A Reconfigurable Mixed-Signal VLSI Implementation of Distributed Arithmetic Used for Finite-Impulse Response Filtering

Erhan Özalevli, *Member, IEEE*, Walter Huang, *Student Member, IEEE*, Paul E. Hasler, *Senior Member, IEEE*, and David V. Anderson, *Senior Member, IEEE*

***Abstract*—A reconfigurable implementation of distributed arithmetic (DA) for post-processing applications is described. The input of DA is received in digital form and its analog coefficients are set by using the floating-gate voltage references. The effect of the offset and gain errors on DA computational accuracy is analyzed, and theoretical results for the limitations of this design strategy are presented. This architecture is fabricated in a 0.5-μm CMOS process, and configured as a *16-tap* finite impulse response (FIR) filter to demonstrate the reconfigurability and computational efficiency. The measurement results for comb, low-pass, and bandpass filters at 32/50-kHz sampling frequencies are presented. This implementation occupies around 1.125 mm^2 of die area and consumes 16 mW of static power. The filter order can be increased at the cost of 0.011 mm^2 of die area and 0.02 mW of power per tap.**

***Index Terms*—Array of tunable FG voltage reference (epot), fistributed arithmetic (DA), finite-impulse response (FIR) filter, floating-gate (FG) transistor, post processing.**

I. Low-Power Signal Processing

THE battery lifetime of portable electronics has become a major design concern as more functionality is incorporated into these devices. Therefore, the shrinking power budget of modern portable devices requires the use of low-power circuits for signal processing applications. The data or media in these devices is generally stored in a digital format but the output is still synthesized as an analog signal. Examples of such devices are flash memory and hard disk based audio players. The signal processing functions employed in these devices include finite-impulse response (FIR) filters, discrete cosine transforms (DCTs), and discrete Fourier transforms (DFTs). The common feature of these functions is that they are all based on the inner product. Digital signal processing (DSP) implementations typically make use of multiply-and-accumulate (MAC) units for the calculation of these operations, and the computation time increases linearly as the length of the input vector grows. In contrast, distributed arithmetic (DA) is an efficient way to compute an inner product. It computes an inner product in a fixed number of cycles, which is determined by the precision of the input data. It has been employed for image coding, vector quantization, discrete cosine transform and adaptive filtering implementations [1]–[4].

DA is computationally more efficient than MAC-based approach when the input vector length is large. However, the trade-off for the computational efficiency is the increased power consumption and area usage due to the use of a large memory. These problems can be alleviated by utilizing mixed-signal circuit implementations for optimized DA performance, power consumption, and area usage. This paper proposes a mixed-signal DA architecture built by utilizing the analog storage capabilities of floating-gate (FG) transistors for reconfigurability and programmability. The circuit compactness is obtained through the application of the iterative nature of the DA computational framework, where many multipliers and adders are replaced with an addition stage, a single gain multiplication, and a coefficient array.

In this paper, the computational efficiency of DA implementation is demonstrated by configuring it as an FIR filter. The low-power implementations of these filters can readily ease the power consumption requirements of portable devices. Also, due to the serial nature of the DA computation, the power and area of this filter increase linearly with its order. Hence, this design approach allows for a compact and low-power implementation of high-order FIR filters.

In the next section, the DA computation is described. Subsequently, the hybrid DA architecture is explained, and the integration of tunable voltage references into the DA implementation is described. After that, the precise programming/tuning of these voltage references is explained. In addition, the theoretical analysis of second order effects in the design is given and the experimental results of this reconfigurable DA for FIR filtering are presented. In the last part of the paper, previously reported FIR filter implementations are given and their design issues are summarized and compared with the proposed implementation.

II. DA Computation

The DA concept was first introduced by Croisier *et al.* [5], and later utilized for the hardware implementation of digital filters using memory and adders instead of multipliers [6]. It is an efficient computational method for computing the inner product

Manuscript received January 20, 2007; revised June 28, 2007. This paper was recommended by Associate Editor M. E. Zaghloul.

E. Özalevli was with the Georgia Institute of Technology. He is now with Texas Instruments, Dallas, TX 75243 USA (e-mail: e-ozalevli@ti.com; ozalevli@ece.gatech.edu).

W. Huang, P. E. Hasler, and D. V. Anderson are with the Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: walter@ece.gatech.edu; phasler@ece.gatech.edu; dva@ece.gatech.edu).

Digital Object Identifier 10.1109/TCSI.2007.913735

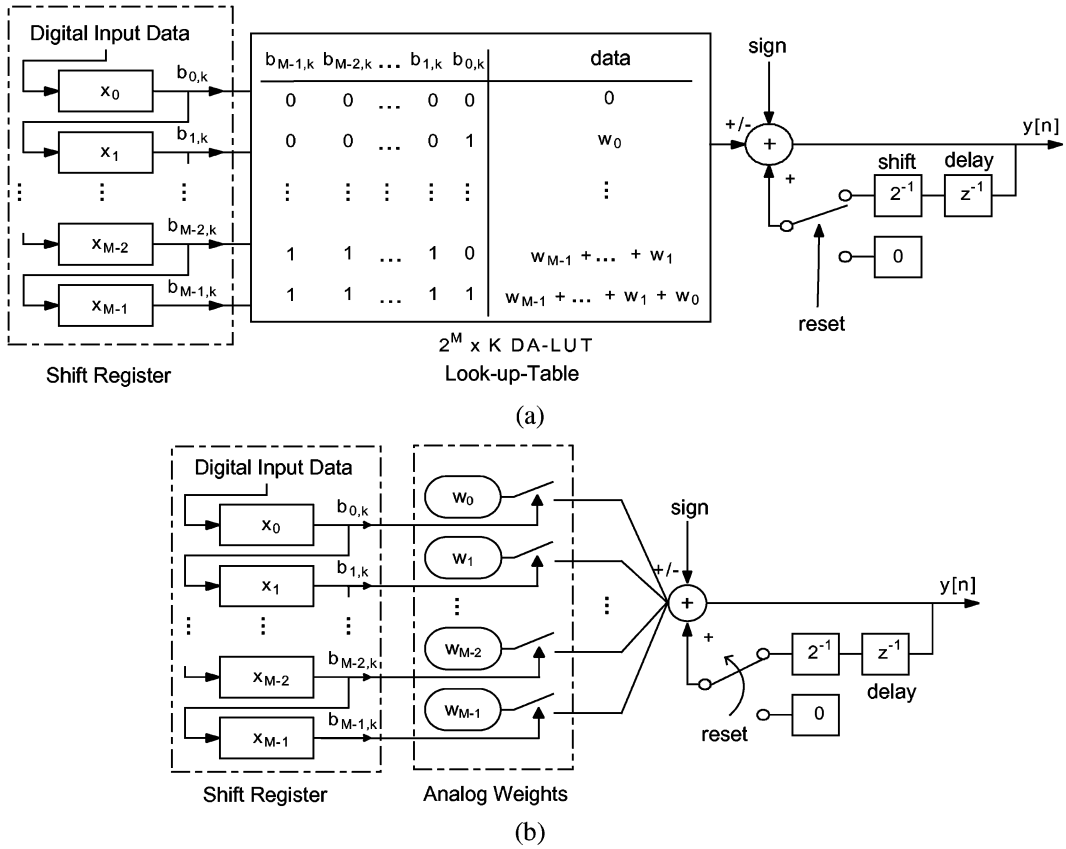


Fig. 1. Basic DA hardware architecture. $b_{i,k}$ is the input bit for kth cycle of operation and $y[n]$ is the output. (a) Digital implementation. (b) Proposed hybrid mixed-signal implementation using digital input data and stored analog weights. Digital input data is processed in the analog domain.

of two vectors in a bit-serial fashion [7]. The operation of DA can be derived from the inner product equation as follows:

$$y[n] = \sum_{i=0}^{M-1} x[n-i]w[i]. \tag{1}$$

In the case of FIR filtering, x is the input vector and w is the weight vector. Using a K-bit 2's-complement representation, x can be written as $x[n-i] = -b_{i0} + \sum_{j=1}^{K-1} b_{ij}2^{-j}$, where b_{i0} is the sign bit, b_{ij} is the jth bit of the ith element in the vector x, and $b_{i(K-1)}$ is the least significant bit. Substituting x into (1), and by reordering the summations and grouping the terms together, (1) can be written as

$$y[n] = -\sum_{i=0}^{M-1} w_i b_{i0} + \sum_{j=1}^{K-1} 2^{-j} \sum_{i=0}^{M-1} w_i b_{ij}. \tag{2}$$

In digital implementations, the summation $\sum_{i=0}^{M-1} w_i b_{ij}$ is pre-computed and stored in a memory for multiplier-less operation and reduced hardware complexity. This is usually achieved by storing 2^M possible combinations of summed weights in the memory, which simplifies the hardware requirements of DA to a bank of input registers, a memory, a delay element, a shifter, a switch, and an adder as illustrated by Fig. 1(a). By reusing the hardware K times, an output sample can be processed in K clock cycles regardless of the number of taps, M, and without using a multiplier. Digital DA architectures obtain significant throughput advantages when M is large.

