EURASIP Journal on Applied Signal Processing 2005:7, 1082–1092

Towards Low-Power on-Chip Auditory Processing

Sourabh Ravindran
Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA
Email: stg@ece.gatech.edu

Paul Smith
Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA
Email: pds@gtronix.com

David Graham
Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA
Email: dgraham@ece.gatech.edu

Varinthira Duangudom
Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA
Email: vduangu@ece.gatech.edu

David V. Anderson
Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA
Email: dva@ece.gatech.edu

Paul Hasler
Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA
Email: phasler@ece.gatech.edu

Received 18 September 2003; Revised 16 August 2004

Machine perception is a difficult problem both from a practical or implementation point of view as well as from a theoretical or algorithmic point of view. Machine perception systems based on biological perception systems show great promise in many areas but they often have processing requirements and/or data flow requirements that are difficult to implement, especially in small or low-power systems. We propose a system design approach that makes it possible to implement complex functionality using cooperative analog-digital signal processing to lower power requirements dramatically over digital-only systems, as well as provide an architecture facilitating the development of biologically motivated perception systems. We show the architecture and application development approach. We also present several reference systems for speech recognition, noise suppression, and audio classification.

Keywords and phrases: low power, noise suppression, classification, speech recognition, cooperative analog digital, feature extraction.

1. INTRODUCTION

This paper describes our efforts toward making on-chip auditory perception systems. With these systems we hope to achieve human-like performance in various auditory tasks. Part of the motivation in this arena comes from the fact that humans outperform machines in most tasks of audio perception; therefore, one way to improve the current machine perception implementations is to mirror biological systems as best as we can in hopes of obtaining comparable results. In keeping with the current mainstream ideas in audio signal processing, the characteristics of biology could be programmed into a digital processing system, and the results would likely be good. However, this is at the cost of high power consumption and longer time requirements due to the computational complexity. It is time we looked at different perspectives of performing these tasks both in terms of hardware innovation and algorithm development [1].

A major consideration while designing on-chip machine perception is the flexibility, size, and power requirements of

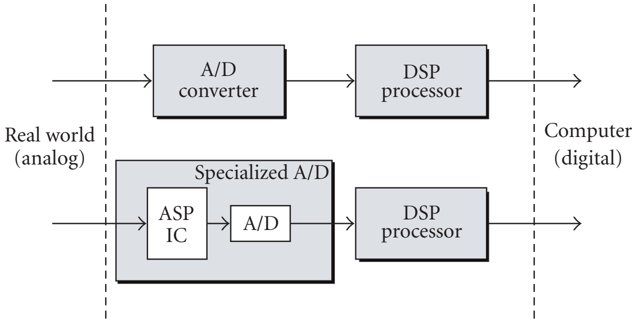


Figure 1: The traditional approach is to put an analog-to-digital (A/D) converter as close to the sensor signals as possible, implementing nearly all functionality in digital systems. An alternate approach is to perform some of the computations using advanced analog signal processing, requiring simpler A/D converters, and reducing the computational load of resulting digital processors. The question of where to put this boundary line strongly depends upon the particular requirements of an application.

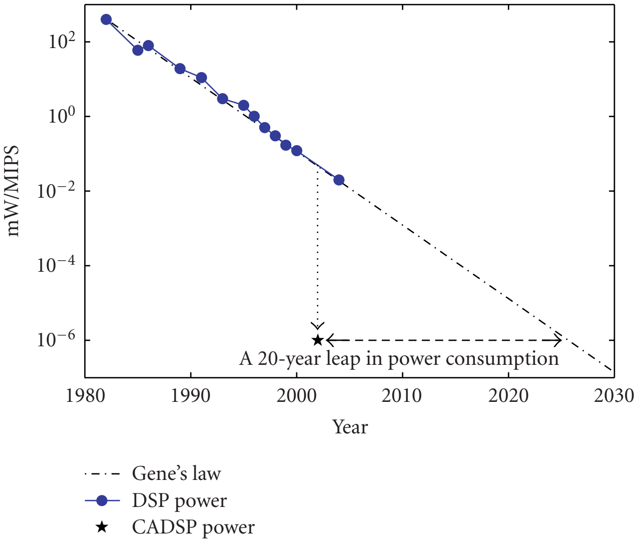


Figure 2: Data showing the power consumption trends in DSP microprocessors [3] along with data taken from a recent analog signal processing IC [4, 5, 6].

the actual hardware. In order to address these issues, we describe a two-pronged approach to developing on-chip solutions using both programmable analog processing and digital signal processing (an approach that we call CADSP for cooperative analog-digital signal processing [2]). The resulting systems are very low-power, small, and can efficiently implement a large class of signal processing systems.

In this paper, we present CADSP as a vehicle for implementing auditory perception on a chip. The rest of the paper is divided into four main sections. Section 2 discusses the system technology. Section 3 introduces signal processing algorithms that can be implemented using this technology and some results. Section 4 gives a comparative analysis between CADSP and digital approaches. Section 5 discusses future work towards the goal of implementing all stages (noise suppression, feature extraction, symbol representation, and higher-end processing such as recognition) in the signal processing pathway on the CADSP platform.

2. CADSP SYSTEM OVERVIEW

Our approach to machine perception is unique in that it introduces a novel hardware processing platform [4, 7] that is low power, programmable, and versatile. In the works is the development of a prototyping platform [8] that would considerably reduce the hardware design cycle. The next section introduces some of the basic ideas and guiding principles of CADSP.

2.1. Introduction to CADSP

The CADSP approach addresses the problem of runaway hardware complexity by adding increased functionality to the analog subsystem. For example, offloading the task of extracting signal features to the analog circuits can reduce the analog-to-digital converter requirements (i.e., reduction in bits of resolution and bandwidth required) and reduce the load on the digital processor. This is in contrast to the recent trend of moving the analog-to-digital converter earlier in the signal path and doing increasingly more computation digitally. However, there are many operations that may be easier, cheaper, and more efficient to perform with analog circuits. CADSP allows for freedom of movement of the partition between the analog and digital systems. By adding functionality to our analog systems, we enhance the capabilities of the controlling digital system, and therefore, the entire product under consideration (see Figure 1). Many of the tradeoffs have yet to be investigated, but we include a brief discussion of power, size, and resolution.

The CADSP approach of building machine perception systems promises increased algorithmic sophistication at substantially reduced power consumption. In particular, a power reduction of 3–4 orders of magnitude over currently available digital signal processing (DSP) systems is expected for certain classes of systems. As shown in Figure 2, such a gain in power efficiency is approximately equal to a 20-year leap ahead relative to the projected power efficiency of digital signal processors.

