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 (SoC) Field Programmable Analog Array (FPAA). We focus on characterizing the FG programming infrastructure and hot-electron injection parameters, MOSFET parameters using the EKV model, and calibrating DACs and ADCs. Also, 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 non-linear classifier block comprising a vector-matrix-multiplier (VMM) and a winner-takes-all (WTA) on three different calibrated chips.**

***Index Terms*—Floating-Gate FPAA, Calibration, 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 SoC systems require a calibration (e.g., a clock speed, bad memory blocks, 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 (e.g., [1]) resulting from the fact that "not all transistors are created equal."[1] Typically an 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 article describes bringing analog computation towards 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 (FPAA), 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 (e.g., [4]).

Figure 1 illustrates the concept of enabling algorithms to be directly downloaded to a large number of FG analog programmable and configurable ICs using a single calibration flow. Our primary need for calibration is to account for the threshold voltage mismatch (V_{T0}) between two pFETs for indirect FG programming [5], where previous characterization initially shows V_{T0} mismatches between these devices [6].

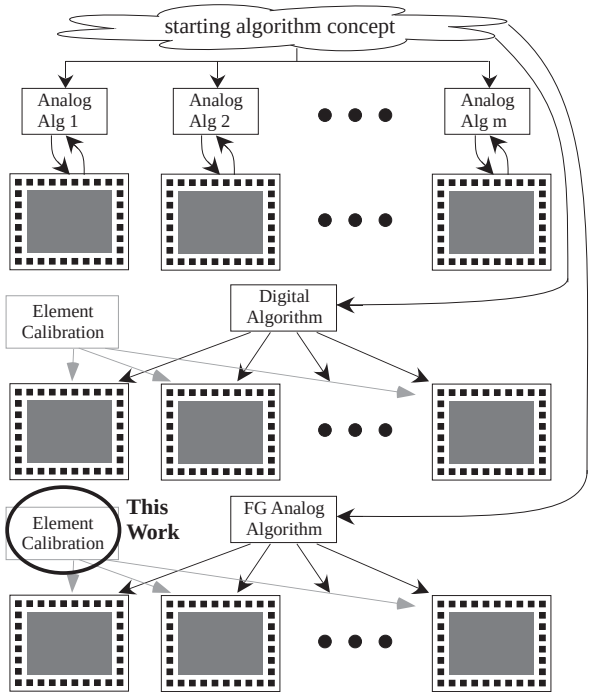


Fig. 1. Separation of calibration and algorithm enables the same algorithm implementations at similar performance levels in both digital and analog systems. Digital systems enable a single algorithm directly downloaded to a large number of ICs (m), however classical analog systems need each algorithm to be *tuned* for each particular application. The digital approach, especially a digital SoC system including μP, SRAM and analog components (e.g., a clock crystal, oscillator [7]) and providing several V_{dd} for low-power consumption, requires a calibration on a clock speed, bad memory blocks, and internal voltage regulators, as well as precision components due to the mismatches [8], whereas this process is independent of the algorithm. This work focuses on developing a single calibration flow to bridge the gap, enabling algorithms directly to be downloaded to (m) FG analog programmable and configurable ICs.

In the following sections, we will discuss our FG SoC FPAA architecture in Section I [3] [9] [10], compilation flow and the FG programming algorithm using Fowler-Nordheim tunneling and hot-electron injection in Section II, and the five steps of calibration flow and the results of our nonlinear classifier in multiple calibrated chips in Section III. Section IV includes the conclusion and discussion.

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]. Figure 2 shows the PCB and IC level

The authors are with the School of Electrical and Computer Engineering (ECE), Georgia Institute of Technology, Atlanta, GA 30332-250 USA (e-mail: jennifer.hasler@ece.gatech.edu).

