High-Level Modeling of Analog Computational Elements for Signal Processing Applications

Craig R. Schlottmann, Student Member, IEEE, and Jennifer Hasler, Senior Member, IEEE

Abstract—Large-scale field-programmable analog array ICs have made analog and analog–digital signal processing techniques accessible to a much wider community. Given this opportunity, we present a framework for considering analog signal processing (ASP) techniques for low-power systems. The core of this paper is the definition of an analog abstraction methodology and the creation of a library of high-level analog computation blocks. By abstracting the analog design, we ensure that users have a similar experience to what they would expect with digital design, thus empowering system-level engineers to take advantage of ASP concepts. The result of this paper is to pull analog computation toward system-level development, comparable with the trend in digital system design over the last 30 years.

Index Terms—Analog signal processing (ASP), fieldprogrammable analog array (FPAA), rapid analog prototyping.

I. INTRODUCTION

THE observation of a fundamental power-efficiency wall for digital circuits [1] has encouraged engineers to consider new processing approaches. This challenge has created a renewed interest in techniques such as neuromorphic computation, pioneered in [2], where the natural physics of the subthreshold transistor is used as a computational primitive. This greatly optimizes the computation that can be performed for a given power or area. As an example, subthreshold analog signal processor (ASP) systems have been shown to be 1000 times more efficient than comparable digital signal processors when it comes with the power needed per a million multiply accumulate cycles a second, effectively a 20-year leap on the Gene's Law curve [3], [4].

Cooperative analog-digital signal processing (CADSP) is the design approach whereby the two domains (analog and digital) are combined to achieve advanced system performance [5]. CADSP does not propose to eliminate digital processing, but rather develop hybrid systems that process according to each domain's strengths, as shown in Fig. 1. The resource-precision curves in [6] show that ASP achieves its largest advantage when a lower resolution is required. Digital implementations are more efficient when a higher precision is required because noise accumulation is not as drastic and it is less prone to offsets and mismatch. To realize the CADSP concept, we need two things: 1) a hardware platform for

C. Schlottmann is with the Georgia Tech Research Institute, Atlanta, GA 30332 USA (e-mail: cschlott@gatech.edu).

J. Hasler is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2013.2280718



1

Fig. 1. Analog processor embedded with a digital processor provides a powerefficient platform. The incoming signal can be processed by the FPAA, the digital signal processing (DSP), or a combination of both. A custom MATLAB toolbox is used to program and control the mixed-mode processor.

mixed-signal processing and 2) software tools to leverage the integration. The field-programmable analog array (FPAA) has provided the hardware to develop and explore novel ASP systems, but the art analog design still imposes a large barrier to entry for the typical system engineer.

The goal of this paper is to define a standard analog abstraction method for the purposes of high-level system design [7]. This abstraction framework is necessary to bridge the analog and digital design for the system engineer. By modeling the circuit blocks as signal processing elements, a much higher level of clarity is conveyed and the noncircuit designer is empowered to take advantage of CADSP techniques. This vision resembles the VLSI revolution of the 1980s, which was triggered when the digital circuit design was presented in an accessible way to digital system architects [8].

The remainder of this paper addresses the key challenges that must be overcome to make analog design accessible to the system designer. Section II covers the technique of analog abstraction and system-level constraints. Section III describes the modeling techniques for analog blocks. Section IV provides several case studies of functional models for analog processing elements. Section V pulls all of the concepts together with the design of an analog classifier system. Finally, Section VI concludes this paper.

II. RECONFIGURABLE ANALOG DESIGN ABSTRACTION

This section describes the FPAA hardware as well as several high-level design choices that were made in creating the CADSP framework.

A. Field-Programmable Analog Array

An FPAA is a reconfigurable platform that allows analog systems to be synthesized and programmed repeatedly. FPAAs

Manuscript received August 15, 2012; revised January 31, 2013 and April 18, 2013; accepted August 13, 2013.

2

IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS

fill the same gaps for analog design that field-programmable gate arrays fill for digital design, namely a fabless means for prototyping and fielding VLSI systems, and the flexibility for updating those designs in the field.

The reconfigurable ASP (RASP) 2.9a FPAA architecture [9], [10] is the base platform used in this paper, although the techniques derived here can be applied to other platforms. The RASP contains hundreds of configurable analog blocks (CABs) and a crossbar switch matrix (SM) composed of tens of thousands of programmable floating-gate (FG) transistors. This device was fabricated in 350-nm CMOS and has an operational bandwidth of up to 1 MHz.

Each CAB contains four operational transconductance amplifiers (OTAs), four n- or p-FETs, four FG multiple-input translinear elements (MITEs), four 500-fF capacitors, and a transmission gate. All of the OTAs are based on a ninetransistor architecture. Two of the OTAs in each CAB have an FG capacitive divider on the input pair, which attenuates the input for a wide range and eliminates (or can introduce a fixed) input offset.