The analog circuits we present are very small—a characteristic made possible by CMOS floating-gate technology which allows for easy tuning and programming of the circuits. Each analog system discussed in this paper occupies less than 2.25 mm^2 in a standard (digital) CMOS 0.5 μm process. These circuits are generally operated in subthreshold mode, yielding tremendous power savings.

It has been previously shown [9] that analog computation has significant advantage if the resolution of incoming information is not too high, typically 10–12 bits or less. We can safely say that doing more in digital hardware generally increases both flexibility and power consumption and, beyond a certain point, can yield increased accuracy, whereas analog implementations of parts of a system generally result in significant power savings and space savings at the expense of flexibility.

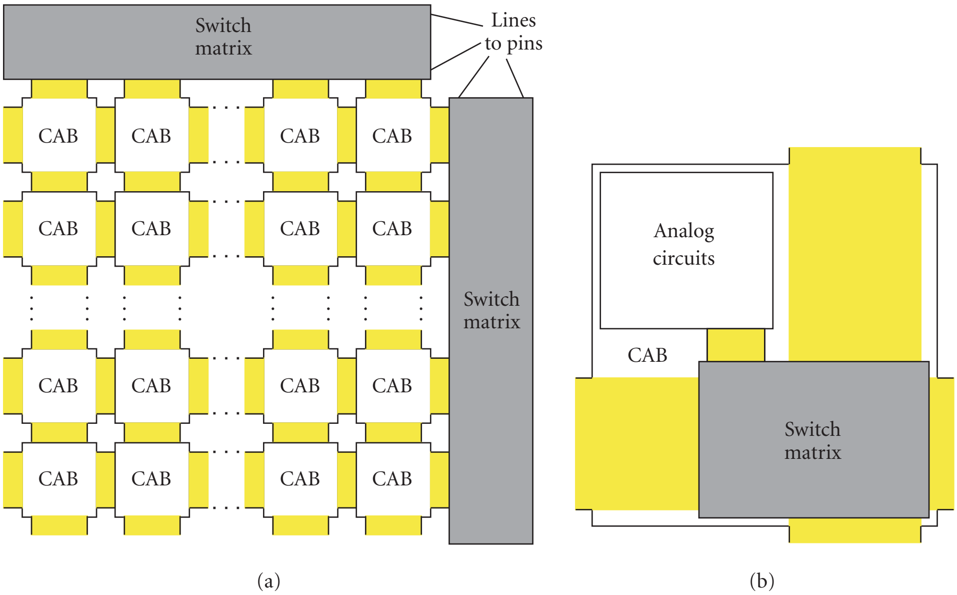


Figure 3: Block diagram of our generic FPAA chip. (a) Architecture diagram of our generic analog programmable computing blocks. Although in many cases, a generic, mesh-connected structure will be inefficient for many computations, small mesh-connected arrays will be important in more optimized implementations. Also, this chip will allow for partitioning between medium-scale analog-digital design problems. We expect to build a generic FPAA block with 100 computational analog blocks (CAB), each with 24 ports, to the mesh-switching matrix. (b) Signal flow in a single CAB.

When implementing machine perception on a chip, the use of low-power analog circuits provides increased computational ability while requiring less space and power. In particular, this is useful for creating biologically motivated perception systems, since the complexity involved in such systems often makes them difficult to implement in low-power systems. For instance, implementation of the biologically motivated modulation spectra representation [10] would be relatively straightforward to implement in a CADSP system but it requires extensive computation when implemented in programmable digital systems.

2.2. Floating gates overview

The core elements of the analog systems are CMOS analog floating-gate devices. These elements add programmability and increased flexibility and power savings to analog VLSI circuits [11]. A floating gate is a polysilicon gate surrounded by silicon dioxide. Charge on the floating gate is stored permanently, providing a long-term memory.

Floating gates are programmable in that the charge can be modified using several methods. Electrons are removed from the gate by applying large voltages across a silicon-oxide capacitor to tunnel electrons through the oxide. Electrons are added to the floating gate using hot-electron injection. Programming of floating gate is accomplished by resetting the floating-gate charge using electron tunneling, and setting positive or negative offsets using hot-electron injection [12].

The floating gate can be used in computation since the voltage, determined by the charge stored on the floating gate, can modulate a channel between a source and drain. Floating-gate devices can compute a wide range of static and dynamic translinear functions by the particular choice of capacitive couplings into floating-gate devices. This programmable analog CMOS technology is the basis for programmable and reconfigurable CADSP system.

2.3. Prototyping platform

The process of designing, fabricating, and testing an analog chip requires certain expertise and is often long and expensive. The process is not unlike designing digital ASICs (application specific integrated circuits) except there are fewer tools and libraries available to the designer. The difficulties in digital ASIC design are largely ameliorated by using field-programmable gate arrays (FPGAs). For digital circuits, FPGAs provide a fast, simple way to implement, test, and eventually compile custom circuits. We have built programmable floating-gate analog devices that are similar to FPGAs in both design and in the benefits they provide. The analog arrays, dubbed field-programmable analog arrays or FPAAs, represent a significant advance in the field of analog signal processing [8]. These chips are based on structures of floating-gate analog circuits and they may contain tens of thousands of analog elements as illustrated in Figure 3.

The arrays of analog circuits on which the FPAAs are based consist of matrix/vector multiplies, filtering, and so forth—not just a few op-amps with passive elements as with commercial and other research FPAAs [13, 14, 15, 16, 17, 18, 19]. Relative to custom-designed analog circuits, a design implemented in an FPAA will always result in higher parasitics as well as increased die area for a given design; therefore,

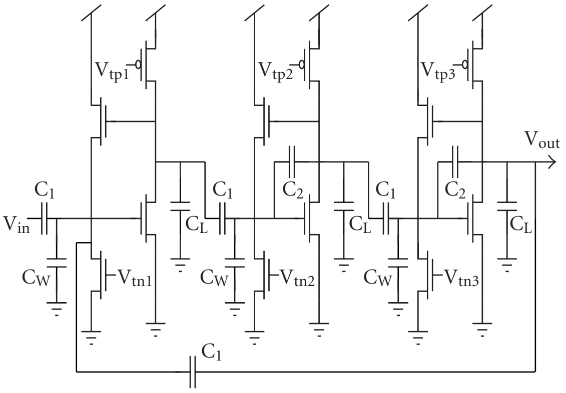


Figure 4: C^4 second-order section circuit diagram.

the design will always possess some inefficiencies. On the other hand, since analog-circuit design is often time consuming, these adverse tradeoffs are well balanced by decreased time to market. These FPAA chips are mixed-mode chips because the data-flow control and the programming and interface control to the floating-gate devices are digital. Currently, we have well-defined digital control and protocols for programming and controlling arrays of floating-gate devices [20]. Figure 3a shows a block diagram of the generic FPAA mesh architecture. Figure 3b shows the interconnection in a single CAB element; we connect the buses from the four edges of the chip and internal bus lines from the analog circuitry through a single switch matrix [8].

