

# Low-Power Discrete Fourier Transform for OFDM: A Programmable Analog Approach

Sangwook Suh, *Student Member, IEEE*, Arindam Basu, *Member, IEEE*, Craig Schlottmann, *Student Member, IEEE*, Paul E. Hasler, *Senior Member, IEEE*, and John R. Barry, *Senior Member, IEEE*

**Abstract**—The modulation and demodulation blocks in an orthogonal frequency-division multiplexing (OFDM) system are typically implemented digitally using a fast Fourier transform circuit. We propose an analog implementation of an OFDM demodulator as a means for reducing power consumption. The proposed receiver implements the discrete Fourier transform (DFT) as a vector–matrix multiplier using floating-gate transistors on a field-programmable analog array (FPAA). The DFT coefficients can be tuned to counteract an inherent device mismatch by adjusting the amount of electrical charge stored in the floating-gate transistors. When compared to a digital field-programmable gate array implementation, the analog FPAA implementation of the DFT reduces power consumption at the cost of a slight performance degradation. Considering the errors in the DFT coefficients as intersymbol interference, the performance degradation can be further mitigated by employing a least mean-square or minimum mean-square-error equalizer.

**Index Terms**—Discrete Fourier transform (DFT), fast Fourier transform (FFT), field-programmable analog array (FPAA), floating-gate transistor, intersymbol interference (ISI), least mean square (LMS), minimum mean square error (MMSE), orthogonal frequency-division multiplexing (OFDM), vector–matrix multiplier (VMM).

## I. INTRODUCTION

ORTHOGONAL frequency-division multiplexing (OFDM) is widely used in numerous wireless communication systems not only because of its spectral efficiency and robustness to multipath fading but also because of its ease of implementation; OFDM modulators and demodulators can be implemented using simple fast Fourier transform (FFT) blocks, typically in digital circuits. However, for mobile devices with limited battery power, replacing these digital circuits with low-power analog circuits can significantly improve the power efficiency of the devices [1], [2]. The cost paid for this reduced power is the long development cycle and lack of flexibility that typifies analog circuit design.

So as to retain the rapid-prototyping capability and flexibility of a field-programmable gate array (FPGA) but with reduced power consumption, an analog counterpart of the FPGA, namely

a field-programmable analog array (FPAA), was proposed in [3], followed by several different FPAA realizations using a switched capacitor [4], [5], a transistor [6], or an operational transconductance amplifier (OTA) with a capacitor [7]. The early FPAAs, however, contained only a few computational elements, and their applications were restricted to analog filters, until floating-gate transistors were used as switches of the FPAA to enable large-scale analog circuit design [2], [8]. Recently, a hexagonal arrangement of computational analog blocks (CABs) has been reported in [9] and [10] to reduce the size and path delay of the FPAA chip. Two decades since its advent, the FPAA is finding viability in space applications as well by imposing self-reconfigurable features [11], [12].

There have been several dedicated nonprogrammable analog implementations of FFT and discrete Fourier transform (DFT) circuits. A voltage-mode analog FFT block was reported in [13] and [14] that uses analog multipliers and dedicated input signals representing the FFT coefficients and analog adders for summing voltage signals. An FFT based on analog current mirrors was proposed in [15], where the FFT coefficients are not reconfigurable but are determined by the  $W/L$  ratio of the output transistor of each mirror. More recently, a numerical simulation for approximating a fast DFT operation with a 2-D lattice of inductors and capacitors has been introduced in [16].

To overcome the drawbacks of previous works, we present in this paper a current-mode analog DFT block implemented as a vector–matrix multiplier (VMM) using the reconfigurable analog signal processor (RASP) 2.9 FPAA chip [8]. Floating-gate transistors inside the FPAA chip are used as partially connected switches to store the DFT coefficients by locking in an appropriate amount of electrical charge in each floating-gate capacitor. Therefore, dedicated input signals representing the DFT coefficients are not required. Furthermore, these coefficients are reconfigurable without changing the circuit structure. The VMM structure using floating-gate transistors as programmable switches also enables tuning the DFT coefficients to compensate for the inherent mismatch between different transistors. Another benefit of our current-mode design over a voltage-mode circuit is the ease with which signals can be added, which is particularly beneficial in systems having multiple inputs and multiple outputs. The RASP 2.9 FPAA chip contains more computational elements than previous FPAA chips, including the commercial products in [17] and [18]. The large number of computational elements and the configurable floating-gate switches make the RASP 2.9 FPAA chip viable in a wide range of applications. Such versatility is an important figure of merit for any programmable circuit.

In Section II, we present the system description and analysis. In Section III, we summarize the FPAA programming procedure. In Section IV, we describe the FPAA measurement and equalization procedure. In Section V, we present our conclusions.

Manuscript received April 22, 2010; revised July 03, 2010; accepted July 30, 2010. This paper was recommended by Associate Editor M. Delgado-Restituto.

S. Suh, C. Schlottmann, P. E. Hasler, and J. R. Barry are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: swsuh@ece.gatech.edu; cschlott@gatech.edu; phasler@ece.gatech.edu; john.barry@ece.gatech.edu).

A. Basu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: arindam.basu@ntu.edu.sg).

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

Digital Object Identifier 10.1109/TCSI.2010.2071950

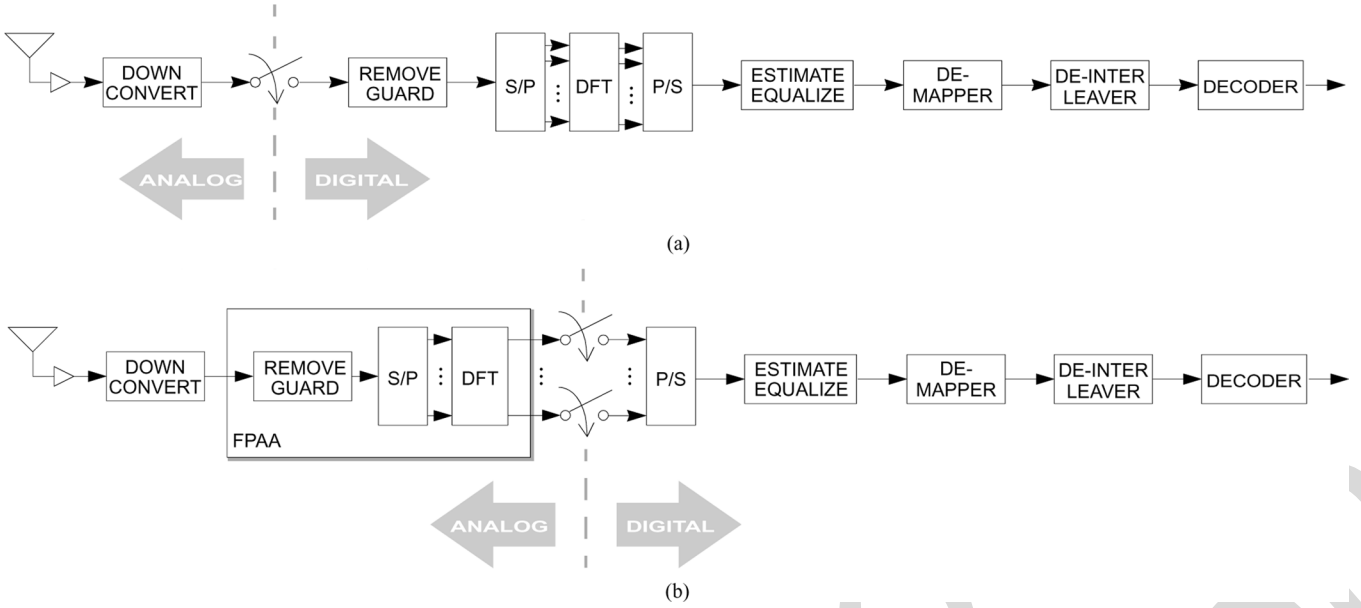


Fig. 1. OFDM receiver. (a) Conventional implementation in which sampling occurs before a digital DFT. (b) Proposed implementation in which sampling occurs after an analog DFT.

## II. SYSTEM DESCRIPTION AND ANALYSIS

### A. Motivation

In Fig. 1(a), we illustrate a simplified block diagram of a conventional OFDM receiver, such as for an 802.11 a/g system, where the received signal is sampled immediately after down-conversion. The OFDM demodulation is performed digitally using a DFT. As an alternative, we propose an analog implementation, as shown in Fig. 1(b), where the downconverter output is fed directly to the FPAA, with no sampling. The DFT functionality is implemented in analog using the FPAA. The  $N$  outputs of the FPAA—one for each subcarrier—are each sampled separately.

Besides the reduced power consumption with an analog implementation of the DFT, an important benefit of the proposed receiver structure in Fig. 1(b) is that it greatly relieves the speed and precision burdens of the analog-to-digital converter (ADC). In particular, the ADC of the conventional receiver shown in Fig. 1(a) would need to sample at a rate equal to the full signal bandwidth, and its precision would need to be high (on the order of 10 bits or more) to accommodate the wide dynamic range and Gaussian-like distribution of OFDM signals. In contrast, the proposed receiver in Fig. 1(b) has not one but  $N$  ADCs, one for each subcarrier, with a sampling rate slower by a factor of  $N$ . Moreover, each DFT output is a finite alphabet signal that can be sampled with significantly less bit precision [14]. Compared to the ADC in Fig. 1(a), each ADC in Fig. 1(b) requires a sampling rate that is lower by a factor of  $N$ , and the number of bits of precision is smaller by a factor of three or more, depending on the modulation alphabet size. Both effects yield a reduction in ADC power consumption, although the exact amount depends on the type of the ADC structure; the ADC power consumption is between a linear and quadratic function of the sampling rate [19].

Thus, the proposed receiver in Fig. 1(b) is beneficial not only because of the reduced power consumption with an analog implementation of the DFT but also because of the additional power savings resulting from the lower speed and bit-precision

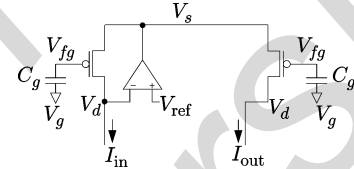


Fig. 2. Current multiplier circuit composed of floating-gate transistors and an OTA. The output current is a scaled multiple of the input current.

requirements of the subsequent ADC. Although our focus here is on the receiver side, we briefly point out that these same advantages are also valid at the transmitter side, where the outputs from the symbol mapper are modulated with an inverse DFT (IDFT) block, which has the same structure as the DFT block with different coefficients. Besides the power savings with an analog IDFT implementation, shifting the IDFT block after the digital-to-analog converter (DAC) enables an OFDM transmitter to replace a full-speed high-precision DAC by  $N$  separate DACs, each operating at a  $1/N$  times lower clock with lower bit precision.

### B. Floating-Gate Transistors and the RASP 2.9 FPAA

Floating-gate transistors can store a nonvolatile electrical charge, so arrays of floating-gate transistors can be programmed as a signal processing block for specific functionality. One application is to use them as a VMM circuit. Fig. 2 shows a current multiplier circuit—the basic element of a VMM circuit—composed of floating-gate transistors and an OTA. Two pMOSFETs are connected at the source, and the gate of each transistor is connected to a capacitor  $C_g$  to be electrically isolated so as to form a *floating gate*. The source voltage  $V_s$  is common for both transistors, and  $V_{fg}$  is the voltage potential at the floating gate.  $V_d$  is the drain voltage of the transistor. The voltages on the other side of the capacitors connected to the gates are set to be the same at the fixed potential  $V_g$ .

The input current  $I_{in}$  and output current  $I_{out}$  are defined as drain currents of the transistors operating in the subthreshold region. Neglecting the Early effects, the input and output currents of the pMOSFET are given by [20]

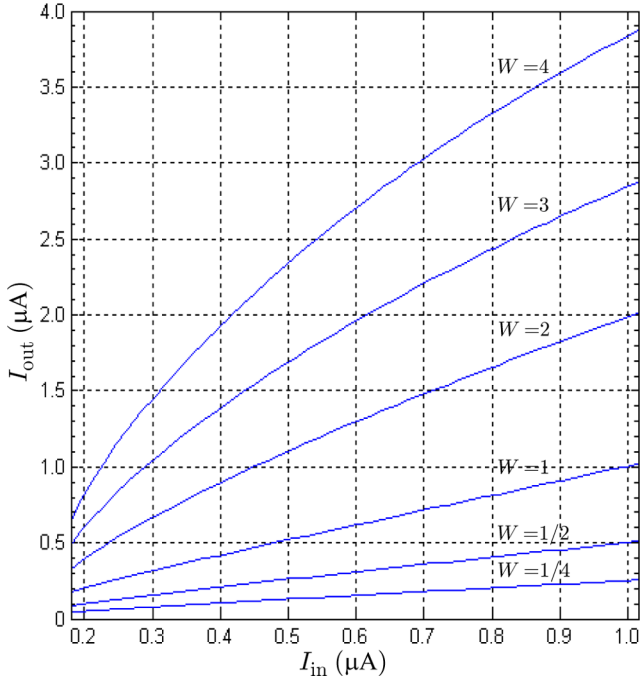


Fig. 3. Output currents with the programmed weights of  $1/4$ – $4$ . The input current is swept from  $0.2$  to  $1.0 \mu\text{A}$ . The straight curves for  $W \leq 1$  show the programmed circuit can be used as a current multiplier in the current range. As  $W$  increases, the output current shows transition toward the strong-inversion region.

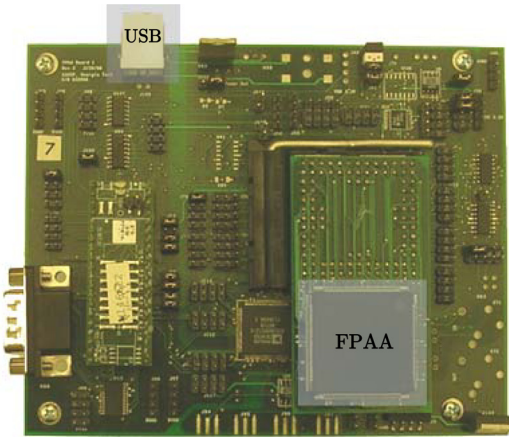


Fig. 4. RASP 2.9 FPAA chip mounted on the board. The chip is fabricated with the  $0.35\text{-}\mu\text{m}$  CMOS process. The board has 56 I/O pins for setting drain voltages of floating-gate transistors and measuring output currents. It is connected to a PC through a USB interface for controlling the FPAA chip programming. The sizes of the chip and the board are  $5 \text{ mm} \times 5 \text{ mm}$  and  $114 \text{ mm} \times 140 \text{ mm}$ , respectively.

$$I_{\text{in}} = I_1 \exp\left(\frac{V_s - \kappa_{\text{eff}} V_g}{U_T}\right) \exp\left(\frac{-\kappa Q_1}{C_t U_T}\right) \quad (1)$$

$$I_{\text{out}} = I_2 \exp\left(\frac{V_s - \kappa_{\text{eff}} V_g}{U_T}\right) \exp\left(\frac{-\kappa Q_2}{C_t U_T}\right) \quad (2)$$

where  $U_T = kT/q$  (where  $T$  is the temperature,  $k$  is the Boltzmann's constant, and  $q$  is the elementary charge) and  $C_t$  is the total capacitance at each gate, including the floating-gate capacitor  $C_g$  and the internal capacitance of the MOSFET.  $Q_1$  and  $Q_2$

are the electrical charges stored in the input and output floating-gate transistors, respectively.  $\kappa$  is the back-gate coefficient, and  $\kappa_{\text{eff}} = \kappa C_g / C_t$ . The parameters  $I_1$  and  $I_2$  are the preexponential factors of the MOSFETs that can be defined as a drain current flowing in each transistor when  $V_s = \kappa_{\text{eff}} V_{\text{fg}}$ . When there is a negligible mismatch between the threshold voltage<sup>1</sup> of the input and output transistors so that  $I_1$  and  $I_2$  are approximately identical, the ratio of the output current to the input current reduces to

$$W = \frac{I_{\text{out}}}{I_{\text{in}}} = \exp\left(\frac{-\kappa(Q_2 - Q_1)}{C_t U_T}\right). \quad (3)$$

Therefore, the weighting coefficient  $W$  is determined by the difference in charge values between the input and output floating-gate transistors, provided that both transistors operate in the sub-threshold region. One can observe in (3) that  $W$  is also a function of temperature. Note that, from (3), only positive weights can be realized. The weight zero can be realized by not connecting the input and output floating-gate transistors.

