Calibration of Floating-Gate SoC FPAA System

Sihwan Kim, Sahil Shah, and Jennifer Hasler, Senior Member, IEEE

Abstract—We present a calibration flow for a large-scale floating-gate (FG) system-on-chip field programmable analog array. We focus on characterizing the FG programming infrastructure and hot-electron injection parameters, MOSFET parameters using the EKV model, and calibrating digital-analog converters and analog-digital converters. In addition, threshold voltage mismatches on FG devices due to their indirect structure are characterized using on-chip measurement techniques. The calibration results in enabling a digital approach, where a design can be programmed without having to deal with the local and global mismatches, on a reconfigurable analog system. This paper shows the results of a compiled nonlinear classifier block comprising a vector-matrix-multiplier and a winner-takes-all on three different calibrated chips.

Index Terms—Calibration, floating-gate (FG) field programmable analog array (FPAA), mismatch.

I. CALIBRATION ON DIGITAL/ANALOG SYSTEMS

Digital system design is enhanced when an algorithm can be directly ported to any number of equivalently designed systems, with effectively the same performance for all devices. Although digital system-on-chip (SoC) systems require a calibration (e.g., a clock speed, bad memory blocks, and internal voltage regulators) and precision components (e.g., a clock crystal, oscillator), this process is independent of the algorithm, performed away from system programmers.

One rarely expects this property in analog systems, even when some form of programmability is possible. Every system is handled in a special way; a mismatch is the primary limiting factor for analog systems [1] resulting from the fact that “not all transistors are created equal.” Typically, an analog-digital converter (ADC) and filters (e.g., Gm-C topologies) utilize programmable elements to deal with mismatches; larger analog systems significantly effect larger levels of algorithm modification. One can reduce calibration via an increased device area to reduce mismatches, resulting in a larger die area and cost, implying higher power consumption as well as lower levels of system integration.

This paper describes bringing analog computation toward the expected (digital) system techniques, where a one-time calibration of a batch of devices enables the same algorithm at similar performance levels to be downloaded to all devices. This project will focus on large-scale field programmable analog arrays (FPAs), with particular focus on the SoC FPAA IC described in [3]. The dense programmable element is a floating-gate (FG) device, found in standard CMOS processes [4].

II. FLOATING-GATE SOC FPAA ARCHITECTURE

The infrastructure for FPAA systems has been integrated onto a chip to increase area efficiency, as well as analog parameter density [11], [12] to enable more complicated applications [3], [13]. Fig. 2 shows the printed circuit board (PCB) and IC level architecture of the latest version of the FG FPAA family [3]. The IC comprises an FPAA fabric
array, an FG programming infrastructure, a \( \mu \)P (open-source MSP430 [14]), and \( 16 \times 16 \) SRAM. The FG programming infrastructure includes a 7-bit gate digital-analog converter (DAC), a 7-bit drain DAC, a pFET diode \( I-V \) converter, and a 14-bit ramp ADC, interfacing with \( \mu \)P through memory mapped registers.

The PCB consists of power components regulating 2.5/3.3 V, charge pump units handling high voltages (6/12 V), and input/output (I/O) pins for external connection (to be used with voltage generators, voltmeters, ammeters, and so on). Some of the external pins are connected to the array to provide direct input or enable measurements, and some are connected to the FG programming infrastructure in calibration mode.

The FPAA array includes computational analog blocks (CABs), computational logic blocks (CLBs), and routing switches composed of connection (C), switch (S), and I/O blocks. Each CAB includes local routing switches for connecting the inputs/outputs of a CAB to its elements, such as operational transconductance amplifiers (OTAs) with and without FG inputs, nFETs, pFETs, capacitors, and T-gates. Each CLB includes local routing switches with basic logic elements lookup table circuits. FG switches can be used for computation [e.g., vector-matrix-multiplier (VMM)] as well as for connections between CAB/CLB/I/O blocks.

III. DESIGN COMPIlATION AND FG PROGRAMMING

Fig. 3(a) and (b) shows the compilation flow from designing a high-level application in Scilab/Xcos (open-source programs similar to MATLAB/Simulink) by a user to measuring the output. When the user compiles the design, each chip's calibration information is integrated with it. As shown in Fig. 3, a switch list refers to an FG mismatch table, an input vector refers to a calibrated DAC table and program assembly codes (prog codes), and lookup tables for programming refer to FG device parameters and program infrastructure characterization tables. These generated files are sent to the FG FPAA IC, which programs the switches and measures data. When the output is sent back, the characterized ADC table is used to map the hex codes to their analog values (e.g., voltages).

