Floating-Gate FPAA Calibration for Analog System Design and Built-In Self Test

Sihwan Kim, Sahil Shah, Jennifer Hasler
Georgia Institute of Technology, Atlanta, US
E-mail: skim819, sshah389@gatech.edu, jennifer.hasler@ece.gatech.edu

***Abstract*—We present a calibration flow for a large-scale Floating-Gate (FG) System-on-Chip (SoC) Field Programmable Analog Array (FPAA) to enable analog system design and built-in self test. We focus on calibration of the FG programming infrastructure, Digital-Analog Converters (DAC) and Analog-Digital Converters (ADC), as well as characterization of hot-electron injection parameters. This paper shows the results of a compiled Winner-Take-All (WTA) circuit on three different calibrated chips.**

***Keywords*—Floating-Gate FPAA, Calibration, Mismatch.**

I. Digital System and FG FPAA System

Digital systems have implemented a wide range of applications based on the fact that an algorithm by a digital system programmer is compiled and downloaded to any number of digital ICs. Calibration of digital IC elements (e.g. clock generator, memory reference voltages, etc.) is a key process enabling this simple algorithm flow, programmers kept away from digital IC variation issues can expect the same performance in all ICs.

Floating-Gate (FG) Field Programmable Analog Arrays (FPAA) group has been working to apply the same concept to analog systems, where an algorithm is compiled and downloaded to any number of FPAA ICs, as well as built-in self test provides the result of the application without using external measurement equipment. Multiple generations of large-scale Reconfigurable Analog Signal Processor (RASP) ICs including on-chip FG programming infrastructure and providing high analog parameter density have been developed [1], an FG programming algorithm to precise target currents has been presented [2]. We provide a high-level design tool [3] to support a graphical design environment and compile it to necessary files (e.g. assembly program codes). Calibration of FPAA ICs remains as the last piece of this puzzle and a key to enable the simple algorithm flow, which helps analog system programmers to tackle more complicated applications.

Figure 1 illustrates the concept of enabling algorithms to be directly downloaded to a large number of ICs in digital systems and FG FPAA systems. The purpose of the present work is to develop a calibration flow of necessary elements for on-chip FG programming and built-in self test, as well as to integrate calibrated information to the compilation flow.

In the following sections, we will discuss FG SoC FPAA architecture and compilation flow in Section II, the FG programming infrastructure calibration and FG injection parameter characterization in Section III, the calibration of DACs and ADCs in Section IV, and the WTA circuit experiments on three calibrated FPAA ICs in Section V. Section VI includes the conclusion.

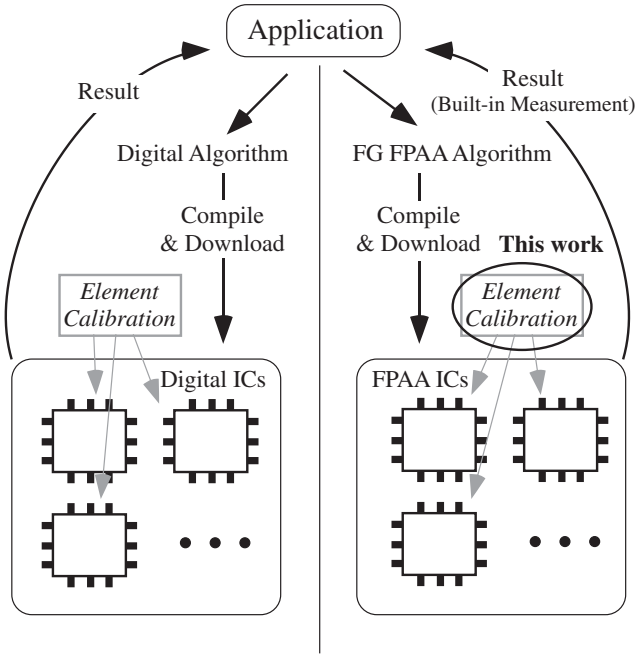

Fig. 1. Algorithm implementation in digital system and Floating-Gate (FG) FPAA system. Calibration of digital IC elements (e.g. clock generator) enables a single digital algorithm for an application to be compiled and downloaded to any number of ICs as well as same results from them. The same concept is applied in the FG FPAA system, where an analog system for an application is designed and tested without any external measuring equipment. This work focuses on the necessary elements' calibration for programming FG devices and built-in self test.