The development and use of FPAAs as described here is an important step toward designing and deploying flexible CADSP systems. We are in the process of building specialized FPAAs for audio signal processing that incorporate microphone inputs, distributed filter banks, distributed analog-computational memory arrays, CAB blocks, and sigma-delta analog-to-digital (A/D) converters. The CAB blocks used with this chip will also contain some special circuitry such as circuits for active noise suppression (as described in the next section) and adaptive filtering.

3. APPLICATIONS

In this section, we present various audio processing applications that can be implemented using CADSP. We present results from fabricated chips for some of the applications and simulation results for the rest.

3.1. Bandpass filtering

Spectrum decomposition is invariably the first step in most audio processing applications. In this section, we present a continuous-time bandpass filter element called the capacitively coupled current conveyer (C^4) (Figure 4). Measurement data from circuits fabricated on a 0.5 μm n-well CMOS process is shown for frequency ranges from 10 Hz to 100 kHz (Figure 5).

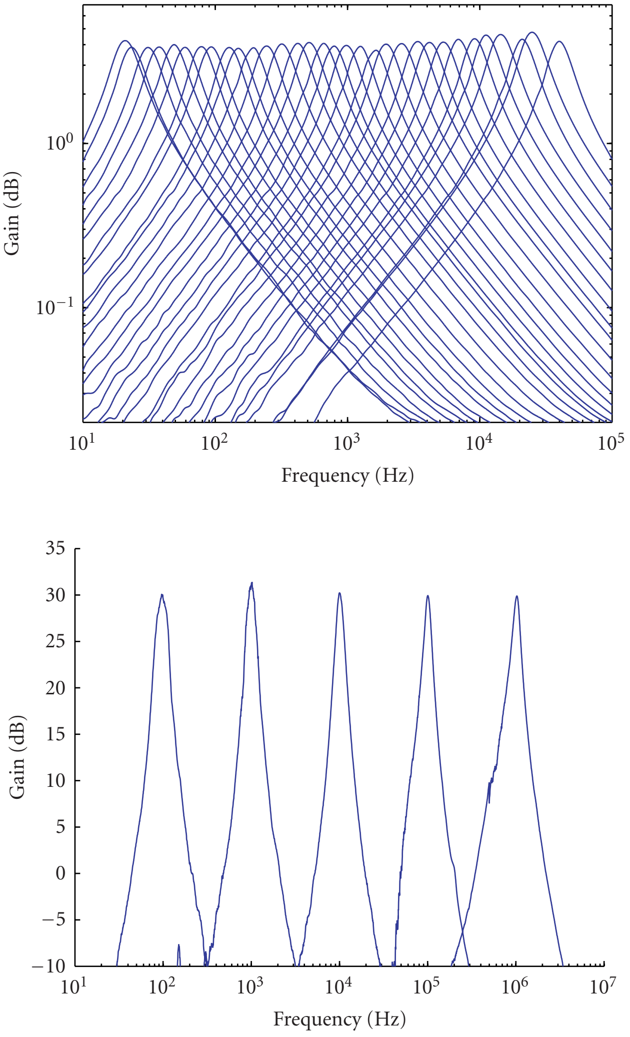
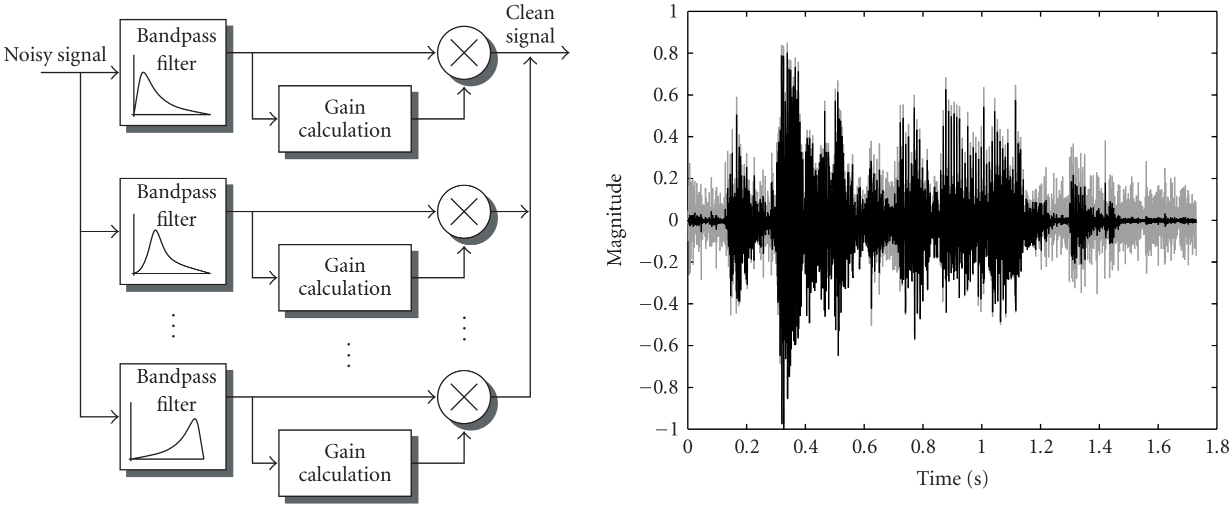


Figure 5: Programmed differential weights to the floating-gate multiplier circuits (a single sine period).

C^4 is a bandpass filter with electronically tunable corner frequencies and ±40 dB/decade or greater rolloffs. The corner frequencies can be set independent of each other; therefore, the bandwidth can be tuned at will. Each corner can have its own Q-peak, or, if the corners are brought close together, a very tight bandwidth with a single Q-peak develops (see Figure 5). This leads to further isolation of any given frequency and is thus useful for signal processing applications. An array of these C^4 second-order sections (SOS) with exponentially spaced center frequencies forms a good model of the human cochlea where signals are decomposed with filtering processes that have $Q \approx 30$ [21].

3.2. Noise suppression

Audio signal enhancement by removing additive background noise from a corrupted noisy signal is not a new concept. While most noise suppression methods are focused on the processing of discrete-time sampled audio signals, we use a



(a)

(b)

Figure 6: Our continuous-time noise suppression system. (a) The overall structure of the system. The incoming noisy signal is divided into exponentially spaced frequency bands using C^4 second-order sections. Next, the optimal gain (gain calculation block) for each band is computed. If the band has sufficient estimated SNR, the signal passes through with maximal gain, otherwise the gain is reduced dependent upon the estimated SNR in that particular band. The resulting gain factor is multiplied with the bandlimited noisy signal to produce a bandlimited "clean" signal. Finally, the output of all of the bands is summed to reconstruct the signal with the noise components significantly reduced. (b) Experimental measurements of noise suppression in one frequency band. The light gray data is the subband noisy speech input signal; the black waveform is the corresponding subband output after the gain function has been applied. The noise-only portions of the signal have been significantly attenuated while a lesser attenuation is applied appropriately to the speech + noise portions.

technique for noise suppression in the continuous-time domain. We are building a system that operates in real time and uses extremely low amounts of power. The result is a system that performs a function normally reserved for digital computation, freeing those resources for other operations in the digital domain.