The FG nature of the SM transistors and MITEs also allow for the programming away of mismatch on the gate, which eliminates a significant challenge with traditional subthreshold design. The FG elements enable analog memory to be stored within the fabric, therefore coefficients are embedded in the data path; separate memory access is not necessary for typical processing. Examples of systems that have been successfully demonstrated on the RASP FPAA include a low-power robot path planner [11], and a speech processor [12].

B. High-Level Analog Design With Simulink

Simulink is used as the top-level design space for ASP with the RASP FPAA. The *Sim2Spice* tool provides a library of analog blocks and a compiler for generating a SPICE netlist from a Simulink project built with the library [13]. The *Grasper* tool is then used to compile the netlist down to programming code for a particular FPAA architecture. At the intermediate netlist level, full analog simulation can be performed to validate the system. The use of Simulink as a top-level design space is important because it is already a familiar tool to many DSP and control-system engineers and its graphical nature makes system-level design intuitive.

A major component of this framework is the Simulink library of analog components. Here, the analog engineer designs and packages ASP blocks to be used by others. In addition to attaching a valid circuit mapping, a model is included for the simulation and to describe the user the operation of the block. This modeling introduces the question of how much abstraction is required. If a large mixed-mode system is to be simulated, the simpler the model is, the faster the simulation will run, the quicker the analog engineer can design and release the blocks, and the easier the system will be for others to understand. The models, however, need to be flexible enough that higher order effects (such as noise and distortion) can be included for better performance analysis.



Fig. 2. System abstraction first involves defining the signal protocol. The analog processing tool is constrained to use only voltage mode between the blocks because it is more similar to digital design and fits into the Simulink framework. Vectorized signals are also important because they take advantage of the analog processor's parallel processing capabilities.

C. Voltage-Mode Systems

The first step in making analog design feel like digital design is to define a standard protocol for the interface between the blocks. Digital design benefits from a very simple convention of high and low voltages. Conversely, analog systems can propagate information by means of intermediate levels of voltage or current signals. These operating domains create advantages for analog systems. As shown in Fig. 2, current-mode systems can broadcast signals to many destinations. Although each domain has its advantages, these choices are exactly what we want to abstract away in the high-level design space so that the things are easy and familiar to the system designer.

At the expense of the current-mode system's efficient summing, the interface of the Simulink blocks is constrained to voltage-mode operation. This constraint resembles the traditional digital design where a single block can fan out to many, but signals must be summed through a device. Full advantage is still taken of current-mode analog processing inside the block, but the interface is exclusively voltage.

The voltage-mode design methodology has implications on the up-front design of each analog block. Many analog systems have a native current-mode interface, in which the conversion stages need to be embedded. The voltage-to-current (V/I) or current-to-voltage (I/V) stages can take many forms, and the best choice depends on the particular application or specification. Within each block, the multiple conversion choices are generally characterized so that the user can select the one they want based on the performance.

D. Vectorized Signals

Frequently in DSP, and in particular, when using MATLAB, the lines between the blocks are vectorized and carry parallel signals. This is common in matrix operations where the inputs and outputs are multidimensional. Although one of the features of analog design is the ability to represent many bits of information on a single wire, vectorized net buses are incorporated into this analog tool structure to accommodate parallel processing of signals. A wire size of unity is often sufficient, but each net can have any size vector dimension. Rather than forcing the user to define every size, the signals are automatically scaled based on the blocks that are used. For example, if an $M \times N$ vector-matrix multiplier (VMM) is instantiated, the input vector is automatically sized to N, and the output is sized to M.

Fig. 2 shows the use of differential mode along with singleended vectorized lines. Often in analog design, differential signals are used to increase SNR or cancel even-order harmonics. To keep the design simple, single-ended or differential mode can be selected by a block's parameter, rather than having to manually add the complimentary overhead.

E. Biasing

A major design element of analog systems is the proper biasing of the blocks. This is a concept that does not manifest itself in digital design, and therefore must be dealt with behind the scenes.

The RASP FPAAs are built on a network of FG switch elements that can also store the bias values for computation. This feature is one of the reasons why such high computational density is achieved. During the synthesis, the bias value is derived from a parameter in the system's function. For instance, in an OTA-C filter, the time constant is given by a C/G_m relation. These hardware mappings are written into the block, so that the user only needs to specify the time constant, and the correct bias is programmed.

III. ANALOG MODELING TECHNIQUES

This section defines the methods used to model analog blocks, whereas larger computational block examples are modeled in Section IV. Here, we derive the expressions for three common analog characteristics: 1) nonlinearities; 2) noise; and 3) the conversion stage transfer function.