In contrast to digital implementations, the addition in the analog domain is much more power and area efficient. Therefore, the high memory usage of digital DA implementations can be eliminated by processing the digital input data in the analog domain as shown in Fig. 1(b). To design such a structure, weights in (1) are stored in the analog domain. For an individual weight, data is processed in a similar way as it is achieved by serial DACs, where the conversion is performed sequentially.

III. Proposed DA Architecture

The hybrid DA architecture consists of four components, which are a 16-bit shift register, an array of tunable FG voltage references (epots) [8], inverting amplifiers (AMP), and sample-and-hold (S/H) circuits, as illustrated in Fig. 2. The timing of the digital data and control bits governs the DA computation and is illustrated in Fig. 3. Digital inputs are introduced to the system by using a serial shift register. These digital input words represent the digital bits, b_{ij} in (2), which selects the epot voltages to form the appropriate sum of weights necessary for the DA computation at the jth bit. The clock frequency of the shift register is dependent on the input data precision K and the length of the filter M and is equal to $M \cdot K$ times the sampling frequency. Once the jth input word is serially loaded into the top shift register, the data from this register is latched at K times the sampling frequency. If the amount of area used

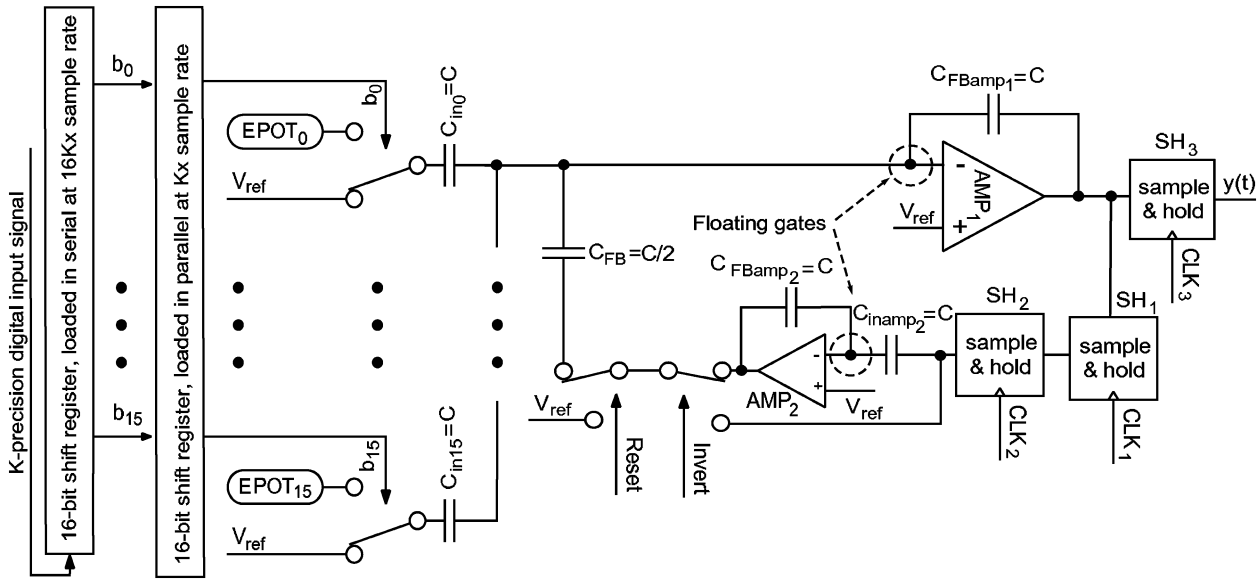


Fig. 2. Implementation of the 16-tap hybrid FIR filter. b_i is the input bit for jth cycle of operation and $y(t)$ is the output. Epots store the analog weights. S/Hs, are used to obtain the delay and hold the computed output voltage.

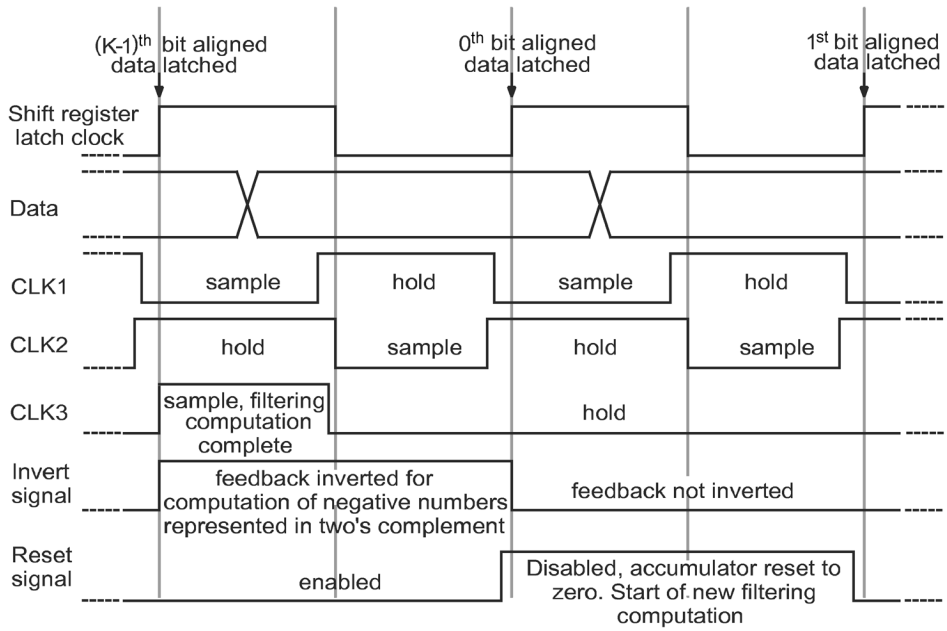


Fig. 3. Digital clock diagram of the filter architecture. For desired sampling frequency, f_s, K-bit precise M-bit digital input data is loaded serially to a shift register at a $K \cdot M \cdot f_s$ clock frequency, and latched at a $K \cdot f_s$ clock frequency. CLK_1, CLK_2, and CLK_3 are the bits used to control S/H_1, S/H_2, and S/H_3, respectively. Invert signal is used to obtain 2's-complement compatibility. Also, Reset signal is used to clear the result of the previous computation.

by the shift registers is not a design concern, then ideally an M-tap FIR filter should have M shift registers. A clock that is K times faster than the sampling frequency would be used for this ideal configuration.

The analog weights of DA are stored by the epots. When selected, these weights are added by employing a charge amplifier structure composed of same size capacitors, and a two-stage amplifier AMP_1. The epot voltages as well as the rest of the analog voltages in the system are referenced to a reference voltage, $V_{\mathrm{ref}} = 2.5$ V. Since the addition operation is performed by using an inverting amplifier, the relative output voltage, when Reset signal is enabled, becomes equal to the negative sum of the selected weights for $C_{\mathrm{in}_i} = C_{\mathrm{FBamp}_1}$. For the first computational cycle, the result of the addition stage represents the summation, $\sum_{i=0}^{m-1} w_i b_{i(K-1)}$ in (2), which is the addition of weights for the LSBs of the digital input data.

In the feedback path of the system, a delay, an invert and a divide-by-two operations are used for the DA computation. For that purpose, S/H circuits S/H_1 and S/H_2 and inverting amplifiers AMP_1 and AMP_2 are employed in the implementation. The S/H circuits store the amplifier output to feed it back to the system for the next cycle of the computation. Nonoverlapping clocks CLK_1 and CLK_2 are used to hold the analog voltage while the next stream of digital data is introduced to the addi-