3.2.1. Structure of suppression system

Figure 6a shows the structure of a continuous-time noise suppression system for real-time analog implementation. The goal is to design a real-time system that generates some optimal estimate of the actual signal from an additive mixture of signal and noise. We assume that the additive noise is stationary over a long time period relative to the short-term nonstationary patterns of normal speech. A filter bank separates the noisy signal into 32 bands that are exponentially spaced in frequency, similar to the human auditory system for frequency domain processing.

After the incoming noisy signal has been bandlimited by the filter bank, a gain factor is calculated based on the the envelopes of each observed subband signal and subband noise signal. The first step in the gain calculation algorithm is to estimate both the levels of the noisy signal and the noise (using a minimum statistics approach). Because one cannot accurately determine the desired signal component of the incoming signal, the noisy signal is accepted as a reasonable estimate. Currents that are representative of the noisy signal level and the noise level are divided to create an estimate for SNR. Within each frequency band, the noisy signal envelope is estimated using a peak detector. Based on the voltage output of the peak detector, the noise level is estimated using a minimum detector operating at a slower rate than the peak detector. The currents representing the noisy signal and noise levels are input to a translinear division circuit, which outputs a current representing the estimated signal-to-noise ratio. A nonlinear function is applied to the SNR current to calculate a gain factor.

Finally, a gain factor is calculated and multiplied with the bandlimited signal to realize the noise suppression. The gain factor may be expressed as a function of the estimated SNR. Several different gain functions may be used but all of them have the general characteristics of low gain for low SNR and high gain (at or near unity) for high SNR, with varying smoothing between these two regions. The output of the gain function is then multiplied with the original bandlimited signal. Finally, the bandlimited signals are summed to reconstruct the full-band signal estimate, with the additive noise components suppressed.

3.2.2. System results

The experimental results presented in this paper are from tests on individual components that have not yet been integrated into a larger system. Figure 6b shows a noisy speech signal that has been processed by the components in our system. The system is effective at adaptively reducing the amplitude of noise-only portions of the signal while leaving the

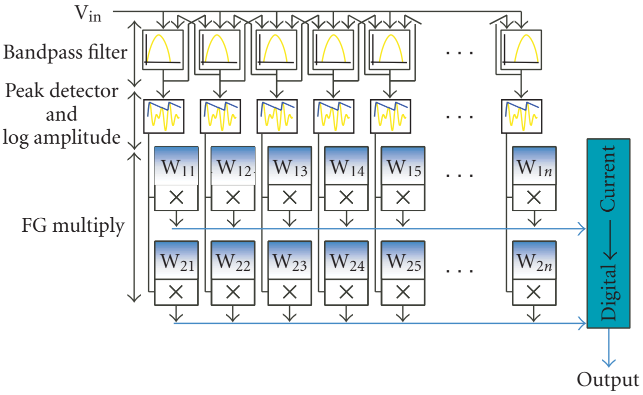


Figure 7: Block diagram of a floating-gate system to perform cepstrum front-end computation for speech processing systems. The system contains 32 frequency taps that can be spaced arbitrarily by programming the corner frequencies for the bandpass filter banks. The peak detectors provide a power spectrum of the input signal for any given time slice.

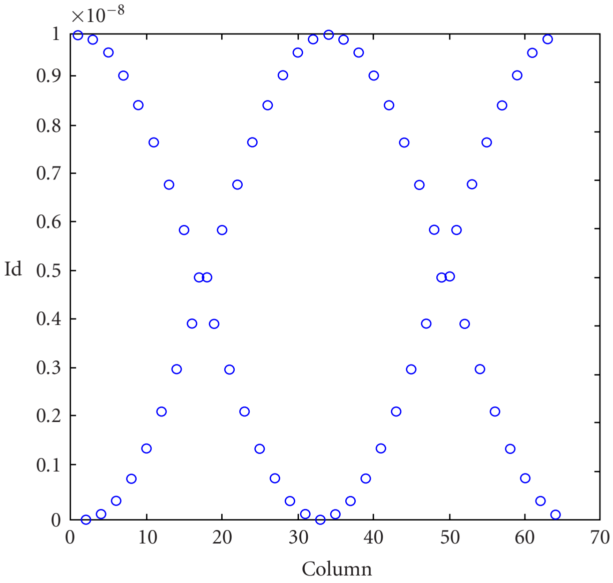


Figure 8: Programmed differential weights to the floating-gate multiplier circuits (a single sine period).

desired portions relatively intact. Any noise or distortion created by the gain calculation circuits minimally affects the output signal because these circuits are not directly in the signal path. While the bandpass filters and the multipliers will inject a certain amount of noise into each frequency band, this noise will be averaged out by the summation of the signals at the output of the system.

3.3. Feature extraction for audio processing

This section discusses our current work on a continuous-time mel-frequency cepstrum encoding IC using analog circuits and floating-gate computational arrays (more detail is given in [6]). We also introduce biologically inspired auditory features that are similar to the standard mel-frequency cepstrum coefficients (MFCCs) but appear to be more robust to noise.

3.3.1. Continuous-time cepstrum

The mel-cepstrum is often computed as the first stage of a speech recognition system [22]. The mel-cepstrum, as used in digital signal processing (DSP), is based on a signal sampled in time and in frequency. Figure 7 shows the block diagram for the analog cepstrum which is an approximation to the mel-cepstrum. The output of each filter contains information similar to the short-time Fourier transform and can likewise be assumed to represent the product of the excitation and vocal tract within that filter band. The primary difference here is that the DSP mel-cepstrum approximates the critical band log-frequency analysis of the human ear by combining DFT bands while the analog system actually performs a critical band-like analysis on the input signal. Thus, higher-frequency critical band energies are effectively computed using shorter basis functions than the lower-frequency bands [6]. This is more in agreement with analysis in the human auditory system and is better suited to identifying transients.

The basic building block of the continuous-time cepstrum implementation begins with a continuous spectrum decomposition and amplitude detection, similar to a discrete fourier transform (DFT). The spectrum decomposition is done using differential C^4 second-order section bandpass filters [21]. The magnitude function (inside the log) is estimated using a peak detector rather than using the true magnitude of the complex spectrum. Finally, we compute a cosine transform on these results using a matrix transform based on arrays of floating-gate multiplier circuits. Figure 8 shows the 32 programmed weight values (difference between a positive and negative weight) for a single row of multipliers programmed to a sine function.