The characterization of FG device parameters and program infrastructure requires an understanding of the FG programming algorithm. A detailed discussion on the algorithm in the FG SoC FPAA system is presented elsewhere [10]; this paper summarizes the algorithm and brings up related parts. The programming of FG devices relies on a combination of electron tunneling and hot-electron injection. Fig. 3(c) shows a program sequence from tunneling to precise injection, and Fig. 3(d) shows the terminal voltage condition of the FG device for each step.
Erasing FG devices is a global operation requiring a sufficiently high voltage (12 V) on the tunneling junction of all the FG devices, which results in a low channel current (≈1 nA). Reverse tunneling, also a global operation, requires a lower voltage (6 V) on all the terminals of the FG device except the tunneling junction, resulting in a current of a few pA, which is at a proper range for injection. During recover injection, each FG is programmed to a current of 1 nA. Since the leakage from the array and drain decoder is several hundreds of pA, the current in the recover injection is measured by using the gate capacitive coupling effect of the FG device; 20–30 nA of current, measured in the recover injection, with \( V_g \) at 0 V, corresponds to 1 nA when measured with \( V_g \) at 0.6 V in the coarse injection, which is the next step. The effective FG capacitive coupling with a different \( V_g \) is characterized in the calibration flow and integrated into the programming algorithm during the compilation.

Hot-electron injection current (\( I_{inj} \)) in subthreshold or near subthreshold operation [15], [16] is \( I_{inj} \propto I_s e^{(\Phi_1/\Phi_{dc})} \), where \( I_s \) is the channel current and \( \Phi_{dc} \) is the drain-to-channel potential. \( Q_{fg} \) (charge on the FG) \( (Q_{fg} = \int I_{inj} dt) \) is a function of time and voltage between source and drain. Coarse injection fixes \( V_d \) at 0 V for fast electron injection and controls the time of drain pulse, requiring characterization of the pulsewidth table to calculate the number of unit pulses (10 µs) to program an FG at a close range from the target current. Precise injection fixes the drain pulsewidth and controls the drain voltage for precise electron injection, requiring characterization of a 7-bit drain DAC.

Fig. 4. Off-chip equipment (voltage generator, voltmeter, and ammeter) is required for steps 1, 2, and 4, but the external measurement device is no longer necessary after the calibration. In particular, the ammeter, which is large and heavy compared with the FG SoC FPAA system, is not in use after step 2. Each step has been automated to enable a mass chip calibration and then integrated into the compilation flow.

**IV. CALIBRATION OF FG SoC FPAA**

This section illustrates five steps of the calibration flow shown in Fig. 4 and shows nonlinear classifier results working in multiple calibrated chips. Off-chip equipment used for the calibration step 1, 2, and 4 includes analog discovery for generating or measuring voltage and Keithley 6485 Picoammeter for measuring currents through the external pins. The automated
Fig. 5. Characterization of the on-chip FG programming infrastructure circuits is shown. The gate DAC, which converts a 7-bit code to an output voltage through a current bank, is measured by an external voltmeter. With two different supply voltages ($V_{dd}$) for the FG injection and current measurement, the gate DAC has the output voltage in roughly 2–5 V with $V_{dd}$ at 6 and 0.6–2 V with $V_{dd}$ at 2.5 V. A 7-bit drain DAC consisting of a current bank, a resistor, and a buffer is characterized by an external voltmeter. Body-source connected two pFET diodes convert the FG current ($I_{prog}$) to $V_{prog}$ and a ramp ADC converts $V_{prog}$ to a 14-bit code. Based on the characterization by an external voltage generator and ammeter, EKV parameters ($\kappa$, $V_{T0}$, and $I_{th}$), and the slope (m) and y-intercept (b) on the ramp ADC of each chip are calculated.