[1] [2], Chapter 5, p. 72

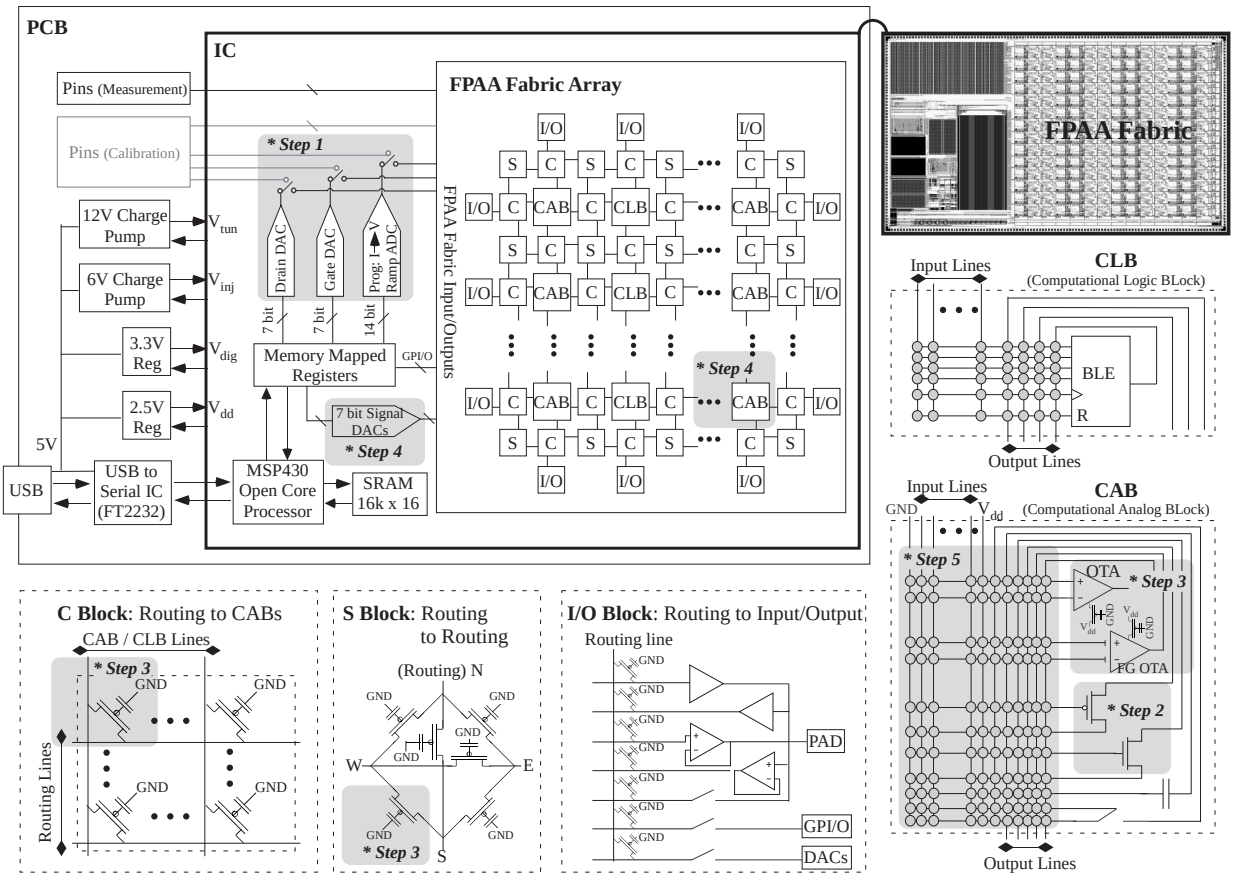


Fig. 2. The FG FPAA system interface between the on-chip μP and external devices (e.g., computer / tablet) is a USB, which provides the system power (5 V) as well. The PCB includes voltage regulators for the power supply (2.5 V / 3.3 V) to the IC, charge pumps to generate 6 V and 12 V for the injection and electron tunneling, and pins for a measurement or calibration. The IC consists of a μP, 16 k $\times$ 16 SRAM, an FPAA fabric array, and an FG program infrastructure comprised of a 7-bit gate DAC, a 7-bit drain DAC, an I-V converter, and a 14-bit ramp ADC. The FPAA fabric array is composed of Computational Analog Blocks (CAB), Computational Logic Blocks (CLB), Connection (C) blocks, Switch (S) blocks, and Input/Output (I/O) blocks. "$*$" indicates each calibration step in Fig. 4

architecture of the latest version of the FG FPAA family [3]. The IC comprises an FPAA fabric array, an FG programming infrastructure, a μP (open-source MSP430 [14]), and 16 k $\times$ 16 SRAM. The FG programming infrastructure includes a 7-bit gate DAC, a 7-bit drain DAC, a pFET diode I-V converter, and a 14-bit ramp ADC, interfacing with the μP through memory mapped registers.

The PCB consists of power components regulating 2.5 V / 3.3 V, charge pump units handling high voltages (6 V / 12 V), and Input/Output pins for external connection (to be used with voltage generators, voltmeters, ammeters, etc.). 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 (CAB), Computational Logic Blocks (CLB) and routing switches composed of Connection (C), Switch (S) and Input/Output (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 (OTA) with and without FG inputs, nFETs, pFETs, capacitors, and T-gates. Each CLB includes local routing switches with BLE lookup table circuits. FG switches can be be used for computation (e.g., VMM) as well as for connections between CAB/CLB/IO blocks.

III. Design Compilation and FG Programming

Figures 3a and 3b show 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 V_{T0} mismatch table, an input vector refers to a calibrated DAC table, 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. Figure 3c shows a program sequence from tunneling to precise injection, and

