A PROGRAMMABLE BANDPASS ARRAY USING FLOATING-GATE ELEMENTS

David W. Graham, Paul D. Smith, Rich Ellis, Ravi Chawla, and Paul E. Hasler

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

In order to meet the demands of low-power real-time audio signal processing, we present a programmable array of bandpass filters using floating-gate transistors. This bank of analog bandpass filters can be placed before an analog-to-digital converter to perform a frequency decomposition of an audio signal and also allow for analog preprocessing before being transformed into the digital domain. Individual filter taps can easily be tuned to any given frequency and bandwidth because capacitively coupled current conveyers (C^4s) are used with floating-gate transistors as biasing elements. Offsets and mismatches within the circuit elements are shown to be inconsequential because they can be accounted for and programmed out.

As portable electronics continue to advance, greater signal processing is required at lower power and at faster rates. While digital systems and DSPs are highly programmable and easy to use, the high power consumption quickly drains batteries and the slow processing can exclude these from real-time situations. However, instead of looking to the digital world for all of life's answers, analog blocks should be used when there are clear advantages. Specifically, MOSFETs running in the subthreshold regime can be used to create extremely low-power, continuous-time systems. Plus, with the addition of floating-gate MOSFETs within this analog circuitry, these analog systems take on one of the main characteristics of digital circuitry – ease of use through programmability.

In this paper, we present a bank of programmable bandpass filters utilizing low-power analog circuits. Since floating-gate devices are used for biasing elements, the individual filter taps can be programmed to have any arbitrary spacing and bandwidth.

By programming the filter bank to have exponentially spaced center frequencies with narrow bandwidths and moderate resonance, an input signal is broken down into its respective frequency components. This type of filter bank is extremely useful in signal processing applications in which the specific algorithm is performed on individual subbands of the input signal, as is often done in auditory processing [1], [2].

Several versions of this programmable array of bandpass filters have been fabricated. All data in this paper were obtained from $0.5\mu m$ processes available through MOSIS.

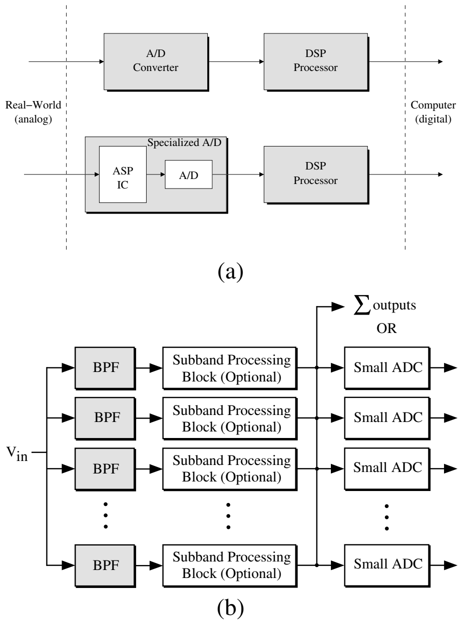


Fig. 1. **(a)** Collaborative analog and digital signal processing blurs the boundary of where to place the conversion from analog to digital. Performing some of the signal processing with low-power real-time analog circuitry alleviates some of the burden of the DSP allowing the DSP to perform more complex computations or a smaller DSP to be used. **(b)** The analog signal processing (ASP) block we present in this paper consists of an array of programmable bandpass filters (shown in grey). Further signal processing can be performed on each subband signal before either recombining them or sending them through small ADCs and then on to the DSP.

1. ANALOG AUDIO PROCESSING ADVANTAGES

The tendency in the signal processing realm for dealing with incoming audio signals is immediately passing the analog audio signal to an analog-to-digital converter (ADC) so that the signal can be manipulated digitally. Typically, the FFT of the signal is performed digitally so that the individual subbands can be manipulated. Digital signal processing is invoked as early as possibly since it has many advantages, and the greatest is the ease of programming a digital system to meet the given requirements. Typically, the FFT of the signal is one of the first digital computations so that the individual subbands can be manipulated.

However, there is another option which is to introduce an analog system that does more than simply convert a signal into a digital version as soon as possible. By placing an analog signal processing block immediately before an ADC, as is shown in Fig. 1a, much of the processing can be done with the low-power and real-time computation of analog circuitry. This, therefore, alleviates a

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

large portion of the digital circuitry's burden. The overall system can either have a smaller digital processing block than was previously required, or it can have the same size digital block allowing for more functionality since the basic processing has already been conducted in analog.