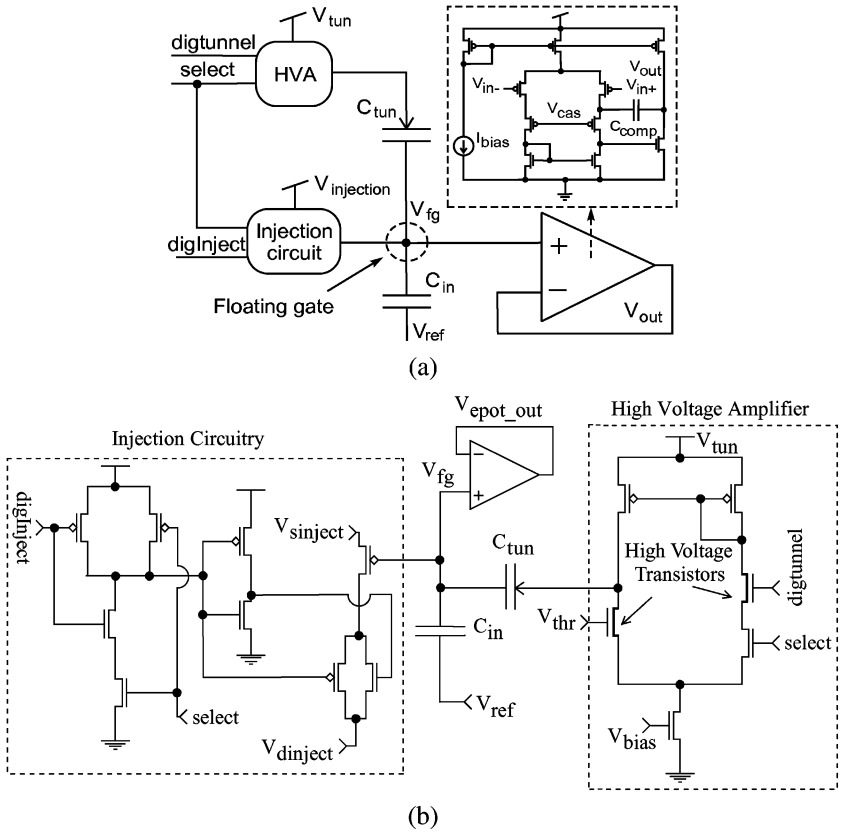


Fig. 4. Modified epot schematic. Epot is selected by a decoder using select signal. Injection and tunnelling is controlled by using $\mathrm{digInject}$ and $\mathrm{digtunnel}$, respectively. V_{fg} and V_{ref} are the floating-gate and reference voltages, respectively. C_{in} is the input capacitor and C_{tun} is the tunnelling junction. (a) Epot is built by using a low-noise amplifier and FG programming circuitry. The epot amplifier is compensated by using C_{comp}. HVA is a high-voltage-amplifier used for tunnelling. (b) Epot programming circuitry. V_{sinject} and V_{dinject} are the source and drain voltages used to control the injection, while V_{tun} and V_{thr} are the tunnelling and the reference voltage of high-voltage amplifier used to control tunnelling mechanism.

tion stage. These clocks have a frequency of K times the sampling frequency. The stored data is then inverted relative to the reference voltage by using the second inverting amplifier AMP_2 to obtain the same sign as the summed epot voltages. AMP_2 is identical to AMP_1, and has the same size input/feedback capacitors. After obtaining the delay and the sign correction, the stored analog data is fed back to the addition stage as delayed analog data. During the addition, it is also divided by two by using $C_{\mathrm{FB}} = C_{\mathrm{FBamp1}}/2 = C/2$, which gives a gain of 0.5 when it is added to the new sum. This operation is repeated until the MSBs of the digital input data is loaded into the shift register. The MSBs correspond to $(K-1)$th bits, and are used to make the computation 2's-complement compatible. This compatibility is achieved by disabling the inverting amplifier in the feedback path during the last cycle of the computation by enabling the Invert signal. As a result, during the last cycle of the computation, the relative output voltage of AMP_1 becomes

$$V_{\mathrm{out}_{\mathrm{amp1}}} - V_{\mathrm{ref}} = -\sum_{i=0}^{M-1} \frac{C_{\mathrm{in}_i}}{C_{\mathrm{FBamp}_1}} (V_{\mathrm{ref}} - V_{\mathrm{epot}_i}) b_{i0} + \sum_{j=1}^{K-1} 2^{-j} \sum_{i=0}^{M-1} \frac{C_{\mathrm{in}_i}}{C_{\mathrm{FBamp}_1}} (V_{\mathrm{ref}} - V_{\mathrm{epot}_i}) b_{ij} \quad (3)$$

where the first term is the result of the calculation with the sign bits. Finally, when the computation of the output voltage in (3) is finished, it is sampled by $\mathrm{S/H}_3$ using CLK_3, which is enabled once every K cycle. $\mathrm{S/H}_3$ holds the computed voltage till the next analog output voltage is ready. The new computation starts by enabling the Reset signal to zero out the effect of the previous computation. Then, the same processing steps are repeated for the next digital input data.

IV. Circuit Description of Computational Blocks

To achieve an accurate computation using DA, the circuit components are designed to minimize the gain and offset errors in the signal path. In this architecture, those components are the epots, the inverting amplifiers, and the S/Hs.

The epot, shown in Fig. 4(a), is modified from its original version [8] to obtain a low-noise voltage reference. It is a dynamically reprogrammable, on-chip voltage reference that uses a low-noise amplifier integrated with FG transistors and programming circuitry to tune the stored analog voltage. The amplifier in the epot circuit is used to buffer the stored analog voltage so that the epot can achieve low noise and low output resistance as well as the desired output voltage range. An array of epots is used for storing the filter weights; and during the programming, individual epots are controlled and read by employing a decoder.

In this architecture, epots and inverting amplifiers are the main blocks that use FG transistors to exploit their analog storage and capacitive coupling properties. A precise tuning

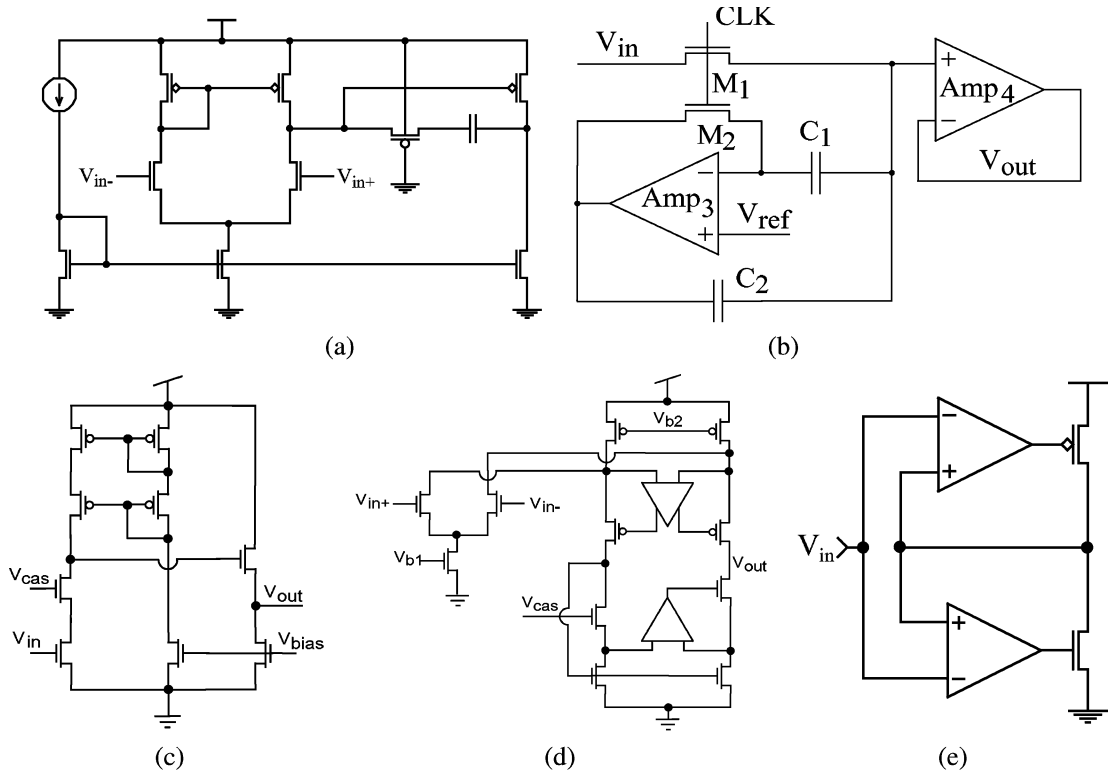


Fig. 5. Circuit components. (a) Inverting amplifier schematic. This circuit is used for AMP_1 and AMP_2 in the DA implementation. (b) S/H circuit schematic. This circuit is employed for S/H_1 and S/H_2 in the DA implementation. (c) AMP_3 in the S/H circuit. (d) AMP_4 in the S/H circuit. (e) Buffer schematic. This circuit is used to drive the sinal off-chip.

of the stored voltage on FG node is achieved by utilizing the hot-electron injection and the Fowler–Nordheim tunnelling mechanisms. The epots employ FG transistors to store the analog coefficients of the inner product. In contrast, the inverting amplifiers use them not only to obtain capacitive coupling at their inverting-node, but also to remove the offset at their FG terminals.

One of the main advantages of exploiting FG transistors in this design is that the area allocated for the capacitors can be dramatically reduced. It is shown in [9] that epots can be utilized to implement a compact programmable charge amplifier DAC. This structure helps to overcome the area overhead, which is mainly due to layout techniques used to minimize the mismatches between the input and feedback capacitors. Similarly in this DA implementation, the unit capacitor C is set to 300 fF, and no layout technique is employed. As expected, due to inevitable mismatches between the capacitors, there will be a gain error contributed from each input capacitor. The stored weights are also used to compensate this mismatch. When the analog weights are stored to the epots, the gain errors are also taken into account to achieve accurate DA computation.