A. Nonlinearities

Electronic devices are inherently nonlinear. We frame our discussion around the MOS device operating in the subthreshold regime, because that is where ultralow power is achieved. The nonlinearity is observed in the subthreshold, saturated $(V_{\rm ds} \ge 4U_T)$ current–voltage equation

$$I_d = I_0 e^{(\kappa V_g - V_s + \sigma V_d)/U_T} \tag{1}$$

where V_g , V_s , and V_d are the gate, source, and drain voltages, respectively, $U_T = kT/q$ is the thermal voltage, q is the charge of an electron (1.6 × 10⁻¹⁹ C), κ is the inverse subthreshold slope, I_0 is a device-dependent preexponential term, and σ is the drain-induced barrier lowering parameter that models subthreshold current versus drain voltage changes [14]. In the rest of this paper, we treat (1) as an ideal current source and ignore the early effects (i.e., $\sigma = 0$). The values for each constant are included in the modeling file that is compiled for each FPAA [13]. To model the nonlinearity, we use the following expansion:

$$e^{x} - 1 \rightarrow x + \frac{x^{2}}{2} + \frac{x^{3}}{6} + O(x^{4}).$$
 (2)

Other common analog nonlinear functions are the hyperbolic tangent and sine [15]. These nonlinear functions appear



Fig. 3. Low-pass filter (LPF) Simulink block. (a) Basic OTA-C implementation of the LPF. (b) LPF block for Simulink simulation. (c) Parameter box asks for a time constant for the first-order response.

when the output current is measured as a function of the input voltage of common transconductors. We use the following expansions to approximate the hyperbolic functions:

$$\tanh(x) \to x - \frac{x^3}{3} + O(x^5)$$
 (3)

$$\sinh(x) \to x + \frac{x^3}{6} + O(x^5).$$
 (4)

These linearization techniques are illustrated in the analysis of the dynamics for the OTA-C LPF, as shown in Fig. 3 (a). Here, the current summed on the output capacitor is as follows:

$$C\frac{dV_{\text{out}}}{dt} = I_b \tanh\left\{\frac{\alpha\kappa}{2U_T} \left[V_{\text{in}}\left(t\right) - V_{\text{out}}\left(t\right)\right]\right\}$$
(5)

where I_b is the OTA bias current and α is an optional attenuation factor. The attenuation term is included if the widelinear range OTA is used, which has a capacitive divider on the input. This equation is easily nondimensionalized to the common form

$$\tau dy/dt = \tanh(x - y) \tag{6}$$

where $x = \alpha \kappa V_{in}(t) / (2U_T)$, $y = \alpha \kappa V_{out}(t) / (2U_T)$, and $\tau = 2CU_T / (\alpha \kappa I_b)$.

We use the expansion in (3) to obtain

$$\tau \dot{y} = (x - y) - \frac{1}{3} (x - y)^3.$$
 (7)

The expansion is useful not only to see the harmonic pattern, but also it reduces the computation when the nonlinearity is small. If we assume that (x - y) is small (i.e., less than 0.1), then we can drop the cubic term. When the attenuation factor IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS



Fig. 4. Simulation and FPAA response of the LPF. (a) Simulink simulation uses the ideal time constant. (b) FPAA step response closely resembles the simulation. The time constant is programmed as a bias current for the OTA given an amount of capacitance at the output. (c) Nonlinear simulation (solid black) matches the step response from the FPAA (color).

С



Fig. 5. Sinh system. (a) Transconductance amplifier implementation of a sinh. (b) Dynamics of the sinh function are demonstrated by a step response. The nonlinearity is apparent by the bowing off of the straight time constant line.

(α) is unity, the resultant differential input voltage should be less than 10 mV. Here, we see one of the tradeoffs of the wide-input-range OTAs: with an attenuation factor of 0.1, the input is linearized up to 100 mV, but the time constant is increased. This decrease in speed can be compensated for by an increase in I_b , at the expense of power. With the constrained step sizes and initial conditions of zero, we can rewrite (7) as $\tau \dot{y} = (x - y)$. In the voltage mode, we are left with the transfer function

$$\frac{V_{\text{out}}}{V_{\text{in}}} = \frac{1}{s\tau + 1} \tag{8}$$

where the linearized model is used. In other filter modeling applications, we typically use the OTA approximation

$$I_{\text{out}} = G_m \left(V_1 - V_2 \right)$$
 (9)

where $G_m = I_b \alpha \kappa / (2U_T)$, and therefore time constants are in the form C/G_m . Fig. 3 (c) shows the parameter window for the LPF with a time constant of 60 μ s. Fig. 4 shows the model simulation, FPAA step response, and nonlinearity of the LPF.

Another common circuit nonlinearity is in the form of a sinh, exemplified by the dynamics of the transconductance amplifier in Fig. 5. For this system, Kirchhoff's Current Law (KCL) at the output gives

$$C\frac{dV_o}{dt} = I_p(t) - I_n(t).$$
⁽¹⁰⁾

With subthreshold currents, it is written as follows:

$$\frac{dV_o}{dt} = I_{\rm op} \exp\left[\frac{\kappa_p}{U_T} \left(V_{\rm DD} - V_{\rm in}\right) + \frac{\sigma}{U_T} \left(V_{\rm DD} - V_{\rm out}\right)\right] - I_{\rm on} \exp\left(\frac{\kappa_n}{U_T} V_{\rm in} + \frac{\sigma}{U_T} V_{\rm out}\right).$$
(11)