In practice, there exists an inherent mismatch between the threshold voltage of different transistors which leads to a mismatch in the preexponential factors  $I_1$  and  $I_2$ , and this, in turn, leads to multiplicative distortion in the programmed weights. However, this mismatch can be compensated for by adjusting the charge values  $Q_1$  and  $Q_2$  while they are programmed into the FPAA chip. Although a floating-gate current mirror can be similarly tuned over a narrow range to compensate for the device mismatch, its weight is fixed by the  $W/L$  ratio of the transistors and cannot be changed widely without changing the circuit structure.

Fig. 3 shows a plot of the output currents versus input current for a set of programmed weights between  $1/4$  and  $4$ . The mismatch effect has been canceled through the weight programming procedure that will be discussed in Section III. The input current is swept from  $0.2$  to  $1.0 \mu\text{A}$ . For  $W \leq 1$ , the linear relation between  $I_{\text{in}}$  and  $I_{\text{out}}$  shows that the programmed circuit can be used as a current multiplier for the range of currents shown. It can be observed that the output current shows a transition to the strong-inversion region as  $W$  increases.

The basic current multiplier circuit in Fig. 2 can be expanded to a multiple-input multiple-output structure to construct a larger size VMM circuit in the FPAA. The RASP 2.9 FPAA [8] consists of 133 744 floating-gate transistors and 84 CABs, each of which contains three OTAs, three capacitors, a transmission gate, and a voltage buffer. The floating-gate transistors can be programmed as a VMM circuit by storing an appropriate amount of charge in each floating-gate capacitor. Fig. 4 depicts the RASP 2.9 FPAA chip mounted on the board. The chip is fabricated with a  $0.35\text{-}\mu\text{m}$  CMOS process. The size of the transistors is  $W = 1.8 \mu\text{m}$  and  $L = 0.6 \mu\text{m}$ . Fig. 5 depicts the wide-output-range OTA inside the CABs, which is used to supply the input and output currents of the VMM circuit. The board has 56 I/O pins that can be used to set drain voltages of the floating-gate transistors and to measure output currents. The programming process for the FPAA chip is controlled through a universal serial bus (USB) interface equipped on the board.

### C. VMM Representation of an Analog DFT

The key to an OFDM receiver is to compute the DFT of a set of complex samples  $\{x(n)\}$ , defined by

<sup>1</sup>The threshold voltage is the gate voltage at which channel formation occurs between the oxide and the body of a transistor.

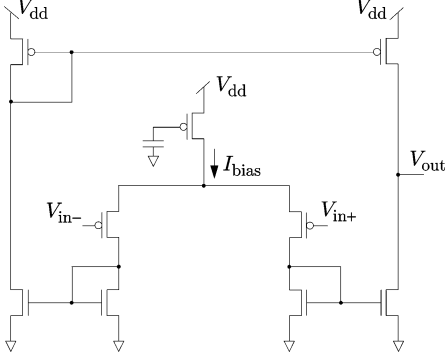


Fig. 5. Wide-output-range OTA inside the CABs of the RASP 2.9 FPAA chip.  $V_{dd}$  is the bias voltage, and  $I_{bias}$  is the bias current. The floating gate transistor in the middle is programmed with the charge value corresponding to the bias current. The targeting bias current is determined at the amount sufficient to provide the input and output currents of the connected floating-gate transistors.

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad (4)$$

where  $N$  is the number of subcarriers and  $k$  is an integer ranging from 0 to  $N - 1$ . By splitting each complex number into real and imaginary parts, we can rewrite (4) as

$$\text{Re}[X(k)] = \frac{1}{N} \sum_{n=0}^{N-1} \left[ \text{Re}[x(n)] \cos\left(\frac{2\pi kn}{N}\right) + \text{Im}[x(n)] \sin\left(\frac{2\pi kn}{N}\right) \right] \quad (5)$$

$$\text{Im}[X(k)] = \frac{1}{N} \sum_{n=0}^{N-1} \left[ -\text{Re}[x(n)] \sin\left(\frac{2\pi kn}{N}\right) + \text{Im}[x(n)] \cos\left(\frac{2\pi kn}{N}\right) \right] \quad (6)$$

which is equivalent to a VMM

$$\mathbf{X} = \mathbf{H}\mathbf{x} \quad (7)$$

where  $\mathbf{X}$  is a  $2N \times 1$  vector consisting of the real and imaginary components of  $\{X(k)\}$ ,  $\mathbf{x}$  is a  $2N \times 1$  vector consisting of the real and imaginary components of  $\{x(n)\}$ , and  $\mathbf{H}$  is a real-valued  $2N \times 2N$  matrix.

The VMM in (7) cannot be implemented directly, as some of the coefficients in  $\mathbf{H}$  are negative. Instead, we represent each signal differentially, and we represent each element of  $\mathbf{H}$  by a  $2 \times 2$  nonnegative differential submatrix by mapping a positive gain  $G$  to  $[G \ 0; 0 \ G]$  and a negative gain  $-G$  to  $[0 \ G; G \ 0]$ . We thus transform the  $2N \times 2N$  matrix  $\mathbf{H}$  into an equivalent  $4N \times 4N$  matrix with nonnegative weights that can be programmed into the FPAA chip.

For a size-4 DFT, we need a  $16 \times 16$  VMM circuit with real-valued nonnegative weights. The schematic of the  $16 \times 16$  VMM structure in the FPAA chip is shown in Fig. 6. As discussed in Section II-B, the weighting coefficients  $W_{1,1} - W_{16,16}$  can be programmed into the FPAA chip by assigning appropriate charge values to the input and output floating-gate transistors. Note that the weights in each column of the VMM circuit correspond to the coefficients in each row of the  $16 \times 16$  VMM matrix.

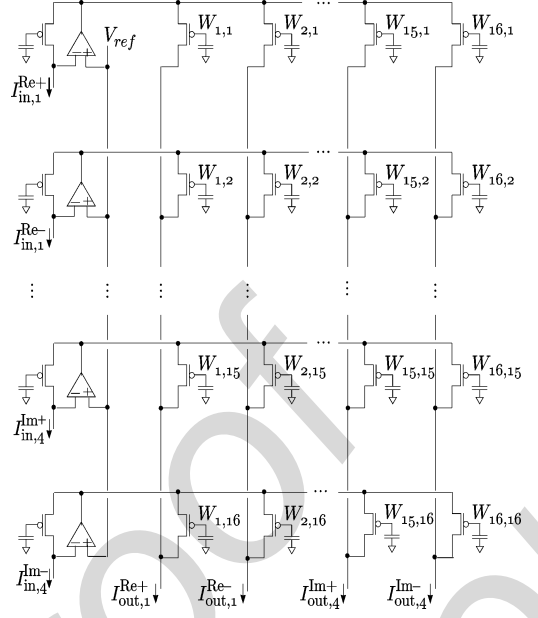


Fig. 6. FPAA implementation of a 4-point DFT as a  $16 \times 16$  VMM circuit. Each input current of the floating-gate transistors on the left determines the output voltage of the corresponding OTA, and this output voltage is broadcast to the source of all the connected floating-gate transistors in each row. At the output floating-gate transistor, this source voltage drives a drain current which is a scaled multiple of the corresponding input current. Then, the scaled currents are added up along each column to give a combined output current per column.

The output port of each OTA is connected to the source of the input floating-gate transistors, and the negative input port is connected to the drain. The positive input port is set to the reference voltage  $V_{ref}$ . Because of the negative feedback of the OTA, the drain voltage of each input floating-gate transistor is also close to  $V_{ref}$ . Therefore, the input currents of the VMM circuit, defined as the drain currents of the input floating-gate transistors, can be controlled by connecting a resistor to the drain of each input floating-gate transistor and then varying the input voltage  $V_{in}$  applied to the resistors from 0 V to  $V_{ref}$ . This configuration is shown in Fig. 7. Due to the nonideal characteristic of the negative feedback OTA, the realistic gain of the negative feedback OTA is finite. Therefore, the negative input voltage of the OTA does not stay close to  $V_{ref}$ , particularly when the input voltage  $V_{in}$  gets close to  $V_{ref}$  or, equivalently, the input current  $I_{in}$  gets close to zero. Hence, the operating range of the input current is chosen at  $0.2 - 1.0 \mu\text{A}$  in order to provide a linear relationship between the input voltage and the corresponding input current so that the received OFDM signals can be linearly mapped to the input currents of the VMM circuit, and to minimize the redundant power consumption while holding a reasonable current resolution and path delay. This range also guarantees that each transistor operates in the subthreshold region for weights less than or equal to one.

The drain current of each input floating-gate transistor determines the output voltage of the corresponding OTA, and this output voltage is broadcast to the source of all the connected floating-gate transistors in each row. When the drain voltage of each output floating-gate transistor is set to  $V_{ref}$ , this source voltage drives a drain current of each output floating-gate transistor, which is a scaled multiple of the corresponding input current. Then, the drain currents from the output floating-gate transistors are added up along each column to give a combined output current per column.

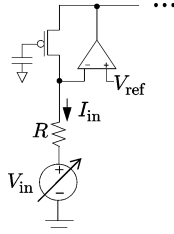


Fig. 7. Input voltage supplied to the drain of an input floating-gate transistor through a resistor. The setup enables the received OFDM signals to be linearly mapped to the input current of the VMM circuit.

Note that, in Fig. 6, each output floating-gate transistor has only one OTA connected to it, so the current level flowing through each output transistor is determined by the input current level and the programmed weights with respect to the connected input. In addition, the DFT coefficients involve a factor of  $1/N$ , as shown in (4), so all the converted nonnegative weights span within the range of  $0 \leq W \leq 1/N$ . These factors guarantee that each transistor of the VMM circuit operates in the subthreshold region, as far as the input current level stays less than the threshold current.

The required operations in the VMM given in (7) are scaling and summing operations. Since the information signals are conveyed in current levels, the summing operations do not require additional circuits. Therefore, the power consumption becomes less than that for the digital circuits where the information signals are conveyed in voltage levels and adding entries requires full adders. The scaling operations do not involve any complex multiplications, as all the signals and weights are real valued. Therefore, the power consumption in the scaling operations is also limited.

Now, we can take into account a butterfly operation in order to reduce the number of computations for a DFT operation. The butterfly operation basically decomposes a DFT matrix into a series of smaller matrices, and each output from the previous stage is handed over to the next stage. This can be viewed as a cascade of VMMs, where the VMM size in each stage gets smaller by a factor of the radix size. Therefore, it is clear that applying butterfly operations in the VMM circuit increases the path delay of the circuit. As the radix size gets lower, the number of stages increases, and consequently, the path delay increases. Moreover, the butterfly operation substitutes copying operations for summing and scaling operations. This is beneficial in voltage-mode circuits, where a summing operation requires a full adder, whereas a copying operation is trivial. However, in a current-mode circuit, a copying operation requires a current mirror or a current multiplier, whereas a summing operation is trivial. It is also claimed in [15] that an FFT design with a higher radix becomes less sensitive to the device mismatch. For these reasons, a full-radix DFT is more preferable for a current-mode analog circuit design.

### III. FPAA PROGRAMMING PROCEDURE

#### A. Programming Platform

To simplify the implementation of an analog DFT in the FPAA, we scale up the DFT matrix by a factor of  $N$ , with the understanding that it can be compensated for by scaling down the outputs of the analog DFT by a reciprocal of the scaling factor. This simplification makes the coefficients span within the  $0 \leq W \leq 1$  range, so each transistor will still operate in the subthreshold region. In particular, the  $16 \times 16$  matrix for a

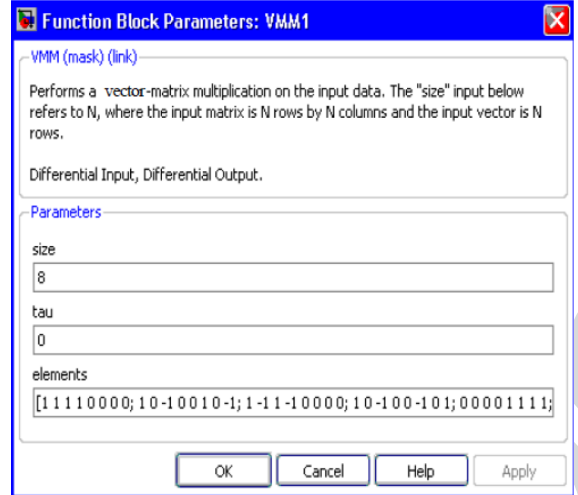
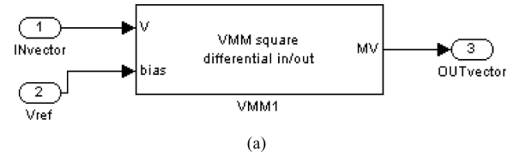


Fig. 8. (a) Custom VMM library block for the Simulink and (b) its block property that contains a field where real-valued  $8 \times 8$  differential weighting coefficients can be defined. The Matlab script is coded to load the VMM block with provided weights to generate a netlist file of the corresponding  $16 \times 16$  VMM analog circuit.

4-point DFT contains only ones and zeros with this simplification. This makes the amount of charge to be stored in each transistor relatively close to each other so as to increase the linearity between the input and output current levels for each transistor. The resulting matrix is then provided to a custom library block for a VMM in the Simulink shown in Fig. 8, and the Matlab script is coded to load the library block with a provided weighting matrix to generate a netlist of the VMM analog circuit [21], [22].

