

# ANALOG SPEECH RECOGNITION PROJECT

Paul D. Smith, Paul Hasler \*

Georgia Institute of Technology  
Department of Electrical and Computer Engineering  
Atlanta, GA 30332

## ABSTRACT

The Analog Speech Recognition project combines low-power analog signal processing and digital signal processing theory to provide low-power and robust speech processing systems. This project looks to bring together multiple analog signal processing (ASP) blocks into one large ASP system. These component blocks include an analog cepstrum, vector quantizer, and analog HMM. Finally, there are power dissipation comparisons made between the analog and digital systems based on the computations performed.

This project focuses on performing various speech recognition processing blocks in analog circuitry. The blocks that have been proposed and developed are basic blocks that are used throughout speech recognition systems. The final goal is to develop an analog system that can work in concert with digital systems to offload some of the computation and allow the DSP to perform higher-level more complex operations.

The processing cores take advantage of the inherent computational abilities of analog circuitry. The blocks exhibit low power dissipation and by using floating-gate technology, each component block can be programmed to both eliminate any offsets and also reconfigure the overall system.

The computational blocks that are described in this paper include: a cepstrum-like computational block that consists of an analog frequency decomposition similar to a Fourier transform [1] in section 1, a vector quantization block that reduces the overall data set for later matching in section 2, and finally, a Hidden-Markov Model (HMM) computational block that was developed using floating-gate diffusor circuits [2] in section 3. Each processing block is a full stand-alone block and can be substituted into any system desiring a particular type of computation. By combining the individual modules, one can begin to form larger, more complex systems.

\*This work was partially supported by grants National Science Foundation (CISE-1068549, ECS (CAREER): 0093915, ECS-9988905) and by corporate donations to the Georgia Tech Analog Consortium by Texas Instruments and Motorola, Inc.

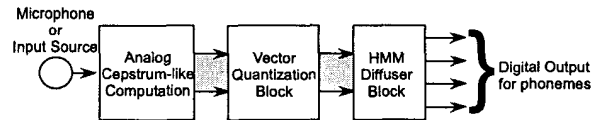


Figure 1. Analog speech recognition system block diagram.

## 1. CEPSTRUM FILTERING

The analog cepstrum encoding IC uses analog circuits and floating-gate computational arrays. This system can act as the front-end for larger digital or analog speech processing systems. It is a novel approach to programmable signal spectrum decomposition, analog frequency transforms, and spectrum compaction. The processing is all done in continuous analog circuitry. By keeping the signals as purely analog signals we are able to neglect noise that is introduced during conversion and also process higher order properties of the signals. Experimental data is presented from circuits fabricated using a  $0.5\mu\text{m}$  nwell CMOS process available through MOSIS [1].

### 1.0.1. Frequency Decomposition

The basic building block of the cepstrum begins with a continuous spectrum decomposition similar to a Fourier Transform. The spacing of the bandpass filters is arbitrary because each can be programmed to have a desired high-frequency corner and low-frequency corner.

### 1.0.2. Log Magnitude

The magnitude of each spectrum passes through a peak detector stage to produce a constant magnitude output. This magnitude is similar to taking the power spectrum density or real spectrum of an input signal. At this point, phase information is unchanged, however the frequency response of the peak detectors must be programmed to its respective frequency band. Each peak detector has an individually

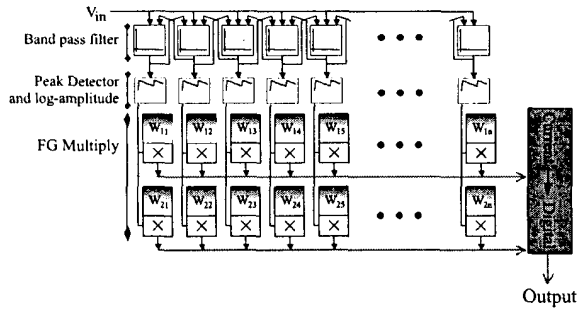


Figure 2. Floating-gate system to perform cepstrum front-end computation for speech processing systems. The system contains 32 frequency taps that can be spaced arbitrarily by programming the corner frequencies for the bandpass filter banks. The peakdetectors provide a power spectrum of the input signal for any given time slice.

programmable corner frequency. Because the output magnitude is continuous, this allows us to capture additional high frequency content within each band.

### 1.0.3. Basis Function Multiplication

The analog differential multiplier is used to multiply the respective frequency bands by a specific basis function. These basis functions can be programmed individually to allow for multiple operations to be performed. For a vanilla Fourier decomposition, each band is multiplied by a cosine factor [3]. These values can be programmed to any arbitrary value [4], but for operations involving spectrum decomposition and transforms, these values are programmed to cosine scale factors.

### 1.0.4. Relation of the Analog Cepstrum to the Real Cepstrum

The cepstrum, as used in digital signal processing (DSP) is based on a signal sampled in time and, if the DFT is used, also in frequency. The analog cepstrum is an approximation to the cepstrum or mel-frequency cepstrum (depending on how the filters are defined) in which frequency is sampled but time is not. 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.