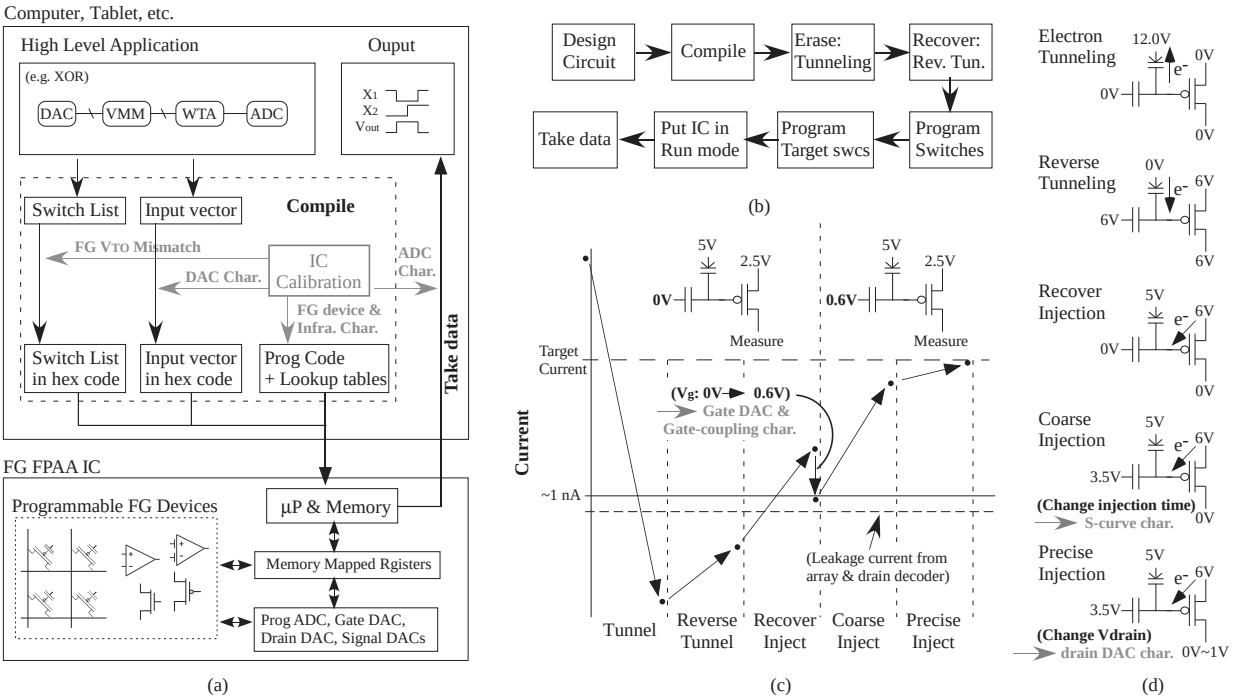


Fig. 3. The design and test flow includes the compilation and programming of FG devices. (a) The design compilation interfaces between high level application designs and the FG FPAA IC. A circuit designed by a user in XCOS is compiled to a switch list, input vectors, and program codes, which are transmitted to and executed by the IC. The calibrated IC information is integrated into the compilation process, including converting the measured data sent by the IC to real values (e.g., voltage). (b) The system employs electron tunneling to erase and hot electron injection to program FG devices. (c) The measured current at the end of the recover injection is set to 1 nA by using the FG's gate capacitive coupling, which is characterized in the calibration flow. (d) It shows the tunneling and injection conditions. Coarse injection, which modulates the pulse width at a fixed drain voltage (0 V), requires S-curve characterization for the pulse width table. Similarly, precise injection, which modulates the drain voltage at a fixed pulse width (10 μs), requires a 7-bit drain DAC characterization.

Fig. 3d 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 ($\sim$fA). 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 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 $I_{inj} \propto I_s e^{f(\Phi_{dc})}$, where I_s is the channel current and Φ_{dc} is the drain-to-channel potential. Q_{fg} (charge on the floating-gate) ($Q_{fg} = \int I_{inj} dt$) 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 pulse width 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 pulse width and controls the drain voltage for precise electron injection, requiring characterization of a 7-bit drain DAC.

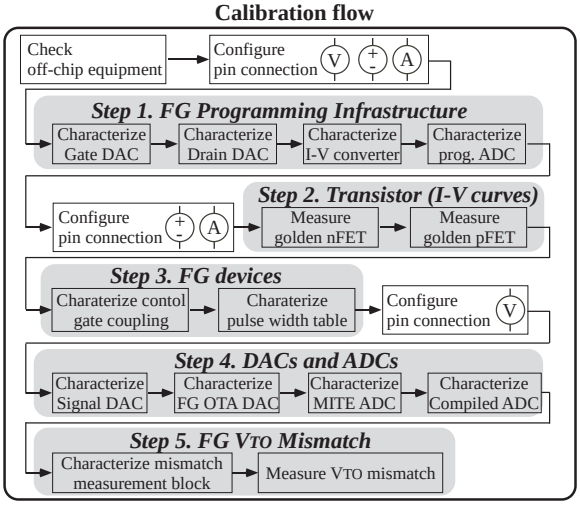


Fig. 4. Off-chip equipment (voltage generator, voltmeter, ammeter) is required for steps 1, 2, and 4, but the external measurement device is no longer necessary after the calibration. In particular, the ammeter, which is large and heavy compared to the FG SoC FPAA system, is not in use after step 2. 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 non-linear classifier results working in multiple calibrated chips. Off-chip equipment used for the

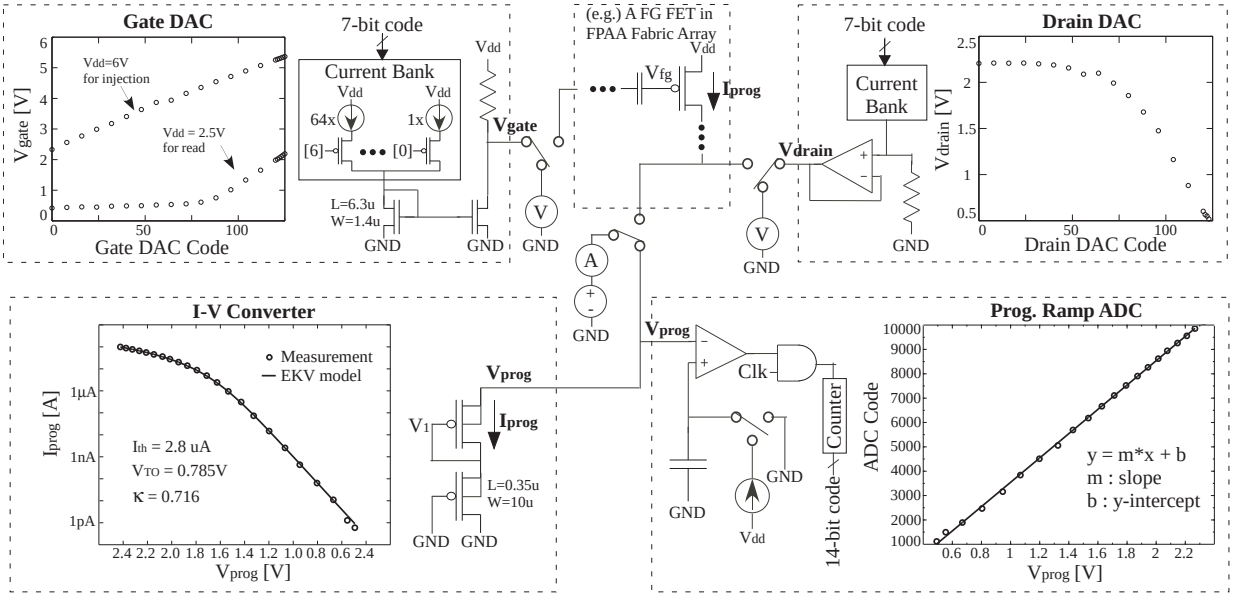


Fig. 5. A characterization of the on-chip FG programming infrastructure circuits is shown. The gate DAC which converts a 7-bit code to an output voltage through a current bank is measured by an external voltmeter. With two different supply voltages (V_{dd}) for the FG injection and current measurement, the gate DAC has the output voltage in roughly 2 V to 5 V with V_{dd} at 6 V and 0.6 V to 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 (κ, V_{T0}, I_{th}) and the slope (m) and y-intercept (b) on the ramp ADC of each chip are calculated.

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 calibration script communicates with those external devices through a USB interface.