The generated netlist is taken by the RASPER tool that places and routes the available components in the FPAA chip [23]. The output file of the RASPER is a list of switch addresses and the targeting current value for each switch. This list is loaded by the Matlab script to be programmed into the FPAA chip. The RASP 2.9 FPAA chip contains the necessary circuitry for tunneling and injecting electrical charges of floating-gate transistors and the circuitry for current measurement. All the stored charges are tunneled before getting programmed, and an appropriate amount of charge value is injected to each floating-gate transistor while targeting on the corresponding current level determined by (1) and (2). The targeting current levels are represented with 10 bits of precision, of which 3 bits are assigned for the exponent and 7 bits for the significant [24]. Even though the information signals are conveyed in unquantized current levels, the accuracy in the programmed weights will impose a limit to the resolution of the analog DFT system.

In order to reduce the circuitry required for measurement and tunneling and injecting charges, the indirect programming method is used to charge the floating-gate transistors [25]. Fig. 9 shows the indirect programming structure for a floating-gate pMOSFET. The floating-gate transistor on the left is connected to the on-chip programming circuitry and is actively programmed. The one on the right is the floating-gate



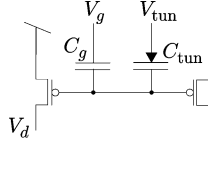


Fig. 9. Indirect programming structure of a pMOSFET. The left transistor is part of the on-chip programming circuitry and is actively programmed. The transistor on the right is the transistor that is used for the VMM circuit and is passively programmed.

transistor that is used for the VMM circuit and is passively programmed.

Due to the inherent mismatch between threshold voltages of different transistors, the indirectly programmed charges in the floating-gate transistors for the VMM circuit can be different from the directly programmed charges in the floating-gate transistors of the programming circuitry. This mismatch also occurs in between the programmed charges in the input and output floating-gate transistors of the VMM circuit. While this mismatch is inherent, we can circumvent this by adjusting the charge value in each input and output floating-gate transistor. We will discuss this process in Section III-B.

### B. Mismatch in FPAA Chips

When there is a mismatch in threshold voltages of different transistors, the preexponential factor of each MOSFET can be different from each other, and thus, the programmed weights can suffer from multiplicative distortion. However, the ratio of the output current to the input current is a function of the relative difference in charge values, as shown in (3), so the mismatch in weights can be compensated for by adjusting the charge values in the floating-gate transistors. This process can be accomplished by targeting on the desired *ratio* of the input and output currents, rather than targeting on the desired input and output currents themselves.

In the VMM circuit, each output current is the sum of the drain currents of the output floating-gate transistors in each column. Due to the different levels of nonlinearity in the  $I$ - $V$  characteristics of the input OTAs, there exist additive offsets between the input and output current levels. Therefore, targeting on the ratio at a single point will not suffice. Instead, we need two points of measurement so that a *slope* of the output current versus input current can be targeted. Thus, the FPAA programming procedure is conducted in the following two steps so as to minimize errors in the programmed weights.

#### 1) Coarse programming

- a) The fully turned-on switches and the input floating-gate transistors are first programmed with the desired charge values by targeting on the corresponding current levels.
- b) The floating-gate transistors inside the OTAs are also programmed with the charge values corresponding to the bias current.
- c) On the other hand, the output floating-gate transistors are programmed with lower charge values than what are desired by targeting on a half of the corresponding current levels.

#### 2) Fine programming

- a) Each output floating-gate transistor is then injected with a small amount of electrons iteratively to increase the stored charge.
- b) In each iteration, the input and output current values are measured at two different input voltages, and then, the slope of the input and output currents is obtained.
- c) The iteration stops when the slope of the input and output currents reaches the desired weight.

After the fine programming, the output currents of the VMM circuit still involve additive offsets, but these offsets do not vary as the input current levels change. Therefore, the sum of these offsets per output node is constant, and it can be easily calculated to be subtracted out from each output current.

## IV. FPAA MEASUREMENT AND EQUALIZATION

### A. FPAA Measurement

We now investigate the measured data of the OFDM receiver with an analog DFT demodulator. The transmitted symbols are randomly generated and mapped to 16 quadratic amplitude modulation (QAM) complex symbols with Gray coding. The generated symbols are then modulated by a size-4 inverse FFT. The guard interval is allocated for 1/4 of the FFT size, and the resulting complex samples are serialized, applied to a DAC, and upconverted to the carrier frequency. On the receiver side, after downconversion and removal of the guard interval, the received OFDM signals are split into real-valued differential pairs and converted to the input currents of the analog DFT within the current range of 0.2–1.0  $\mu$ A, as discussed in Section II-C. The converted 16 input currents are then fed into the  $16 \times 16$  VMM analog circuit implemented in the FPAA to demodulate the OFDM signals. The resulting 16 output currents of the FPAA are sampled, then reverted back to the voltage levels, and reassembled to yield four complex single-ended demodulated OFDM signals. These are then fed into a 16-QAM demapper to recover the transmitted symbols.

Fig. 10(a)–(b) shows the  $I$ - $Q$  plots of the demodulated symbols without injecting any channel noise. The color maps are used to illustrate the density of the occurrence. It can be observed in Fig. 10(a) that there is some dispersion in the demodulated symbols even without any channel noise, which results in a performance penalty as a price for the reduced power consumption in the analog DFT. The performance degradation arises for multiple reasons, including the following:

- 1) errors in the programmed weights due to the limit on the bit precision for targeting current values;
- 2) temperature sensitivity of the programmed weights;
- 3) nonlinear mapping between the input voltage and input current caused by the nonideal characteristic of the OTA;
- 4) thermal noise;
- 5) parasitic capacitance between routed paths.

Despite the dispersion, however, the 16-QAM demapper in the Matlab determined all the demodulated symbols correctly. This implies that the error rate will still converge to zero in a noisy channel as the signal-to-noise ratio (SNR) increases.

The processing speed of the analog DFT in the FPAA is limited by the settling time of the VMM circuit. Fig. 11 shows the step response of the  $16 \times 16$  VMM circuit for a size-4 analog DFT implemented in the FPAA (RASP 2.9) while the step input changes from 0.2 to 1.0  $\mu$ A. To measure the accurate settling time of the VMM circuit, an  $I$ - $V$  conversion circuit shown in Fig. 12 was included to the programming netlist so that the

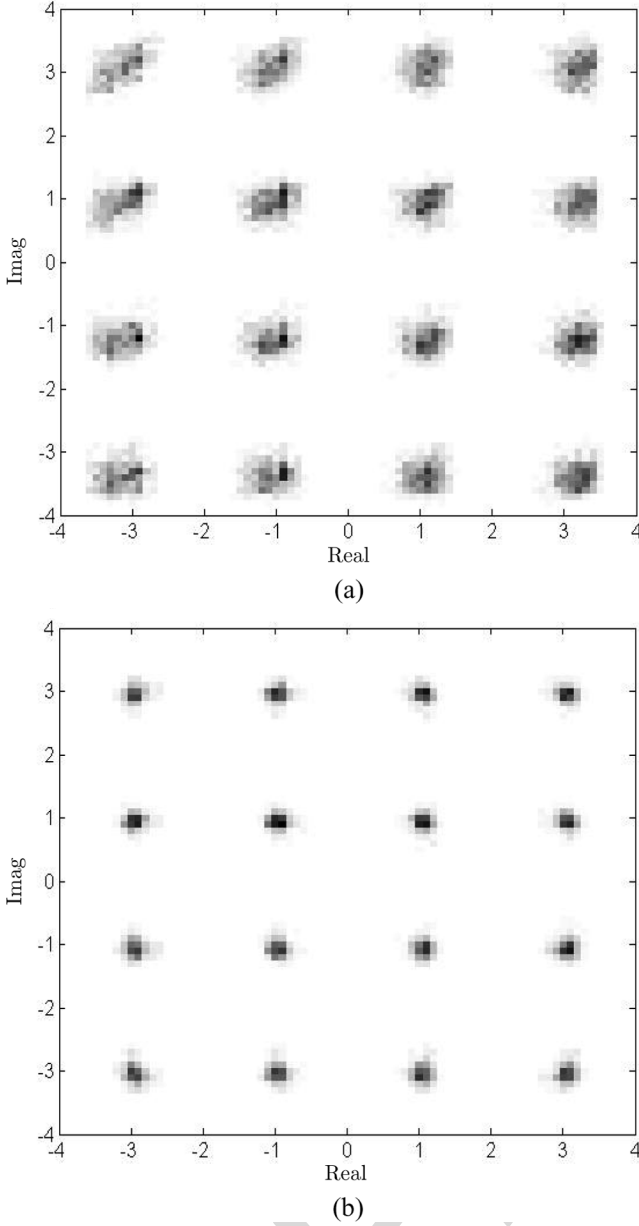


Fig. 10. Constellations of the demodulated symbols for 16 QAM without channel noise. (a) Before equalization. (b) After MMSE equalization. The gradation depicts the density of the occurrence in each pixel.

output voltage signals can be measured by a high-frequency oscilloscope. It can be observed in Fig. 11 that the settling time of the VMM circuit is around  $4 \mu\text{s}$ , which is close to a typical OFDM symbol duration for the IEEE 802.11 a/g with a 64-point FFT. Note that the measured settling time includes an additional delay caused by the auxiliary  $I$ - $V$  conversion circuit itself, so the actual settling time of the VMM circuit will be less than the measured value. For a size-4 digital DFT implemented in the FPGA with an 8-bit data width, the minimum data path delay is 49.6 ns for Xilinx Virtex2Pro (device: XC2VP30; package: ff896; speed: -7) and 101.6 ns for Xilinx Virtex (device: XCV50; package: fg256; speed: -5). However, when the number of subchannels increases and, thus, the OFDM symbol duration increases, the path delay in the digital DFT increases due to the larger number of complex multiplications and

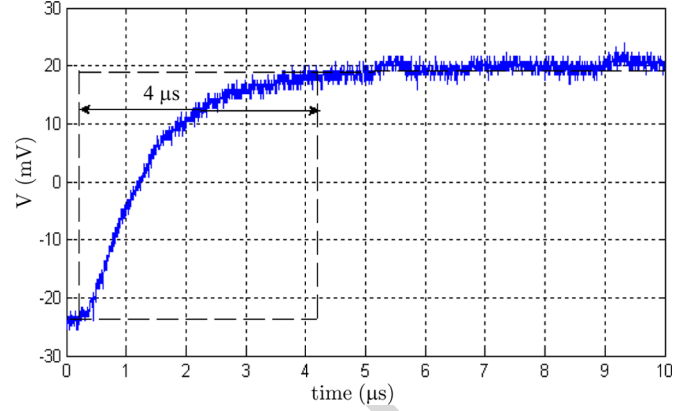


Fig. 11. Step response of the VMM circuit implemented in the RASP 2.9 FPAA.

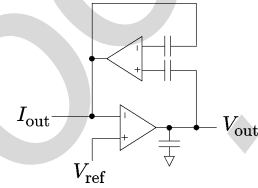


Fig. 12. On-chip  $I$ - $V$  conversion circuit. This is attached to the output node of the VMM circuit to eliminate delays caused by the measurement setup.

the hierarchical structure of the digital adders, whereas, in current-mode analog circuits, it stays almost the same because of the parallelized structure of the real-valued VMM operation. In [10], an analog filter implemented in a FPAA with a  $0.13\text{-}\mu\text{m}$  CMOS process is reported to achieve a frequency range up to 135 MHz, thus showing a potential increase in the processing speed of an analog DFT implemented in a FPAA with a smaller CMOS process.

The total power consumed in the analog 4-point DFT of the FPAA is measured to be 13.4 mW. This measured power is larger than the theoretically expected value for the  $16 \times 16$  VMM circuit that can be obtained by  $16.2 \cdot I_{\text{bias}} \cdot V_{\text{dd}} \approx 3 \text{ mW}$ , where the bias current of the OTA  $I_{\text{bias}}$  is  $40 \mu\text{A}$  and the bias voltage  $V_{\text{dd}}$  is 2.4 V. This difference may have been caused by the imperfect isolation of the VMM circuit from the rest of the chip and board. The digital 4-point DFT in the Virtex2Pro FPGA required 247 mW at the maximum speed and 105 mW at the same speed as the analog DFT. For the Virtex FPGA, it required 219 mW at the maximum speed and 34 mW at the same speed as the RASP 2.9 FPAA. Therefore, the power consumption required for the 4-point DFT operation is significantly reduced by 8.9 dB and 4.0 dB, respectively, for the same speed. Table I shows the comparisons of the measurements for the FPGA and FPAA. Aside from the power saving in the DFT block itself, implementing a DFT block in an analog circuit allows the ADC to be placed after the DFT block at the receiver, thus effectively reducing the overall power consumption by relieving the speed and bit-precision requirements of the ADC block. The application-specific integrated circuit (ASIC) implementation of an analog DFT in [15] was reported to consume lower power than the FPAA implementation, where the full-radix analog 256-point DFT implemented in an ASIC with a 180-nm CMOS process was claimed to consume 1.6 mW. Despite the higher power consumption compared to the ASIC implementation, the benefit of the FPAA implementation

TABLE I  
POWER AND DELAY COMPARISONS FOR FPGA AND FPAA

Chipset	Virtex2Pro FPGA	Virtex FPGA	RASP2.9 FPAA
Power consumption @ Processing delay	247 mW @ 49.6 ns	219 mW @ 101.6 ns	13.4 mW @ 4 $\mu$ s
	105 mW @ 4 $\mu$ s	34 mW @ 4 $\mu$ s	

The upper power consumption values for the FPGAs are measured when operating at its own fastest speed. The lower values are measured when operating at the same speed as the RASP 2.9 FPAA.

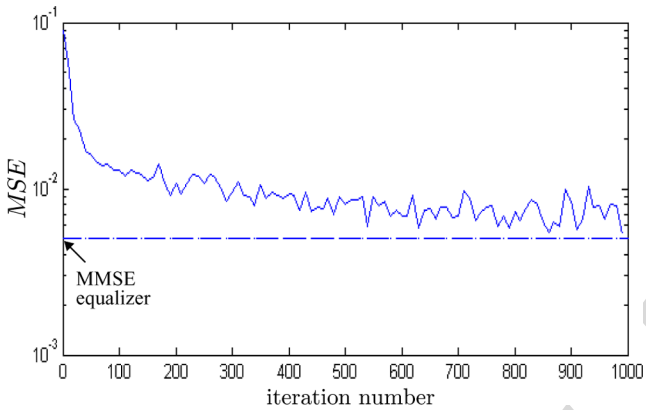


Fig. 13. MSE trace of a  $16 \times 16$  LMS equalizer.

with floating-gate transistors is its ability to tune the DFT coefficients caused by the mismatch in transistors without changing the circuit structure.

### B. Equalization of FPAA Outputs

Any residual errors in the programmed weights of the DFT will prevent it from perfectly separating the symbols for the different subcarriers, leading to a form of intersymbol interference (ISI). These errors can be mitigated by applying an equalizer to each output of the analog DFT block. The equalizer coefficients can be obtained by injecting training symbols. As the errors in the programmed weights are independent with each other, each output of the equalizer has 16 taps, and the 16 parallel outputs are equalized separately. Therefore, the  $k$ th output  $z_k$  of the equalizer (for  $k = 1, \dots, 16$ ) is given by the inner product  $z_k = \mathbf{c}_k^T \mathbf{X}$  of the  $k$ th equalizer coefficient vector  $\mathbf{c}_k = [c_{k,1} \dots c_{k,16}]^T$  and the output vector  $\mathbf{X} = [X_1 \dots X_{16}]^T$  of the DFT block.