II. Floating-Gate SoC FPAA Architecture and Design Compilation

The FG SoC FPAA system enables complicated applications (e.g. speech recognition) by integrating programming infrastructure onto a chip and increasing parameter density and area efficiency [1], [4], [5]. In the latest version of the FG FPAA family (Fig. 2), the FPAA IC consists of μP (open-source MSP430 [6]), 16 k $\times$ 16 SRAM, FPAA fabric array, and FG programming infrastructure including 7-bit gate DAC, 7-bit drain DAC, I-V converter, and 14-bit ramp ADC. μP interfaces with ADCs, DACs, and FPAA array through memory mapped registers in the FPAA IC. The FPAA array comprises several functional blocks. Connection (C) and Switch (S) blocks consist of routing FG switches, Input / Output (I/O)

978-1-4673-6853-7/17/$31.00 ©2017 IEEE

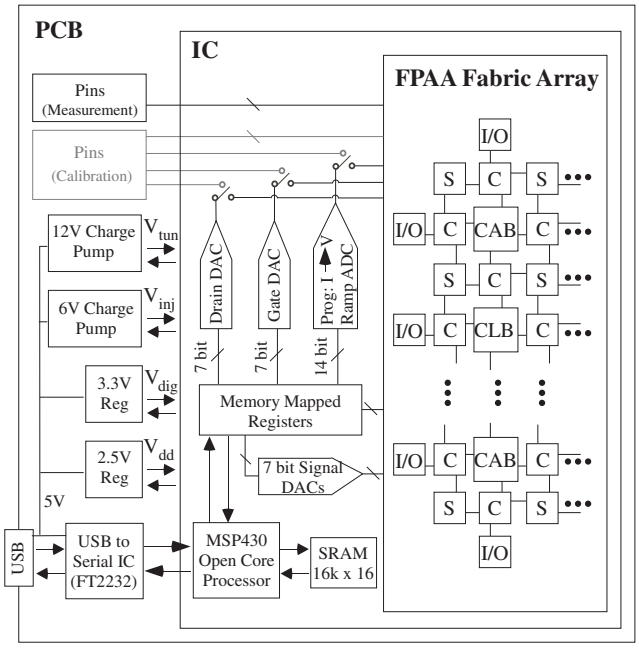

Fig. 2. The PCB and FG FPAA IC architecture [1]. USB connection provides the system power (5 V) as well as the interface between external devices (e.g. computer / tablet) and on-chip μP. The FG FPAA IC includes a μP, 16 k $\times$ 16 SRAM, FG program infrastructure, and an FPAA fabric array comprised of Computational Analog Blocks (CAB), Computational Logic Blocks (CLB), Connection (C) blocks, Switch (S) blocks, and Input/Output (I/O) blocks. FG program infrastructure, composed of a 7-bit gate DAC, a 7-bit drain DAC, an I-V converter, and a 14-bit ramp ADC, is calibrated for each IC. Power supply (2.5 V / 3.3 V) regulators, injection (6 V) and electron tunneling (12 V) charge pumps, pins for measurement or calibration are on PCB.