Unlike switched-capacitor amplifiers, the addition in this implementation is achieved without resetting the inverting node of the amplifiers. This is because the floating-gate inverting-node of the amplifiers allow for the continuous-time operation. This design approach eliminates the need for multi-phase clocking or resetting. Inverting amplifiers are implemented by using a two-stage amplifier structure [10], shown in Fig. 5(a), to obtain a high gain and a large output swing. Similar to the epots, the charge on the FG node of these amplifiers is precisely programmed by monitoring the amplifier output while the system operates in the reset mode. In this mode, the shift registers are cleared and the Reset signal is enabled. Therefore, all the input voltages to the input capacitors including the voltage to the feedback capacitor, C_{FB}, are set to the reference voltage. These conditions ensure that the amplifier output becomes equal to the reference voltage when the charge on the FG is compensated. The charge on the FG terminal is tuned using the hot-electron injection and the Fowler–Nordheim tunnelling mechanisms. By using this technique, the offset at the amplifier output is reduced to less than 1 mV.

Lastly, S/H circuits need to be designed to simultaneously achieve high sampling speed and high sampling precision due to the bit-serial nature of the DA computation. Therefore, these circuits are implemented by utilizing the S/H technique using Miller hold capacitance [11], as illustrated in Fig. 5(b). This compact circuit minimizes the signal dependent error, while maintaining the sampling speed and precision by using the Miller capacitance technique together with AMP_3 shown in Fig. 5(c). For simplification, if we assume there is no coupling between M_1 and M_2, and amplifier AMP_3, has a large gain, then the pedestal error contributed from turning switches (M_1 and M_2) off can be written as

$$\Delta V_{S1} + \Delta V_{S2} = \frac{\Delta Q_1 (C_2 + C_{2B})}{C_{2B}(C_1 + C_2) + C_1 C_2 (A+1)} + \frac{\Delta Q_2}{C_2} \tag{4}$$

where ΔQ_1 and ΔQ_2 are the charges injected by M_1 and M_2, respectively. Also, A and C_{2B} are the gain and input capaci-

tance of the amplifier, AMP_3. ΔQ_2 is independent of the input level, therefore ΔV_{S2} can treated as an offset. In addition, the error contributed by M_1, ΔV_{S1}, can be minimized by the Miller feedback, and this error decreases as A increases [11]. Due to serial nature of the DA computation offset in the feedback path is attenuated as the precision of the digital input data increases. Therefore, AMP_3 is designed to minimize mainly the signal dependent error ΔV_{S1}.

Moreover, a gain-boosting technique [12] is incorporated into the S/H amplifier AMP_4, as shown in Fig. 5(d), to achieve a high gain and fast settling. Two S/H circuits are used in the feedback path to obtain the fixed delay for the sampled analog voltage. In addition, the third S/H is utilized to S/H the final computed output once every K cycles. This S/H uses a negative-feedback output stage [13], shown in Fig. 5(e), to be able to buffer the output voltage off-chip. Due to the performance requirements of the system, these S/H circuits consume more power than the rest of the system.

V. Tuning of the Stored Weights: Epot Programming

The epots are incorporated into the design not only to store the weights of DA, but also to obtain reconfigurability/tunability. In this section, programming of these epots is described.

The epot programming circuitry is shown in Fig. 4(b). The stored voltage is tuned by the using Fowler–Nordheim tunnelling and the hot-electron injection mechanisms. The tunnelling is utilized for coarse programming of the epot voltage, and used to reach 200 mV below the target voltage. The purpose of undershooting is to avoid the coupling effect of the tunnelling junction on the floating gate when tunnelling is turned off. The tunnelling mechanism decreases the number of electrons, thus increasing the epot voltage. After selecting the desired epot by enabling its select signal, the tunnelling bit, $\mathrm{digtunnel}$, is activated and a high voltage across the tunnelling junction is created. During programming, the high-voltage amplifier is powered with 14 V.

In contrast to the coarse programming, the precise programming is achieved by using the hot-electron injection. The hot-electron injection mechanism decreases the epot voltage by increasing the number of electrons on the FG terminal. It is performed by pulsing a 6.5 V across the drain and the source terminals of a FG, as illustrated in Fig. 4(b). As the FG voltage, V_{fg}, decreases, the injection efficiency drops exponentially since the injection transistor has better injection efficiency for smaller source-to-gate voltages. By keeping the FG potential at a constant voltage, the number of injected electrons, hence the output voltage change, is accurately controlled. To keep the FG at a constant potential, the input voltage of the epot V_{ref} is modulated during programming based on the epot voltage output, since the epot output is approximately at the same potential as V_{fg}.

After the programming, the tunnelling and injection voltages are set to ground to decrease the power consumption, and minimize the coupling to the floating-gate terminal. Also, V_{ref} is set to 2.5 V to have the same reference voltage for all parts of the system. The epot voltage is programmed with respect to this voltage reference with an error less than 1 mV for a 4-V output range. The amount of charge that needs to be stored at an epot depends on the targeted weight and the gain error introduced by the input/feedback capacitors at the addition stage. During the programming, the Reset signal is enabled and all other capacitor inputs are connected to V_{ref} while periodically switching the targeted epot to find the voltage difference when epot is selected and unselected. This voltage is used to find the approximate value of the stored weight.

VI. Analysis of Error Sources

DA is typically implemented in digital, therefore an analysis of the error sources generated by the analog components of the proposed circuit must be performed. These error sources include gain and offset errors, nonideal weights, and noise in the signal path. In this section, we identify the source of these errors and analyze their contribution on DA. The effect of nonideal weights mostly depends on the application that DA is used for. In this paper, we analyze their effect on FIR filtering applications.

A. Computational Errors

As in serial digital-to-analog converters (DACs), the gain and offset errors determines the accuracy of DA computation. If the error at the addition stage due to the weight errors in the epots and the mismatch errors between the input capacitors, C_{in_i} (for $i = 1, 2, \ldots$), is assumed to be negligible, then the gain/offset errors and the noise in the data paths become the main sources of error. In this architecture, the inverting amplifiers, AMP_1 and S/H_2, introduce gain and offset errors, and the S/H circuits S/H_1 and S/H_2 cause offset errors. In addition, the mismatch between C_{FB} and C_{FBamp_1} as well as between C_{FBamp_2} and C_{inamp_2} cause gain errors. In this paper, the effects of gain, offset, and random errors in the system were analyzed.

1) Gain Error: Unlike in the digital domain where a division by two is simply a shift of a bit, in the analog domain this operation is achieved by employing an analog circuit. This circuit implementation usually introduces an error and the result of the division becomes 0.5 plus a gain error Δ. The effect of Δ on the output of a DA computation, $y[n]$, is modelled by

$$y[n] = -\sum_{i=0}^{M-1} w_i b_{i0} + \sum_{j=1}^{K-1} (0.5+\Delta)^j \sum_{i=0}^{M-1} w_i b_{ij}. \quad (5)$$

The output error caused by Δ can be found by computing the difference of (2) and (5). For simplification, the term $\sum_{i=0}^{M-1} w_i b_{ij}$ is set to α. Therefore, the output error, ε, reduces to the difference of two geometric sums and can be expressed as

$$\begin{aligned}\varepsilon &= \alpha \frac{1-(0.5+\Delta)^K}{1-(0.5+\Delta)} - \alpha \frac{1-0.5^K}{1-0.5} \\ &= \alpha \frac{1-(0.5+\Delta)^K}{0.5-\Delta} - \alpha \frac{1-0.5^K}{0.5}. \end{aligned} \quad (6)$$

A plot of the output error due to the gain error normalized by α for varying values of Δ and K is given in Fig. 6(a). Since this system converts the digital input data to an analog output, the output error due to quantization is also provided. The intersection of output error (due to gain error) and quantization error curves provides the minimum achievable output error of