To find the dynamics of this system, we find that at the dc operating point, $V_{in} = V_{out}$, and we define the quiescent current as I_b . With a dynamic input signal, we decompose the input and output into dc and time-varying components

$$V_{\text{in}}(t) = V_{\text{DC}} + v_i(t), \ V_{\text{out}}(t) = V_{\text{DC}} + v_o(t).$$
 (12)

With the dynamic function now in the form

$$C\frac{dv_o}{dt} = I_b \exp\left[-\frac{\kappa}{U_T}v_i(t) - \frac{\sigma}{U_T}v_o(t)\right] - I_b \exp\left[\frac{\kappa}{U_T}v_i(t) + \frac{\sigma}{U_T}v_o(t)\right]$$
(13)

we can nondimensionalize the equation with

$$x = \frac{\kappa}{U_T} v_i(t), \quad y = -\frac{\sigma}{U_T} v_o(t)$$

$$\frac{dy}{dt} = -\frac{\sigma}{U_T} \frac{dv_o}{dt}, \quad \tau = \frac{CU_T}{2\sigma I_b}.$$
(14)

After plugging in these values, we are left with

$$2\tau \dot{y} = \exp(x - y) - \exp[-(x - y)]$$
(15)

and we can use trigonometric identities to write it as

$$\tau \dot{y} = \sinh\left(x - y\right). \tag{16}$$

Using our expansion from (4), we can write this nonlinear dynamic equation as follows:

$$\tau \dot{y} = (x - y) + (x - y)^3 / 6. \tag{17}$$

This form makes it easier to see how the system is acting. Again, for small inputs, we can neglect the cubic term and model the function as a simple difference of output and input. Fig. 5 (b) shows the step response for small and large inputs. For larger inputs, the output is shown to speed up and deviate from the linear time constant. SCHLOTTMANN AND HASLER: HIGH-LEVEL MODELING OF ANALOG COMPUTATIONAL ELEMENTS



Fig. 6. (a) Embedding V/I and I/V stages into the analog blocks allows the system interconnects to be voltage mode. There are multiple implementations of the V/I and I/V stages, such as the (b) diode-connected FET, (c) differential pair, (d) wide-range OTA, and (e) TIA.

B. Noise

The performance of analog circuits is highly susceptible to noise. Noise is a very important consideration in ASP because the noise will accumulate from block to block, unlike the roundoff error in digital systems. Basic noise analysis is provided here as an example to illustrate the process of adding a noise component to the analog models. Noise modeling for other blocks will follow basic principles, which can be found in common analog textbooks. Modern FPAAs include n-/p-FETs and capacitors, so the major contributors are the channel and kT/C noise.

In the LPF example, a noise source can be added in series with the output. The output voltage noise density is modeled as kT/C, which we rewrite as qU_T/C to use the global parameters. Fig. 4 (a) shows the LPF with noise enabled.

The current power of the thermal noise in a subthreshold transistor is given as follows [16]:

$$\hat{I}^2 = 2qI\Delta f \tag{18}$$

where I is the dc current and Δf is the bandwidth. At small current levels, the flicker-noise current power $(KI^2\Delta f/f)$ is negligible even at low frequencies because of the square term.

C. Voltage In to Voltage Out

Many powerful analog blocks are inherently current-mode systems. To conform the voltage-mode system protocol, we need interface blocks: V/I and I/V. These interface blocks are embedded into the system block, as shown in Fig. 6.

The simplest V/I source is a single FET, which produces a current according to (1). The complement I/V is the diodeconnected FET, shown in Fig. 6 (b), which has the following relation:

$$V_{\text{out}} = \frac{U_T}{\kappa} \ln\left(\frac{I_{\text{in}}}{I_0}\right). \tag{19}$$

This pair of converters is advantageous in its simplicity and works well for single-ended designs. There are three major considerations when using these blocks: 1) they are nonlinear, so they are most useful when used together around a fully current-mode block; 2) the input converter is exponentially expansive and the output converter logarithmically compressive, therefore the analog block should have a large dynamic range; and 3) the currents are unidirectional.

For differential systems, we can use a differential pair in the place of the single FET for the V/I stage, as shown in Fig. 6 (c). The differential current is in the form well known from OTAs

$$I_1 - I_2 = I_b \tanh\left[\frac{\kappa}{2U_T} (V_1 - V_2)\right].$$
 (20)

The attenuation factor (α) can be added to this equation using FG FETs (the MITEs), which have a capacitive divider on the gate. If we assume small differential voltage, the tanh can be linearized as follows:

$$I_1 - I_2 = I_b \frac{\kappa}{2U_T} \left(V_1 - V_2 \right).$$
 (21)

This topology has the useful feature that its bias (I_b) can be programmed independently of the system operation. This is useful because the bias current often sets the time constant of the current-mode circuit. For differential diode-connected I/V stages, we use the convention that the output currents are small swings around the bias current: $I_{out} = I_b (1 + \Delta I_{out}/I_b)$. Therefore, the differential diodes produce

$$V_1 - V_2 = \frac{U_T}{\kappa} \left[\ln \left(1 + \Delta I_1 / I_b \right) - \ln \left(1 + \Delta I_2 / I_b \right) \right] \quad (22)$$

which can be shown for small $\Delta I_{out}/I_b$ to reduce to

$$V_{\text{out1}} - V_{\text{out2}} = \frac{U_T}{\kappa} \frac{1}{I_b} (I_{\text{out1}} - I_{\text{out2}}).$$
 (23)