The minimum mean-square-error (MMSE) coefficients that minimize  $MSE = E((c_k^T \mathbf{X} - a_k)^2)$ , where  $a_k$  represents the training symbols, are [26]

$$\mathbf{c}_k = \mathbf{R}^{-1} \mathbf{p}_k \quad (8)$$

where  $\mathbf{R} = \mathbf{E}(\mathbf{X}\mathbf{X}^T)$  and  $\mathbf{p}_k = E(a_k \mathbf{X})$ . Therefore, each of the 16 parallel outputs from the DFT block can be equalized with the 16 tap coefficients in (8). Fig. 10(b) depicts the equalized symbols while using a  $16 \times 16$  MMSE equalizer. It can be observed that demodulated symbols are located tighter in the  $I$ - $Q$  map by applying equalization to the outputs of the DFT.

For a least mean square (LMS) equalizer, the equalizer coefficients are updated along the steepest decent direction using [26]

$$\mathbf{c}_k[n+1] = \mathbf{c}_k[n] - \mu(z_k - a_k) \mathbf{X}[n] \quad (9)$$

where  $\mu$  is the step size. Fig. 13 exhibits the trace of the  $MSE$  for an LMS equalizer when the step size is  $\mu = 5 \times 10^{-4}$  and the initial coefficient vector for each output is set to each row of the  $16 \times 16$  identity matrix. It can be seen from Fig. 13 that the convergence occurs within 500 iterations with these parameters. This iteration can be also applied to an adaptive programming scheme by charging floating-gate transistors with an updated amount of injection based on the measured current level.

### C. BER Performance in AWGN Channels

We now consider the case when the modulated OFDM signals are passed through a noisy channel to see how the channel noise affects the performance of the analog DFT demodulator. Note that there is a certain range of an input voltage that is allowed to be fed into the FPAA chip, effectively  $0 - V_{\text{ref}}$ , but due to the nature of the high peak-to-average power ratio in OFDM signals, some received OFDM signals from a noisy channel can be converted to the input voltage levels beyond the allowed range. To avoid this case, the converted input voltage levels that are lower than 0 V are set to 0 V. This results in clipping distortions when the input current value is high, but it happens at rare peak voltages of the OFDM signals.

Fig. 14 demonstrates the measured bit error rate (BER) versus  $E_b/N_0$  (SNR per bit) for a 16-QAM OFDM demodulator implemented in an analog DFT, assuming additive white Gaussian noise (AWGN). The performance with the MMSE and LMS equalization is also shown. These results are compared to the theoretical BER for 16 QAM with Gray mapping [27]

$$BER \approx \frac{3}{4} Q \left( \sqrt{\frac{4}{5} E_b/N_0} \right). \quad (10)$$

The measurement was iterated for 2500 cycles, so the sample size is 10 000 symbols or 40 000 bits for each  $E_b/N_0$  value. As can be observed from the plots in Fig. 14, the demodulated OFDM symbols with an analog DFT suffer a performance penalty of 2 dB compared to the theoretical BER curve. This is because the remaining errors in the programmed weights produce an ISI across the parallel outputs of the analog DFT block. However, applying equalization to the outputs of the FPAA significantly relieves the penalty by mitigating the errors in the programmed weights, and the  $E_b/N_0$  gap between the equalized outputs and the theoretical values becomes less than 1 dB.

For a digital DFT demodulator with an 8-bit data width, the measured BER for the same sample size converged to the theoretical values of a 16 QAM regardless of the existence of the IDFT/DFT blocks in between. Therefore, there is a tradeoff between performance and power consumption, but the power saving of the analog circuit outweighs the performance penalty without equalization.

## V. CONCLUSION

We have proposed a low-power analog DFT implemented on an FPAA as an alternative to a conventional OFDM demodulator based on a digital DFT. The analog DFT is implemented



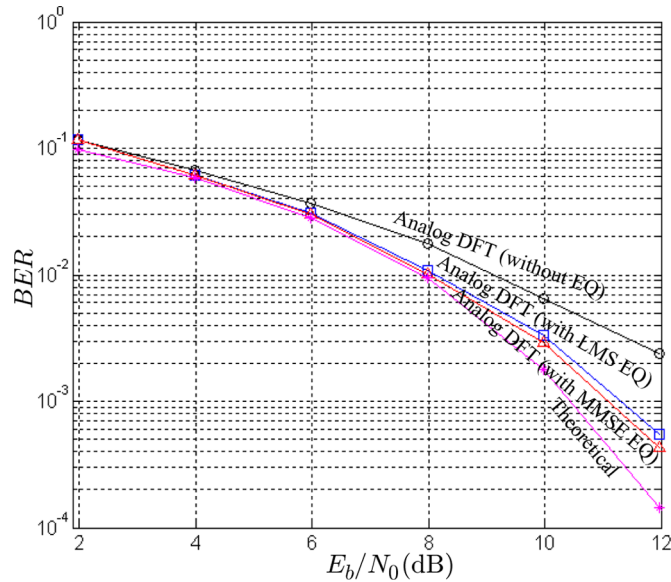


Fig. 14. Performance of 16-QAM OFDM demodulator using an analog DFT, with and without equalization. When compared to theory, the penalty after equalization is less than 1 dB.

as a VMM using floating-gate transistors. The floating-gate transistors of the FPAA are used not only to configure the VMM circuit connections as fully turned-on switches but also to store the DFT coefficients by locking in an appropriate amount of charge in each floating-gate capacitor. The analog DFT in the FPAA consumed 8.9 dB less power than a digital implementation using a Virtex2Pro FPGA, and it consumed 4.0 dB less power than a digital implementation using a Virtex FPGA. This power reduction, although significant, does not reflect the additional power savings that come from the fact that an analog DFT reduces the speed and precision requirements of the ADCs. The price paid for this power reduction was a 2-dB performance degradation. We have also shown that this performance loss can be mitigated by exploiting an equalizer technique as a top-down approach to tackle the device mismatch problem. Furthermore, the unquantized output signals from the analog DFT block enable the real soft inputs to the subsequent decoding block at the OFDM receiver.

#### ACKNOWLEDGMENT

The authors would like to thank S. Brink for his editorial support on the reconfigurable analog signal processor 2.9 field-programmable analog array chip.

#### REFERENCES

- [1] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nW/MHz,  $128 \times 32$  current-mode programmable analog vector-matrix multiplier with over 2 decades of linearity," in *Proc. IEEE Custom Integr. Circuits Conf.*, Oct. 2004, pp. 651–654.
- [2] T. S. Hall, C. M. Twigg, J. D. Gray, P. Hasler, and D. V. Anderson, "Large-scale field-programmable analog arrays for analog signal processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 11, pp. 2298–2307, Nov. 2005.
- [3] E. K. F. Lee and P. G. Gulak, "A CMOS field-programmable analog array," *IEEE J. Solid-State Circuits*, vol. 26, no. 12, pp. 1860–1867, Dec. 1991.

- [4] H. Kutuk and S.-M. Kang, "A field-programmable analog array (FPAA) using switched-capacitor techniques," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1996, pp. 41–44.
- [5] E. K. F. Lee and W. L. Hui, "A novel switched-capacitor based field-programmable analog array architecture," *Analog Integr. Circuits Signal Process.*, vol. 17, no. 1/2, pp. 35–50, Sep. 1998.
- [6] E. K. F. Lee and P. G. Gulak, "A transconductor-based field-programmable analog array," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 1995, pp. 198–199.
- [7] B. Pankiewicz, M. Wojcikowski, S. Szczepanski, and Y. Sun, "A field programmable analog array for CMOS continuous-time OTA-C filter applications," *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 125–136, Feb. 2002.
- [8] A. Basu, C. M. Twigg, S. Brink, P. Hasler, C. Petre, S. Ramakrishnan, S. Koziol, and C. Schlottmann, "RASP 2.8: A new generation of floating-gate based field programmable analog array," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2008, pp. 213–216.
- [9] J. Becker, F. Henrici, S. Trendelenburg, M. Ortmanns, and Y. Manoli, "A field-programmable analog array of 55 digitally tunable OTAs in a hexagonal lattice," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2759–2768, Dec. 2008.
- [10] F. Henrici, J. Becker, S. Trendelenburg, D. DeDorigo, M. Ortmanns, and Y. Manoli, "A field programmable analog array using floating gates for high resolution tuning," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 265–268.
- [11] A. Stoica, D. Keymeulen, M. Mojarradi, R. Zebulum, and T. Daud, "Progress in the development of field programmable analog arrays for space applications," in *Proc. IEEE Aerosp. Conf.*, Mar. 2008, pp. 1–9.
- [12] D. Keymeulen, A. Stoica, R. Zebulum, S. Katkooi, P. Fernando, H. Sankaran, M. Mojarradi, and T. Daud, "Self-reconfigurable analog array integrated circuit architecture for space applications," in *Proc. NASA/ESA Conf. Adapt. Hardw. Syst.*, Jun. 2008, pp. 83–90.
- [13] M. Lehne and S. Raman, "An analog/mixed-signal FFT processor for wideband OFDM systems," in *Proc. IEEE Sarnoff Symp.*, Mar. 2006, pp. 1–4.
- [14] M. Lehne and S. Raman, "A prototype analog/mixed-signal fast Fourier transform processor IC for OFDM receivers," in *Proc. IEEE Radio Wireless Symp.*, Jan. 2008, pp. 803–806.
- [15] N. Sadeghi, V. C. Gaudet, and C. Schlegel, "Analog DFT processors for OFDM receivers: Circuit mismatch and system performance analysis," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 2123–2131, Sep. 2009.
- [16] E. Afshari, H. S. Bhat, and A. Hajimiri, "Ultrafast analog Fourier transform using 2-D LC lattice," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 8, pp. 2332–2343, Sep. 2008.
- [17] "AN231E04 Datasheet Rev. 1.0, Dynamically Reconfigurable dpASP," Analogm, Oak Park, CA, 2007.
- [18] "ispPAC81 Datasheet, In-System Programmable Analog Circuit," Lattice Semicond., Hillsboro, OR, 2001.
- [19] B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, "Analog-to-digital converters," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 69–77, Nov. 2005.
- [20] M. Kucic, A. Low, P. Hasler, and J. Neff, "A programmable continuous-time floating-gate Fourier processor," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 1, pp. 90–99, Jan. 2001.
- [21] C. Petre, C. Schlottmann, and P. Hasler, "Automated conversion of Simulink designs to analog hardware on an FPAA," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 500–503.
- [22] C. Schlottmann, C. Petre, and P. Hasler, "Vector matrix multiplier on field programmable analog array," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2010, pp. 1522–1525.
- [23] F. Baskaya, S. Reddy, S. K. Lim, and D. V. Anderson, "Placement for large-scale floating-gate field-programmable analog arrays," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 8, pp. 906–910, Aug. 2006.
- [24] A. Basu and P. E. Hasler, "A fully integrated architecture for fast and accurate programming of floating gates over six decades of current," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2010, to be published.
- [25] D. W. Graham, E. Farquhar, B. Degnan, C. Gordon, and P. Hasler, "Indirect programming of floating-gate transistors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 5, pp. 951–963, May 2007.
- [26] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 2001, pp. 203–207, 231–238.
- [27] J. Proakis, *Digital Communications*. New York: McGraw-Hill, 2000, pp. 276–280.



**Sangwook Suh** (S'06) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1998, and the M.S. degree in electrical engineering from the Polytechnic Institute of New York University, New York, in 2005. He is currently working toward the Ph.D. degree in electrical and computer engineering at the Georgia Institute of Technology, Atlanta.

In lieu of military service, he was with Sein Electronics Company, Ltd., from 1998 to 2000, and with Corecess Inc. from 2000 to 2002. In 2002, he was a Software Engineering Intern with Microsoft Corporation, Seoul, Korea. From 2005 to 2006, he was a Software Engineer with Samsung Electronics Company, Ltd., Suwon, Korea. His research interests include adaptive equalization, low-power signal processing, and soft-input analog decoders.



**Arindam Basu** (S'06–M'10) received the B.Tech and M.Tech degrees in electronics and electrical communication engineering from the Indian Institute of Technology Kharagpur (IIT Kharagpur), Kharagpur, India, in 2005, and the M.S. degree in mathematics and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in 2009 and 2010, respectively.

He is currently an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore. His research interests include nonlinear dynamics and chaos, modeling neural dynamics, low-power analog IC design, and programmable circuits and devices.

Dr. Basu was a recipient of the Jagadis Bose National Science Talent Search Award in 2000 and the Prime Minister of India Gold Medal from IIT Kharagpur in 2005. He was a recipient of the Best Student Paper Award at the IEEE Ultrasonics Symposium in 2006 and the Best Live Demonstration Award at the IEEE International Symposium on Circuits and Systems (ISCAS) in 2010 and was also a Best Student Paper Award finalist at ISCAS in 2008.



**Craig Schlottmann** (S'07) received the B.S. degree in electrical engineering from the University of Florida, Gainesville, in 2007, and the M.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in 2009. He is currently working toward the Ph.D. degree in electrical engineering at the Georgia Institute of Technology.

His research interests include low-power analog signal processing, multiple-input translinear elements, floating-gate transistor circuits, and analog IC design.



**Paul E. Hasler** (S'87–M'01–SM'03) 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.

In 2002, he cofounded GTronix, Inc., which was acquired by National Semiconductor in 2010. He is currently an Associate Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His current research interests include low-power electronics, mixed-signal system ICs, floating-gate MOS transistors, adaptive information processing systems, smart interfaces for sensors, cooperative analog–digital signal processing, device physics related to submicrometer devices or floating-gate devices, and analog very large scale integration models of on-chip learning and sensory processing in neurobiology.

Dr. Hasler was a recipient of the National Science Foundation CAREER Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He was also a recipient of the Paul Rappaport Best Paper Award from the IEEE Electron Devices Society in 1997, the Best Paper Award at the Multiconference on Systemics, Cybernetics, and Informatics in 2001, the Best Sensor Track Paper at the IEEE International Symposium on Circuits and Systems in 2005, the Best Student Paper Award at the IEEE Custom Integrated Circuits Conference in 2006, the Best Student Paper Award at the IEEE Ultrasound Symposium in 2006, and the Best Demonstration Paper Award at the IEEE International Symposium on Circuits and Systems in 2010.



**John R. Barry** (S'85–M'87–SM'04) received the B.S. degree in electrical engineering from the State University of New York, Buffalo, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, Berkeley, in 1987 and 1992, respectively.

Since 1992, he has been with the Georgia Institute of Technology, Atlanta, where he is currently a Professor with the School of Electrical and Computer Engineering. His research interests include wireless communications, equalization, and multiuser communications. He is a coauthor with E. A. Lee and D. G. Messerschmitt of *Digital Communication* (Springer, 2004) and the author of *Wireless Infrared Communications* (Kluwer, 1994).