Table I

<table>
<thead>
<tr>
<th>Programming Infrastructure Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chip 1</strong></td>
</tr>
<tr>
<td>I–V converter</td>
</tr>
<tr>
<td>$I_{th}$</td>
</tr>
<tr>
<td>$V_{T0}$</td>
</tr>
<tr>
<td>Ramp ADC</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The drain of FG device is connected to the $I$–$V$ converter when measuring current ($I_{prog}$). The $I$–$V$ converter consists of two pFETs that have their body connected to the source. The two pFET diode connected transistors are characterized through an external voltage generator and ammeter, which results in the $I_{prog}$–$V_{prog}$ curve. When we assume that the FG transistor is matched with two pFET diode connected transistors in the $I$–$V$ converter, the relationship between $V_{fg}$ and $V_{prog}$ [10] is given by $V_{prog} = 2(V_{dd} - V_{fg})$. The source current of the FG pFET is given in

$$I_{prog} = I_{th} \ln^2 (1 + e^{\kappa(V_{dd}-V_{fg}-V_{T0})/2U_T})$$

where $\kappa$ ("kappa") is the fractional change in the surface potential due to a fractional change in the applied gate voltage, $U_T$ is the thermal voltage, $V_{T0}$ is the threshold voltage, and $I_{th}$ is the threshold current. $\kappa$, $V_{T0}$, and $I_{th}$ are calculated from the measured $I_{prog}$–$V_{prog}$ curve.

A ramp ADC, which interfaces with $\mu$P, converts $V_{prog}$ to a 14-bit code. The slope and y-intercept are calculated based on the 14-bit code–$V_{prog}$ measurement. Table I shows programming infrastructure parameters in multiple chips.

B. Step 2: EKV Modeling of Golden FETs

Modeling of MOSFET devices’ transconductance characteristics is essential for a high-level analog system simulation before the measurement. It also provides an environment to the user that does not need an ammeter. The EKV-model [17], [18] is well known as an MOS transistor model to illustrate an FET’s behavior. The equation of nFET $I_d$ in
the target program. Capacitive coupling offset required for the recovery injection in external equipment. Fig. 7(a) shows the calibration of the gate injection characterization.

C. Step 3: Gate Coupling Offset and Injection Characterization

FG programming parameters are calibrated without any external equipment. Fig. 7(a) shows the calibration of the gate capacitive coupling offset required for the recovery injection in the target program. $V_{\text{out}}$, the output voltage of the two pFET diodes, is measured with $V_g$ at 0 and 0.6 V, while applying a 10-µs injection pulse with $V_d$ at 0 V. $k_{eq} = kC/C_0$, which is proportional to $\Delta V_{\text{out}}$ measured at different $V_g$ values, decreases as $V_{\text{out}}$ increases, since the MOSFET depletion capacitance increases. The slope changes around the boundary of the subthreshold and above-threshold currents (~0.7 µA), since the current with $V_g = 0$ V is in the above threshold region although the current with $V_g = 0.6$ V is still in the subthreshold region.

Fig. 7(b) shows the calibration of the coarse injection characteristic, i.e., $S$-curve, which is measured in the loop of injection with $V_d$ at 0 V and current measurement with $V_g$ at 0.6 V. The injection current in the $S$-curve, which exponentially grows from an unstable equilibrium for the sub/near threshold and exponentially converges toward a stable equilibrium, forms two linear lines crossing at the current of 2.1 µA on the loop injection (final)–loop injection (start) plot [10]. The pulsewidth table, which shows the number of injection pulses to reach $V_{\text{out}}$ (final) from $V_{\text{out}}$ (start), is calculated based on the $S$-curve measurement. Fig. 7(c) shows the FG device structure in an FG FPAA array. Five kinds of FG devices exist; indirect and direct switches for connection or computation (e.g., VMM), an FG device for OTA bias, an FG device at the input of the FG OTA, and an input bias FG for multiple-input translinear element (MITE). The gate coupling offset and the pulsewidth table for each FG device are calibrated, respectively, in each chip, as shown in Tables III and IV.

D. Step 4: Signal DACs and Compiled DAC/ADC Blocks

Fig. 8 shows the calibration of DACs and ADCs, which provides a mixed-signal design environment for users and eliminates the need for external equipment for measurement. Signal DACs, consisting of a current bank and a resistor,
Fig. 7. FG devices require a characterization of the FG programming parameters. (a) Gate capacitive coupling offsets between $V_{out}$ measured with $V_g$ at 0 and 0.6 V in the injection and current-measurement loop are calculated and set for each chip’s recover injection. As $V_{out}$ increases, the offset decreases due to the increase of the MOSFET depletion capacitor. (b) $S$-curves are measured for the pulsewidth table in the coarse program. The injection-measurement loop starts from $V_{out}$ corresponding to 1 nA in current. The pulsewidth table is calculated based on the linear relation on $V_{out}$ (final) - $V_{out}$ (start). (c) We have five kinds of FG devices in the FG SoC FPAA. Each gate capacitive coupling offset and pulsewidth table for each FG device is measured in an automated calibration script.