Finally, if the bidirectional currents are desired, we can use a wide-range OTA V/I, as shown in Fig. 6 (d), and a transimpedance amplifier (TIA) I/V, as shown in Fig. 6 (e). The wide-range OTA has a single bidirectional output current that is the difference of the two differential pair currents, and the inclusion of (α) helps to linearize the tanh

$$I_{\text{out}} = I_b \frac{\alpha \kappa}{2U_T} \left(V_1 - V_2 \right). \tag{24}$$

The output stage TIA has the transfer function: $V_{\text{out}} = V_{\text{ref}} - I_{\text{in}}/G_m$. This pair of converters is linear and provides a bias current to the system to set the time constant. The linearity makes them useful individually for the blocks that only need conversion on one port. They are also the choice for single-ended bidirectional systems.

Additional overhead costs of the converter stages are power, noise, and speed. These effects can be modeled by following the techniques previously described. For instance, the added power of each OTA is $2I_bV_{dd}$, where I_b is the signal bias defined by the time constant and V_{dd} is the power supply (2.4 V in the FPAA). IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS



Fig. 7. VMM Simulink block. (a) Diagram of the VMM shows that the output channels are the sums of products of the input channels. (b) Circuit implementation is very compact with FG elements performing the weights. (c) Final design is packaged into a single Simulink block. (d) Parameters for the VMM block show options for the weights, time constant, differential signals, and voltage structure.

IV. HIGH-LEVEL MODELING

Now that the basic modeling parameters have been defined, they can be applied to the basic analog processing blocks. The goal here is to model the blocks as simply and in as low-entropy form as possible, while maintaining the functional fidelity of the block. This method provides the highest intuition to the system designer and consumes the least amount of computational resources.

A. Vector-Matrix Multiplier

The VMM is a core component in many signal processing applications [17]. Vector–matrix multiplication is commonly performed in finite-impulse response filters, 2-D block image transforms, convolution, correlation, and classification [18].

As described in [17], the analog VMM can perform one- two- or four-quadrant multiplication. Fig. 7 shows the circuit-level implementation of a four-quadrant cell. The three important things to recognize in the circuit are: 1) the inputs and outputs are both current-mode signals; 2) the operation is performed with differential signals because the currents are unidirectional; and 3) the multiplier weights are programmed as FG values.

The four-quadrant cell performs the following matrix operation

$$\begin{bmatrix} w_+ & w_- \\ w_- & w_+ \end{bmatrix} \begin{bmatrix} I_{\text{in}+} \\ I_{\text{in}-} \end{bmatrix} = \begin{bmatrix} I_{\text{out}+} \\ I_{\text{out}-} \end{bmatrix}$$
(25)

where the differential weights are referenced to a base (w_B)

$$w_{+} = w_{B} + \Delta w, \ w_{-} = w_{B} - \frac{\Delta w}{2}.$$
 (26)

The transfer function for the current-mode VMM cell is thus

$$I_{\text{out}+} - I_{\text{out}-} = (w_+ - w_-) \cdot (I_{\text{in}+} - I_{\text{in}-}).$$
 (27)

To complete the voltage-mode transfer function, we need to add the embedded conversion to the equation. The differential nature of the VMM calls for the use of a differential-pair input stage. The voltage-mode function is formed by cascading the VMM with the differential pair from (21) and the differential diodes from (23), resulting in

$$V_{\text{out1}} - V_{\text{out2}} = \Delta w \left(V_{\text{in1}} - V_{\text{in2}} \right).$$
 (28)

The system dynamics are a composition of the responses from the input stage, the VMM stage, and the output stage. To make the modeling as simple as possible, we approximate the whole function as a single-pole system, using the lowest frequency pole of the three stages. The dynamics of the VMM stage are primarily set by the bandwidth of the log-amp, which has a dominant pole is at $AI_b/(C_{in}U_T)$ [19]. The factor A is due to the active feedback in the log-amp and increases the effective transconductance (and thus speed) by about a factor of 100. The pole at the output stage is set by the transconductance of a single subthreshold FET, $I_b\kappa/(CU_T)$. Because the pole at the output is at the lowest frequency, we use it for our single-pole approximation for the system. It is clear that the bias current of the system sets the speed of each stage, and thus should be parameterized in the modeling.

The VMM dialog box in Fig. 7(d) contains five parameters that are used in the circuit compilation, whereas only the first two (elements and tau) are used in the Simulink model. The first one (elements) sets the multiplier weights and is programmed into the FG mesh. The time constant is implemented as the dc bias current of the input stage. The last three parameters tell the circuit compiler if the differential signals are needed and which conversion stage to use.