# Low-Power Discrete Fourier Transform for OFDM: A Programmable Analog Approach

Sangwook Suh, *Student Member, IEEE*, Arindam Basu, *Member, IEEE*, Craig Schlottmann, *Student Member, IEEE*, Paul E. Hasler, *Senior Member, IEEE*, and John R. Barry, *Senior Member, IEEE*

**Abstract**—The modulation and demodulation blocks in an orthogonal frequency-division multiplexing (OFDM) system are typically implemented digitally using a fast Fourier transform circuit. We propose an analog implementation of an OFDM demodulator as a means for reducing power consumption. The proposed receiver implements the discrete Fourier transform (DFT) as a vector–matrix multiplier using floating-gate transistors on a field-programmable analog array (FPAA). The DFT coefficients can be tuned to counteract an inherent device mismatch by adjusting the amount of electrical charge stored in the floating-gate transistors. When compared to a digital field-programmable gate array implementation, the analog FPAA implementation of the DFT reduces power consumption at the cost of a slight performance degradation. Considering the errors in the DFT coefficients as intersymbol interference, the performance degradation can be further mitigated by employing a least mean-square or minimum mean-square-error equalizer.

**Index Terms**—Discrete Fourier transform (DFT), fast Fourier transform (FFT), field-programmable analog array (FPAA), floating-gate transistor, intersymbol interference (ISI), least mean square (LMS), minimum mean square error (MMSE), orthogonal frequency-division multiplexing (OFDM), vector–matrix multiplier (VMM).

## I. INTRODUCTION

ORTHOGONAL frequency-division multiplexing (OFDM) is widely used in numerous wireless communication systems not only because of its spectral efficiency and robustness to multipath fading but also because of its ease of implementation; OFDM modulators and demodulators can be implemented using simple fast Fourier transform (FFT) blocks, typically in digital circuits. However, for mobile devices with limited battery power, replacing these digital circuits with low-power analog circuits can significantly improve the power efficiency of the devices [1], [2]. The cost paid for this reduced power is the long development cycle and lack of flexibility that typifies analog circuit design.

So as to retain the rapid-prototyping capability and flexibility of a field-programmable gate array (FPGA) but with reduced power consumption, an analog counterpart of the FPGA, namely

a field-programmable analog array (FPAA), was proposed in [3], followed by several different FPAA realizations using a switched capacitor [4], [5], a transistor [6], or an operational transconductance amplifier (OTA) with a capacitor [7]. The early FPAAs, however, contained only a few computational elements, and their applications were restricted to analog filters, until floating-gate transistors were used as switches of the FPAA to enable large-scale analog circuit design [2], [8]. Recently, a hexagonal arrangement of computational analog blocks (CABs) has been reported in [9] and [10] to reduce the size and path delay of the FPAA chip. Two decades since its advent, the FPAA is finding viability in space applications as well by imposing self-reconfigurable features [11], [12].

There have been several dedicated nonprogrammable analog implementations of FFT and discrete Fourier transform (DFT) circuits. A voltage-mode analog FFT block was reported in [13] and [14] that uses analog multipliers and dedicated input signals representing the FFT coefficients and analog adders for summing voltage signals. An FFT based on analog current mirrors was proposed in [15], where the FFT coefficients are not reconfigurable but are determined by the  $W/L$  ratio of the output transistor of each mirror. More recently, a numerical simulation for approximating a fast DFT operation with a 2-D lattice of inductors and capacitors has been introduced in [16].

To overcome the drawbacks of previous works, we present in this paper a current-mode analog DFT block implemented as a vector–matrix multiplier (VMM) using the reconfigurable analog signal processor (RASP) 2.9 FPAA chip [8]. Floating-gate transistors inside the FPAA chip are used as partially connected switches to store the DFT coefficients by locking in an appropriate amount of electrical charge in each floating-gate capacitor. Therefore, dedicated input signals representing the DFT coefficients are not required. Furthermore, these coefficients are reconfigurable without changing the circuit structure. The VMM structure using floating-gate transistors as programmable switches also enables tuning the DFT coefficients to compensate for the inherent mismatch between different transistors. Another benefit of our current-mode design over a voltage-mode circuit is the ease with which signals can be added, which is particularly beneficial in systems having multiple inputs and multiple outputs. The RASP 2.9 FPAA chip contains more computational elements than previous FPAA chips, including the commercial products in [17] and [18]. The large number of computational elements and the configurable floating-gate switches make the RASP 2.9 FPAA chip viable in a wide range of applications. Such versatility is an important figure of merit for any programmable circuit.

In Section II, we present the system description and analysis. In Section III, we summarize the FPAA programming procedure. In Section IV, we describe the FPAA measurement and equalization procedure. In Section V, we present our conclusions.

Manuscript received April 22, 2010; revised July 03, 2010; accepted July 30, 2010. This paper was recommended by Associate Editor M. Delgado-Restituto.

S. Suh, C. Schlottmann, P. E. Hasler, and J. R. Barry are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: swsuh@ece.gatech.edu; cschlott@gatech.edu; phasler@ece.gatech.edu; john.barry@ece.gatech.edu).

A. Basu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: arindam.basu@ntu.edu.sg).

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

Digital Object Identifier 10.1109/TCSI.2010.2071950

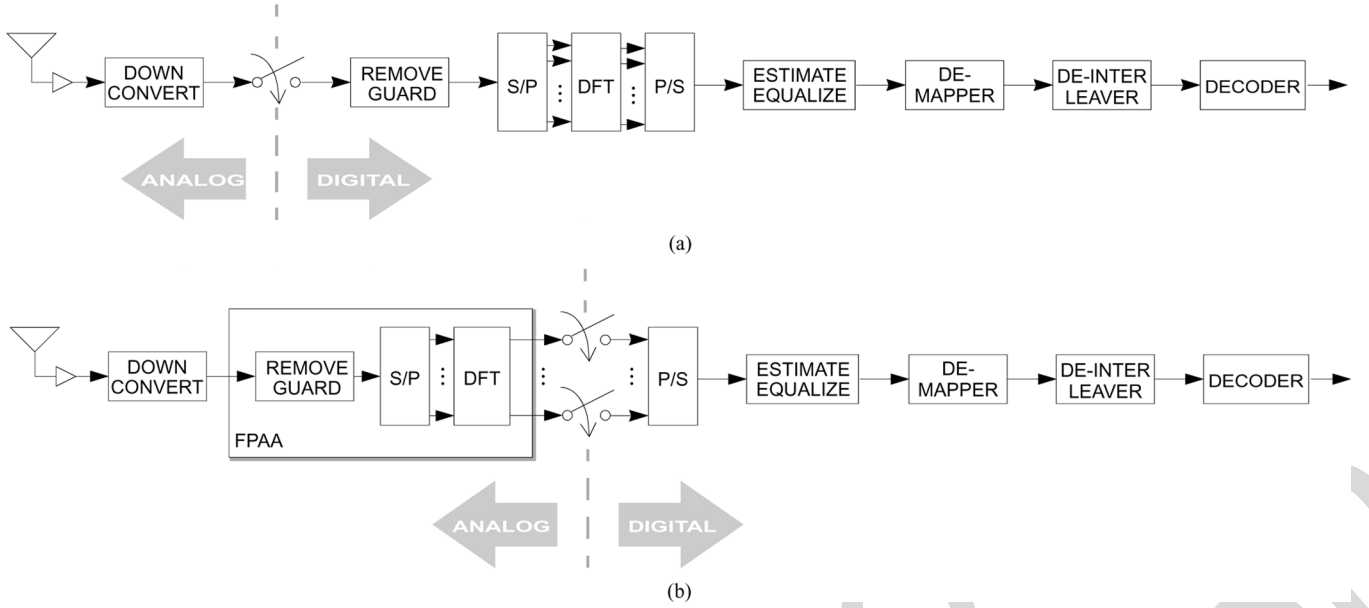


Fig. 1. OFDM receiver. (a) Conventional implementation in which sampling occurs before a digital DFT. (b) Proposed implementation in which sampling occurs after an analog DFT.

## II. SYSTEM DESCRIPTION AND ANALYSIS

### A. Motivation

In Fig. 1(a), we illustrate a simplified block diagram of a conventional OFDM receiver, such as for an 802.11 a/g system, where the received signal is sampled immediately after down-conversion. The OFDM demodulation is performed digitally using a DFT. As an alternative, we propose an analog implementation, as shown in Fig. 1(b), where the downconverter output is fed directly to the FPAA, with no sampling. The DFT functionality is implemented in analog using the FPAA. The  $N$  outputs of the FPAA—one for each subcarrier—are each sampled separately.

Besides the reduced power consumption with an analog implementation of the DFT, an important benefit of the proposed receiver structure in Fig. 1(b) is that it greatly relieves the speed and precision burdens of the analog-to-digital converter (ADC). In particular, the ADC of the conventional receiver shown in Fig. 1(a) would need to sample at a rate equal to the full signal bandwidth, and its precision would need to be high (on the order of 10 bits or more) to accommodate the wide dynamic range and Gaussian-like distribution of OFDM signals. In contrast, the proposed receiver in Fig. 1(b) has not one but  $N$  ADCs, one for each subcarrier, with a sampling rate slower by a factor of  $N$ . Moreover, each DFT output is a finite alphabet signal that can be sampled with significantly less bit precision [14]. Compared to the ADC in Fig. 1(a), each ADC in Fig. 1(b) requires a sampling rate that is lower by a factor of  $N$ , and the number of bits of precision is smaller by a factor of three or more, depending on the modulation alphabet size. Both effects yield a reduction in ADC power consumption, although the exact amount depends on the type of the ADC structure; the ADC power consumption is between a linear and quadratic function of the sampling rate [19].

Thus, the proposed receiver in Fig. 1(b) is beneficial not only because of the reduced power consumption with an analog implementation of the DFT but also because of the additional power savings resulting from the lower speed and bit-precision

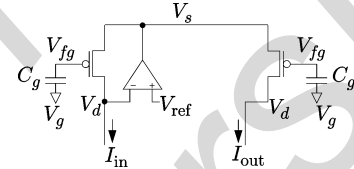


Fig. 2. Current multiplier circuit composed of floating-gate transistors and an OTA. The output current is a scaled multiple of the input current.

requirements of the subsequent ADC. Although our focus here is on the receiver side, we briefly point out that these same advantages are also valid at the transmitter side, where the outputs from the symbol mapper are modulated with an inverse DFT (IDFT) block, which has the same structure as the DFT block with different coefficients. Besides the power savings with an analog IDFT implementation, shifting the IDFT block after the digital-to-analog converter (DAC) enables an OFDM transmitter to replace a full-speed high-precision DAC by  $N$  separate DACs, each operating at a  $1/N$  times lower clock with lower bit precision.

### B. Floating-Gate Transistors and the RASP 2.9 FPAA

Floating-gate transistors can store a nonvolatile electrical charge, so arrays of floating-gate transistors can be programmed as a signal processing block for specific functionality. One application is to use them as a VMM circuit. Fig. 2 shows a current multiplier circuit—the basic element of a VMM circuit—composed of floating-gate transistors and an OTA. Two pMOSFETs are connected at the source, and the gate of each transistor is connected to a capacitor  $C_g$  to be electrically isolated so as to form a *floating gate*. The source voltage  $V_s$  is common for both transistors, and  $V_{fg}$  is the voltage potential at the floating gate.  $V_d$  is the drain voltage of the transistor. The voltages on the other side of the capacitors connected to the gates are set to be the same at the fixed potential  $V_g$ .

The input current  $I_{in}$  and output current  $I_{out}$  are defined as drain currents of the transistors operating in the subthreshold region. Neglecting the Early effects, the input and output currents of the pMOSFET are given by [20]

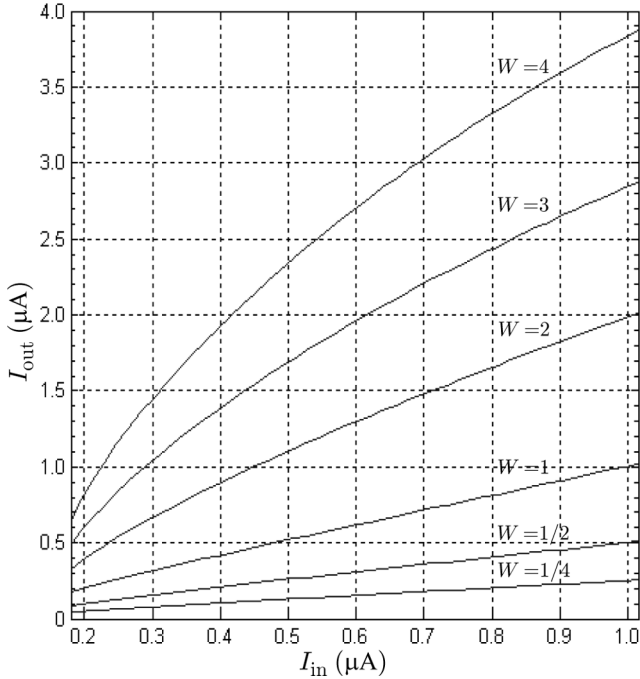


Fig. 3. Output currents with the programmed weights of 1/4–4. The input current is swept from 0.2 to 1.0  $\mu\text{A}$ . The straight curves for  $W \leq 1$  show the programmed circuit can be used as a current multiplier in the current range. As  $W$  increases, the output current shows transition toward the strong-inversion region.

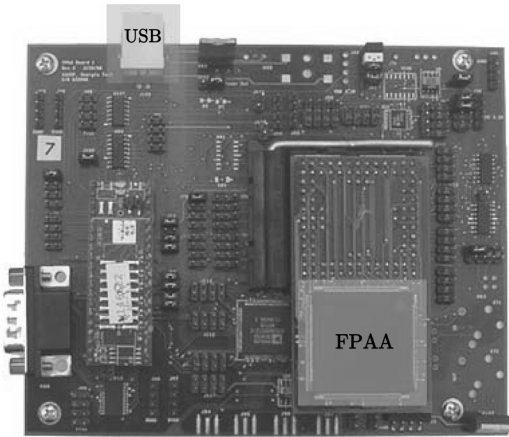


Fig. 4. RASP 2.9 FPAA chip mounted on the board. The chip is fabricated with the 0.35- $\mu\text{m}$  CMOS process. The board has 56 I/O pins for setting drain voltages of floating-gate transistors and measuring output currents. It is connected to a PC through a USB interface for controlling the FPAA chip programming. The sizes of the chip and the board are 5 mm  $\times$  5 mm and 114 mm  $\times$  140 mm, respectively.

$$I_{\text{in}} = I_1 \exp\left(\frac{V_s - \kappa_{\text{eff}} V_g}{U_T}\right) \exp\left(\frac{-\kappa Q_1}{C_t U_T}\right) \quad (1)$$

$$I_{\text{out}} = I_2 \exp\left(\frac{V_s - \kappa_{\text{eff}} V_g}{U_T}\right) \exp\left(\frac{-\kappa Q_2}{C_t U_T}\right) \quad (2)$$

where  $U_T = kT/q$  (where  $T$  is the temperature,  $k$  is the Boltzmann's constant, and  $q$  is the elementary charge) and  $C_t$  is the total capacitance at each gate, including the floating-gate capacitor  $C_g$  and the internal capacitance of the MOSFET.  $Q_1$  and  $Q_2$

are the electrical charges stored in the input and output floating-gate transistors, respectively.  $\kappa$  is the back-gate coefficient, and  $\kappa_{\text{eff}} = \kappa C_g / C_t$ . The parameters  $I_1$  and  $I_2$  are the preexponential factors of the MOSFETs that can be defined as a drain current flowing in each transistor when  $V_s = \kappa_{\text{eff}} V_{\text{fg}}$ . When there is a negligible mismatch between the threshold voltage<sup>1</sup> of the input and output transistors so that  $I_1$  and  $I_2$  are approximately identical, the ratio of the output current to the input current reduces to

$$W = \frac{I_{\text{out}}}{I_{\text{in}}} = \exp\left(\frac{-\kappa(Q_2 - Q_1)}{C_t U_T}\right). \quad (3)$$

Therefore, the weighting coefficient  $W$  is determined by the difference in charge values between the input and output floating-gate transistors, provided that both transistors operate in the sub-threshold region. One can observe in (3) that  $W$  is also a function of temperature. Note that, from (3), only positive weights can be realized. The weight zero can be realized by not connecting the input and output floating-gate transistors.

