2.384 MIPS. The ADSP2181 processor has 29 MIPS available. Training takes only a few seconds.

Area results

To evaluate the area that nonlinear devices consume, we developed analog components using a 0.8-micron CMOS technology. Table 3 shows these area results. As you can see, for a purely digital technology, the area of the nonlinear devices is much smaller than that of the linear devices.

In more advanced fabrication processes, you can achieve a sevenfold gain by using a gate-to-channel capacitance rather than a poly1-poly2 capacitance, in a 0.25-micron process.[9] Also, a capacitor formed by a sandwich of metal and poly layers is 35 times larger than the gate-to-channel equivalent capacitor formed by a sandwich of five metal layers alone.

ALTHOUGH WE HAVE NOT YET prototyped a complete system-level device, this work demonstrates promising results. The compensated test cases can configure the set amplifier and filter, and become part of a mixed-signal application, such as data acquisition from a sensor system. The need for a digital filter might seem to be a problem, but you can easily develop the filter as a specific software block inside a microprocessor. Alternatively, you can form the filter as an array of cells in the digital part of an FPGA itself, as we have shown here.

Another issue is the area occupied by the digital adaptive filter. Designers must develop the area tradeoff at system level. This way, although including a signal processor increases the area, the use of nonlinear

components reduces the analog part. A simple, nonlinear capacitor is at least one order of magnitude smaller than its linear equivalent. Moreover, from the point of view of an FPMS, a processor is readily available, and the main tradeoff involves speed versus accuracy.

Table 3. Capacitor area comparison (area in square microns).

Capacitor value (fF)	Linear poly1-poly2	Linear metal1-metal2	Linear metal1-poly	Nonlinear
500	581	13,605	9,067	229
1,000	1,162	27,603	18,347	458

The use of nonlinear adaptive filters is limited in the sense that, during power up, a training phase must exist. The training phase must use white noise to excite the user-specified analog circuit and the compensating filter. Although easy to achieve, this excitation requires a D/A converter, so the architecture of the mixed-signal programmable system must include one of these devices. Also, because the compensation methodology requires a digital filter, the speed of the set analog-converter compensating filter limits the whole system's frequency response.

Our future work involves a concrete evaluation of the area-performance tradeoff for a mixed-signal programmable system as proposed here. We also plan to complete the layout of the field-programmable analog part. Moreover, we must also study a specific test methodology, because having analog and digital parts on the same chip will impose new challenges in this field. ■

References

1. G. De Micheli and R.K. Gupta, "Hardware/Software Co-Design," *Proc. IEEE*, vol. 85, no. 3, Mar. 1997, pp. 349-365.
2. J.K. Adams and D.E. Thomas, "The Design of Mixed Hardware/Software Systems," *Design Automation Conf.* (DAC 96), ACM Press, New York, 1996, pp. 515-521.
3. E.K.F. Lee and W.L. Hui, "A Novel Switched-Capacitor-Based Field-Programmable Analog Array Architecture," *J. Analog Integrated Circuits and Signal Processing*," vol. 17, no. 1/2, Sept. 1998, pp. 35-50.
4. H. Kutuk and S. Kang, "A Switched Capacitor Approach to Field-Programmable Analog Array (FPAA) Design," *J. Analog Integrated Circuits and Signal Processing*, vol. 17, no. 1/2, Sept. 1998, pp. 51-66.
5. A. Bratt and I. Macbeth, "DPAD2—A Field Programmable Analog Array," *J. Analog Integrated Circuits and Signal Processing*, vol. 17, no. 1/2, Sept. 1998, pp. 67-90.
6. S. Pavan, Y.P. Tsividis, and K. Nagaraj, "Widely Programmable High-Frequency Continuous-Time Filters in Digital CMOS Technology," *IEEE J. Solid-State Circuits*, vol. 35, no. 4, Apr. 2000, pp. 503-511.
7. Y. Tsividis, M. Banu, and J. Khoury, "Continuous-Time MOSFET-C Filters in VLSI," *IEEE J. Solid-State Circuits*, vol. 21, no. 1, Feb. 1986, pp. 15-30.
8. J.C.M. Bermudez, M.C. Schneider, and C.G. Montoro, "Linearity of Switched Capacitor Filters Employing Nonlinear Capacitors," *IEEE Int'l Symp. Circuits and Systems,* IEEE Press, Piscataway, N.J., 1992, pp. 1211-1214.
9. H. Yoshizawa et al., "MOSFET-Only Switched-Capacitor Circuits in Digital CMOS Technology," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, June 1999, pp. 734-747.
10. K. Bult and G.J.G.M. Geelen, "An Inherently Linear and Compact MOST-Only Current Division Technique," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, Dec. 1992, pp. 1730-1735.
11. Y. Tsividis, "Externally Linear, Time-Invariant Systems and Their Application to Companding Signal Processors," *IEEE Trans. Circuits and Systems, II: Analog and Digital Signal Processing*. vol. 44, no. 2, Feb. 1997, pp. 65-85.
12. B. Widrow and E. Walach, *Adaptive Inverse Control*, Prentice Hall, Upper Saddle River, N.J., 1996, p. 508.

Luigi Carro is a professor in the Electrical Engineering Department of Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, and a lecturer in the Computer Science Department at UFRGS. His research interests include mixed-signal design, digital signal processing, and rapid system prototyping. Carro has a BS in electrical engineering and a PhD in computer science from UFRGS. He is a member of the IEEE and the Brazilian Computing Society.

Marcelo Negreiros is an associate researcher in the Signal Processing Laboratory of the Electrical Engineering Department at UFRGS, where he is also working toward a PhD in computer science. His research interests include digital signal processing and adaptive systems. Negreiros has a BS in electrical engineering and an MS in engineering from UFRGS.

Gabriel Parmegiani Jahn is a telecommunications engineer at Telefonica Celular—RS in the area of cell planning. His research interests include digital signal processing, power measurement and quality analysis, and telecommunications, particularly cell planning and data transmission. Jahn has a BS in electrical engineering from UFRGS.

Adão Antônio de Souza Jr. is an associate researcher in the Hardware Prototyping and Testing Laboratory at UFRGS, where he is also working toward a PhD in computer science. His research interests include evolutive hardware, hardware implementation of adaptive systems, and field-programmable mixed-signal arrays. Souza has a BS in electrical engineering and an MSc in engineering from UFRGS.

Denis Teixeira Franco is a lecturer in the Physics Department of Fundação Universidade Federal do Rio Grande (FURG), Brazil. He participated in this research as a student at UFRGS. His research interests include computer architecture, digital systems design, and configurable computing technology. Franco has a BS in electrical engineering from the Catholic University of Pelotas, and an MSc in computer science from UFRGS.

■ Direct questions and comments about this article to Luigi Carro, Departamento de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Av. Osvaldo Aranha 103, Porto Alegre RGS, Brasil, CEP 90035-190; carro@eletro.ufrgs.br.