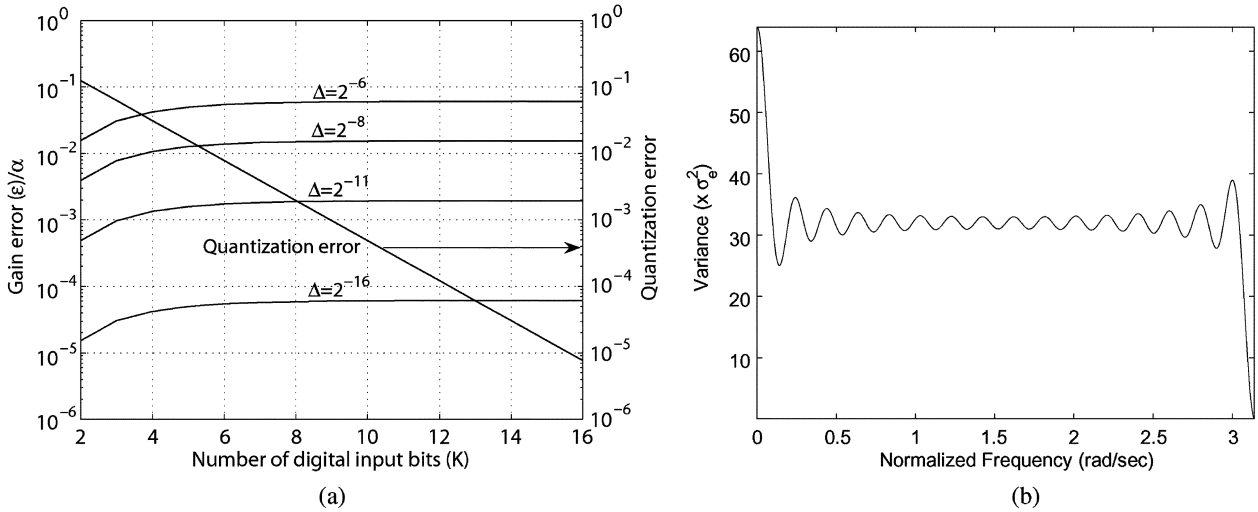


Fig. 6. (a) Computational error ε/α of the system due to quantization error and gain nonideality, Δ. (b) Frequency response of the variance for symmetric offset error $M = 32$.

the proposed system and determines the precision of an equivalent digital system. For example, when $\Delta = 2^{-11}$, the two curves intersect at $\varepsilon/\alpha = 0.002$ and $K = 8$. This intersection point represents the minimum error when $\Delta = 2^{-11}$ and that proposed system is equivalent to using an 8-bit digital DA. Also note that as K becomes large, ε approaches a limit which is equal to $2\Delta/(\Delta - 0.5)$.

2) Offset Feedback Error: Another source of error is the offset error. It is modelled as a constant error δ added to each jth summation of weights, $\sum_{i=0}^{M-1} w_i b_{ij}$, as follows:

$$y[n] = -\left[\delta + \sum_{i=0}^{M-1} w_i b_{i0}\right] + \sum_{j=1}^{K-1} 2^{-j}\left[\delta + \sum_{i=0}^{M-1} w_i b_{ij}\right]. \quad (7)$$

After distributing $\sum_{j=1}^{K-1} 2^{-j}$ and then grouping the δ into one term, the error due to offset can be written as the summation of a geometric series

$$\text{error}_{\text{offset}} = \delta\frac{1-0.5^K}{1-0.5} - 2\delta = \delta \cdot 2^{-(K-1)}. \quad (8)$$

As K increases, the offset error in the feedback loop decreases, which is a byproduct of how DA handles two's complement numbers. In DA, the last summation of weights, $\sum_{i=0}^{M-1} w_i b_{i0}$, is subtracted rather than added. This system decreases the offset error especially when the K is large. For $K = 8$ and $\delta = 100\ \text{mV}$, the offset error becomes 0.7813 mV.

3) Random Feedback Error: The random error is assumed to be Gaussian and is represented by X_j. The random variable X_j is added to the summation of weights at each jth iteration, and all X_j's are independent and identically distributed

$$y[n] = -\left[X_0 + \sum_{i=0}^{M-1} w_i b_{i0}\right] + \sum_{j=1}^{K-1} 2^{-j}\left[X_j + \sum_{i=0}^{M-1} w_i b_{ij}\right]. \quad (9)$$

Once the term $\sum_{j=1}^{K-1} 2^{-j}$ is distributed and the $X_j's$ are collected into one summation, the mean and variance of $y[n]$ can be written as $\mu_Y = \mu_X((1-0.5^K)/(1-0.5)) - 2\mu_X$ and $\sigma_Y^2 = \sigma_X^2((1-0.25^K)/(1-0.25))$, respectively. As K approaches infinity, the mean of the random error approaches zero and the maximum variance of the random error becomes $(4/3)\sigma^2$.

B. Nonideal Weight Errors for FIR Filters

The errors due to nonideal filter weights, such as random offset error, are caused by the limited precision of the epot programming and the epot noise. The effects of these errors are similar to the quantization effects in the digital domain which causes the linear difference equation of an FIR filter to become a nonlinear [14]. In addition, an analysis of the random time-varying error of the filter weights is provided.

1) Offset Error: An analysis similar to the one presented in [14] is given. Although the analysis provided is for a Type 2 FIR filter, the analysis can be generalized to any type of symmetric FIR filter [14]. An analogous analysis can be performed for nonsymmetric FIR filters.

a) Symmetric Offset Error: The frequency response for $e[n]$ can be written as $E(w) = \sum_{n=0}^{M-1} e[n]e^{-jwn}$. Assuming the FIR filter is Type-2 and the offset errors are of the same symmetry as the filter, $E(w)$ can be rewritten as a summation of cosines

$$E(w) = e^{-jw\frac{M-1}{2}} \sum_{n=0}^{\frac{M}{2}-1} 2e[n]\cos\left(w\left(\frac{2n+1}{2}\right)\right). \quad (10)$$

Treating $e[n]$ as a random variable with a variance of σ_e^2 and using some trigonometric identities and Euler's rule, the variance of $E(w)$ can be written as follows:

$$\sigma_E^2(w) = \sigma_e^2\left(M + \frac{\sin(wM)}{\sin(w)}\right). \quad (11)$$

$\sigma_E^2(w)$ can vary from zero to $2M\sigma_e^2$. Its frequency response for $M = 32$ is illustrated in Fig. 6(b). The effects of the symmetrical offset errors are similar to the effects of coefficient quantization in symmetrical digital FIR filters. These effects are reduced pass-band width, increased pass-band ripple, increased transition-band, and reduced minimum stop-band attenuation [14].

TABLE I
IDEAL AND ACTUAL (PROGRAMMED EPOT VOLTAGES, V_{EP}) COEFFICIENTS OF THE COMB, LOW-PASS, AND BANDPASS FILTERS. $V_{\rm IEP}$ IS THE IDEAL EPOT VOLTAGE

Filter	Comb			LPF			BPF		
Coeff.	Ideal	V_{EP}(V)	V_{IEP}(V)	Ideal	V_{EP}(V)	V_{IEP}(V)	Ideal	V_{EP}(V)	V_{IEP}(V)
1	0.4	2.0996	2.1	-0.0190	2.5192	2.519	0.033	2.4670	2.467
2	0	2.4994	2.5	-0.0390	2.5393	2.539	-0.064	2.5639	2.564
3	0	2.4994	2.5	0.0260	2.4738	2.474	-0.053	2.5530	2.553
4	0	2.5007	2.5	0.0160	2.4835	2.484	0.038	2.4617	2.462
5	0	2.5005	2.5	-0.0240	2.5239	2.524	0.047	2.4528	2.453
6	0	2.5000	2.5	-0.0360	2.5362	2.536	-0.054	2.5541	2.554
7	0	2.4999	2.5	0.0600	2.4401	2.440	-0.056	2.5561	2.556
8	0	2.4994	2.5	0.1800	2.3201	2.320	0.057	2.4425	2.443
9	0	2.4997	2.5	0.1800	2.3201	2.320	0.057	2.4427	2.443
10	0	2.5002	2.5	0.0600	2.4391	2.440	-0.056	2.5560	2.556
11	0	2.4998	2.5	-0.0360	2.5358	2.536	-0.054	2.5535	2.554
12	0	2.5002	2.5	-0.0240	2.5240	2.524	0.047	2.4527	2.453
13	0	2.5001	2.5	0.0160	2.4853	2.484	0.038	2.4616	2.462
14	0	2.5001	2.5	0.0260	2.4743	2.474	-0.053	2.5526	2.553
15	0	2.4997	2.5	-0.0390	2.5389	2.539	-0.064	2.5638	2.564
16	0.4	2.0996	2.1	-0.0190	2.5184	2.519	0.033	2.4669	2.467

b) Nonsymmetric Offset Error: Unlike the previous analysis, $E(w)$ cannot be rewritten as a summation of cosines because the offset error is not symmetrical. Assuming $e[n]$ is a random variable with a variance of σ_e^2, the variance of $E(w)$, σ_E^2, is equal to $M\sigma_e^2$ for an M-tap FIR filter. Unlike the variance for symmetrical offset errors which varies with frequency, the variance for nonsymmetric offset errors is constant.