In practice, there exists an inherent mismatch between the threshold voltage of different transistors which leads to a mismatch in the preexponential factors  $I_1$  and  $I_2$ , and this, in turn, leads to multiplicative distortion in the programmed weights. However, this mismatch can be compensated for by adjusting the charge values  $Q_1$  and  $Q_2$  while they are programmed into the FPAA chip. Although a floating-gate current mirror can be similarly tuned over a narrow range to compensate for the device mismatch, its weight is fixed by the  $W/L$  ratio of the transistors and cannot be changed widely without changing the circuit structure.

Fig. 3 shows a plot of the output currents versus input current for a set of programmed weights between 1/4 and 4. The mismatch effect has been canceled through the weight programming procedure that will be discussed in Section III. The input current is swept from 0.2 to 1.0  $\mu\text{A}$ . For  $W \leq 1$ , the linear relation between  $I_{\text{in}}$  and  $I_{\text{out}}$  shows that the programmed circuit can be used as a current multiplier for the range of currents shown. It can be observed that the output current shows a transition to the strong-inversion region as  $W$  increases.

The basic current multiplier circuit in Fig. 2 can be expanded to a multiple-input multiple-output structure to construct a larger size VMM circuit in the FPAA. The RASP 2.9 FPAA [8] consists of 133 744 floating-gate transistors and 84 CABs, each of which contains three OTAs, three capacitors, a transmission gate, and a voltage buffer. The floating-gate transistors can be programmed as a VMM circuit by storing an appropriate amount of charge in each floating-gate capacitor. Fig. 4 depicts the RASP 2.9 FPAA chip mounted on the board. The chip is fabricated with a 0.35- $\mu\text{m}$  CMOS process. The size of the transistors is  $W = 1.8 \mu\text{m}$  and  $L = 0.6 \mu\text{m}$ . Fig. 5 depicts the wide-output-range OTA inside the CABs, which is used to supply the input and output currents of the VMM circuit. The board has 56 I/O pins that can be used to set drain voltages of the floating-gate transistors and to measure output currents. The programming process for the FPAA chip is controlled through a universal serial bus (USB) interface equipped on the board.

### C. VMM Representation of an Analog DFT

The key to an OFDM receiver is to compute the DFT of a set of complex samples  $\{x(n)\}$ , defined by

<sup>1</sup>The threshold voltage is the gate voltage at which channel formation occurs between the oxide and the body of a transistor.



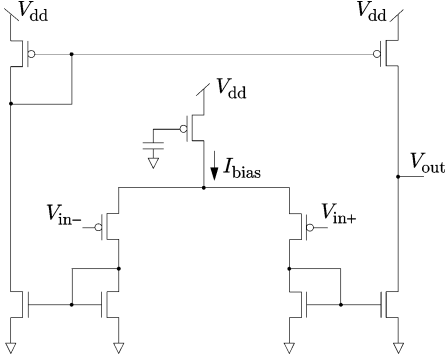


Fig. 5. Wide-output-range OTA inside the CABs of the RASP 2.9 FPAA chip.  $V_{dd}$  is the bias voltage, and  $I_{bias}$  is the bias current. The floating gate transistor in the middle is programmed with the charge value corresponding to the bias current. The targeting bias current is determined at the amount sufficient to provide the input and output currents of the connected floating-gate transistors.

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad (4)$$

where  $N$  is the number of subcarriers and  $k$  is an integer ranging from 0 to  $N - 1$ . By splitting each complex number into real and imaginary parts, we can rewrite (4) as

$$\begin{aligned} \text{Re}[X(k)] = \frac{1}{N} \sum_{n=0}^{N-1} & \left[ \text{Re}[x(n)] \cos\left(\frac{2\pi kn}{N}\right) \right. \\ & \left. + \text{Im}[x(n)] \sin\left(\frac{2\pi kn}{N}\right) \right] \quad (5) \end{aligned}$$

$$\begin{aligned} \text{Im}[X(k)] = \frac{1}{N} \sum_{n=0}^{N-1} & \left[ -\text{Re}[x(n)] \sin\left(\frac{2\pi kn}{N}\right) \right. \\ & \left. + \text{Im}[x(n)] \cos\left(\frac{2\pi kn}{N}\right) \right] \quad (6) \end{aligned}$$

which is equivalent to a VMM

$$\mathbf{X} = \mathbf{H}\mathbf{x} \quad (7)$$

where  $\mathbf{X}$  is a  $2N \times 1$  vector consisting of the real and imaginary components of  $\{X(k)\}$ ,  $\mathbf{x}$  is a  $2N \times 1$  vector consisting of the real and imaginary components of  $\{x(n)\}$ , and  $\mathbf{H}$  is a real-valued  $2N \times 2N$  matrix.

The VMM in (7) cannot be implemented directly, as some of the coefficients in  $\mathbf{H}$  are negative. Instead, we represent each signal differentially, and we represent each element of  $\mathbf{H}$  by a  $2 \times 2$  nonnegative differential submatrix by mapping a positive gain  $G$  to  $[G \ 0; 0 \ G]$  and a negative gain  $-G$  to  $[0 \ G; G \ 0]$ . We thus transform the  $2N \times 2N$  matrix  $\mathbf{H}$  into an equivalent  $4N \times 4N$  matrix with nonnegative weights that can be programmed into the FPAA chip.

For a size-4 DFT, we need a  $16 \times 16$  VMM circuit with real-valued nonnegative weights. The schematic of the  $16 \times 16$  VMM structure in the FPAA chip is shown in Fig. 6. As discussed in Section II-B, the weighting coefficients  $W_{1,1} - W_{16,16}$  can be programmed into the FPAA chip by assigning appropriate charge values to the input and output floating-gate transistors. Note that the weights in each column of the VMM circuit correspond to the coefficients in each row of the  $16 \times 16$  VMM matrix.

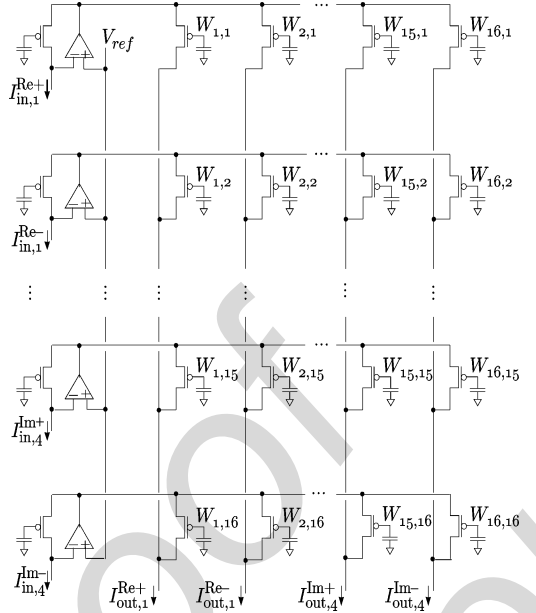


Fig. 6. FPAA implementation of a 4-point DFT as a  $16 \times 16$  VMM circuit. Each input current of the floating-gate transistors on the left determines the output voltage of the corresponding OTA, and this output voltage is broadcast to the source of all the connected floating-gate transistors in each row. At the output floating-gate transistor, this source voltage drives a drain current which is a scaled multiple of the corresponding input current. Then, the scaled currents are added up along each column to give a combined output current per column.

The output port of each OTA is connected to the source of the input floating-gate transistors, and the negative input port is connected to the drain. The positive input port is set to the reference voltage  $V_{ref}$ . Because of the negative feedback of the OTA, the drain voltage of each input floating-gate transistor is also close to  $V_{ref}$ . Therefore, the input currents of the VMM circuit, defined as the drain currents of the input floating-gate transistors, can be controlled by connecting a resistor to the drain of each input floating-gate transistor and then varying the input voltage  $V_{in}$  applied to the resistors from 0 V to  $V_{ref}$ . This configuration is shown in Fig. 7. Due to the nonideal characteristic of the negative feedback OTA, the realistic gain of the negative feedback OTA is finite. Therefore, the negative input voltage of the OTA does not stay close to  $V_{ref}$ , particularly when the input voltage  $V_{in}$  gets close to  $V_{ref}$  or, equivalently, the input current  $I_{in}$  gets close to zero. Hence, the operating range of the input current is chosen at  $0.2 - 1.0 \mu\text{A}$  in order to provide a linear relationship between the input voltage and the corresponding input current so that the received OFDM signals can be linearly mapped to the input currents of the VMM circuit, and to minimize the redundant power consumption while holding a reasonable current resolution and path delay. This range also guarantees that each transistor operates in the subthreshold region for weights less than or equal to one.

The drain current of each input floating-gate transistor determines the output voltage of the corresponding OTA, and this output voltage is broadcast to the source of all the connected floating-gate transistors in each row. When the drain voltage of each output floating-gate transistor is set to  $V_{ref}$ , this source voltage drives a drain current of each output floating-gate transistor, which is a scaled multiple of the corresponding input current. Then, the drain currents from the output floating-gate transistors are added up along each column to give a combined output current per column.

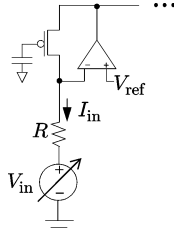


Fig. 7. Input voltage supplied to the drain of an input floating-gate transistor through a resistor. The setup enables the received OFDM signals to be linearly mapped to the input current of the VMM circuit.

Note that, in Fig. 6, each output floating-gate transistor has only one OTA connected to it, so the current level flowing through each output transistor is determined by the input current level and the programmed weights with respect to the connected input. In addition, the DFT coefficients involve a factor of  $1/N$ , as shown in (4), so all the converted nonnegative weights span within the range of  $0 \leq W \leq 1/N$ . These factors guarantee that each transistor of the VMM circuit operates in the subthreshold region, as far as the input current level stays less than the threshold current.

The required operations in the VMM given in (7) are scaling and summing operations. Since the information signals are conveyed in current levels, the summing operations do not require additional circuits. Therefore, the power consumption becomes less than that for the digital circuits where the information signals are conveyed in voltage levels and adding entries requires full adders. The scaling operations do not involve any complex multiplications, as all the signals and weights are real valued. Therefore, the power consumption in the scaling operations is also limited.

Now, we can take into account a butterfly operation in order to reduce the number of computations for a DFT operation. The butterfly operation basically decomposes a DFT matrix into a series of smaller matrices, and each output from the previous stage is handed over to the next stage. This can be viewed as a cascade of VMMs, where the VMM size in each stage gets smaller by a factor of the radix size. Therefore, it is clear that applying butterfly operations in the VMM circuit increases the path delay of the circuit. As the radix size gets lower, the number of stages increases, and consequently, the path delay increases. Moreover, the butterfly operation substitutes copying operations for summing and scaling operations. This is beneficial in voltage-mode circuits, where a summing operation requires a full adder, whereas a copying operation is trivial. However, in a current-mode circuit, a copying operation requires a current mirror or a current multiplier, whereas a summing operation is trivial. It is also claimed in [15] that an FFT design with a higher radix becomes less sensitive to the device mismatch. For these reasons, a full-radix DFT is more preferable for a current-mode analog circuit design.

### III. FPAA PROGRAMMING PROCEDURE

#### A. Programming Platform

To simplify the implementation of an analog DFT in the FPAA, we scale up the DFT matrix by a factor of  $N$ , with the understanding that it can be compensated for by scaling down the outputs of the analog DFT by a reciprocal of the scaling factor. This simplification makes the coefficients span within the  $0 \leq W \leq 1$  range, so each transistor will still operate in the subthreshold region. In particular, the  $16 \times 16$  matrix for a

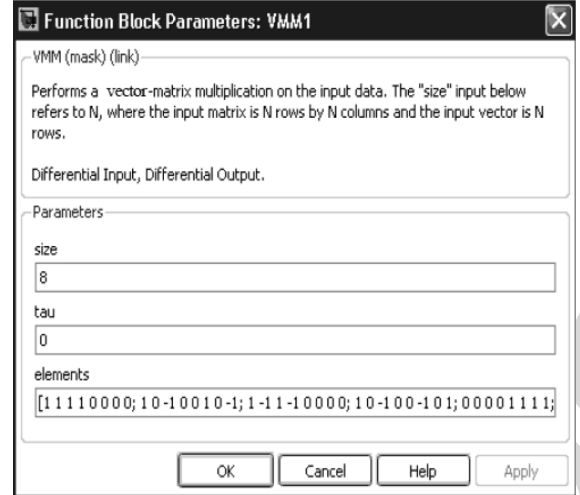
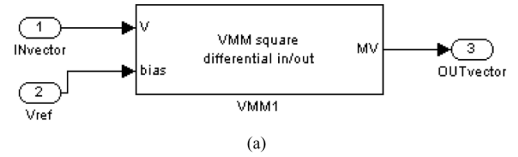


Fig. 8. (a) Custom VMM library block for the Simulink and (b) its block property that contains a field where real-valued  $8 \times 8$  differential weighting coefficients can be defined. The Matlab script is coded to load the VMM block with provided weights to generate a netlist file of the corresponding  $16 \times 16$  VMM analog circuit.

4-point DFT contains only ones and zeros with this simplification. This makes the amount of charge to be stored in each transistor relatively close to each other so as to increase the linearity between the input and output current levels for each transistor. The resulting matrix is then provided to a custom library block for a VMM in the Simulink shown in Fig. 8, and the Matlab script is coded to load the library block with a provided weighting matrix to generate a netlist of the VMM analog circuit [21], [22].