This cepstrum processor can act as the front end for larger digital or analog speech processing systems. Early data from a related project gives confidence that this approach will improve the state of the art at a given power-dissipation level [23].

3.3.1.1. Simulation results

The cepstral coefficients were extracted using a Matlab simulation of the analog cepstrum chip. A hidden Markov model-(HMM-)based recognizer was used to perform continuous word recognition on the TIDIGITS database. The tests yielded a 98.2% word accuracy, which is comparable to that obtained with standard MFCCs.

3.3.2. MFCC-like auditory features

This section discusses the extraction of biologically inspired features for the task of audio classification.

Yang et al. [24, 25] have presented a biophysically defensible mathematical model of the early auditory system. The mathematical model consists of three stages—the analysis stage that performs the filtering in the cochlea; the transduction stage which mimics the inner-hair-cell stage and is

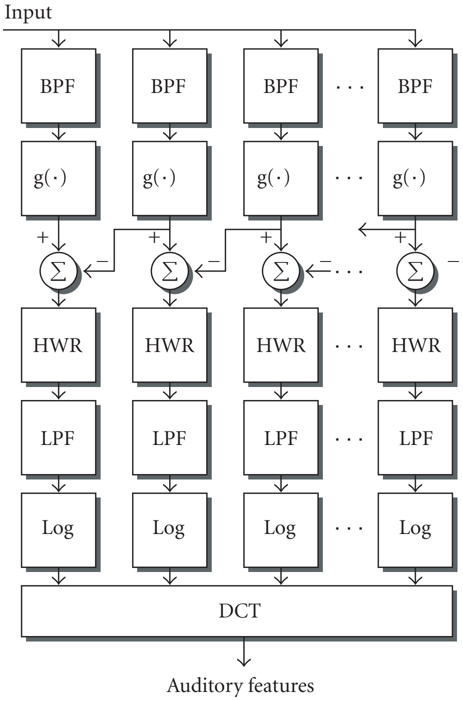


Figure 9: The bandpass filtered version of the input is nonlinearly compressed. The differencing between lower and higher channels approximates a spatial derivative. The half-wave rectification followed by the smoothing filter picks out the peak. DCT is performed to decorrelate the signal.

comprised of a time derivative, a nonlinear compression, and a lowpass filter; and the reduction stage which consists of a spatial derivative, a spatial lowpass filter, a half-wave rectifier, and a temporal integrator. The reduction stage models the lateral inhibition in the cochlear nucleus (CN) and also the inability of the CN neurons to respond to fast temporal changes.

When viewing the way the cochlea acts on signals of different frequencies from an engineering perspective, it can be seen that the cochlea has bandpass frequency responses for each location. Ideally, one would have a bandpass filter with center frequency corresponding to the resonant frequency of every point along the cochlea—the cochlea has about 3000 inner hair cells acting as transduction points. In practice, 10–20 filters per octave are considered an adequate approximation.

The block diagram for extraction of MFCC-like auditory features is shown in Figure 9. The input speech signal is passed through the bandpass filter bank, the 40 dB/decade rolloff on the low frequency along with modeling the cochlear filter also provides a time differentiation of the input signal which models the inner hair cell coupling to the traveling wave. Each of the bandpass filters is built with C^4 SOS. Following the bandpass filtering, each bandpassed signal is then nonlinearly compressed followed by a difference with the adjacent channels. The difference between channels is an approximation of the spatial derivative that models the lateral inhibition of the CN neurons. This is followed by half-wave rectification and a smoothing filter. We then proceed to compute a cosine transform of the logarithm of the output of the smoothing filter.

3.3.3. Simulation results

A Matlab simulation of the above-described setup was used to extract the features. The new features were used for a four class audio classification problem to test their discriminant capabilities. The database used for the classification consisted of 4325 training files and 1124 testing files each of a duration of one second. The four classes considered were speech, music, noise, and animal sounds. A Gaussian mixture model (GMM) was used to perform the classification and four Gaussians were used for modeling each of the classes. A classification accuracy of 92.97% was achieved. Tests at different SNRs have shown these features to be robust to noise [26].

3.4. Continuous-time VQ

The following section discusses our current work on continuous-time vector quantization IC. Experimental data is presented from circuits fabricated on a 0.5 μm n-well CMOS process available through MOSIS. In this section, we provide an overview of vector quantization (VQ), which is typically used in data compression and in classifying signals to symbols [27]. A VQ system will compute how faraway a particular input vector is from the desired target vectors, and pick the code vector that is *closest* to the input vector. For VQ, some information is lost in the representation, but the goal is that it should be a sufficient representation for the problem at hand.

Figure 10 shows the circuit and measured data from the VQ classifier array. Each cell in the array compares the value of that column's input to the value it has memorized; the output current flows out of the V_{out} node. This circuit is a variation on the bump circuit [28], which compares the two inputs to this circuit; this cell returns a large current if the two values match (minimal difference). This system outputs a measure of the similarity; therefore, the outputs of all the elements can be added (by KCL) and the largest output is the vector with the maximum similarity. The sum of these current outputs is sent through a winner-take-all circuit that outputs the N largest results, where N can be 1 or more [29].

We utilize floating gate elements [30] at the inputs to provide the ability to store and subtract off each cell's mean value. Figure 10b shows that the means in a VQ array can be programmed to an arbitrary level.

Figure 10c shows the circuit and architecture for our adaptive VQ system [31, 32]; we adapt the floating-gate charge to the mean of the input signal by allowing an equilibrium to be set as a result of continuous tunneling and continuous injection currents. This equilibrium sets the input offset voltage equal to the incoming signal mean. To get a stable adaptive differential pair in both differential and common mode, both drain voltages of the differential pair should be connected to transistor current sources. We stabilize the current sources using a common-mode feedback (CMFB) circuit. We incorporate our CMFB circuit on the bottom of