One other difference between the analog cepstrum described herein and the real cepstrum described in Eq. ?? is that the magnitude function (inside the log) is estimated using a peak detector rather than using the true magnitude of the complex spectrum.

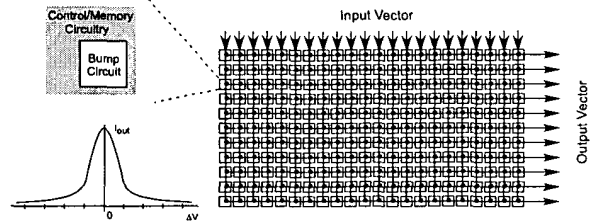


Figure 3. Analog Vector Quantization system that performs a distance measure in parallel on a given input vector. The block is continuous and the vector input is a continuous signal. The individual outputs are a sum of the distance measures for each vector element. The largest output is used to select the vector that is represented on the input.

## 2. VECTOR QUANTIZATION

Vector Quantization is typically used in data compression. Within the speech processing domain, it will be used to reduce the set of detectable spectrum vectors to a set that is considerably more manageable. For our application, we are representing speech spectra and storing the spectra of various vocal parameters/characteristics as vectors. These vectors will represent a reduced set of spectrum data based on actual input data sets. Vector Quantization, just like Scalar Quantization, is a lossy compression. This is because an error is introduced each time a distance measure is used to match an input window to a specific vector. At the same time, within a speech signal, the major speech components often differ greatly and therefore a data representation compression is very beneficial.

The input vector is a spectrum decomposition of an input speech signal. The input data is continuously varying and therefore the input to each vector quantizer cell is continuous. The analog bump circuit computes a continuous distance measure of the analog voltage inputs. For the initial systems, a vector depth of 32 was chosen in order to match the data output from the spectrum decomposition block. The computational blocks are modular and operate in parallel, therefore, the system can be as large or as small as needed.

In the neural network approach to pattern recognition, an N-dimensional input vector  $i$  is mapped into the scalar output variable by means of the scalar product  $a^T i$ , where  $a$  is a weight vector. This operation is typically followed by a nonlinear functional mapping by some nonlinear function[5].

The weighting vector  $a$  is a spectrum of a particular sound. The analog circuitry operates this continuous input and each input "window" is compared using a modified bump cell [6] which gives an output proportional to the difference between the input vector and the stored weighting vector. This output is then sent through a nonlinear discrim-

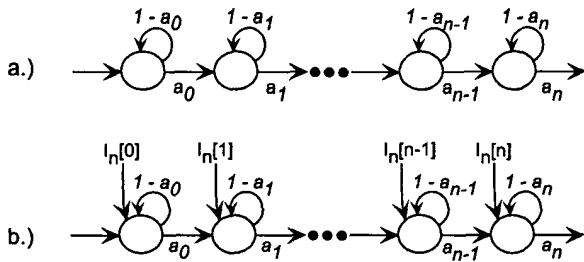


Figure 4. a.) Wave propagation diagram for a lossless path with zero input. b.) Wave propagation diagram for lossless propagation with input at each stage. Each node may require an input in order offset signal degradation as it propagates down the path.

inator called a winner-take-all circuit that eliminates all outputs except the  $N$  largest, where  $N$  can be 1 or more [7, 8]. These outputs can either be processed separately or used as excitation inputs to a HMM diffusor block to produce a time-dependent classification function.

### 3. ANALOG HIDDEN MARKOV MODELS

Hidden Markov Models (HMMs) have become very popular in signal processing because they have proven to be useful in modelling short-term stationary signals. Perhaps the most widely known use of HMMs is speech recognition, where they are used in phoneme recognition, word recognition, and sentence reconstruction. Other applications of HMMs include many pattern recognition problems such as face recognition, handwriting recognition, and molecular sequencing.

The core of analog HMM decoders are the programmable diffusors [2]. We will store a wave directionality term on each floating-gate element and as a wave propagates down a particular path, these stored values determine the amount of additional driving force a wave receives as it passes through a node. Each node has a single input source and it is the coincidence of the passing wave and the input which determines the wave behavior.

Hidden Markov Model decoders on an analog chip. The core structure is a small diffusor cell containing 2-3 transistors and capacitors. The compact structure allows us to implement very long HMM diffusor lines and/or many HMM diffusor lines. This capability is important since each pattern that we desire to recognize must have an HMM decoder line associated with it.

When applying this continuous-time HMM decoder to problems where the rate of the state sequence under observation varies greatly, it may be desirable to implement multiple HMM decoders for each state sequence. For example, if this HMM decoder is applied to speech recognition then

two or three HMM decoders may be used for each phoneme to optimally identify the same phoneme spoken fast or slow. The nature of the continuous-time HMM is such that moderate variations in the rate of state changes do not present a problem but large changes may defeat the method of identification using coincident waves. It should also be noted that disadvantage of placing multiple HMM decoders for each sequence on a chip is far outweighed by the small size of each decoder.