The generated netlist is taken by the RASPER tool that places and routes the available components in the FPAA chip [23]. The output file of the RASPER is a list of switch addresses and the targeting current value for each switch. This list is loaded by the Matlab script to be programmed into the FPAA chip. The RASP 2.9 FPAA chip contains the necessary circuitry for tunneling and injecting electrical charges of floating-gate transistors and the circuitry for current measurement. All the stored charges are tunneled before getting programmed, and an appropriate amount of charge value is injected to each floating-gate transistor while targeting on the corresponding current level determined by (1) and (2). The targeting current levels are represented with 10 bits of precision, of which 3 bits are assigned for the exponent and 7 bits for the significant [24]. Even though the information signals are conveyed in unquantized current levels, the accuracy in the programmed weights will impose a limit to the resolution of the analog DFT system.

In order to reduce the circuitry required for measurement and tunneling and injecting charges, the indirect programming method is used to charge the floating-gate transistors [25]. Fig. 9 shows the indirect programming structure for a floating-gate pMOSFET. The floating-gate transistor on the left is connected to the on-chip programming circuitry and is actively programmed. The one on the right is the floating-gate

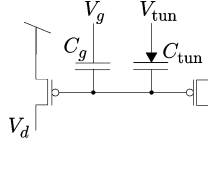


Fig. 9. Indirect programming structure of a pMOSFET. The left transistor is part of the on-chip programming circuitry and is actively programmed. The transistor on the right is the transistor that is used for the VMM circuit and is passively programmed.

transistor that is used for the VMM circuit and is passively programmed.

Due to the inherent mismatch between threshold voltages of different transistors, the indirectly programmed charges in the floating-gate transistors for the VMM circuit can be different from the directly programmed charges in the floating-gate transistors of the programming circuitry. This mismatch also occurs in between the programmed charges in the input and output floating-gate transistors of the VMM circuit. While this mismatch is inherent, we can circumvent this by adjusting the charge value in each input and output floating-gate transistor. We will discuss this process in Section III-B.

### B. Mismatch in FPAA Chips

When there is a mismatch in threshold voltages of different transistors, the preexponential factor of each MOSFET can be different from each other, and thus, the programmed weights can suffer from multiplicative distortion. However, the ratio of the output current to the input current is a function of the relative difference in charge values, as shown in (3), so the mismatch in weights can be compensated for by adjusting the charge values in the floating-gate transistors. This process can be accomplished by targeting on the desired *ratio* of the input and output currents, rather than targeting on the desired input and output currents themselves.

In the VMM circuit, each output current is the sum of the drain currents of the output floating-gate transistors in each column. Due to the different levels of nonlinearity in the  $I-V$  characteristics of the input OTAs, there exist additive offsets between the input and output current levels. Therefore, targeting on the ratio at a single point will not suffice. Instead, we need two points of measurement so that a *slope* of the output current versus input current can be targeted. Thus, the FPAA programming procedure is conducted in the following two steps so as to minimize errors in the programmed weights.

#### 1) Coarse programming

- a) The fully turned-on switches and the input floating-gate transistors are first programmed with the desired charge values by targeting on the corresponding current levels.
- b) The floating-gate transistors inside the OTAs are also programmed with the charge values corresponding to the bias current.
- c) On the other hand, the output floating-gate transistors are programmed with lower charge values than what are desired by targeting on a half of the corresponding current levels.

#### 2) Fine programming

- a) Each output floating-gate transistor is then injected with a small amount of electrons iteratively to increase the stored charge.
- b) In each iteration, the input and output current values are measured at two different input voltages, and then, the slope of the input and output currents is obtained.
- c) The iteration stops when the slope of the input and output currents reaches the desired weight.

After the fine programming, the output currents of the VMM circuit still involve additive offsets, but these offsets do not vary as the input current levels change. Therefore, the sum of these offsets per output node is constant, and it can be easily calculated to be subtracted out from each output current.

## IV. FPAA MEASUREMENT AND EQUALIZATION

### A. FPAA Measurement

We now investigate the measured data of the OFDM receiver with an analog DFT demodulator. The transmitted symbols are randomly generated and mapped to 16 quadratic amplitude modulation (QAM) complex symbols with Gray coding. The generated symbols are then modulated by a size-4 inverse FFT. The guard interval is allocated for 1/4 of the FFT size, and the resulting complex samples are serialized, applied to a DAC, and upconverted to the carrier frequency. On the receiver side, after downconversion and removal of the guard interval, the received OFDM signals are split into real-valued differential pairs and converted to the input currents of the analog DFT within the current range of 0.2–1.0  $\mu\text{A}$ , as discussed in Section II-C. The converted 16 input currents are then fed into the  $16 \times 16$  VMM analog circuit implemented in the FPAA to demodulate the OFDM signals. The resulting 16 output currents of the FPAA are sampled, then reverted back to the voltage levels, and reassembled to yield four complex single-ended demodulated OFDM signals. These are then fed into a 16-QAM demapper to recover the transmitted symbols.

Fig. 10(a)–(b) shows the  $I-Q$  plots of the demodulated symbols without injecting any channel noise. The color maps are used to illustrate the density of the occurrence. It can be observed in Fig. 10(a) that there is some dispersion in the demodulated symbols even without any channel noise, which results in a performance penalty as a price for the reduced power consumption in the analog DFT. The performance degradation arises for multiple reasons, including the following:

- 1) errors in the programmed weights due to the limit on the bit precision for targeting current values;
- 2) temperature sensitivity of the programmed weights;
- 3) nonlinear mapping between the input voltage and input current caused by the nonideal characteristic of the OTA;
- 4) thermal noise;
- 5) parasitic capacitance between routed paths.

Despite the dispersion, however, the 16-QAM demapper in the Matlab determined all the demodulated symbols correctly. This implies that the error rate will still converge to zero in a noisy channel as the signal-to-noise ratio (SNR) increases.

The processing speed of the analog DFT in the FPAA is limited by the settling time of the VMM circuit. Fig. 11 shows the step response of the  $16 \times 16$  VMM circuit for a size-4 analog DFT implemented in the FPAA (RASP 2.9) while the step input changes from 0.2 to 1.0  $\mu\text{A}$ . To measure the accurate settling time of the VMM circuit, an  $I-V$  conversion circuit shown in Fig. 12 was included to the programming netlist so that the

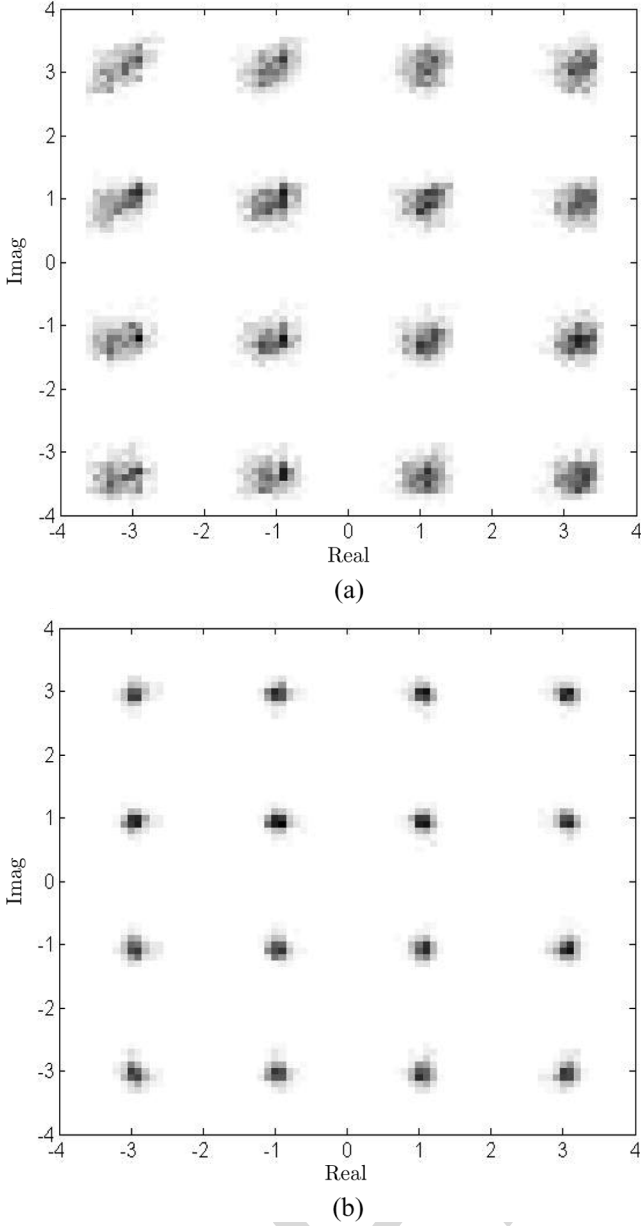


Fig. 10. Constellations of the demodulated symbols for 16 QAM without channel noise. (a) Before equalization. (b) After MMSE equalization. The gradation depicts the density of the occurrence in each pixel.

output voltage signals can be measured by a high-frequency oscilloscope. It can be observed in Fig. 11 that the settling time of the VMM circuit is around  $4 \mu\text{s}$ , which is close to a typical OFDM symbol duration for the IEEE 802.11 a/g with a 64-point FFT. Note that the measured settling time includes an additional delay caused by the auxiliary  $I$ - $V$  conversion circuit itself, so the actual settling time of the VMM circuit will be less than the measured value. For a size-4 digital DFT implemented in the FPGA with an 8-bit data width, the minimum data path delay is 49.6 ns for Xilinx Virtex2Pro (device: XC2VP30; package: ff896; speed: -7) and 101.6 ns for Xilinx Virtex (device: XCV50; package: fg256; speed: -5). However, when the number of subchannels increases and, thus, the OFDM symbol duration increases, the path delay in the digital DFT increases due to the larger number of complex multiplications and

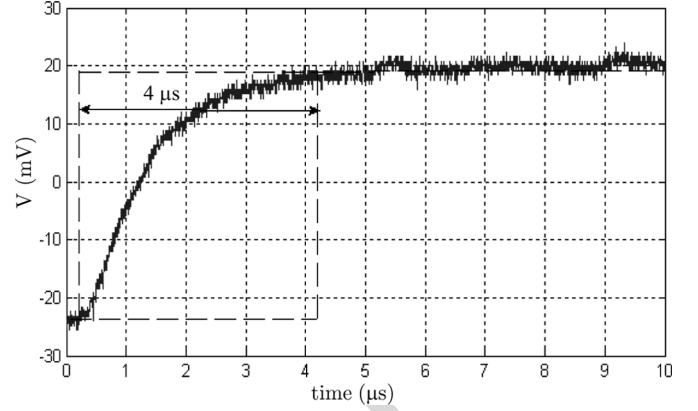


Fig. 11. Step response of the VMM circuit implemented in the RASP 2.9 FPAA.

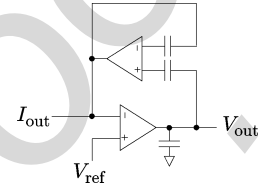


Fig. 12. On-chip  $I$ - $V$  conversion circuit. This is attached to the output node of the VMM circuit to eliminate delays caused by the measurement setup.

the hierarchical structure of the digital adders, whereas, in current-mode analog circuits, it stays almost the same because of the parallelized structure of the real-valued VMM operation. In [10], an analog filter implemented in a FPAA with a  $0.13\text{-}\mu\text{m}$  CMOS process is reported to achieve a frequency range up to 135 MHz, thus showing a potential increase in the processing speed of an analog DFT implemented in a FPAA with a smaller CMOS process.

The total power consumed in the analog 4-point DFT of the FPAA is measured to be 13.4 mW. This measured power is larger than the theoretically expected value for the  $16 \times 16$  VMM circuit that can be obtained by  $16.2 \cdot I_{\text{bias}} \cdot V_{\text{dd}} \approx 3 \text{ mW}$ , where the bias current of the OTA  $I_{\text{bias}}$  is  $40 \mu\text{A}$  and the bias voltage  $V_{\text{dd}}$  is 2.4 V. This difference may have been caused by the imperfect isolation of the VMM circuit from the rest of the chip and board. The digital 4-point DFT in the Virtex2Pro FPGA required 247 mW at the maximum speed and 105 mW at the same speed as the analog DFT. For the Virtex FPGA, it required 219 mW at the maximum speed and 34 mW at the same speed as the RASP 2.9 FPAA. Therefore, the power consumption required for the 4-point DFT operation is significantly reduced by 8.9 dB and 4.0 dB, respectively, for the same speed. Table I shows the comparisons of the measurements for the FPGA and FPAA. Aside from the power saving in the DFT block itself, implementing a DFT block in an analog circuit allows the ADC to be placed after the DFT block at the receiver, thus effectively reducing the overall power consumption by relieving the speed and bit-precision requirements of the ADC block. The application-specific integrated circuit (ASIC) implementation of an analog DFT in [15] was reported to consume lower power than the FPAA implementation, where the full-radix analog 256-point DFT implemented in an ASIC with a 180-nm CMOS process was claimed to consume 1.6 mW. Despite the higher power consumption compared to the ASIC implementation, the benefit of the FPAA implementation

TABLE I  
POWER AND DELAY COMPARISONS FOR FPGA AND FPAA

Chipset	Virtex2Pro FPGA	Virtex FPGA	RASP2.9 FPAA
Power consumption @ Processing delay	247 mW @ 49.6 ns	219 mW @ 101.6 ns	13.4 mW @ 4 $\mu$ s
	105 mW @ 4 $\mu$ s	34 mW @ 4 $\mu$ s	

The upper power consumption values for the FPGAs are measured when operating at its own fastest speed. The lower values are measured when operating at the same speed as the RASP 2.9 FPAA.

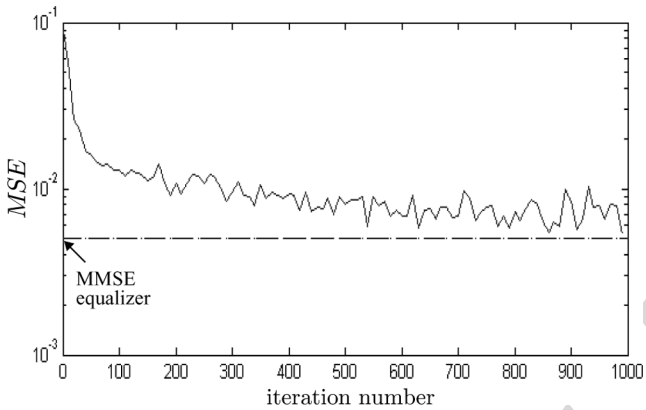


Fig. 13. MSE trace of a  $16 \times 16$  LMS equalizer.

with floating-gate transistors is its ability to tune the DFT coefficients caused by the mismatch in transistors without changing the circuit structure.

### B. Equalization of FPAA Outputs

Any residual errors in the programmed weights of the DFT will prevent it from perfectly separating the symbols for the different subcarriers, leading to a form of intersymbol interference (ISI). These errors can be mitigated by applying an equalizer to each output of the analog DFT block. The equalizer coefficients can be obtained by injecting training symbols. As the errors in the programmed weights are independent with each other, each output of the equalizer has 16 taps, and the 16 parallel outputs are equalized separately. Therefore, the  $k$ th output  $z_k$  of the equalizer (for  $k = 1, \dots, 16$ ) is given by the inner product  $z_k = \mathbf{c}_k^T \mathbf{X}$  of the  $k$ th equalizer coefficient vector  $\mathbf{c}_k = [c_{k,1} \dots c_{k,16}]^T$  and the output vector  $\mathbf{X} = [X_1 \dots X_{16}]^T$  of the DFT block.