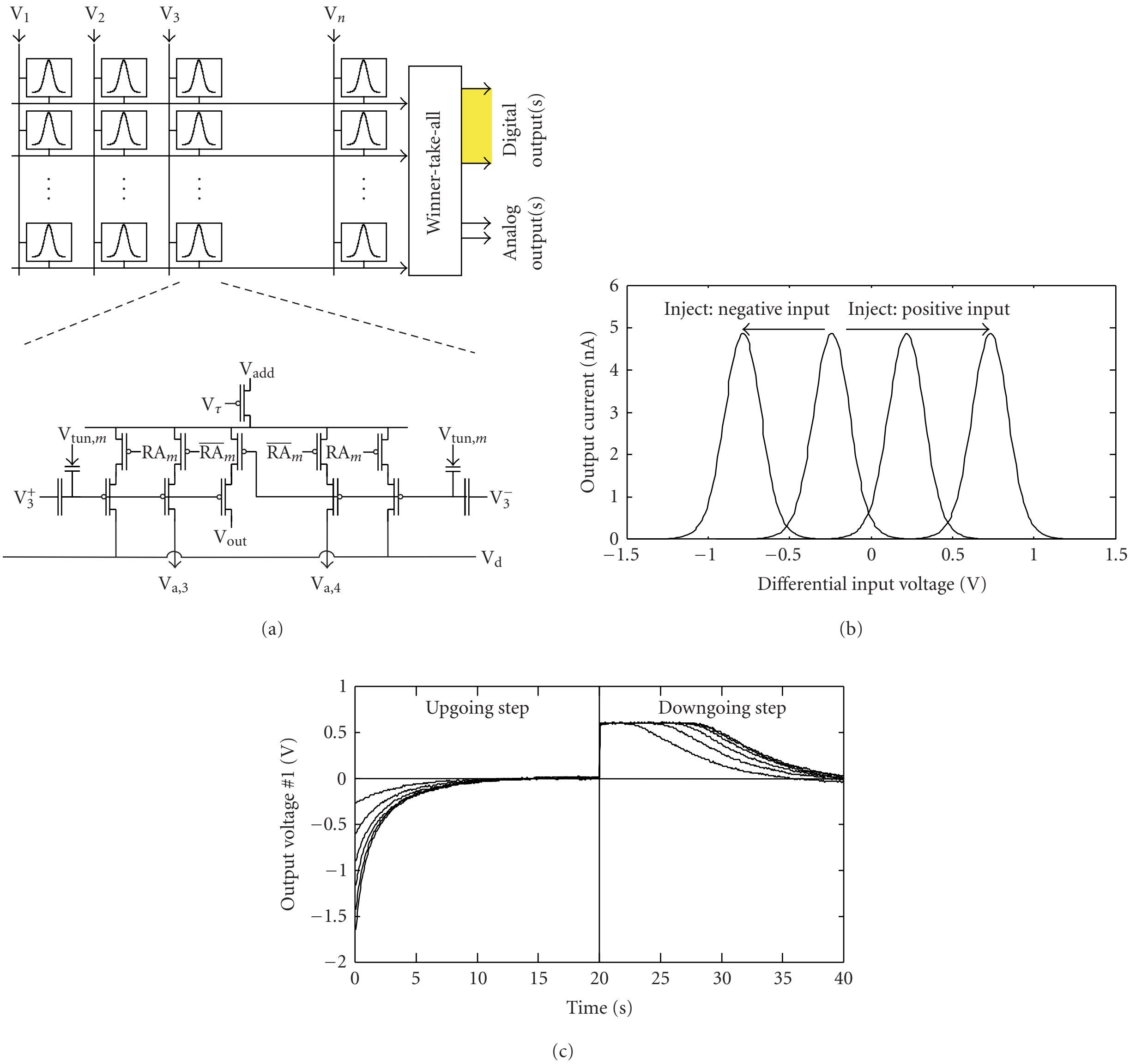


FIGURE 10: Basic circuit, architecture, and measurements from the VQ circuit. (a) The core cell is built from a floating-gate bump circuit, which allows the target mean value to be stored and subtracted from the broadcast input signal. (b) Results from programming the VQ circuit. We see that output current (experimental measurements) of the middle leg of the bump circuit reaches a maximum at its center value, and falls off exponentially as one moves from that center value. This output current is summed together with the output from other bump circuits. We use the RA_m signal to select between adaptation or computation/programming along a given row; if only programming and computation are required, then the circuit can be significantly reduced. We reconfigure the VQ circuit so that it fits within the standard floating-gate programming architecture and algorithms [12]. We reset the floating-gate charge using electron tunneling, and program positive or negative offsets using hot-electron injection. If we inject the floating gate associated with the positive input terminal, then we increase the offset. If we inject the floating gate associated with the negative input terminal, then we decrease the offset. (c) The results of adapting the input signal mean. The common-mode feedback (CMFB) circuitry is switched in from the bottom of the array. We show experimental measurements showing the convergence of the floating-gate bump element with the CMFB circuitry. We show one drain voltage when connected to the CMFB circuitry; if the drain voltage reaches equilibrium between the operating rails, then the circuit has converged to the signal mean.

the array, and connect to the selected element for adaptation; typically we would only be adapting one element at a given time. This approach requires some circuit reconfiguration at the core cell; if only adaptation or programming would be used, then the circuit remains simpler than shown in Figure 10a.

4. COMPARATIVE ANALYSIS

In this section, we discuss the tradeoffs between analog and digital implementations. We also provide a comparative analysis of our CADSP approach with the standard digital approach for an audio feature extraction application.

Sarpeshkar [33] presents a good comparative study of digital and analog systems. Power and area gains in analog systems come primarily from the fact that one wire represents many bits of information, but a digital system is restricted to one bit per wire. By using the device physics such as the Kirchoff current laws, operations like 8-bit addition can be made very cheap, essentially using only one wire. In digital it would require an 8-bit adder. Similarly an 8-bit multiply takes about 3000 transistors to implement in digital logic using static CMOS whereas in analog it can be implemented using 4–8 transistors. Since digital computations use more number of devices, there is more wiring and computation overhead which in turn takes up more area and dissipates power.

The disadvantage of analog systems is that of noise. The noise in analog systems is essential due to thermal noise and in a cascade of components the thermal noise adds up. Additional compensation circuitry needs to be added to counter these offsets and these additional circuits can get expensive in terms of power and area. Typically for SNRs lower than 60 dB, analog computation is more efficient compared to its digital implementation. CADSP is the optimal approach in that it allows us to harness the low power and smaller area of analog computations to perform low signal-to-noise ratio (SNR) computations and thereby reducing the load on the digital processor that follows.

MFCCs are the standard features used in state-of-the-art audio signal processing. MFCC computation by a digital processor would involve as a first step, converting the incoming signal to a digital representation, computing the fast fourier transform (FFT), weighting the frequencies to group them on the mel-scale, taking a logarithm, and computing the discrete cosine transform (DCT). If we assume a sampling rate of 16 kHz and frame length for feature extraction as 20 msec with 50% overlap, the digital approach would take about 9×10^5 fixed-point operations per second. On a DSP processor performing at 20 MIPS/mW, this would result in a power dissipation of 45 μW. But in order to get about 10 bits of resolution in some of the frequency channels, we would need a 16-bit A/D converter at the front end.