A. Step1: Gate & Drain DACs, I-V Converter, and Ramp ADC

The characterization of the on-chip programming infrastructure in Fig. 5 is the first step of the FG SoC FPAA IC calibration. The gate of an FG device is controlled by a 7-bit gate DAC consisting of a current bank and a resistor with a current mirror, where the 7-bit code steers currents, and the mirrored current and resistor set the DAC output voltage. A current bank includes seven kinds of current sources and seven pFETs controlling the amount of the current based on the code. The gate DAC is calibrated through external voltmeter with two different supply voltages (V_{dd}), 6 V for injection and 2.5 V for current measurements. The output voltage is in a range from 2 V to 5 V with a V_{dd} of 6V and in a range from 0.6 V to 2 V with a V_{dd} of 2.5 V. The 7-bit drain DAC has a structure similar to the gate DAC, but the resistor is connected to ground without a current mirror and it has a buffer to drive the drain line. The drain DAC is also calibrated through an external voltmeter, which has an output voltage in the range of 0.5 V to 2.2 V.

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

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

where κ ("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, I_{th} is the threshold current. κ, V_{T0}, and I_{th} are calculated from the measured I_{prog}-V_{prog} curve.

A Ramp ADC, which interfaces with the μP, converts V_{prog} to a 14-bit code. The slope and y-intercept is calculated based on the 14-bit code - V_{prog} measurement. Table I shows programming infrastructure parameters in multiple chips.

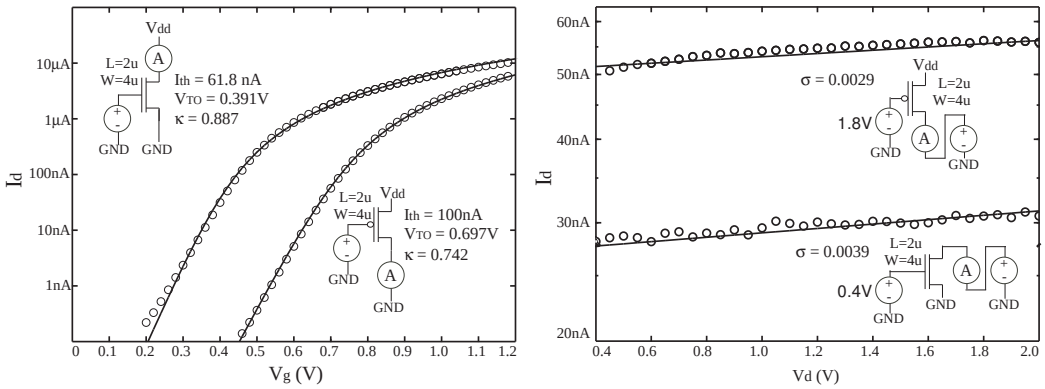
TABLE I
PROGRAMMING INFRASTRUCTURE PARAMETERS

		Chip 1	Chip 2	Chip 3
I-V converter	κ	0.716	0.707	0.699
	I_{th}	2.8μA	3.1μA	3.2μA
	V_{T0}	0.785V	0.847V	0.828V
Ramp ADC	m	4490	5709	5474
	b	-1445	-1991	-1679

B. Step2: 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 a MOS transistor model to illustrate a FET's behavior. The equation of nFET I_d in the EKV model is

$$\begin{aligned} I_d =& I_{th} \ln^2 \left(1 + e^{(\kappa(V_g - V_{T0}) - V_s + \sigma(V_d - V_s))/2U_T}\right) \\ & - I_{th} \ln^2 \left(1 + e^{(\kappa(V_g - V_{T0}) - V_d - \sigma(V_d - V_s))/2U_T}\right) \end{aligned} \qquad (2)$$



	sub V_{th}	above V_{th}
Ohmic	$I_{th}e^{\kappa(V_g-V_{T0})/U_T}\left(e^{-V_s/U_T}-e^{-V_d/U_T}\right)$	$\frac{I_{th}}{4U_T^2}\left((\kappa(V_g-V_{T0})-V_s)^2-(\kappa(V_g-V_{T0})-V_d)^2\right)$
Saturation	$I_{th}e^{(\kappa(V_g-V_{T0})-V_s+\sigma V_d)/U_T}$	$\frac{I_{th}}{4U_T^2}(\kappa(V_g-V_{T0})-V_s+\sigma V_d)^2$

Fig. 6. A golden set of nFET and pFET, compiled at a specified location in the FPAA fabric array of each chip, is modeled with EKV parameters (I_{th}, V_{T0}, κ, and σ). An ammeter is no longer required for the rest of the calibration steps or for data measurement in a user's design. V_{T0}, κ, and σ are calculated from the measured I_d–V_g and I_d–V_d data. It also shows the transistor equations of the ohmic/saturation current in the sub/above threshold region.

TABLE II
nFET, pFET EKV Parameters

		Chip 1	Chip 2	Chip 3
nFET	κ	0.887	0.781	0.856
	I_{th}	61.8nA	64.1nA	86.9nA
	V_{T0}	0.391V	0.390V	0.418V
	σ	0.0039	0.00049	0.0023
pFET	κ	0.742	0.772	0.723
	I_{th}	100nA	107nA	118.41nA
	V_{T0}	0.697V	0.714V	0.705V
	σ	0.0029	0.0022	0.0029

σ is U_T/V_A, where V_A it the Early voltage. (2) includes all equations of the ohmic/saturation current in the sub/above threshold region shown in Fig. 6.

Figure 6 shows EKV parameters (κ, I_{th}, V_{T0}, and σ) which are extracted from the measured I-V curves taken from a golden set, compiled at a fixed location in each chip, of the nFET and the pFET. Characterizing the golden nFET and pFET means one can always figure out the relationship between current and voltage, as well as calibrate between different devices. κ, I_{th}, and V_{T0} for nFET and pFET are calculated based on I_d-V_g curves sweeping V_g with a fixed V_d and V_s [19]. First, each starting value for V_{T0} and I_{th} is set to the x-axis intercept in a linear line extracted from $\sqrt{I_d}$ - V_g curve and twice the value of I_d when V_g is V_{T0} via a cubic-spline interpolation, respectively. Then, the optimal I_{th} to minimize the curvature of the EKV model inverse expression is found in the interval between one tenth and ten times the initial value of I_{th}, which results in κ and the final V_{T0}. σ for nFET and pFET is calculated from $\sqrt{I_d}$-V_d curves sweeping V_d with a fixed V_g and V_s. In each characterization, V_g and V_d are set by external voltage generators, and I_d is measured through an external ammeter. Table II shows measured nFET and pFET EKV parameters in multiple chips.