Many options exist for an analog signal processing block. However, if frequency decomposition is possible with analog circuitry, then this is a clear choice for the front end analog block. In addition, more signal processing could be performed with analog circuits on the subband signals before they are recombined or sent through individual, smaller ADCs, as is illustrated in Fig. 1b.

In order to build a useful analog frequency-decomposition block, a bank of bandpass filters has to be created in which the basic bandpass filter element must be relatively small so that it can be placed in an array, the bandpass element must be easily tunable so that multiple elements can cover the entire auditory spectrum, and the center frequencies of the bandpass elements must follow an orderly spacing. Typically, for frequency decomposistion, exponential spacing is desired. Having a moderate amount of resonance ($Q \approx 30$) is also desirable for better isolation of the center frequency. The rest of this paper discusses how we have gone about designing and building this type of programmable bandpass array.

2. TUNABLE BANDPASS FILTER ELEMENT

The filter used in the programmable filter array is based on the capacitively coupled current conveyer (C^4) that has been presented [3] and characterized [4] elsewhere and is shown in Fig. 2a. A summary of the C^4 is as follows.

The C^4 is a capacitively based bandpass filter with electronically tunable corner frequencies that are independent of each other. The frequency response of the C^4 is governed by

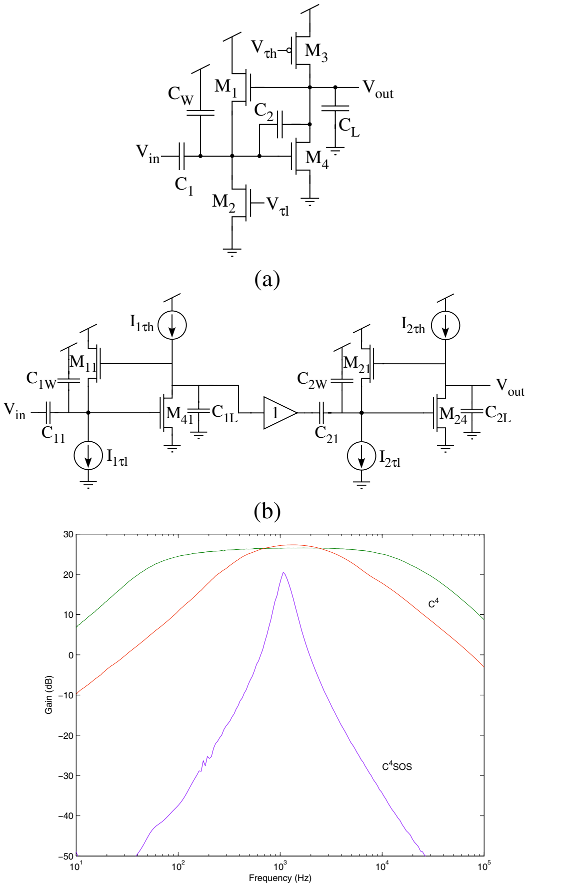
$$\frac{V_{out}}{V_{in}} = -\frac{C_1}{C_2}\frac{s\tau_l(1-s\tau_f)}{s^2\tau_h\tau_l + s(\tau_l + \tau_f(\frac{C_o}{\kappa C_2}-1))+1} \qquad (1)$$

where the time constants are given by

$$\tau_l = \frac{C_2 U_T}{\kappa I_{\tau_l}} \qquad \tau_f = \frac{C_2 U_T}{\kappa I_{\tau_h}} \qquad \tau_h = \frac{C_T C_O - C_2^2}{C_2}\frac{U_T}{\kappa I_{\tau_h}}$$

and the total capacitance, C_T, and the output capacitance, C_O, are defined as $C_T = C_1 + C_2 + C_W$ and $C_O = C_2 + C_L$. The currents I_{τ_l} and I_{τ_h} are the currents through M_2 and M_3, respectively in Figure 2a. With normal usage, τ_f is so fast that the zero it produces lies far outside of the operating range. Hence, the C^4 takes on the form of a bandpass filter within the region of interest with ±20 dB/decade slopes outside the passband. The midband gain is $-C_1/C_2$.

Removing the feedback capacitor C_2, a configuration called a *vanilla* C^4, transforms the C^4 into a high-gain filter with a much larger Q peak value [4]. The C^{4}SOS, shown in Fig. 2b, is simply cascade of two vanilla C^{4}s isolated by a buffer. By tuning each of the vanilla C^{4}s comprising the C^{4}SOS to have identical time constants, the overall response of the C^{4}SOS has ±40 dB/decade slopes outside the passband and a potentially large Q peak. Tuning of the bias currents is accomplished by programming floating-gate transistors to the desired current which are shown in Fig. 2b as current sources.