B. C⁴ Bandpass Filter

Filtering for spectral decomposition is a common front-end function for many low-power sensor applications. The capacitively coupled current conveyor (C^4) filter is a programmable, continuous-time bandpass filter that is power efficient and can cover a wide range of frequencies [20]. It is commonly used to create a bank of narrow passband filters (Fig. 8), such as in a Fourier processor system [21]. The defining parameters for such a block are the center frequency (f_{center}) and the quality factor (Q).





Fig. 8. The C^4 bandpass (a) system implementation, (b) Simulink parameter box, (c) schematic, and (d) step response with four outputs. Each tap is tuned for a center frequency of 1 kHz and a Q vector of [0.5, 1, 2, 3].

The dimension of the f_{center} parameter array sets the port dimension of the output and is interpreted as the number of parallel filter channels. If the Q is input as a scalar, that value is applied to each filter. Alternatively, if the Q is input as a vector matching the dimension of the f_{center} , each filter will have the respective Q value. The block also provides an option for a common input, which specifies if the input should be a vectorized bus or a single input line that fans out to each filter element.

The C⁴ schematic diagram is shown in Fig. 8 (c), where we use OTAs rather than individual FETs because it is a more efficient use of the elements available in the FPAA. A benefit of this filter implementation is that the high and low corners can be tuned independently of each other and programmed as the bias of the OTAs. The general transfer function for the C⁴ filter in the Q > 1 region is given in [20] and is

$$\frac{V_{\text{out}}}{V_{\text{in}}} = -\frac{C_1}{C_2} \frac{sC_2/G_{m1}}{1 + s\left(\frac{C_2}{G_{m1}} + \frac{C_0}{G_{m2}}\right) + s^2 \frac{C_0 C_T}{G_{m1} G_{m2}}}$$
(29)

where $C_T = C_1 + C_2 + C_W$ and $C_0 = C_2 + C_L$. The center frequency is thus set by

$$f_{\text{center}} = \frac{1}{2\pi \tau} = \frac{\sqrt{G_{m1}G_{m2}}}{2\pi \sqrt{C_0 C_T}}.$$
 (30)

When this architecture is mapped to the FPAA, capacitors C_2 and C_w do not need to be explicitly placed because the line



Fig. 9. Peak detector. (a) Schematic and (b) Simulink simulation. The output shows the block vectorized for four outputs with a common sine input. Of the four signals, two are MAX followers and two are MIN followers, each with decay rates of 1e3 and 2e3.



Fig. 10. VMM and WTA combine to create a single library block, allowing the internal I/V-V/I to cancel.

capacitance can be characterized and leveraged. This equation provides us with the algorithmic method for generating the circuit netlist from a vector of center frequencies, given that we use Q_{max} .

The OTA structure makes the transfer function easy to visualize as follows:

$$\frac{V_{\text{out}}}{V_{\text{in}}} = -\frac{\tau_1 s}{1 + \tau_2 s + \tau_1 \tau_2 s^2}$$
(31)

where $\tau_1 = C_1/G_{m1}$ and $\tau_2 = C_L/G_{m2}$. Given the canonical form of a second-order filter, we can see the frequency and Q map to these time constants as $\tau_2 = \tau/Q$ and $\tau_1 = \tau^2/\tau_2$, where $\tau = 1/(2\pi f_{\text{center}})$.

Following the discussion in Section III-A, it is useful to understand the nonlinear dynamics of the C^4 system. In most of the cases, the model in (31) is the most efficient to simulate and extract the understanding of the function; however, having an option to include the nonlinear dynamics provides a closer match to the dynamic range of the real analog circuits.

Deriving the system equations from the implementation in Fig. 8 (c), we see that the OTA really provides a tanh rather that a linear transconductance. The resulting model is as



Fig. 11. Simulink simulation from the classifier system for a linear chirp signal. (a) C^4 bandpass filer is set with three blocks to pass different sections of the chirp. (b) Peak detector tracks the envelope of the three channels. (c) VMM–WTA classifier creates an output where only one channel is high at a time. The matrix is [2, 1, 0.5; 0.5, 2, 1; 1, 0.5, 2] to demonstrate each channel winning.

follows:

$$\dot{V}_{\text{out}} = -\frac{I_{b2}}{C_L} \tanh\left(\frac{\kappa}{2U_T}V_2\right) \tag{32}$$

$$\dot{V}_2 = \frac{I_{b1}}{C_1} \tanh\left(\frac{\kappa}{2U_T} \left(V_{\text{out}} - V_2\right)\right) + \dot{V}_{\text{in}} \qquad (33)$$

where I_{b1} and I_{b2} are the bias currents for OTAs G_{m1} and G_{m2} , respectively. In both cases, standard OTAs are used, so α is not included. Note that all the voltages are referenced to V_{ref} and the steady-state condition is that all the node voltages are equal.

C. Peak Detector

The peak detector block frequently follows the spectral deconstruction stage in a signal processing. The peak detector block tracks the envelope of the signal in each band, which is useful for the classification. The implementation of the peak detector is shown in Fig. 9.