2) Random Error: The effects of time-varying random error on DA computation can be modelled as

$$y[n] = -\sum_{i=0}^{M-1}(w_i+e_{i0})b_{i0}+\sum_{j=1}^{K-1}2^{-j}\sum_{i=0}^{M-1}(w_i+e_{ij})b_{ij}. \quad (12)$$

Assuming each e_{ij} is a random variable that is independent and identically distributed, the error can be expressed as

$$\text{error} = -\sum_{i=0}^{M-1}e_{i0}b_{i0}+\sum_{j=1}^{K-1}2^{-j}\sum_{i=0}^{M-1}e_{ij}b_{ij}. \quad (13)$$

Since the above equation is just a summation of random variables, the parameter of significance for this analysis is the maximum variance of the random error $\sigma^2_{\rm error}$. For simplification, the analysis assumed that b_{ij} for all i and j is equal to 1. First, the variance of $-\sum_{i=0}^{M-1}e_{i0}b_{i0}$ is computed as $M\sigma^2$. Then, the variance of $\sum_{j=1}^{K-1}2^{-j}\sum_{i=0}^{M-1}e_{ij}b_{ij}$ is calculated as $M\sigma^2((1-0.5^K)/(1-0.5))$. These two variances results in a total variance, $\sigma^2_{\rm error} = M\sigma^2(3-0.5^{K-1})$, which approaches $3M\sigma^2$ when K is large.

VII. MEASUREMENT RESULTS

In this section, we present the experimental results from the proposed DA architecture, which is configured as an FIR filter. The measurement results are obtained from the chips that were fabricated in a 0.5-μm CMOS process. This CMOS process was chosen for prototyping and to prove the proposed concept since the programming and retention characteristics of the CMOS floating-gate devices are well characterized in this process. In this process, it was shown that the stored charge on the floating-gate drifts around $10^{-3}\%$ over the period of 10 years at 25 °C [15], and this makes the use of floating-gate transistors suitable for the DA implementation. The 16-tap FIR filter is designed to run at 32/50-kHz sampling frequency depending on the desired performance. The precision of the digital input data is set to 8 for these experiments. To meet this sampling rate, the data is loaded into the upper shift register at a rate of 3.84 MHz for a 32-kHz sampling frequency or 6.4 MHz for a 50-kHz sampling frequency.

To demonstrate the reconfigurability, the filter is configured as a comb, a low-pass, and a bandpass filter. The coefficients of these filters are shown in Table I. Ideal coefficients are given to illustrate how close the epots are programmed to obtain the actual coefficients. The epots are programmed relative to a reference voltage $V_{\rm ref}$, which is set to 2.5 V. The error of the stored epot voltages are kept below 1 mV to minimize the effect of weight errors on the filter characteristics. Also, it was shown that epot exhibits a temperature coefficient of less than 37 $\rm ppmm/^{\circ}C$ [16], and therefore it can be readily utilized in this implementation to retain the characteristics of the configured filter across temperature.

An 858-Hz sinusoidal output of the low-pass filter at a 50-kHz sampling rate is illustrated in Fig. 7(a). The spurious-free-dynamic-range (SFDR) of this signal is measured to be 43 dB. For the comb filter with a 22-kHz input signal frequency, it is observed that the SFDR does not degrade as shown in Fig. 7(b). Although the input precision was set to 8 bits, the gain error in the system as well as noise in the experimental set-up limits the maximum achievable SFDR.

The second experiment is performed to characterize the magnitude and phase responses of the filters. For that purpose, a sinusoidal wave at a fixed sampling rate, 32/50 kHz, is generated using the digital data, and the magnitude and phase responses are measured by sweeping the frequency of the input sine wave from dc to 16/25 kHz. For this experiment, 256 data points are collected to accurately measure the frequency response of these filters. These responses follow the ideal responses closely even if the sampling rate is increased as illustrated in Fig. 8(a)–(c).

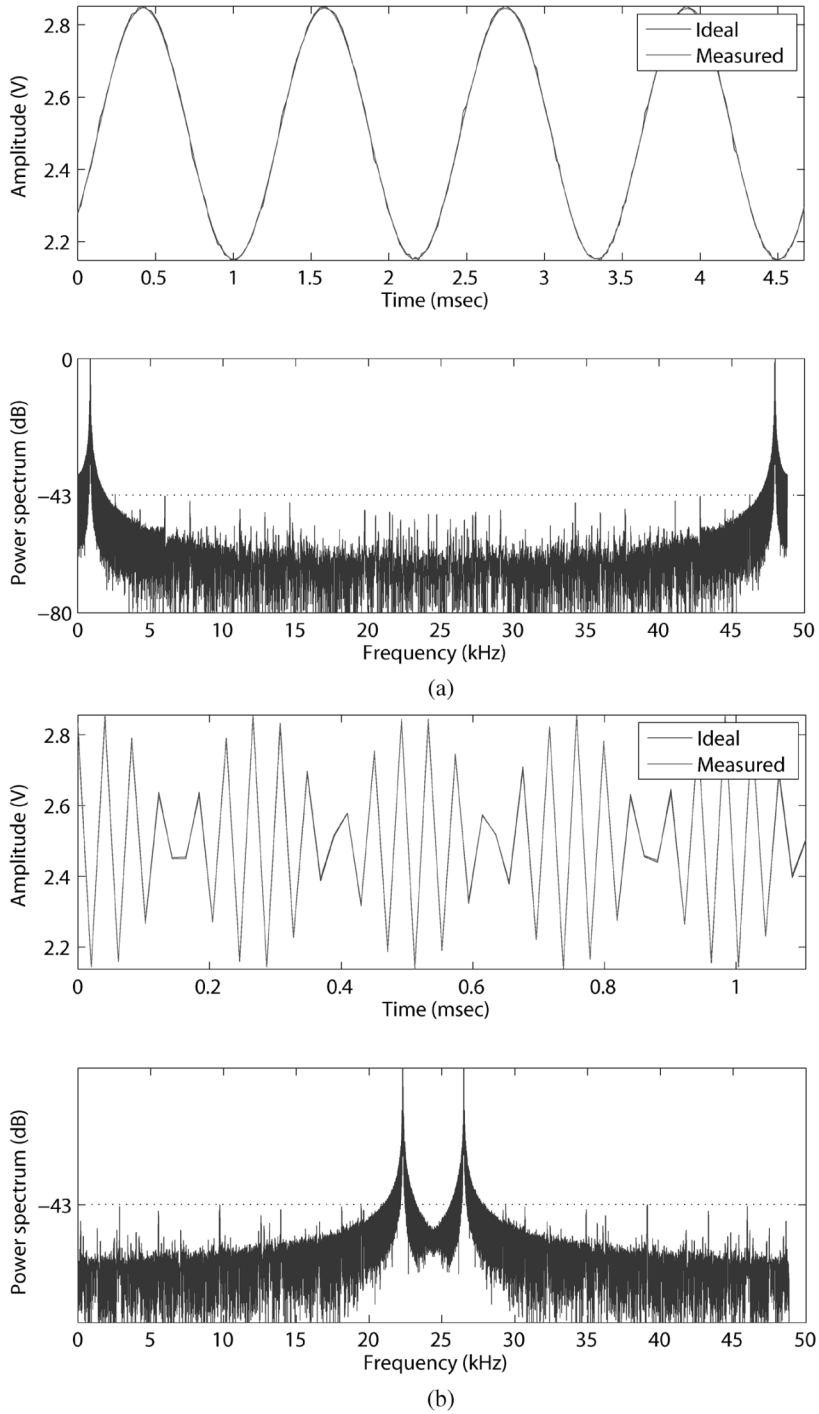


Fig. 7. Transient responses for 50-kHz sampling frequency and their power spectrums. (a) Low-pass filter output has a frequency of 858 Hz. (b) Comb filter output has a frequency of 22 kHz.

Any variation in the frequency response as the sampling rate increases is caused by the noise and offset in the feedback path as well as due to the performance degradation of the circuits. As the output signal amplitude becomes very low, the experimental set-up limits the resolvable magnitude and phase. As expected for a symmetrical FIR filter, the measured phase responses of comb, low-pass, and bandpass filters are linear.

As illustrated in Figs. 7 and 8, the measured data follows the ideal data very closely. Although the computational errors in these measurements are not quantified individually, the noise floor and quantization error in Fig. 7 and the deviation from the phase and frequency responses in Fig. 8 are mainly attributed to the gain/offset errors and the noise in the data paths and the setup. In order to make sure that the quantization error, whose effect on the total error is illustrated in Fig. 6(a) and theoretically analyzed in Section VI-A, is not the limiting factor, the input precision is increased from 8 to 16, but there is no improvement is observed. As it is also expected from the theoretical analysis, the gain error is measured to be main source of error in addition to the noise in the experimental setup. In ad-