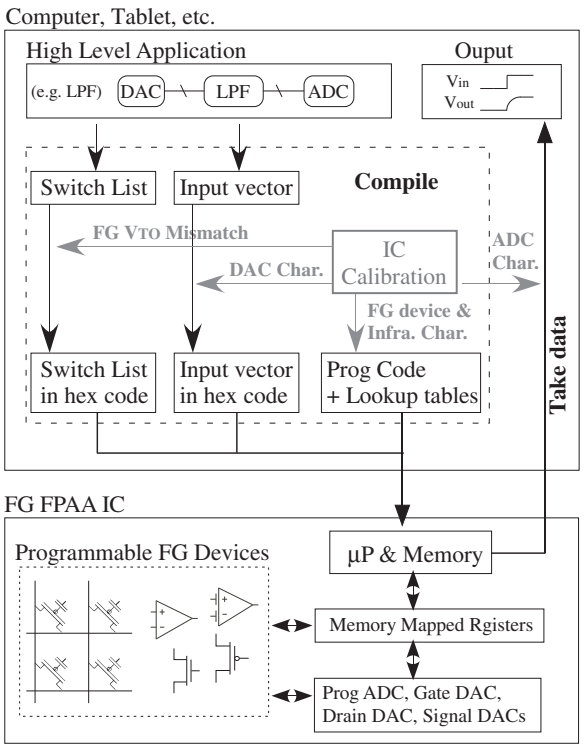

Fig. 3. Analog system design and test flow. High level application designed in XCOS is compiled to switch list, input vectors, and program codes, then FG FPAA IC measures and transmits the output after programming switches. Compilation integrates each chip's calibration information, which also converts the measured output data into voltage values.

blocks consist of FG switches with analog / digital buffers. Computational Analog Block (CAB) and Computational Logic Block (CLB) include analog and logic elements with local routing switches, respectively. Power components and high voltage charge pumps remain on Printed Circuit Board (PCB). External pins on PCB allow the direct access to the array as well as measurement of on-chip programming infrastructure for calibration.

Figure 3 shows analog system design and test flow in the FG FPAA system. A high-level application designed by a user in Scilab/Xcos (open-source programs similar to MATLAB/Simulink) [3] is compiled to necessary files to be transmitted to the IC. Each chip's calibration information is integrated into this compilation; switch list, input vector, and program files look up FG V_{T0} mismatch table, calibrated DAC table, and FG parameters and program infrastructure characterization table, respectively. Data measured on the IC and sent back to the user is converted to voltage values based on the ADC characterization table. In the next sections, we focus on calibration of the FG programming infrastructure, FG parameters, and ADCs / DACs in the system. V_{T0} mismatch calibration is beyond the scope of this paper, it will be discussed elsewhere.

III. Calibration of Programming Infrastructure and FG Injection Parameters

We use the electron tunneling effect to erase and hot-electron injection to program FG devices. To program some of the FG devices to target currents, a precise FG programming algorithm has been presented in [2], which has a sequence of recover injection, coarse injection, and fine injection after tunneling and reverse tunneling. The current of the FG device results in 1 nA, some current close to the target current, and exact target current after recover, coarse, and fine injection, respectively.

Programming infrastructure in Fig. 4 provides necessary injection voltages for each step through 7-bit gate and 7-bit drain DACs, as well as measures the current of the FG device through I-V converter and 14-bit ramp ADC. The gate DAC consists of a 7-bit code current bank and a resistor with a current mirror, setting the DAC output voltage resulting in a range from 2 V to 5 V in injection (V_{dd}= 6 V) and in a range from 0.6 V to 2 V in current measurement (V_{dd}= 2.5 V). The drain DAC, including a 7-bit code current bank, a resistor, and a buffer, drives the drain line of FG device in injection. The gate and drain DACs are calibrated through an external voltmeter and modeled with poly-fit functions.

Two body-source connected pFET diodes, converting the current of the FG device to voltage, are characterized by applying external voltage and measuring current through an ammeter. Assuming that the two pFET diodes are matched with FG device, the relationship between FG node voltage (V_{fg}) and drain current (I_{prog}), based on the EKV model [7], is given by (1) [2].