C. Step3: Gate coupling offset and Injection characterization

FG programming parameters are calibrated without any external equipment. Figure 7a shows the calibration of the gate capacitive coupling offset required for the recover injection in the target program. V_{out}, the output voltage of the two pFET diodes, is measured with V_g at 0 V and 0.6 V, while applying a 10 μs injection pulse with V_d at 0 V. $\kappa_{eff}(=\kappa C/C_T)$, which is proportional to ΔV_{out} measured at different V_g, decreases as V_{out} increases since the MOSFET depletion capacitance increases. The slope changes around the boundary of the sub- and above-threshold currents ($\sim$ 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.

TABLE III
Gate Coupling parameters

	FG	Chip 1	Chip 2	Chip 3
ΔV_{out} @1nA	SWC (Ind.)	0.190V	0.224V	0.243V
	SWC (Dir.)	0.205V	0.268V	0.256V
	OTA	0.226V	0.270V	0.282V
	FG OTA	0.317V	0.383V	0.388V
	MITE	0.358V	0.429V	0.426V

Figure 7b illustrates 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 towards a stable equilibrium, forms two linear lines crossing at the current of 2.1 μA on the V_{out}(final)-V_{out}(start) plot [10]. The pulse width table, which shows the number of injection pulses to reach V_{out}(final) from V_{out}(start), is calculated based on the S-curve measurement. Figure 7c 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 pulse width table for each FG device are calibrated respectively in each chip, shown in Table III and IV.

D. Step4: Signal DACs and Compiled DAC/ADC blocks

Figure 8 shows the calibration of DACs and ADCs, which provides a mixed-signal design environment for users and

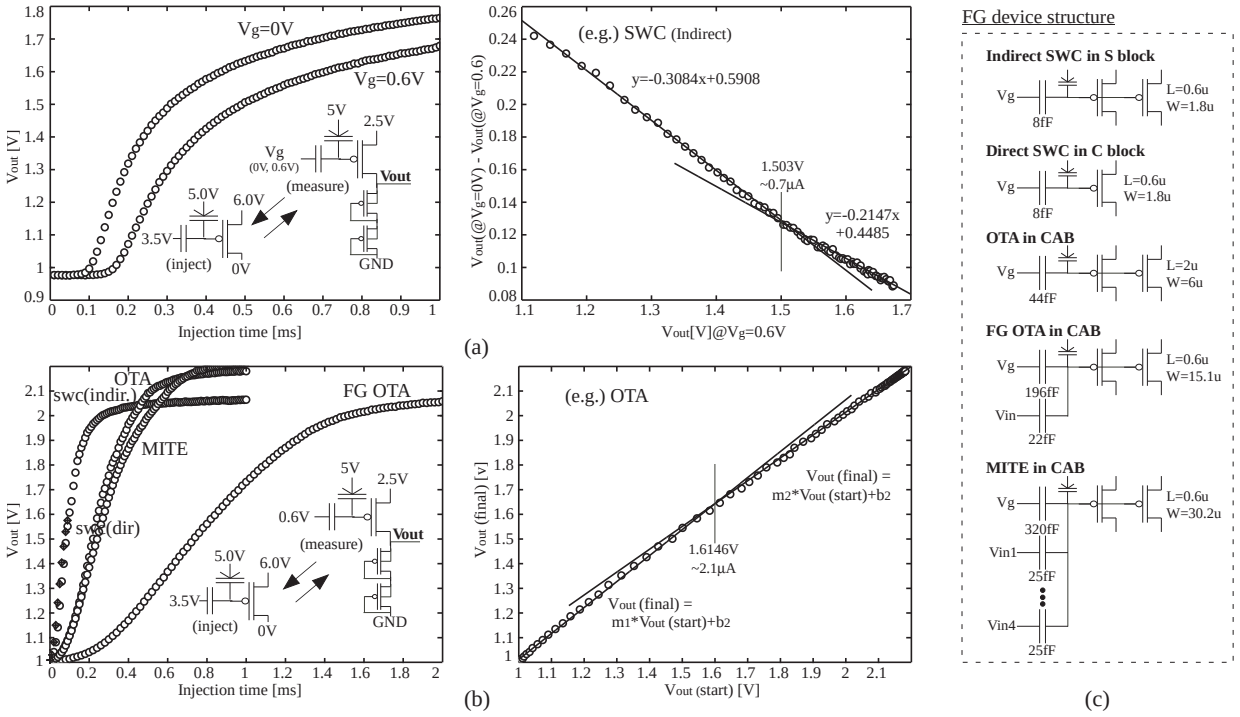


Fig. 7. FG devices require a characterization of the FG programming parameters. (a) Gate capacitive coupling offsets between V_{out} measured with V_g at 0 V and 0.6 V in the injection and current-measurement loop are calculated and set for each chip's recover injection. As V_{out} increases, the offset decreases due to the increase of the MOSFET depletion capacitor. (b) S-curves are measured for the pulse width table in the coarse program. The injection-measurement loop starts from the V_{out} corresponding to 1 nA in current. The pulse width 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 pulse width table for each FG device is measured in an automated calibration script.

eliminates the need for external equipment for measurement. Signal DACs, consisting of a current bank and a resistor, interface with the μP through memory mapped registers. Signal DACs could be used as arbitrary waveform generators by the user. The input is compiled as a vector on the SRAM. The run-mode assembly code sends the input vector uploaded on SRAM to a memory mapped register at a given frequency. A signal DAC is calibrated by connecting V_{out} to an external voltmeter through an I/O block in the array.