### TABLE III

**GATE COUPLING PARAMETERS**

<table>
<thead>
<tr>
<th>FG</th>
<th>Chip 1</th>
<th>Chip 2</th>
<th>Chip 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWC (Ind.)</td>
<td>0.190V</td>
<td>0.223V</td>
<td>0.243V</td>
</tr>
<tr>
<td>SWC (Dir.)</td>
<td>0.205V</td>
<td>0.208V</td>
<td>0.206V</td>
</tr>
<tr>
<td>OTA</td>
<td>0.226V</td>
<td>0.270V</td>
<td>0.282V</td>
</tr>
<tr>
<td>FG OTA</td>
<td>0.317V</td>
<td>0.383V</td>
<td>0.388V</td>
</tr>
<tr>
<td>MITE</td>
<td>0.358V</td>
<td>0.429V</td>
<td>0.426V</td>
</tr>
</tbody>
</table>

### TABLE IV

**PULSEWIDTH PARAMETERS**

<table>
<thead>
<tr>
<th>$m_{1/b1}$</th>
<th>SWC (Ind.)</th>
<th>0.955/0.114</th>
<th>0.945/0.121</th>
<th>0.894/0.228</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWC (Dir.)</td>
<td>0.880/0.200</td>
<td>0.873/0.199</td>
<td>0.805/0.318</td>
<td></td>
</tr>
<tr>
<td>OTA</td>
<td>1.056/0.050</td>
<td>1.045/0.036</td>
<td>1.026/0.015</td>
<td></td>
</tr>
<tr>
<td>FG OTA</td>
<td>1.081/0.077</td>
<td>1.029/0.009</td>
<td>1.001/0.032</td>
<td></td>
</tr>
<tr>
<td>MITE</td>
<td>1.049/0.038</td>
<td>1.021/0.003</td>
<td>1.007/0.018</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$m_{2/b2}$</th>
<th>SWC (Ind.)</th>
<th>0.930/0.145</th>
<th>0.938/0.121</th>
<th>0.947/0.111</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTA</td>
<td>0.941/0.130</td>
<td>0.978/0.047</td>
<td>0.964/0.076</td>
<td></td>
</tr>
<tr>
<td>FG OTA</td>
<td>0.973/0.059</td>
<td>0.944/0.117</td>
<td>0.924/0.166</td>
<td></td>
</tr>
<tr>
<td>MITE</td>
<td>0.959/0.093</td>
<td>0.965/0.077</td>
<td>0.957/0.095</td>
<td></td>
</tr>
</tbody>
</table>

An MITE ADC is implemented with an MITE [20] block in a CAB and the programming infrastructure. The surface potential of the MITE FG pFET is capacitively coupled by $V_{in}$. By measuring the increase/decrease in the current through the $I-V$ converter and the program ramp ADC, $V_{in}$ with analog voltage is converted to a 14-bit digital code. A previously calibrated signal DAC is applied to $V_{in}$ to minimize the use of external equipment.

An FG OTA DAC, a compiled block in a CAB to set a dc voltage, comprises an FG OTA in a unity-gain follower configuration. $V_{in(+)}$ is connected to $V_{dd}$, and $V_{in(-)}$ is connected to $V_{out}$. $V_{fg(-)}$ is

$$V_{fg(-)} = V_{fg(+)} + Q_{inj}/C_T + V_{out} \cdot C/C_T$$  \hspace{1cm} (3)

where $Q_{inj}$ is the injected charge to the FG node and $C_T$ is the total capacitance of the FG. $V_{out}$ is

$$V_{out} = -\frac{Q_{inj}}{C_T} \left(1 + \frac{C}{C_T}\right)$$  \hspace{1cm} (4)

where $A_v$ is the gain of an FG OTA. A digital input dc voltage set by the user in the Xcos design is converted to a corresponding value of $Q_{inj}/C_T$ based on the $Q_{inj}/C_T$ - $V_{out}$ curve, calibrated through an external voltmeter for calibration.