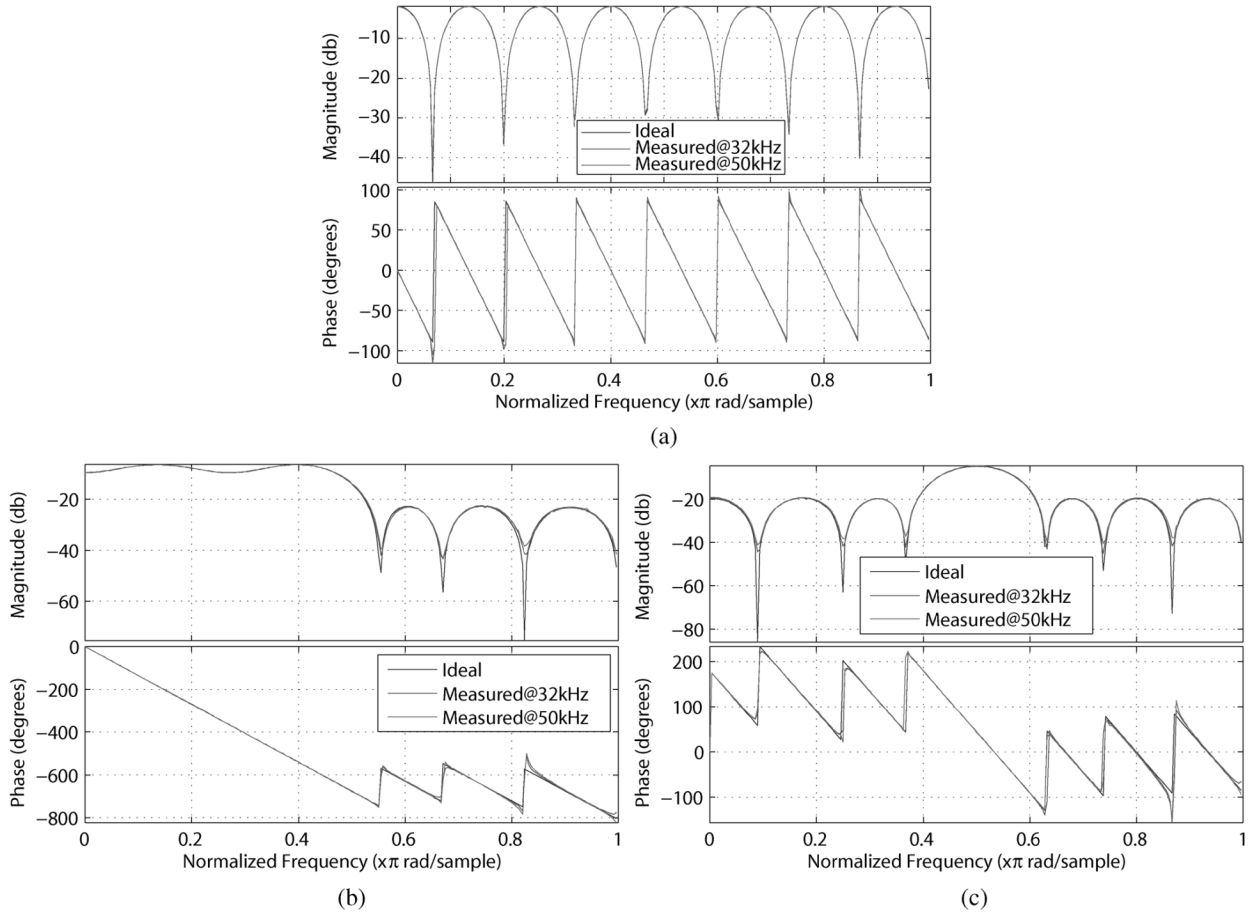


Fig. 8. Magnitude and phase responses at 32/50-kHz sampling rates. (a) Comb filter. (b) Low-pass filter. (c) Bandpass filter.

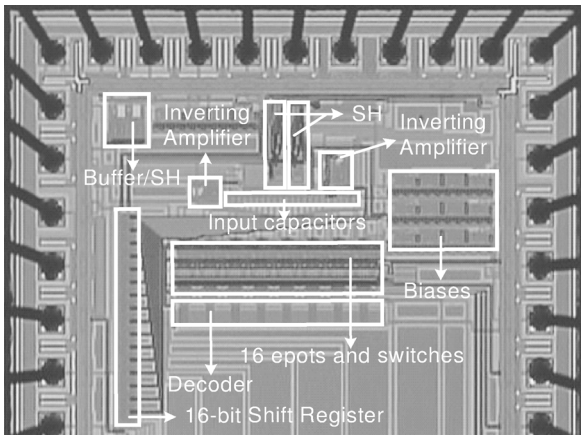


Fig. 9. Die photo of the DA based FIR filter chip.

TABLE II
PERFORMANCE AND DESIGN PARAMETERS OF THE FIR FILTER

Process	$0.5\mu m$, $2-poly$ CMOS
Power supply	$5V$
Reference voltage	$2.5V$
Epot Programming Resolution	100μV
Programming Mechanisms	Hot-Electron Injection and Electron Tunneling
Unit capacitor	$300fF$
Sampling frequency	$30/50KHz$
Input data precision	8
Number of filter taps	16
Increase in the power per tab	$0.02mW$
Increase in the area per tab	$0.011mm^2$
Total static power consumption	$16mW$
Used chip area	$\sim 1.125mm^2$

dition, the nonideal weight errors are observed to be least of a problem since the nonideal effects on the FIR filter behavior, such as reduced passband width and increased passband ripple, listed in Section VI-B was minimal.

The static power consumption of the fabricated chip is measured as 16 mW. Most of the power is consumed by the S/H and inverting amplifier circuits. The die photo of the designed chip is shown in Fig. 9. The system occupies around half of the $1.5 \cdot 1.5$ mm^2 die area. The cost to increase the filter order is 0.011 mm^2 of die area and 0.02 mW of power for each additional filter tap. This readily allows for the implementation of high-order filters. Lastly, the performance of the filter is summarized in Table II.

VIII. DISCUSSION

To obtain a programmability in the analog domain, a variety of design strategies has been suggested [17], [18]. The analog implementations of FIR filters have been generally designed for pre-processing applications by employing switched-capacitor and switch-current techniques.

Switched-capacitor techniques are suitable for FIR filter implementations and offer precise control over the filter coefficients. However, these techniques pose different design challenges depending upon the implementation. To avoid the power and speed trade-off in the switched-capacitor FIR filter implementations, a transposed FIR filter structure is usually employed [17]. Also, a parallel filter concept is suggested to increase the sample-rate-to-corner-frequency ratio of FIR filters [19]. In addition, a rotating switch matrix is used to eliminate the error accumulation [20]. Alternatively, these problems can be partially alleviated by employing over-sampling design techniques [18], [21], [22]. The filter implementations with these techniques offer design flexibility by allowing for coefficient and/or input modulation [23], [24]. However, this design approach requires the use of higher clock rates to obtain high over-sampling ratios.

The programmability in analog FIR filter implementations can also be obtained by utilizing switched-current techniques. These techniques allow for the integration of the digital coefficients through the use of the current division technique [25] or multiplying DACs (MDAC) [26], [27]. Moreover, a circular buffer architecture can be utilized to ease the problems associated with analog delay stages and to avoid the propagation of both offset voltage and noise [28]–[31]. Recently, a switched-current FIR filter based on DA has also been suggested for preprocessing applications to decrease the hardware complexity and area requirements of the FIR filters [32].

Some of these techniques can be employed for post-processing by using them after a DAC. However, the use of a high-resolution and/or high-speed DAC in addition to the FIR filter implementation causes an increase in the area and power consumption. Moreover, although these approaches solve some of the design problems, the implementations of these techniques still suffer from high area overhead and power consumption especially in the case of high-order FIR filters.

The proposed DA structure which can be used for FIR filtering circumvents some of these problems by employing DA for signal processing and utilizing the analog storage capabilities of FG transistors to obtain programmable analog coefficients for reconfigurability. In this way, the DAC is used as a part of the DA implementation, which helps achieving digital-to-analog conversion and signal processing at the same time.

Compared to the switched-capacitor implementations, which have their coefficients set by using different capacitor ratios, the proposed implementation offer more design flexibility since its coefficients can be set by tuning the stored weights at the epots. Also, offset accumulation and signal attenuation make it difficult to implement long tapped delay lines with these approaches. In the proposed implementation, we showed that DA processing decreases the offset as the precision of the digital input data increases. Also, the gain error in this implementation is mainly caused by the two inverting stages (implemented using AMP_1 and AMP_2), and can be minimized using special layout techniques only at these stages. The measurement results illustrated that the output signal of the filter follows the ideal response very closely. This is mainly because it is mostly insensitive to the number of filter taps and most of the computation is performed in the feedback path. Also, the power and area of the proposed design increases linearly with the number of taps due to the serial nature of the DA computation. Therefore, this design approach is well suited for compact and low-power implementations of high-order filters for post-processing applications. In the proposed design inverting amplifiers, S/Hs, and buffers consume 98.75% of the total power consumption. Each inverting amplifier uses less than 10% of the consumed power, therefore, the optimization of the power consumption requires minimizing the power consumed by the buffer and S/Hs.

The programmable analog coefficients of this filter will enable the implementation of adaptive systems that can be used in applications such as adaptive noise cancellation and adaptive equalization. Since DA is an efficient computation of an inner product, this architecture can also be utilized for signal processing transforms such as a modified discrete cosine transform.

REFERENCES

[1] S. N. Merchant and B. V. Rao, "Distributed arithmetic architecture for image coding," in *Proc. IEEE Int. Conf. TENCON'89*, Nov. 1989, pp. 74–77.