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}, $V_{in}(-)$ is connected to V_{out}. $V_{fg}(-)$ is

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

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

$$V_{out} = -\frac{Q_{inj}}{C_T}/(\frac{1}{A_v} + \frac{C}{C_T}) \qquad (4)$$

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.

A MITE ADC is implemented with a Multiple-Input Translinear Element (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.

A compiled Ramp ADC includes two FG pFETs, a capacitor, an nFET, and an OTA in a CAB. The μP resets the ramp ADC by turning the nFET on and counts clock cycles until the output of the OTA is flipped from V_{dd} to gnd. The slope of the ADC depends on the capacitor's size and the bias current of the FG pFETs. The compiled Ramp ADC has an 8-bit code.

E. Step5: 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. Figure 9 shows a V_{T0} mismatch characterization of FG devices. The indirect pFET's drain is connected to the mismatch measurement block in Fig. 9a. 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 ($\sim$10), is measured by a 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_{out} at 1.25 V. Then, the V_{T0} mismatch, causing the difference between I_{meas} and I_{meas}(ref), is calculated from ΔV_{out}. ΔV_{T0} is

$$\Delta V_{T0} = \frac{\Delta V_{out}}{A_v \cdot \kappa} \qquad (5)$$

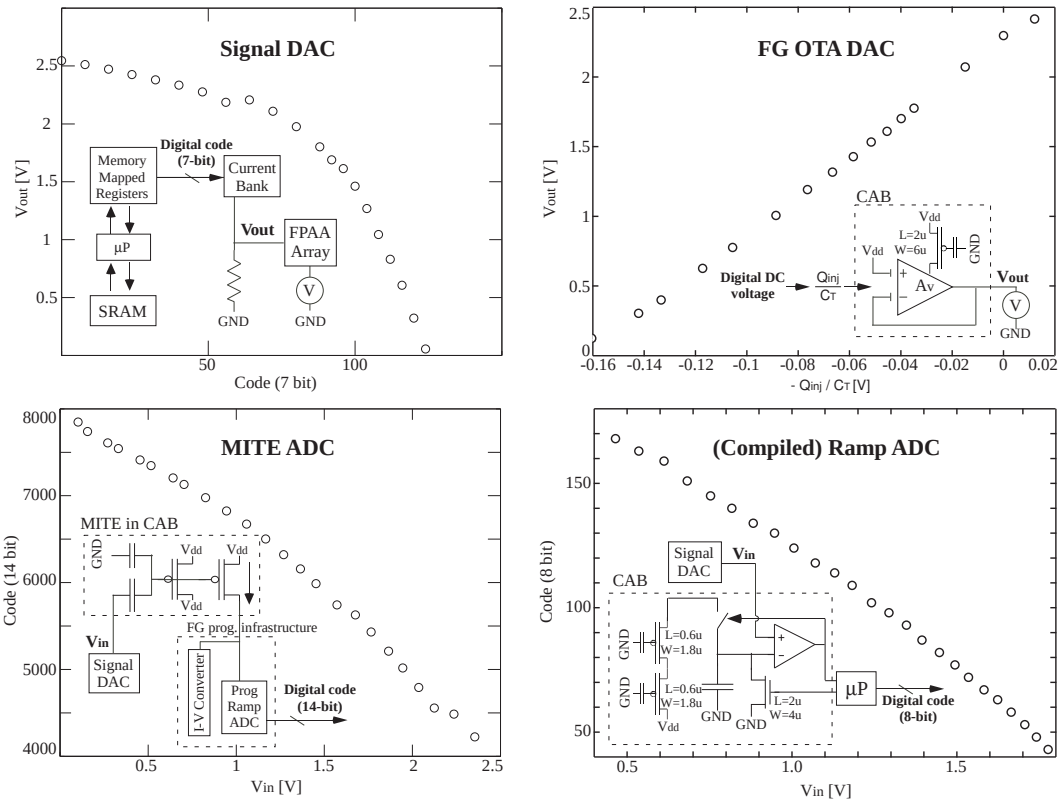


Fig. 8. A signal DAC is a dedicated circuit in the IC, but other DAC and ADCs are compiled blocks in CABs. **Signal DAC:** V_{out} of 16 7-bit on-chip signal DACs are calibrated by an external voltmeter through I/O blocks in the array. **FG OTA DAC:** A feedback FG OTA in a CAB operates as a DC DAC. V_{out} is set by Q_{inj}, the offset of injected charge on two input FG nodes. The FG OTA DAC block is calibrated through an external voltmeter. **MITE ADC:** V_{in} of a Multiple-Input Translinear Element (MITE) device in a CAB couples V_{fg}, which is measured by a pFET diode I-V converter and a 14-bit ramp ADC in the program infrastructure. A calibrated signal DAC is used to apply V_{in}. **(Compiled) Ramp ADC:** A compiled ramp ADC block including two FG pFETs, an nFET, a capacitor, an OTA in a CAB converts V_{in} to 8-bit codes, interacting with the μP through GPIO.

TABLE IV
PULSE WIDTH PARAMETERS

	FG	Chip 1	Chip 2	Chip 3
m_1/b_1	SWC (Ind.)	0.953/0.114	0.945/0.121	0.894/0.228
	SWC (Dir.)	0.880/0.200	0.873/0.199	0.805/0.318
	OTA	1.060/-0.050	1.045/-0.036	1.026/-0.015
	FG OTA	1.081/-0.077	1.029/-0.009	1.001/0.032
	MITE	1.049/-0.038	1.021/-0.003	1.007/0.0184
m_2/b_2	SWC (Ind.)	0.930/0.145	0.938/0.121	0.947/0.111
	OTA	0.941/0.130	0.978/0.047	0.964/0.076
	FG OTA	0.973/0.059	0.944/0.117	0.924/0.166
	MITE	0.959/0.093	0.965/0.077	0.957/0.095

TABLE V
MISMATCH MAP

	Chip 1	Chip 2	Chip 3
σ_{start}	14.3mV	15.2mV	13.1mV
σ_{final}	1.04mV	1.77mV	1.21mV

Figure 9b 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 δV_{T0} between the two transistors.