The minimum mean-square-error (MMSE) coefficients that minimize  $MSE = E((c_k^T \mathbf{X} - a_k)^2)$ , where  $a_k$  represents the training symbols, are [26]

$$\mathbf{c}_k = \mathbf{R}^{-1} \mathbf{p}_k \quad (8)$$

where  $\mathbf{R} = \mathbf{E}(\mathbf{X}\mathbf{X}^T)$  and  $\mathbf{p}_k = E(a_k \mathbf{X})$ . Therefore, each of the 16 parallel outputs from the DFT block can be equalized with the 16 tap coefficients in (8). Fig. 10(b) depicts the equalized symbols while using a  $16 \times 16$  MMSE equalizer. It can be observed that demodulated symbols are located tighter in the  $I$ - $Q$  map by applying equalization to the outputs of the DFT.

For a least mean square (LMS) equalizer, the equalizer coefficients are updated along the steepest decent direction using [26]

$$\mathbf{c}_k[n+1] = \mathbf{c}_k[n] - \mu(z_k - a_k) \mathbf{X}[n] \quad (9)$$

where  $\mu$  is the step size. Fig. 13 exhibits the trace of the  $MSE$  for an LMS equalizer when the step size is  $\mu = 5 \times 10^{-4}$  and the initial coefficient vector for each output is set to each row of the  $16 \times 16$  identity matrix. It can be seen from Fig. 13 that the convergence occurs within 500 iterations with these parameters. This iteration can be also applied to an adaptive programming scheme by charging floating-gate transistors with an updated amount of injection based on the measured current level.

### C. BER Performance in AWGN Channels

We now consider the case when the modulated OFDM signals are passed through a noisy channel to see how the channel noise affects the performance of the analog DFT demodulator. Note that there is a certain range of an input voltage that is allowed to be fed into the FPAA chip, effectively  $0 - V_{\text{ref}}$ , but due to the nature of the high peak-to-average power ratio in OFDM signals, some received OFDM signals from a noisy channel can be converted to the input voltage levels beyond the allowed range. To avoid this case, the converted input voltage levels that are lower than 0 V are set to 0 V. This results in clipping distortions when the input current value is high, but it happens at rare peak voltages of the OFDM signals.

Fig. 14 demonstrates the measured bit error rate (BER) versus  $E_b/N_0$  (SNR per bit) for a 16-QAM OFDM demodulator implemented in an analog DFT, assuming additive white Gaussian noise (AWGN). The performance with the MMSE and LMS equalization is also shown. These results are compared to the theoretical BER for 16 QAM with Gray mapping [27]

$$BER \approx \frac{3}{4} Q \left( \sqrt{\frac{4}{5} E_b/N_0} \right). \quad (10)$$

The measurement was iterated for 2500 cycles, so the sample size is 10 000 symbols or 40 000 bits for each  $E_b/N_0$  value. As can be observed from the plots in Fig. 14, the demodulated OFDM symbols with an analog DFT suffer a performance penalty of 2 dB compared to the theoretical BER curve. This is because the remaining errors in the programmed weights produce an ISI across the parallel outputs of the analog DFT block. However, applying equalization to the outputs of the FPAA significantly relieves the penalty by mitigating the errors in the programmed weights, and the  $E_b/N_0$  gap between the equalized outputs and the theoretical values becomes less than 1 dB.

For a digital DFT demodulator with an 8-bit data width, the measured BER for the same sample size converged to the theoretical values of a 16 QAM regardless of the existence of the IDFT/DFT blocks in between. Therefore, there is a tradeoff between performance and power consumption, but the power saving of the analog circuit outweighs the performance penalty without equalization.

## V. CONCLUSION

We have proposed a low-power analog DFT implemented on an FPAA as an alternative to a conventional OFDM demodulator based on a digital DFT. The analog DFT is implemented



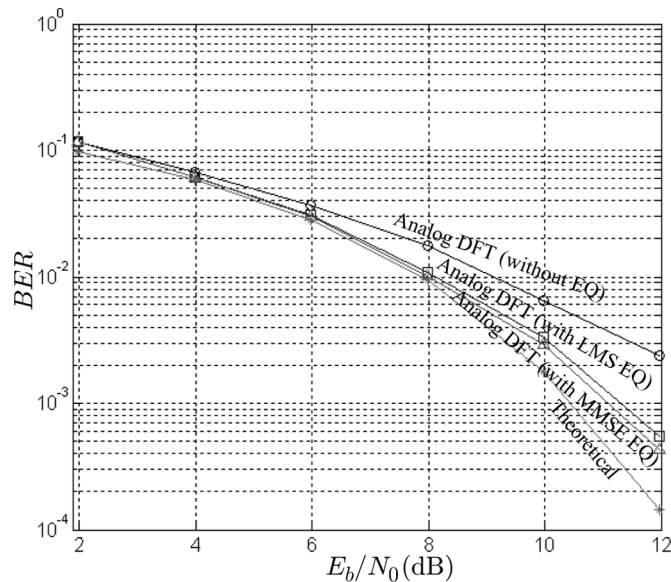


Fig. 14. Performance of 16-QAM OFDM demodulator using an analog DFT, with and without equalization. When compared to theory, the penalty after equalization is less than 1 dB.

as a VMM using floating-gate transistors. The floating-gate transistors of the FPAA are used not only to configure the VMM circuit connections as fully turned-on switches but also to store the DFT coefficients by locking in an appropriate amount of charge in each floating-gate capacitor. The analog DFT in the FPAA consumed 8.9 dB less power than a digital implementation using a Virtex2Pro FPGA, and it consumed 4.0 dB less power than a digital implementation using a Virtex FPGA. This power reduction, although significant, does not reflect the additional power savings that come from the fact that an analog DFT reduces the speed and precision requirements of the ADCs. The price paid for this power reduction was a 2-dB performance degradation. We have also shown that this performance loss can be mitigated by exploiting an equalizer technique as a top-down approach to tackle the device mismatch problem. Furthermore, the unquantized output signals from the analog DFT block enable the real soft inputs to the subsequent decoding block at the OFDM receiver.

#### ACKNOWLEDGMENT

The authors would like to thank S. Brink for his editorial support on the reconfigurable analog signal processor 2.9 field-programmable analog array chip.

#### REFERENCES

- [1] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nW/MHz,  $128 \times 32$  current-mode programmable analog vector-matrix multiplier with over 2 decades of linearity," in *Proc. IEEE Custom Integr. Circuits Conf.*, Oct. 2004, pp. 651–654.
- [2] T. S. Hall, C. M. Twigg, J. D. Gray, P. Hasler, and D. V. Anderson, "Large-scale field-programmable analog arrays for analog signal processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 11, pp. 2298–2307, Nov. 2005.
- [3] E. K. F. Lee and P. G. Gulak, "A CMOS field-programmable analog array," *IEEE J. Solid-State Circuits*, vol. 26, no. 12, pp. 1860–1867, Dec. 1991.

- [4] H. Kutuk and S.-M. Kang, "A field-programmable analog array (FPAA) using switched-capacitor techniques," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1996, pp. 41–44.
- [5] E. K. F. Lee and W. L. Hui, "A novel switched-capacitor based field-programmable analog array architecture," *Analog Integr. Circuits Signal Process.*, vol. 17, no. 1/2, pp. 35–50, Sep. 1998.
- [6] E. K. F. Lee and P. G. Gulak, "A transconductor-based field-programmable analog array," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 1995, pp. 198–199.
- [7] B. Pankiewicz, M. Wojcikowski, S. Szczepanski, and Y. Sun, "A field programmable analog array for CMOS continuous-time OTA-C filter applications," *IEEE J. Solid-State Circuits*, vol. 37, no. 2, pp. 125–136, Feb. 2002.
- [8] A. Basu, C. M. Twigg, S. Brink, P. Hasler, C. Petre, S. Ramakrishnan, S. Koziol, and C. Schlottmann, "RASP 2.8: A new generation of floating-gate based field programmable analog array," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2008, pp. 213–216.
- [9] J. Becker, F. Henrici, S. Trendelenburg, M. Ortmanns, and Y. Manoli, "A field-programmable analog array of 55 digitally tunable OTAs in a hexagonal lattice," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 2759–2768, Dec. 2008.
- [10] F. Henrici, J. Becker, S. Trendelenburg, D. DeDorigo, M. Ortmanns, and Y. Manoli, "A field programmable analog array using floating gates for high resolution tuning," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 265–268.
- [11] A. Stoica, D. Keymeulen, M. Mojarradi, R. Zebulum, and T. Daud, "Progress in the development of field programmable analog arrays for space applications," in *Proc. IEEE Aerosp. Conf.*, Mar. 2008, pp. 1–9.
- [12] D. Keymeulen, A. Stoica, R. Zebulum, S. Katkooi, P. Fernando, H. Sankaran, M. Mojarradi, and T. Daud, "Self-reconfigurable analog array integrated circuit architecture for space applications," in *Proc. NASA/ESA Conf. Adapt. Hardw. Syst.*, Jun. 2008, pp. 83–90.
- [13] M. Lehne and S. Raman, "An analog/mixed-signal FFT processor for wideband OFDM systems," in *Proc. IEEE Sarnoff Symp.*, Mar. 2006, pp. 1–4.
- [14] M. Lehne and S. Raman, "A prototype analog/mixed-signal fast Fourier transform processor IC for OFDM receivers," in *Proc. IEEE Radio Wireless Symp.*, Jan. 2008, pp. 803–806.
- [15] N. Sadeghi, V. C. Gaudet, and C. Schlegel, "Analog DFT processors for OFDM receivers: Circuit mismatch and system performance analysis," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 2123–2131, Sep. 2009.
- [16] E. Afshari, H. S. Bhat, and A. Hajimiri, "Ultrafast analog Fourier transform using 2-D LC lattice," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 8, pp. 2332–2343, Sep. 2008.
- [17] "AN231E04 Datasheet Rev. 1.0, Dynamically Reconfigurable dpASP," Analogm, Oak Park, CA, 2007.
- [18] "ispPAC81 Datasheet, In-System Programmable Analog Circuit," Lattice Semicond., Hillsboro, OR, 2001.
- [19] B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, "Analog-to-digital converters," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 69–77, Nov. 2005.
- [20] M. Kucic, A. Low, P. Hasler, and J. Neff, "A programmable continuous-time floating-gate Fourier processor," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 48, no. 1, pp. 90–99, Jan. 2001.
- [21] C. Petre, C. Schlottmann, and P. Hasler, "Automated conversion of Simulink designs to analog hardware on an FPAA," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 500–503.
- [22] C. Schlottmann, C. Petre, and P. Hasler, "Vector matrix multiplier on field programmable analog array," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2010, pp. 1522–1525.
- [23] F. Baskaya, S. Reddy, S. K. Lim, and D. V. Anderson, "Placement for large-scale floating-gate field-programmable analog arrays," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 8, pp. 906–910, Aug. 2006.
- [24] A. Basu and P. E. Hasler, "A fully integrated architecture for fast and accurate programming of floating gates over six decades of current," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2010, to be published.
- [25] D. W. Graham, E. Farquhar, B. Degnan, C. Gordon, and P. Hasler, "Indirect programming of floating-gate transistors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 5, pp. 951–963, May 2007.
- [26] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 2001, pp. 203–207, 231–238.
- [27] J. Proakis, *Digital Communications*. New York: McGraw-Hill, 2000, pp. 276–280.



**Sangwook Suh** (S'06) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1998, and the M.S. degree in electrical engineering from the Polytechnic Institute of New York University, New York, in 2005. He is currently working toward the Ph.D. degree in electrical and computer engineering at the Georgia Institute of Technology, Atlanta.

In lieu of military service, he was with Sein Electronics Company, Ltd., from 1998 to 2000, and with Corecess Inc. from 2000 to 2002. In 2002, he was a Software Engineering Intern with Microsoft Corporation, Seoul, Korea. From 2005 to 2006, he was a Software Engineer with Samsung Electronics Company, Ltd., Suwon, Korea. His research interests include adaptive equalization, low-power signal processing, and soft-input analog decoders.



**Arindam Basu** (S'06–M'10) received the B.Tech and M.Tech degrees in electronics and electrical communication engineering from the Indian Institute of Technology Kharagpur (IIT Kharagpur), Kharagpur, India, in 2005, and the M.S. degree in mathematics and the Ph.D. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in 2009 and 2010, respectively.

He is currently an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore. His research interests include nonlinear dynamics and chaos, modeling neural dynamics, low-power analog IC design, and programmable circuits and devices.

Dr. Basu was a recipient of the Jagadis Bose National Science Talent Search Award in 2000 and the Prime Minister of India Gold Medal from IIT Kharagpur in 2005. He was a recipient of the Best Student Paper Award at the IEEE Ultrasonics Symposium in 2006 and the Best Live Demonstration Award at the IEEE International Symposium on Circuits and Systems (ISCAS) in 2010 and was also a Best Student Paper Award finalist at ISCAS in 2008.



**Craig Schlottmann** (S'07) received the B.S. degree in electrical engineering from the University of Florida, Gainesville, in 2007, and the M.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, in 2009. He is currently working toward the Ph.D. degree in electrical engineering at the Georgia Institute of Technology.

His research interests include low-power analog signal processing, multiple-input translinear elements, floating-gate transistor circuits, and analog IC design.



**Paul E. Hasler** (S'87–M'01–SM'03) 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.

In 2002, he cofounded GTronix, Inc., which was acquired by National Semiconductor in 2010. He is currently an Associate Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. His current research interests include low-power electronics, mixed-signal system ICs, floating-gate MOS transistors, adaptive information processing systems, smart interfaces for sensors, cooperative analog–digital signal processing, device physics related to submicrometer devices or floating-gate devices, and analog very large scale integration models of on-chip learning and sensory processing in neurobiology.

Dr. Hasler was a recipient of the National Science Foundation CAREER Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He was also a recipient of the Paul Rappaport Best Paper Award from the IEEE Electron Devices Society in 1997, the Best Paper Award at the Multiconference on Systemics, Cybernetics, and Informatics in 2001, the Best Sensor Track Paper at the IEEE International Symposium on Circuits and Systems in 2005, the Best Student Paper Award at the IEEE Custom Integrated Circuits Conference in 2006, the Best Student Paper Award at the IEEE Ultrasound Symposium in 2006, and the Best Demonstration Paper Award at the IEEE International Symposium on Circuits and Systems in 2010.



**John R. Barry** (S'85–M'87–SM'04) received the B.S. degree in electrical engineering from the State University of New York, Buffalo, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, Berkeley, in 1987 and 1992, respectively.

Since 1992, he has been with the Georgia Institute of Technology, Atlanta, where he is currently a Professor with the School of Electrical and Computer Engineering. His research interests include wireless communications, equalization, and multiuser communications. He is a coauthor with E. A. Lee and D. G. Messerschmitt of *Digital Communication* (Springer, 2004) and the author of *Wireless Infrared Communications* (Kluwer, 1994).