(a)

(b)

(c)

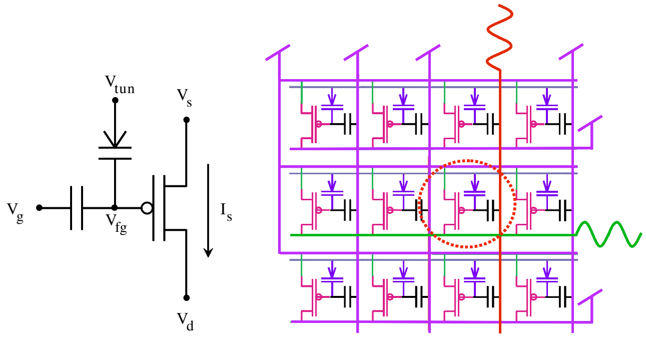
Fig. 2. **(a)** Capacitively coupled current conveyer (C^4). **(b)** C^4 second-order section (C^{4}SOS). **(c)** Magnitude frequency responses illustrating the functionality of the C^4 and the C^{4}SOS.

3. FLOATING-GATE PROGRAMMABILITY

A single floating-gate element is shown in Fig. 3a. Simply put, a floating-gate transistor is a MOSFET device with only capacitors connected to the gate. Charge on the gate is fixed due to the oxide surrounding and insulating the gate. Hence, any charge on the gate is responsible for establishing the amount of current flowing through the transistor. While the charge on the gate will not change on its own, that amount of charge can be modified by three processes – UV photo injection, electron potential boring, and hot electron injection – and the last two are the primary means of programming floating-gate circuits.

Electron potential boring is also known as tunneling. Through this process, a very large voltage is placed on the tunneling capacitor that is shown in Fig. 3a. As this large tunneling voltage is increased, the effective width of the barrier is decreased. This can allow some electrons to breach the gap without adversely affecting the insulator. Through this process, electrons can be removed from the gate in a controlled manner.

Hot electron injection has two requirements. First, there must be an appreciable amount of current flowing through the device. Second, there must be a large source-to-drain voltage. When both of these criteria are met, holes in a pFET that are flowing through the channel can build up sufficiently large enough energy to impact ionize an electron-hole pair. The electron can have enough energy to pass through the insulator and onto the floating gate. Therefore,



(a) (b)

Fig. 3. **(a)** A floating-gate pFET transistor. **(b)** The architecture used for programming arrays of floating-gate devices. Two conditions must exist for injection to occur: (1) a high source-to-drain field to make the electron "hot" and (2) a channel for device current to flow. Because these conditions can be created orthogonally to each other with source-to-drain voltage and gate voltage (to modulate the channel), a single element can be selected for injection or measurement.

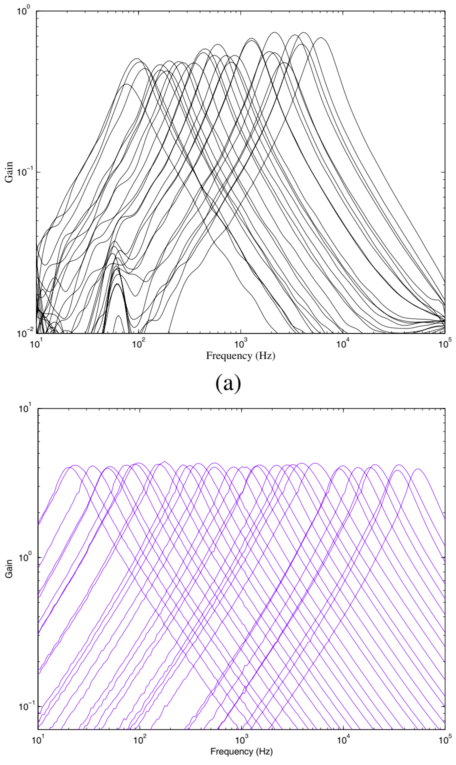
hot-electron injection is responsible for putting electrons onto the floating gate in a controlled manner.