Intuitive analysis of this block shows that it acts much like a source follower. When the input rises, the output tracks it while charging the capacitor on the output node. When the output is, however, decreasing, the bias transistor (M2) discharges the capacitor at a fixed rate. This behavior allows the circuit to track the rising peaks, then decay slowly until it hits the next rising peak.

To create the full dynamic model for this block, we start with KCL at the output node

$$C\dot{V}_{\text{out}} = I_0 e^{[A\kappa(V_{\text{in}} - V_{\text{out}}) - V_{\text{out}}]/U_T} - I_0 e^{\kappa V_b/U_T}.$$
 (34)

We see that at dc (i.e., at $\dot{V}_{out} = 0$), the current in the top branch is balanced by the bottom branch, each passing I_b .

Following the dynamics analysis in Section III-A, the output changes at a rate of

$$\dot{V}_{\text{out}} = \frac{I_b}{C} \{ e^{[A\kappa(v_i(t) - v_o(t)) - v_o(t)]/U_T} - 1 \}.$$
(35)

This equation is consistent with our intuitive analysis. When V_{in} exceeds V_{out} , the rate of growth on the output becomes exponential. The output voltage quickly rises and tracks the input voltage. As V_{in} falls below V_{out} , the exponential becomes

small and the term inside the braces reduces to -1. The rate of decay is a constant I_b/C .

We will use (35) to create a block macromodel for this circuit. The main parameter is the rate of decay, which is tuned by the user to the expected frequency of the incoming signal. If the decay is set too slowly, the block will simply find the largest amplitude and hold it through other cycles. This system is easily converted into a minimum detector using a p-FET source follower rather than the n-FET described.

V. CASE STUDY: CLASSIFIER SYSTEM

To bring together the process of creating a whole signal processing system with the design platform, we use the classifier system in Fig. 10 as a circuit example. This example highlights two important aspects of functional-level analog design: 1) using blocks from the predefined library of ASP elements and 2) using inherent mixed-mode computation to create optimal blocks.

The complete system is a chain of five processing blocks: 1) a C⁴ filter bank; 2) a peak detector; 3) VMM; 4) winner-take-all (WTA); and 5) an encoder. The overall system takes an input waveform, spectrally decomposes it into multiple bands as specified by the C⁴ bank, and uses the peak detector to track the envelope of each channel. Next, the VMM projects the spectral channels against multiple classification basis, and the WTA picks the largest output [22].

The WTA is so commonly used with a VMM on the front end as a classifier that we combined the two into the larger classifier block shown in the dashed box of Fig. 10. This is an efficient structure for a classifier and is much more compact than a two-layer neural network, which would have required two VMMs and sigmoid blocks.

Merging the VMM and WTA into a new block has another advantage. The WTA has a current-mode input, so we would need to add a V/I stage. The VMM, however, has a native current-mode output, so by combining the two, we can cancel an I/V-V/I conversion and allow all the current-mode processing on the internal nets. This provides a more efficient and compact realization. When modeling the WTA for the sake of Simulink simulations, we need to pick among progressively more detailed models. The simplest model of the WTA is the MAX function, which can be programmed easily with the MATLAB toolbox. This model simulates the quickest, but misses a lot of the dynamics involved in the analog implementation. The complete dynamics can be found in [22].

Fig. 11 shows the Simulink simulation results for the classifier system. The three plots show the output of each block of the system at the bottom of Fig. 10. For this simulation, the input signal is a linear chirp with a bandwidth of 10 kHz over 10 s. The C⁴ filter block is set with three center frequencies, resulting in three parallel filters in the bank. The peak detector block has three parallel paths that track the envelope of the C⁴ output, and the VMM is set with a 3×3 matrix that gives each output a chance to win. Of the three output channels, the one corresponding to the largest envelope has a high value. Hardware results from this system are presented in [23].

VI. CONCLUSION

In this paper, we have demonstrated the concept of highlevel abstraction and modeling of analog systems. With the drastic increase in size and complexity of modern reconfigurable analog ICs, high-level tools are a necessity. We demonstrated how analog abstraction techniques are a powerful tool for making analog system design easy for noncircuit experts. A key element of this abstraction approach is the creation of high-level analog libraries. We introduced our methodology for analog macromodeling that looks at the function being performed, rather than each low-level element. This paper introduces a new level of intuition into the field of analog system design.