For the analog approach, we use a 32-channel bandpass filter bank with exponentially spaced corner frequencies for the frequency decomposition. The frequency decomposition is followed by a peak detector circuit and a log compression. The cosine transform is performed by storing weights on the nodes of a floating gate multiplier. The filter bank and peak detector circuit consume about 1.5 μW and 3 μW, respectively. The log operation is essentially free in the analog implementation. The multiplier consumes about 30 nW of power. Since we used a bandpass filter upfront the bandwidth and resolution requirements of the A/D converters are reduced. We just need to use 32 10-bit A/D converters operating at 100 Hz (or one converter operating at 3200 Hz) since we need to sample just one coefficient per frame. Assuming that the 16-bit and 10-bit A/D converters have to maintain the same figure of merit defined by [34]

$$F = \frac{2^R f_{\text{samp}}}{P_{\text{diss}}}, \qquad (1)$$

where R is the resolution of the ADC in bits, f_{samp} is the sampling frequency, and P_{diss} is the power dissipated. It is clear that we have a reduction in power by a factor of approximately 2×10^3.

5. FUTURE WORK

By implementing the auditory processing in CADSP circuitry, we achieve enough power and computational gain to start thinking about implementing algorithms and techniques hitherto restricted due to these constraints.

Biologically inspired feature extraction leads to a much richer feature space. MFCC-like auditory features are just the first step towards harnessing the strengths of biological representation of signals. Wang et al. [35] have shown that modelling the auditory cortex yields features that open up tremendous possibilities of future applications. This leads us to consider designing back-end systems that can fully exploit this representation. Towards this goal we plan to develop HMM-like recognizers that are inspired by the dendritic computations in biological systems, and we have made initial strides in this direction. An ambitious yet plausible aim would be to build an entire biologically inspired automatic speech recognition system on a CADSP platform.

6. CONCLUSION

In this paper, we introduced CADSP systems based on programmable analog floating-gate circuits as a means for implementing audio perception on a chip. The CADSP system approach makes it possible to realize increasingly complex biological models of perception without paying a large price in power or size. System results for several different audio tasks and results of the building blocks of others were presented to illustrate the potential of this approach. A rapid prototyping platform that is currently under development was also shown; this is an important part of designing future systems. By using the systems and design approach described, we hope to make complex machine perception on a chip possible.

ACKNOWLEDGMENT

This research was funded in part by HP Labs and grants from NSF.

REFERENCES

[1] C.-H. Lee, "On automatic speech recognition at the dawn of the 21st century," *IEICE Transactions on Information and Systems*, vol. E86-D, no. 3, pp. 377–396, 2003.

[2] P. Hasler and D. V. Anderson, “Cooperative analog-digital signal processing,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP ’02)*, vol. 4, pp. 3972–3975, Orlando, Fla, USA, May 2002.

[3] G. Frantz, “Digital signal processor trends,” *IEEE Micro*, vol. 20, no. 6, pp. 52–59, 2000.

[4] P. Hasler, P. Smith, R. Ellis, D. Graham, and D. V. Anderson, “Biologically inspired auditory sensing system interfaces on a chip,” in *Proc. IEEE Sensors Conference*, vol. 1, pp. 669–674, Orlando, Fla, USA, June 2002, invited paper.

[5] R. Ellis, H. Yoo, D. Graham, P. Hasler, and D. V. Anderson, “A continuous-time speech enhancement front-end for microphone inputs,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS ’02)*, vol. 2, pp. 728–731, Phoenix-Scottsdale, Ariz, USA, May 2002.

[6] P. Smith, M. Kucic, R. Ellis, P. Hasler, and D. V. Anderson, “Mel-frequency cepstrum encoding in analog floating-gate circuitry,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS ’02)*, vol. 4, pp. 671–674, Phoenix-Scottsdale, Ariz, USA, May 2002.

[7] P. Hasler, A. Bandyopadhyay, and D. V. Anderson, “High fill-factor imagers for neuromorphic processing enabled by floating-gate circuits,” *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 7, pp. 676–689, 2003.

[8] T. S. Hall, P. Hasler, and D. V. Anderson, “Field-programmable analog arrays: a floating-gate approach,” in *Proc. 12th International Conference on Field Programmable Logic and Applications (FPL ’02)*, pp. 424–433, Montpellier, France, September 2002.

[9] R. Sarpeshkar, *Efficient precise computation with noisy components: extrapolating from an electronic cochlea to the brain*, Ph.D. thesis, California Institute of Technology, Pasadena, Calif, USA, 1997.

[10] L. Atlas and S. A. Shamma, “Joint acoustic and modulation frequency,” *Eurasip Journal on Applied Signal Processing*, vol. 2003, no. 7, pp. 668–675, 2003.

[11] P. Hasler and T. S. Lande, “Special issue on floating-gate devices, circuits, and systems,” *IEEE Trans. Circuits Syst. II*, vol. 48, no. 1, 2001.

[12] P. Smith, M. Kucic, and P. Hasler, “Accurate programming of analog floating-gate arrays,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS ’02)*, vol. 5, pp. 489–492, Phoenix-Scottsdale, Ariz, USA, May 2002.

[13] K. F. E. Lee and P. G. Gulak, “A transconductor-based field-programmable analog array,” in *Proc. IEEE International Solid-State Circuits (ISSCC ’95): Digest of Technical Papers*, pp. 198–199, San Francisco, Calif, USA, February 1995.

[14] K. F. E. Lee and P. G. Gulak, “A CMOS field-programmable analog array,” in *Proc. IEEE International Solid-State Circuits (ISSCC ’95): Digest of Technical Papers*, pp. 186–188, San Francisco, Calif, USA, February 1991.

[15] S. T. Chang, B. R. Hayes-Gill, and C. J. Paul, “Multi-function block for a switched current field programmable analog array,” in *Proc. 39th Midwest Symposium on Circuits and Systems*, Iowa State University, Ames, Iowa, USA, August 1996.

[16] D. V. Anderson, C. Marcjan, D. Bersch, et al., “A field programmable analog array and its application,” in *Proc. IEEE Custom Integrated Circuits Conference (CICC ’97)*, pp. 555–558, Santa Clara, Calif, USA, May 1997.

[17] Lattice Semiconductor Corporation, Hillsboro, Ore, USA, *ispPAC Overview*, September 1999.

[18] X. Quan, S. H. K. Embabi, and E. Sanchez-Sinencio, “A current-mode based field programmable analog array architecture for signal processing applications,” in *Proc. IEEE Custom Integrated Circuits Conference (CICC ’98)*, pp. 277–280, Santa Clara, Calif, USA, May 1998.

[19] Fast Analog Solutions, Ltd., Oldham, UK, *Totally reconfigurable analog circuit (TRAC)*, March 1999, issue 2.