[2] H. Q. Cao and W. Li, "VLSI implementation of vector quantization using distributed arithmetic," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1996, vol. 2, pp. 668–671.

[3] M. T. Sun, T. C. Chen, and A. M. Gotlieb, "VLSI implementation of a 16x16 discrete cosine transform," *IEEE Trans. Circuits Syst.*, vol. 36, no. 6, pp. 610–617, Jun. 1989.

[4] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, "LMS adaptive filters uisng distributed arithmetic for high throughput," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 7, pp. 1327–1337, Jul. 2005.

[5] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital Filter for PCM Encoded Signals," U.S. Patent 3 777 130, 1973.

[6] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-22, no. 6, pp. 456–462, Jun. 1974.

[7] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE Trans. Acoust, Speech, Signal Process.*, vol. 37, no. 1, pp. 4–19, Jan. 1989.

[8] R. R. Harrison, J. A. Bragg, P. Hasler, B. A. Minch, and S. Deweerth, "A CMOS programmable analog memory cell array using floating-gate circuits," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 1, pp. 4–11, Jan. 2001.

[9] E. Özalevli and P. Hasler, "10-bit programmable voltage-output digital-analog converter," in *Proc. IEEE Int. Symp. Circuits Syst.*, Kobe, Japan, 2005, vol. 6, pp. 5553–5556.

[10] W. C. Black, D. J. Allstot, and R. A. Reed, "A high performance low power CMOS channel filter," *IEEE J. Solid-State Circuits*, vol. 15, no. 6, pp. 929–938, Dec. 1980.

[11] P. J. Lim and B. A. Wooley, "A high-speed sample-and-hold technique using a Miller hold capacitance," *IEEE J. Solid-State Circuits*, vol. 26, no. 5, pp. 643–651, Oct. 1991.

[12] K. Bult and G. J. G. M. Geelen, "A fast-settling CMOS opamp for SC circuits with 90-dB dc gain," *IEEE J. Solid-State Circuits*, vol. 25, no. 6, pp. 1379–1384, Dec. 1990.

[13] K. E. Brehmer and J. B. Wieser, "Large swing CMOS power amplifier," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 624–629, 1983.

[14] S. K. Haykin, *Digital Signal Processing: A Computer-Based Approach*. New York: McGraw-Hill, 2001.

[15] E. Özalevli, H. Dinc, H. J. Lo, and P. E. Hasler, "Design of a binary-weighted resistor DAC using tunable linearized floating-gate CMOS resistors," in *Proc. Custom Integr. Circuits Conf.*, San Jose, CA, 2006, pp. 149–152.

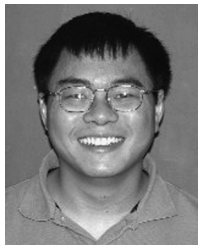
[16] E. Özalevli, W. Huang, P. Hasler, and D. Anderson, "VLSI implementation of a reconfigurable mixed-signal finite impulse response filter," in *Proc. IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, 2007, pp. 2168–2171.

[17] B. C. Rothenberg, S. H. Lewis, and P. J. Hurst, "A 20-Msample/s switched-capacitor finite-impulse-response filter using a transposed structure," *IEEE J. Solid-State Circuits*, vol. 30, no. 12, pp. 1350–1356, Dec. 1995.

[18] P. W. Wong and R. M. Gray, "FIR filters with sigma-delta modulation encoding," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 38, no. 6, pp. 979–990, Dec. 1990.
[19] S. K. Berg, P. J. Hurst, S. H. Lewis, and P. T. Wong, "A switched-capacitor filter in 2-μm CMOS using parallelism to sample at 80 MHz," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 1994, pp. 62–63.
[20] Y. S. Lee and K. W. Martin, "A switched-capacitor realization of multiple FIR filters on a single chip," *IEEE J. Solid-State Circuits*, vol. 24, no. 3, pp. 542–562, Jun. 1989.
[21] Q. Huang and G. Moschytz, "Analog FIR filters with an oversampled sigma-delta modulator," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 6, pp. 658–663, Jun. 1992.
[22] Q. Huang, G. Moschytz, and T. Burger, "A 100 tap FIR/IIR analog linear-phase lowpass filter," in *Dig. Tech. Papers VLSI Circuits Symp.*, 1995, pp. 91–92.
[23] N. Benvenuto, L. Franks, and F. Hill, Jr., "Realization of finite impulse response filters using coefficients $+1$, 0, and -1," *IEEE Trans. Commun.*, vol. COM-33, no. 10, pp. 1117–1125, Oct. 1985.
[24] S. S. Abeysekera and K. P. Padhi, "Design of multiplier free FIR filters using a LADF sigma-delta modulator," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2000, vol. 2, pp. 65–68.
[25] K. Bult and G. J. G. M. Geelen, "An inherently linear and compact MOST-only current division technique," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1730–1735, Dec. 1992.
[26] A. M. Chiang, "Low-power adaptive filter," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 1994, pp. 90–91.
[27] W. Figueroa, D. Hsu, and C. Diorio, "A mixed-signal approach to high-performance low-power linear filters," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 816–822, May 2001.
[28] S. Lyle, G. Worstell, and R. Spencer, "An analog discrete-time transversal filter in 2.0-μm CMOS," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 1992, vol. 2, pp. 970–974.
[29] Y. L. Cheung and A. Buchwald, "A sampled-data switched-current analog 16-tap FIR filter with digitally programmable coefficients in 0.8 μm CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 1997, pp. 54–55.
[30] X. Wang and R. R. Spencer, "A low-power 170-MHz discrete-time analog FIR filter," *IEEE J. Solid-State Circuits*, vol. 33, no. 3, pp. 417–426, Mar. 1998.
[31] F. A. Farag, C. Galup-Montoro, and M. C. Schneider, "Digitally programmable switched-current FIR filter for low-voltage applications," *IEEE J. Solid-State Circuits*, vol. 35, no. 4, pp. 637–641, Apr. 2000.
[32] P. Sirisuk, A. Worapishet, S. Chanyavilas, and K. Dejhan, "Implementation of switched-current FIR filter using distributed arithmetic technique: Exploitation of digital concept in analogue domain," in *Proc. IEEE Int. Symp. Commun. Inf. Technol.*, 2004, vol. 1, pp. 143–148.

Erhan Özalevli (S'04–M'06) received the B.Sc. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2001, the M.Sc. degree in electrical and computer engineering from the University of Arizona, Tucson, AZ, in 2003, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2006.

He is currently a Design Engineer at Texas Instruments, Dallas, TX. His research interests include mixed-mode multichip VLSI systems for bio-inspired and neuromorphic applications, tunable floating-gate CMOS resistors and their applications, and data converters for embedded systems.



Walter Huang (S'01) received the B.S. degree in electrical engineering and the M.S. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, in 1999, and 2003, respectively. He is currently working toward the Ph.D. degree at the same university.

His interests are in rapid hardware prototyping, computer architecture, and hardware implementations of digital signal processing algorithms on reconfigurable systems and processors. His current research topic is the design and application of distributed arithmetic for fixed and adaptive signal processing.

Paul E. Hasler (M'95–SM'04) received the B.S.E. and M.S. degrees in electrical engineering from Arizona State University, Tempe, in 1991, and the Ph.D. degree in computation and neural systems from the California Institute of Technology, Pasadena, in 1997.

He is an Associate Professor in the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta. His current research interests include low-power electronics, mixed-signal system integrated circuits, floating-gate MOS transistors, adaptive information processing systems, "smart" interfaces for sensors, cooperative analog-digital signal processing, 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 National Science Foundation CAREER Award in 2001, and the Office of Naval Research YIP award in 2002. He received the Paul Raphorst Best Paper Award, IEEE Electron Devices Society, 1997, and a Best Paper Award at SCI 2001.

David V. Anderson (S'92–M'98–SM'03) received the B.S. and M.S. degrees from Brigham Young University, Provo, UT, and the Ph.D. degree from Georgia Institute of Technology (Georgia Tech), Atlanta, in 1993, 1994, and 1999, respectively.

He is an Associate Professor in the School of Electrical and Computer Engineering at Georgia Tech and an Associate Director of the Center for Research in Embedded Systems Technology. His research interests include audio and psycho-acoustics, signal processing in the context of human auditory characteristics, and the real-time application of such techniques using both analog and digital hardware. His research has included the development of a digital hearing aid algorithm that has now been made into a successful commercial product. He has over 60 technical publications and five patents/patents pending. He has been actively involved in the development and promotion of computer enhanced education.

Dr. Anderson was awarded the National Science Foundation CAREER Award for excellence as a young educator and researcher in 2004 and the 2004 Presidential Early Career Award for Scientists and Engineers (PECASE). He is a member of the IEEE, the Acoustical Society of America, ASEE, and Tau Beta Pi.