REFERENCES

- [1] B. Marr, B. Degnan, P. Hasler, and D. Anderson, "Scaling energy per operation via an asynchronous pipeline," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 1, pp. 147–151, Jan. 2013.
- [2] C. Mead, Analog VLSI and Neural Systems. Reading, MA, USA: Addison-Wesley, 1989.
- [3] P. Hasler, "Low-power programmable signal processing," in Proc. Int. Workshop Syst. Chip Real-Time Appl., 2005, pp. 413–418.
- [4] G. Frantz, "Digital signal processor trends," *IEEE Micro*, vol. 20, no. 6, pp. 52–59, Nov./Dec. 2000.
- [5] P. Hasler and D. Anderson, "Cooperative analog-digital signal processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2002, pp. 3972–3975.
- [6] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neural Comput.*, vol. 10, no. 7, pp. 1601–1638, 1998.
- [7] C. Schlottmann and P. Hasler, "FPAA empowering cooperative analogdigital signal processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2012, pp. 5301–5304.
- [8] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA, USA: Addison-Wesley, 1979.
- [9] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. Twigg, and P. Hasler, "A floating-gate-based field-programmable analog array," *IEEE J. Solid-State Circuits*, vol. 45, no. 9, pp. 1781–1794, Sep. 2010.
- [10] C. Schlottmann, S. Shapero, S. Nease, and P. Hasler, "A digitallyenhanced dynamically-reconfigurable analog platform for low-power signal processing," *IEEE J. Solid-State Circuits*, vol. 47, no. 9, pp. 2174–2184, Sep. 2012.
- [11] S. Koziol and P. Hasler, "Reconfigurable analog VLSI circuits for robot path planning," in *Proc. NASA/ESA Conf. Adapt. Hardw. Syst.*, Jun. 2011, pp. 36–43.

- [12] S. Ramakrishnan, A. Basu, L. K. Chiu, J. Hasler, D. Anderson, and S. Brink, "Speech processing on a reconfigurable analog platform," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* [Online]. Available: http://dx.doi.org/10.1109/TVLSI.2013.2241089
- [13] C. Schlottmann, C. Petre, and P. Hasler, "A high-level Simulink-based tool for FPAA configuration," *IEEE Trans. Very Large Scale Integr.* (VLSI) Syst., vol. 20, no. 1, pp. 10–18, Jan. 2012.
- [14] C. Enz, F. Krummenacher, and E. Vittoz, "An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications," *Analog Integr. Circuits Signal Process.*, vol. 8, no. 1, pp. 83–114, 1995.
- [15] K. Odame and P. Hasler, "A bandpass filter with inherent gain adaptation for hearing applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 3, pp. 786–795, Apr. 2008.
- [16] R. Sarpeshkar, T. Delbruck, and C. Mead, "White noise in MOS transistors and resistors," *IEEE Circuits Devices Mag.*, vol. 6, no. 9, pp. 23–29, Nov. 1993.
- [17] C. Schlottmann and P. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The FPAA implementation," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 1, no. 3, pp. 403–411, Sep. 2011.
- [18] S. Chakrabartty and G. Cauwenberghs, "Sub-microwatt analog VLSI trainable pattern classifier," *IEEE J. Solid-State Circuits*, vol. 42, no. 5, pp. 1169–1179, May 2007.
- [19] A. Basu, K. Odame, and P. Hasler, "Dynamics of a logarithmic transimpedance amplifier," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 1673–1676.
- [20] D. Graham, P. Hasler, R. Chawla, and P. Smith, "A low-power programmable bandpass filter section for higher order filter applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 6, pp. 1165–1176, Jun. 2007.
- [21] M. Kucic, A. Low, P. Hasler, and J. Neff, "A programmable continuous-time floating-gate fourier processor," *IEEE Trans. Circuits Syst. I, Analog Digit. Signal Process.*, vol. 48, no. 1, pp. 90–99, Jan. 2001.
- [22] J. Lazzaro, S. Ryckebusch, M. Mahowald, and C. Mead, "Winner-takeall networks of O(n) complexity," in *Advances in Neural Information Processing Systems*, vol. 1. New York, NY, USA: Springer-Verlag, pp. 703–711, 1989.
- [23] S. Ramakrishnan and J. Hasler, "Vector-matrix multiply and winner-take-all as an analog classifier," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* [Online]. Available: http://dx.doi.org/10.1109/TVLSI.2013.2245351

Craig R. Schlottmann (S'07) received the B.S. degree in electrical engineering from the University of Florida, Gainesville, FL, USA, in 2007, and the Ph.D. and M.S. degrees in electrical engineering from the Georgia Institute of Technology (Georgia Tech), Atlanta, GA, USA, in 2012 and 2009, respectively.

He is currently a Research Engineer with Georgia Tech. During his graduate studies, he interned with MIT's Lincoln Laboratory in the field of embedded and high performance computing. His current

research interests include low-power analog signal processing, mixed-signal IC design, and low-power embedded systems.

Dr. Schlottmann was a recipient of the Best Live Demo Award from ISCAS in 2010.

Jennifer Hasler (SM'04) received the B.S.E. and M.S. degrees in electrical engineering from Arizona State University, Phoenix, AZ, USA, in 1991, and the Ph.D. degree in computation and neural systems from the California Institute of Technology, Pasadena, CA, USA, in 1997.

She is currently a Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Her current research interests include low-power electronics, mixed-signal system ICs, floating gate MOS transistors, adaptive information processing systems, smart interfaces for sensors, cooperative analog-digital signal processing, device physics related to submicrometer devices or floating-gate devices, and analog VLSI models of on-chip learning and sensory processing in neurobiology.