To program a large amount of floating-gate devices, as would be required for a programmable filter bank, the floating-gate transistors are arranged in an array for ease of programming as is shown in Fig. 3b [5], [6]. Tunneling can be used to program currents accurately, but selectivity is not completely controllable. When one element is tunneled, all the other devices in the array will have their charge altered. As a result, the tunneling operation is reserved for "erasing" charge that is already on the floating gate and not for very accurate programming.

However, hot electron injection allows complete selectivity of an individual element. Hence, injection is used for precise and accurate programming of floating-gate arrays [6]. The method of programming by injection is depicted in Fig. 3b and is explained as follows. After selecting a specific floating-gate device, all the columns not containing that device have their gate lines connected to V_{DD}. Then all the drains of the rows not containing the selected element are connected to V_{DD}, as well. Therefore, the gates or the drains or both of all the non-selected elements are connected to V_{DD} ensuring that no appreciable current will flow in any of the other devices, thus meaning that they cannot be injected, and their floating-gate charge cannot be changed. A voltage can be supplied to the input of the selected element, allowing current to flow. Finally, the drain of the selected element is pulsed down so that the source-to-drain voltage is temporarily large. The two criteria for injection are met, so electrons will be added to the floating gate of the selected element. Any such element in the array can be chosen and programmed, and when all the currents have been set to the desired values, the terminals of the transistor are connected to the rest of the circuit in which they are operating as fully functional transistors.

4. PROGRAMMED FILTER BANK

An array of 32 C^{4}SOSs was fabricated with a $0.5\mu m$ process available through MOSIS. While the bandpass filter elements can be programmed to any desired center frequency spacing and bandwidth, we will use the example of exponentially spaced center frequencies with narrow bandwidths and moderate Qs as this is a highly advantageous configuration in audio signal processing. This type of configuration closely models the biology of the human cochlea, and it lends itself to allow subbands of frequency to be independently manipulated since real-time frequency decomposition is occurring.



(a)

(b)

Fig. 4. Array of 32 programmable vanilla C^{4}s **(a)** Original array [7] using large resistive line to bias the transistors. **(b)** New array in which the time constants are set by floating-gate transistors that were programmed with exponentially spaced corner frequencies within 95% accuracy. This programmed array shows a marked improvement over the original, non-programmable array.

Figure 4 shows the frequency response of each of the 32 filter taps. Only the output from the first stage (a single C^4) is shown. These filters were programmed so that the $I_{\tau l}$s and $I_{\tau h}$s for each filter tap had exponentially spaced currents that were programmed within 95% of their desired values.

Figure 4b illustrates a fundamental design issue with analog circuits. No matter how accurately biases can be set, circuit performance is affected by mismatches that occur during the fabrication process. The 32 traces from the programmable bandpass array are monotonically spaced, which is a difficult task and has only been overcome by very clever circuit design [8]. In contrast, this monotonicity and spacing was simply programmed by the floating-gate biases. However, inspection of Fig. 4b shows that corner frequencies are not *perfectly* spaced. This is of no concern, though, because these errors due to mismatch of transitor and capacitor sizes can be simply programmed out with the floating gates.

In order to determine the exact bias current that should be pro-

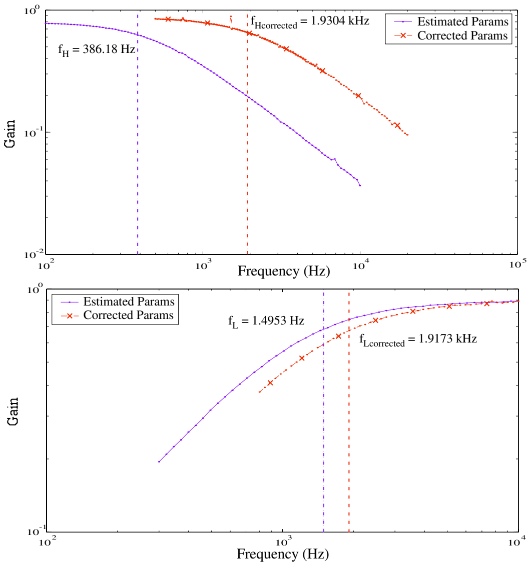


Fig. 5. Example of applying correction factor to **(a)** high frequency **(b)** low frequency corners. The correction factor is calculated as a ratio of the actual time constant to the desired time constant and is used in determining the correct bias current to program. In the case of the high frequency corner, the target corner frequency was 2.0 kHz; after one correction step, the corner frequency was measured as 1.93 kHz – a 4% error which was within the 5% tolerance characteristic of the programming algorithm.