### 4. SNR VS. PRECISION

In order for us to accurately compare a digital signal processing system to an analog signal processing system, we must look at a comparison based on signal representation. In a digital system we can theoretically represent a number using infinite precision but as we move to lower-power systems such as the Texas Instruments C5409 series, these systems perform fixed point computations and are thus subject to fixed point arithmetic errors. Within the realm of analog systems, there are no fixed-point arithmetic errors but there is what is known as a noise floor which gives the minimum signal that can be discerned from the noise present in the system. The representation of signal-to-noise ratio (SNR) can be used within both digital and analog systems. However, the underlying components that produce the noise signals are different. In order to form a valid comparison we must look at signal representation in each system and bring out of this representation an SNR representation for comparison.

### 5. POWER CONSUMPTION

#### 5.1. CMOS Power Consumption

Power dissipation in CMOS logic is caused as charge flows onto and off of the gate as the gate voltage changes. CMOS devices are inherently very low power dissipation devices because they ideally switch from one voltage rail to the other.

$$Q_{gate} = C_{gate} X V_{DD} \quad (1)$$

Repeated switching generates a current proportional to the switching frequency.

$$I = Q_{gate} \cdot f = (C_{gate} \cdot V_{DD}) \cdot f \quad (2)$$

$$Power = (C_{gate} \cdot V_{DD}^2) \cdot f \quad (3)$$

#### 5.2. Computational Power Consumption

Since typical systems include millions of transistors, an explicit calculation using individual capacitances is not used.

Computation	Power ( $\mu$ W)	
	DSP	Analog
FIR Filter	175,000	75
256-point FFT	150,000	75
Cepstrum Filtering	600,000	75

Table 1. Comment about us being able to perform all three operations on the same chip. DSP computations assumes 50MIPS and 3.3 Volts power supply [9].

The current under different conditions is determined empirically by calculating the current for a specific algorithm then scaling the current to different conditions. Current was measured then scaled to represent a frequency-dependent current factor usually expressed as milliamperes per Megahertz, mA per MIPS, or in the analog case, mA per computational block.

Power supply current in typical DSP systems can be broken into two categories: system-related and device/algorithm-related factors. Of all the system-related factors, operating frequency is the most significant. Because CMOS devices dissipate most of their power during switching, by doubling the switching frequency, power dissipation also doubles. Of the device/algorithm related factors, the program activity is the most significant.

For analog systems, power consumption is much easier to determine. It is definitely measurable and separable across multiple computational blocks. The DC operating current can be computed and power consumption simply becomes a multiplication of operating current by transistor count and DC supply voltage.

By separating the power consumption into computational tasks, we can segregate and simplify the interdomain comparison between analog and digital computation. DSP systems can further separated the power consumption based on algorithm and this most closely compares to the analog power consumption because power consumption within a DSP system depends on multiple operational factors such as computation/CPU usage, temperature and peripheral/data bus activity.

Power dissipation for double-precision arithmetic operations is 1mA per MIP, at 50MIPS we get a total of 50mA of current being used. For power calculations, the IC is running at 3.3 Volts which gives a power consumption of 165 mW of power. We see similar numbers for an FIR filter or 256 point FFT calculations.

## 6. REFERENCES

- [1] Paul D. Smith, Matt Kucic, Richard Ellis, Paul Hasler, and David V. Anderson, "Cepstrum frequency encoding in analog floating-gate circuitry," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2002, In Review.
- [2] Paul D. Smith, Paul Hasler, and David V. Anderson, "Programmable floating-gate diffusors and their similarities to hms," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2002, In Review.
- [3] M. Kucic, A. Low, J. Neff, and P. Hasler, "A programmable continuous-time floating-gate fourier processor," *IEEE Journal of Circuits and Systems II*, January 2001.
- [4] Paul D. Smith, Matt Kucic, and Paul Hasler, "Accurate programming of analog floating-gate arrays," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2002, In Review.
- [5] Jurgen Schurmann, *Pattern Classification, A Unified View of Statistical and Neural Approaches*, John Wiley and Sons, Inc., New York, 1996.
- [6] T. Delbrück, "Bump circuits for computing similarity and dissimilarity of analog voltages," in *Proc. IJCNN*, Seattle, WA, July 8-12 1991, vol. I, pp. 475-479.
- [7] J. Lazzaro, S. Ryckebusch, M.A. Mahowald, and C.A. Mead, "Winner-take-all networks of O(N) complexity," in *Advances in Neural Information Processing Systems 1*, David Touretzky, Ed., vol. 1, pp. 703-711. Morgan Kaufman Publishers, San Mateo, CA, 1988.
- [8] W. Fritz Kruger, Paul Hasler, Bradley A. Minch, and Christof Koch, "An adaptive wta using floating gate technology," in *Advances in Neural Information Processing Systems 9*, Michael C. Mozer, Ed., vol. 9, pp. 720-726. MIT Press, Cambridge, MA, 1997.
- [9] Clay Turner, "Calculation of TMS320LC54x power dissipation," Application Report SPRA164, Texas Instruments, Digital Signal Processing Solutions - Semiconductor Group, June 1997.