[20] M. Kucic, P. Hasler, J. Dugger, and D. V. Anderson, “Programmable and adaptive analog filters using arrays of floating-gate circuits,” in *Proc. Conference on Advanced Research in VLSI (ARVLSI ’01)*, E. Brunvand and C. Myers, Eds., pp. 148–162, IEEE Computer Society, Salt Lake City, Utah, USA, March 2001.

[21] D. Graham and P. Hasler, “Capacitively-coupled current conveyer second-order section for continuous-time bandpass filtering and cochlea modeling,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS ’02)*, vol. 5, pp. 485–488, Phoenix-Scottsdale, Ariz, USA, May 2002.

[22] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*, Macmillan, Englewood Cliffs, NJ, USA, 1993.

[23] T. M. Massengill, D. M. Wilson, P. Hasler, and D. Graham, “Empirical comparison of analog and digital auditory preprocessing for automatic speech recognition,” in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS ’02)*, vol. 5, pp. 77–80, Phoenix-Scottsdale, Ariz, USA, May 2002.

[24] X. Yang, K. Wang, and S. Shamma, “Auditory representations of acoustic signals,” *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 824–839, 1992.

[25] K. Wang and S. Shamma, “Self-normalization and noise-robustness in early auditory representations,” *IEEE Trans. Speech Audio Processing*, vol. 2, no. 3, pp. 421–435, 1994.

[26] S. Ravindran, D. V. Anderson, and M. Slaney, “Low-power audio classification for ubiquitous sensor networks,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP ’04)*, vol. 4, pp. 337–340, Montreal, Canada, May 2004.

[27] J. Schürmann, Ed., *Pattern Classification: a Unified View of Statistical and Neural Approaches*, John Wiley & Sons, New York, NY, USA, 1996.

[28] T. Delbruck, “Bump circuits for computing similarity and dissimilarity of analog voltages,” in *Proc. International Joint Conference on Neural Networks (IJCNN ’91)*, vol. 1, pp. 475–479, Seattle, Wash, USA, July 1991.

[29] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead, *Advances in Neural Information Processing Systems 1*, chapter Winner-take-all networks of $O(N)$ complexity, pp. 703–711, Morgan Kaufman Publishers, San Mateo, Calif, USA, 1988.

[30] P. Hasler and T. S. Lande, “Overview of floating-gate devices, circuits, and systems,” *IEEE Trans. Circuits Syst. II*, vol. 48, no. 1, pp. 1–3, 2001.

[31] P. Hasler, P. Smith, C. Duffy, C. Gordon, J. Dugger, and D. V. Anderson, “A floating-gate vector-quantizer,” in *Proc. IEEE Midwest Circuits and Systems*, Tulsa, Okla, USA, August 2002.

[32] P. Hasler, “Continuous-time feedback in floating-gate MOS circuits,” *IEEE Trans. Circuits Syst. II*, vol. 48, no. 1, pp. 56–64, 2001.

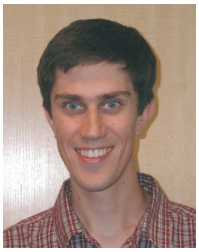
[33] R. Sarpeshkar, “Analog versus digital: extrapolating from electronics to neurobiology,” *Neural Computation*, vol. 10, no. 7, pp. 1601–1638, 1998.

[34] R. H. Walden, “Analog-to-digital converter survey and analysis,” *IEEE J. Select. Areas Commun.*, vol. 17, no. 4, pp. 539–550, 1999.

[35] K. Wang and S. Shamma, “Spectral shape analysis in the central auditory system,” *IEEE Trans. Speech Audio Processing*, vol. 3, no. 5, pp. 382–395, 1995.

Sourabh Ravindran received the B.E. degree in electronics and communication engineering from Bangalore University, Bangalore, India, in October 2000, and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology (Georgia Tech.), Atlanta, Georgia, in August 2003. He is currently pursuing the Ph.D. degree at Georgia Tech. His research interests include audio classification, auditory modelling, and speech recognition. He is a Student Member of IEEE.

Paul Smith received his B.S. degree in electrical engineering from the Illinois Institute of Technology in December 1996. He received his M.S. degree from the Georgia Institute of Technology in December 1998, and his Ph.D. in electrical engineering from the Georgia Institute of Technology in August 2004. His research focus involves analog integrated circuits for speech processing and recognition applications. Paul is a Member of Tau Beta Pi, a Member of Eta Kappa Nu, and a Member of the IEEE.



David Graham received the B.A. degree in natural science from Covenant College, Lookout Mountain, Georgia, in 2001. He also received both the B.S. and M.S. degrees in electrical engineering from the Georgia Institute of Technology, Atlanta, Georgia, in 2001 and 2003, respectively. He is currently pursuing the Ph.D. degree in electrical engineering at Georgia Tech. His research interests include analog circuits for audio signal processing and silicon models of the human cochlea.

Varinthira Duangudom received her B.S. degree in electrical engineering from North Carolina State University, Raleigh, North Carolina, in 2002, and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, Georgia, in 2004. She is currently pursuing the Ph.D. degree in electrical and computer engineering. Her research interests are in auditory processing, including audio perception and biological modeling.

David V. Anderson was born and raised in La Grande, Oregon. He received the B.S. degree in electrical engineering (magna cum laude) and the M.S. degree from Brigham Young University in August 1993 and April 1994, respectively, where he worked on the development of a digital hearing aid. He received the Ph.D. degree from the Georgia Institute of Technology (Georgia Tech), Atlanta, in March 1999. He is currently on the faculty at Georgia Tech. His research interests include audition and psychoacoustics, signal processing in the context of human auditory characteristics, and the real-time application of such techniques. He is also actively involved in the development and promotion of computer-enhanced education. Dr. Anderson is a Member of the Acoustical Society of America, Tau Beta Pi, and the American Society for Engineering Education.

Paul Hasler is an Associate Professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. Dr. Hasler received his M.S. and B.S.E. degree in electrical engineering from Arizona State University in 1991, and received his Ph.D. degree from the California Institute of Technology (computation and neural systems) in 1997. His current research interests include low-power electronics, mixed-signal system ICs, cooperative analog-digital signal processing (CADSP), floating-gate MOS transistors, adaptive information processing systems, "smart" interfaces for sensors, device physics related to submicron devices or floating-gate devices, and analog VLSI models of on-chip learning and sensory processing in neurobiology. Dr. Hasler received the NSF CAREER Award in 2001, and the ONR YIP Award in 2002. Dr. Hasler received the Paul Raphorst Best Paper Award, IEEE Electron Devices Society, 1997, and a Best Paper Award at SCI 2001. Dr. Hasler has been an author on over 90 journal and refereed conference papers.