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

A boolean function XOR using a VMM and WTA, showing a non-linear classification, is tested with the calibrated FG SoC FPAA system. Figure 10a shows the circuit, weight information, input, and expected output logic. The 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 to the other WTAs. The input voltage by signal DACs to represent "1" and "0" is set to 2.5V and 2.3V 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 XOR without a mismatch compensation results in an incorrect classification. Figure 10b shows a measured hyperplane, where V_{out} corresponding to X_1 and X_2 is presented with grayscaled values. It is clear that the V_{T0} mismatch compensation enables decision boundaries for XOR function resulting in "1" when X_1 and X_2 is "1", "0" or "0", "1." Figure 10c shows results of three different ICs for the XOR classification. Results without a mismatch compensation shows failures due to the V_{T0} mismatches, where the expected

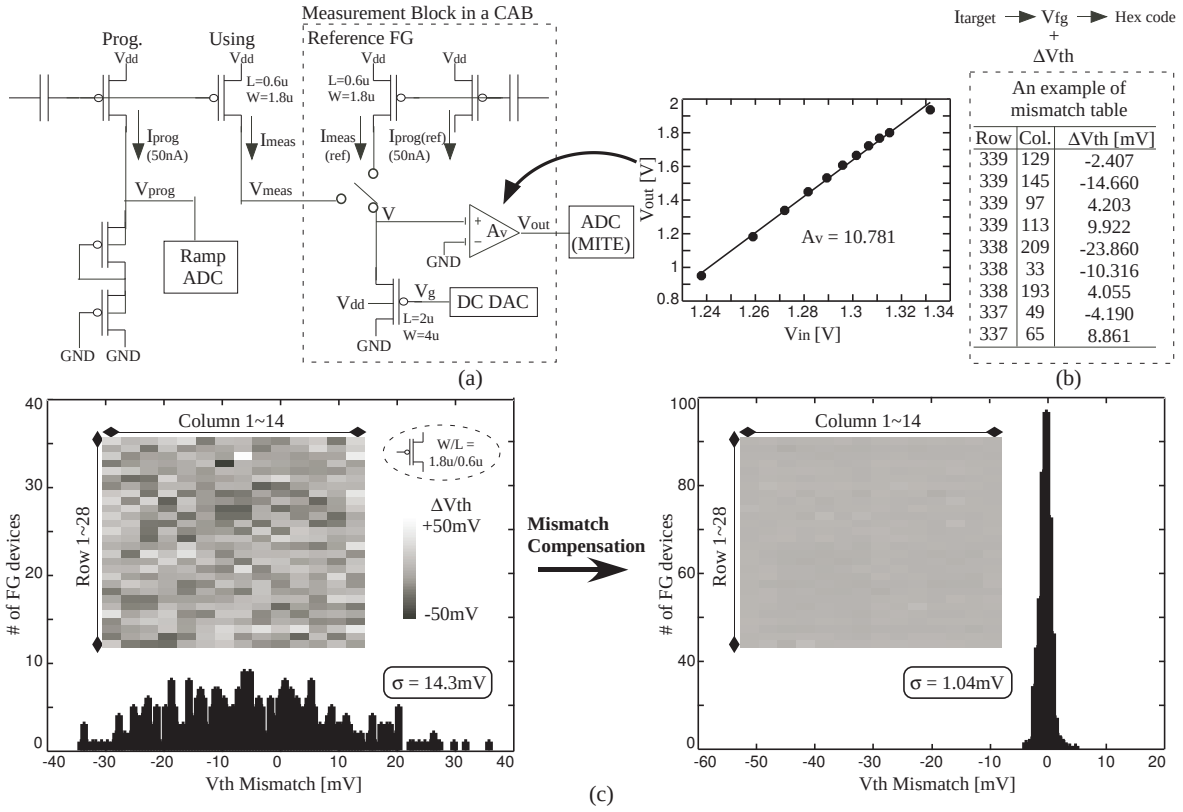


Fig. 9. The characterized mismatch table compensates V_{T0} mismatches effectively. (a) A compiled block in a CAB measures V_{T0} mismatch. After FG devices are programmed at a fixed current (e.g., 50 nA), the current difference between I_{meas} and $I_{meas}(ref)$ is converted to a voltage by pFET, then amplified by FG OTA having a gain of ∼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 X 28 columns) in a CAB. In the grayscale map and mismatch distribution graph, a wide range of V_{T0} mismatches (σ = 14.3 mV) due to the small size of FG pFETs are compensated by the mismatch map, resulting in σ = 1.04 mV.

output is "1010". The V_{out} with a mismatch compensation shows the expected XOR results in multiple chips.

V. Summary and Discussion

A calibration flow for an integrated FG programming system for a large-scale Field Programmable Analog Array (FPAA) has been presented. We focused 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, calibrating the compiled DAC and ADC blocks that interfaces between the on-chip μP 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_{T0} 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. The μP 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