**E. Step 5: $V_{T0}$ Mismatch Map**

A threshold voltage ($V_{T0}$) mismatch due to the indirect FG structure [5] and small device sizes causes errors in the analog computation. Especially, since FG switches are used for computation (e.g., VMM), as well as connections
between analog/digital elements, it is essential to measure and compensate for \( V_{T0} \) mismatches. Fig. 9 shows a \( V_{T0} \) mismatch characterization of FG devices. The indirect pFET’s drain is connected to the mismatch measurement block in Fig. 9(a). A compiled mismatch measurement block includes a reference FG device, a pFET, an FG OTA DAC, and an open-loop FG OTA in a CAB. The FG OTA’s gain, \( A_V \) (~10), is measured by an MITE ADC ahead of the mismatch characterization. The FG OTA DAC and the FG OTA’s input offset between (+) and (−) are set to have \( V_{\text{out}} \) at 1.25 V. Then, the \( V_{T0} \) mismatch, causing the difference between \( I_{\text{meas}} \) and \( I_{\text{meas(ref)}} \), is calculated from \( \Delta V_{\text{out}} = \Delta V_{T0} \times A_V/(A_V \cdot \kappa) \).

Fig. 9(b) shows an example of a mismatch table. The first and second elements are the row and column address of an FG device, respectively. Each \( V_{T0} \) mismatch value in the third column is directly added to \( V_{fg} \) of each FG device, which was calculated from the target current in the switch list and will be converted to a hex code. This allows the algorithm to compensate for \( \delta V_{T0} \) between the two transistors.

Fig. 9(c) shows a mismatch distribution and gray-scale map before and after mismatch compensation. Due to the small size \( (W/L = 1.8 \mu / 0.6 \mu) \) of the FG device, FG devices have a wide range of \( V_{T0} \) mismatches from −35 to 36 mV. The mismatch table compensates those \( V_{T0} \) mismatches; as a result, the standard deviation (\( \sigma \)) decreases from 14.3 to 1.04 mV. Table V shows that the \( V_{T0} \) mismatch compensation effectively decreases \( \sigma \) values in multiple chips.

A Boolean function \( \text{XOR} \) using a VMM and winner-takes-all (WTA), showing a nonlinear classification, is tested with the calibrated FG SoC FPAA system. Fig. 10(a) shows the circuit, weight information, input, and expected output logic. The \( \text{XOR} \), the third WTA’s output, functions as a combination of the input voltage \( (X_1, X_2) \) and weights. The WTA drives the output low when it has a higher current compared with the other WTAs. The input voltage by signal DACs to represent “1” and “0” is set to 2.5 and 2.3 V, respectively. The experiment includes the calibrated on-chip DACs and ADC as an input and output, as well as utilizes the characterized programming infrastructure, FG parameters, and the \( V_{T0} \) mismatch table.

Due to the \( V_{T0} \) mismatches on the weights and pFET biases, the \( \text{XOR} \) without a mismatch compensation results in an incorrect classification. Fig. 10(b) shows a measured hyperplane, where \( V_{\text{out}} \) corresponding to \( X_1 \) and \( X_2 \) is presented with gray-scaled values. It is clear that the \( V_{T0} \) mismatch compensation enables decision boundaries for \( \text{XOR} \) function resulting in “1” when \( X_1 \) and \( X_2 \) are “1”, “0” or “0”, “1.” Fig. 10(c) shows results of three different ICs for the \( \text{XOR} \) classification. Results without a mismatch compensation show failures due to the \( V_{T0} \) mismatches, where the expected output is “1010.” \( V_{\text{out}} \) with a mismatch compensation shows the expected \( \text{XOR} \) results in multiple chips.

| TABLE V |
|-----------------|----------|----------|----------|
| \( \sigma_{\text{start}} \) | Chip 1 | Chip 2 | Chip 3 |
| 14.3mV | 15.2mV | 13.1mV |
| \( \sigma_{\text{final}} \) | 1.04mV | 1.77mV | 1.21mV |

V. CONCLUSION AND DISCUSSION
A calibration flow for an integrated FG programming system for a large-scale FPAA has been presented. We focused
Fig. 9. Characterized mismatch table compensates $V_{T0}$ mismatches effectively. (a) Compiled block in a CAB measures $V_{T0}$ mismatch. After FG devices are programmed at a fixed current (e.g., 50 nA), the current difference between $I_{mea}$ and $I_{mea}(ref)$ is converted to a voltage by pFET, then amplified by FG OTA having a gain of $\sim 10$. A $V_{T0}$ mismatch value is calculated from the measured $V_{out}$. (b) In an example of a mismatch table, the first two elements represent the row and column address of FG devices. The third element indicates each $V_{T0}$ mismatch value. (c) It compares the results of the $V_{T0}$ mismatch compensation on 392 FG devices (14 rows $\times$ 28 columns) in a CAB. In the gray-scale map and mismatch distribution graph, a wide range of $V_{T0}$ mismatches ($\sigma = 14.3$ mV) due to the small size of FG pFETs are compensated by the mismatch map, resulting in $\sigma = 1.04$ mV.