grammed into each floating gate of a C^4, a measure of the total effective mismatch within a filter must be obtained. The process of calculating and using this correction factor is as follows. Using the time constant equations and estimated values for the device constants and capacitances, an initial current is programmed into each bias point so that the upper and lower corner frequencies are widely spread and do not influence each other. The magnitude of the frequency response for the filter is then measured. The time constants can then be easily extracted by transforming these data and performing a linear regression. The transformation for highpass and lowpass responses (and thus low and high corners, respectively) can be written as follows:

$$y_L(x = \frac{1}{\omega^2}) = \frac{1}{|H(jw)|^2} = \frac{1}{A^2} + \frac{1}{\tau_L^2 A^2} \quad x$$

$$y_H(x = \omega^2) = \frac{1}{|H(jw)|^2} = \frac{1}{A^2} + \frac{\tau_H^2}{A^2} \quad x \qquad (2)$$

where A is the passband gain. Using the extracted frequency response parameters, a correction constant is calculated as a ratio of the actual time constant and the target time constant. This correction factor is then multiplied with the original target current and the floating-gate devices are reprogrammed.

This method of estimation and correction has proven very successful. Data from a programmable C^4 circuit fabricated through MOSIS are shown in Fig. 5. The plots show both the original and the corrected frequency response curves of the low and high frequency corners. After the correction factor had been applied, the new corner frequencies were within the 5% tolerance allowed for in the programming algorithm. Increasing the accuracy of the programming algorithm would result in further increases in the accuracy of the tuning algorithm.

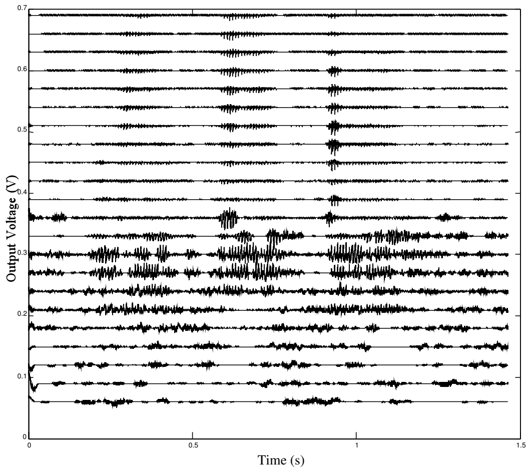


Fig. 6. The output of the middle subbands for a speech signal.

5. CONCLUSION

We have shown that an array of analog bandpass filters can be highly programmed by floating gates. This type of filter bank is extremely useful in audio signal processing where frequency decomposition is desirable. Figure 6 shows the output of the filter bank for a speech data file where the output of each filter tap was obtained in real time while the array was running at only the very low power required by subthreshold MOSFETs.

6. REFERENCES

[1] P. Hasler, P. Smith, R. Ellis, D. Graham, and D. Anderson, "Biologically inspired sensing system interfaces on a chip," in *Proceedings of the IEEE Sensors 2002*, Orlando, FL, June 2002.

[2] T. Massengill, D. Wilson, P. Hasler, and D. Graham, "Empirical comparison of analog and digital auditory preprocessing for automatic speech recognition," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Scottsdale, AZ, May 2002.

[3] P. Hasler, M. Kucic, and B. Minch, "A transistor-only circuit model of the autozeroing floating-gate amplifier," in *IEEE Midwest Symposium on Circuits and Systems*, Las Cruces, August 1999, pp. 157–160.

[4] P. Smith, D. Graham, R. Chawla, and P. Hasler, "A five-transistor bandpass filter element," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Submitted 2004.

[5] P. Hasler, C. Diorio, B. Minch, and C. Mead, "Single transistor learning synapses," in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. Touretzky, and T. Leen, Eds., pp. 817–824. MIT Press, Cambridge, MA, 1995.

[6] P. Smith, M. Kucic, and P. Hasler, "Accurate programming of analog floating-gate arrays," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Scottsdale, AZ, May 2002.

[7] D. Graham and P. Hasler, "Capacitively-coupled current conveyer second-order sections for continuous-time bandpass filtering and cochlea modeling," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Scottsdale, AZ, May 2002.

[8] A. Van Schaik, E. Fragniere, and E. Vittoz, "A silicon model of amplitude modulation detection in the auditory brainstem," in *Advances in Neural Information Processing Systems 9*, M. Mozer, M. Jordan, and T. Petsche, Eds., Cambridge, MA, 1997, pp. 741–747, MIT Press.