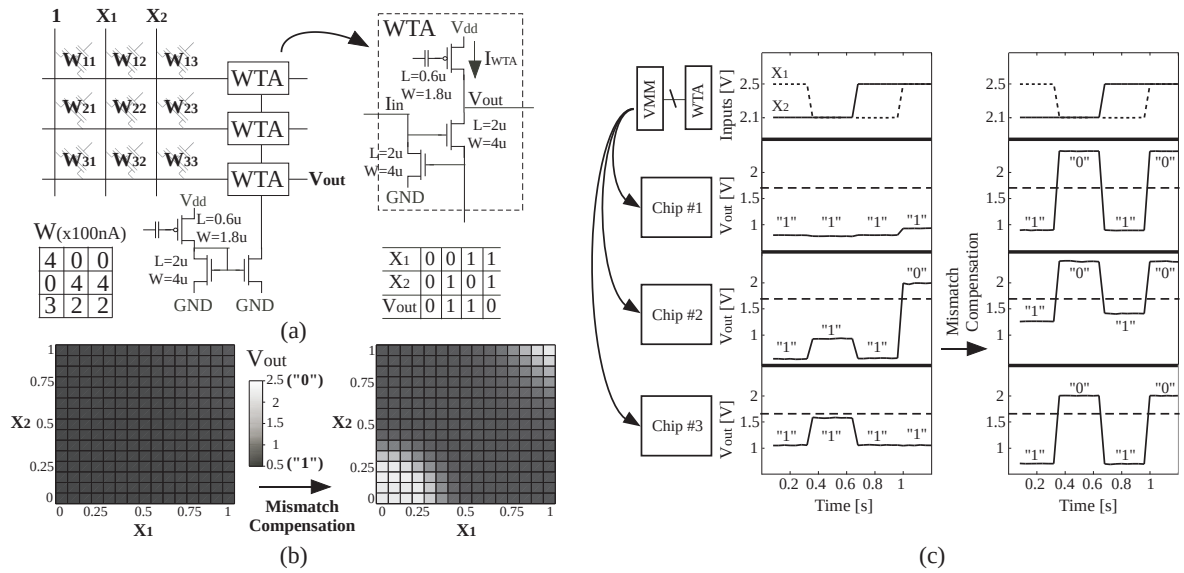


Fig. 10. A nonlinear classifier is tested on multiple chips. (a) A boolean function XOR, as an example of a nonlinear classifier, is implemented with a VMM+WTA structure [21]. A combination of inputs and Weights (W) determines the WTAs' output voltage, in which the winner has a low voltage ("1"). V_{T0} mismatches on the VMM weights and FG pFETs for WTA bias currents (I_{WTA}) cause a malfunction. (b) The 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.

[8]) as if they are designing digital circuits.

VI. Acknowledgment

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

References

[1] V.Srinivasan, D. W.Graham, and P.Hasler, "Floating-gates transistors for precision analog circuit design: an overview," in *48th Midwest Symposium on Circuits and Systems, 2005.*, Aug 2005, pp. 71–74 Vol. 1.

[2] C.Mead, *Analog VLSI and Neural Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[3] S.George *et al.*, "A programmable and configurable mixed-mode FPAA SoC," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2253–2261, June 2016.

[4] P.Hasler, B. A.Minch, and C.Diorio, "Adaptive circuits using pFET floating-gate devices," in *Advanced Research in VLSI, 1999. Proceedings. 20th Anniversary Conference on*, Mar 1999, pp. 215–229.

[5] D. W.Graham *et al.*, "Indirect programming of floating-gate transistors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 5, pp. 951–963, May 2007.

[6] S.Shapero and P.Hasler, "Mismatch characterization and calibration for accurate and automated analog design," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 3, pp. 548–556, March 2013.

[7] T.Wu, K.Mayaram, and U. K.Moon, "An on-chip calibration technique for reducing supply voltage sensitivity in ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 4, pp. 775–783, April 2007.

[8] J.Hasler and H.Marr, "Finding a roadmap to achieve large neuromorphic hardware systems," *Frontiers in Neuroscience*, vol. 7, p. 118, 2013.

[9] M.Collins, J.Hasler, and S.George, "An open-source tool set enabling analog-digital-software co-design," *Journal of Low Power Electronics and Applications*, vol. 6, no. 1, p. 3, 2016.

[10] S.Kim, J.Hasler, and S.George, "Integrated floating-gate programming environment for system-level ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2244–2252, June 2016.

[11] G. E. R.Cowan, R. C.Melville, and Y. P.Tsividis, "A VLSI analog computer/digital computer accelerator," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 42–53, Jan 2006.

[12] J.Becker *et al.*, "A field-programmable analog array of 55 digitally tunable OTAs in a hexagonal lattice," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2759–2768, Dec 2008.

[13] S.Shah *et al.*, "A proof-of-concept classifier for acoustic signals from the knee joint on a fpaa," in *2016 IEEE SENSORS*, Oct 2016, pp. 1–3.

[14] *OpenMSP430 Project: Open Core MSP430, 2015, [online] Available: online http://opencores.org/projectopenmsp430, accessed Nov.1. 2015.*

[15] P.Hasler *et al.*, "Impact ionization and hot-electron injection derived consistently from Boltzmann transport," *VLSI Design*, vol. 8, no. 1-4, pp. 454–461, 1998.

[16] P.Hasler, A.Basu, and S.Kozil, "Above threshold pFET injection modeling intended for programming floating-gate systems," in *2007 IEEE International Symposium on Circuits and Systems*, May 2007, pp. 1557–1560.

[17] C. C.Enz, F.Krummenacher, and E. A.Vittoz, "An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications," *Analog integrated circuits and signal processing*, vol. 8, no. 1, pp. 83–114, 1995.

[18] A.Low and P.Hasler, "Cadence-based simulation of floating-gate circuits using the EKV model," in *Circuits and Systems, 1999. 42nd Midwest Symposium on*, vol. 1, 1999, pp. 141–144 vol. 1.

[19] *Available: online http://madvlsi.olin.edu, accessed May.24. 2017.*

[20] B. A.Minch, P.Hasler, and C.Diorio, "The multiple-input translinear element: a versatile circuit element," in *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, vol. 1, May 1998, pp. 527–530 vol.1.

[21] S.Ramakrishnan and J.Hasler, "Vector-matrix multiply and winner-take-all as an analog classifier," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 353–361, Feb 2014.

[22] C. R.Schlottmann and J.Hasler, "High-level modeling of analog computational elements for signal processing applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 1945–1953, Sept 2014.

[23] S.Suh *et al.*, "Low-power discrete fourier transform for OFDM: A programmable analog approach," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 2, pp. 290–298, Feb 2011.

[24] C.Huang, N.Lajnef, and S.Chakrabartty, "Calibration and characterization of self-powered floating-gate usage monitor with single electron per second operational limit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 3, pp. 556–567, March 2010.

Sihwan Kim photo and bio not available during submission

Sahil Shah photo and bio not available during submission

Jennifer Hasler photo and bio not available during submission