Fig. 10. Nonlinear classifier is tested on multiple chips. (a) Boolean function $XOR$, as an example of a nonlinear classifier, is implemented with a VMM+WTA structure [21]. A combination of inputs and weights (W) determines the WTA’s output voltage, in which the winner has a low voltage (“1”). $V_{T0}$ mismatches on the VMM weights and FG pFETs for WTA bias currents ($I_{WTA}$) cause a malfunction. (b) $V_{T0}$ mismatch compensation integrated into the compilation of the FG FPAA system brings the decision boundary to the right operation range in the measured hyperplane. (c) $V_{out}$ with the $V_{T0}$ mismatch compensation shows the same results with the $XOR$ truth table in all three chips.

on characterizing the FG programming infrastructure and hot-electron injection parameters in the integrated SoC FPAA, calculating the EKV model parameters for the golden FETs, and calibrating the compiled DAC and ADC blocks that interfaces between the on-chip $\mu$P and compiled analog circuits in the array. $V_{T0}$ mismatches due to the indirect FG structure are characterized through a compiled mismatch measurement block. A compiled classifier implementing $XOR$ function using a VMM and WTA on different chips shows the effectiveness of the $V_{T0}$ mismatch-map compensation integrated into the compilation flow.

In our recent work, we have been focusing on an implementation of FG SoC FPAA ICs including an on-chip FG programming infrastructure and providing a high analog parameter density [3], developing an FG programming algorithm to achieve precise target currents [10], and providing a high-
level design tool supporting a graphical design environment and compiling it to necessary files (e.g., assembly program codes) [9]. The standardized and automated calibration method in the system, remained as the last piece of this puzzle, is required to enable users to design analog circuits without considering the device variation; even users with little exposure to an analog circuit and system design (e.g., users from the signal processing community) can design function blocks with abstracted blocks for a top-level design [22].

An iterative approach for measuring the input and output voltages of a VMM to find the $V_{TH}$ mismatch based on calculated output currents was implemented in [23]. However, the iterative approach requires new calibration routine for each specific application. A calibration flow to characterize hot-electron injection parameters in a mechanical usage monitoring the system employing FG devices was shown in [24]. A previous work [6] modeled FG devices’ mismatch and characterized some of the analog devices in a CAB, providing an inspiration for the fully implemented system-level automated calibration presented here. The proposed calibration method in this paper includes all necessary parts for the FG SoC FPAA system from characterization of the programming infrastructure, MOSFETs, threshold voltage mismatch, and FG devices to the compiled DAC and ADC blocks. $\mu$P and SRAM integrated into the SoC IC simplified the calibration scripts by allowing the use of compact and efficient assembly codes, which enabled calibration at a more complicated system level. Since the calibrated information is integrated into the compilation in the analog design flow, users can focus on more complicated applications (e.g., large neuromorphic systems [8]) as if they are designing digital circuits.

ACKNOWLEDGMENT
The authors would like to thank B. A. Minch for providing a source code extracting EKV parameters, which has been modified for our system.

REFERENCES


Sihwan Kim received the B.S. and M.S. degrees from the University of Tokyo, Tokyo, Japan, in 2006 and 2008, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Georgia Institute of Technology, Atlanta, GA, USA.

His current research interests include low-power analog integrated circuit design, programmable circuits and devices, and bio-inspired circuits.

Sahil Shah received the B.S. degree from Manipal University, Manipal, India, and the M.S. degree from Arizona State University, Tempe, AZ, USA. He is currently pursuing the Ph.D. degree in electrical engineering with the Georgia Institute of Technology, Atlanta, GA, USA.

His current research interests include the design of analog- and mixed-signal integrated circuits, embedded machine learning, and real-time signal processing with field-programmable analog arrays.

Jennifer Hasler (S’87–M’01–SM’04) received the B.S.E. and M.S. degrees in electrical engineering from Arizona State University, Tempe, AZ, USA, in 1991 and the Ph.D. degree in electrical and computer systems and neural systems from the California Institute of Technology, Pasadena, CA, USA, in 1997. She is currently a Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Her 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-to-digital signal processing, device physics related to submicrometer devices or floating-gate devices, and analog VLSI models of on-chip learning and sensory processing in neurobiology.