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

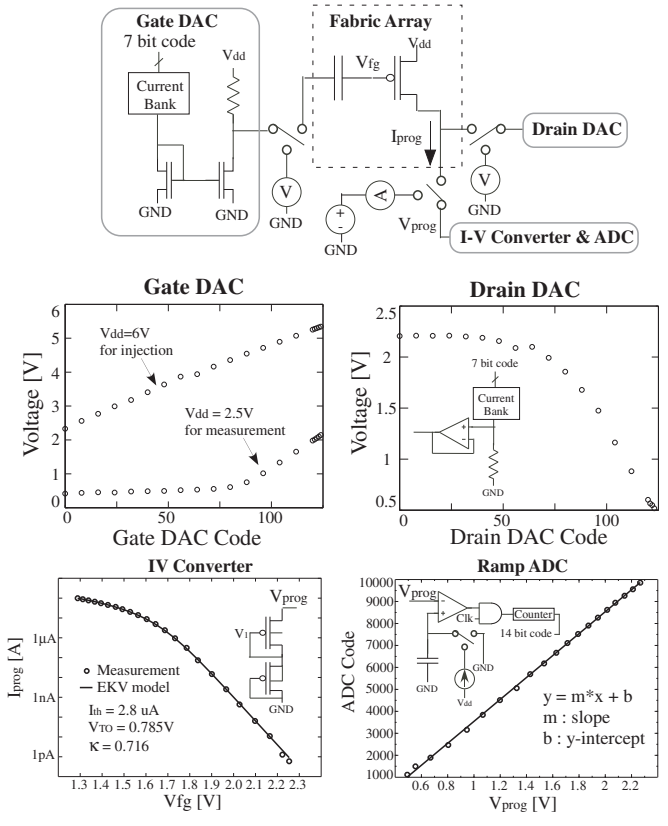

Fig. 4. Calibration of on-chip FG programming infrastructure circuits connected to FG devices. Injecting electrons to FG and measuring current utilize 7-bit gate DAC, 7-bit drain DAC, I-V converter composed of body-source connected two pFET diodes, and a 14-bit dedicated ramp ADC. An automated script calibrates the infrastructure circuits through usb-interfaced external equipment such as voltage generator / ammeter / voltmeter.

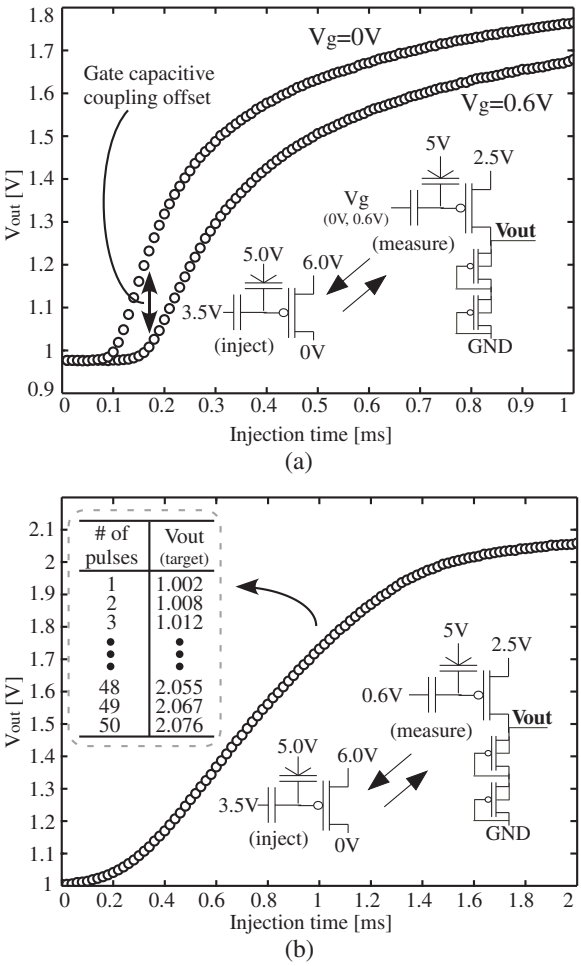

Fig. 5. Characterization of FG parameters. (a) Gate capacitive coupling offset for recover injection. The offset between V_{out} with two different gate voltages is measured in the loop of injection and current-measurement. (b) S-curve measurement for coarse injection. A pulse width table, indicating necessary number of pulses to reach a target current, is characterized by measuring each current after each unit time injection.

where I_{th} is threshold current, V_{T0} is threshold voltage, U_T is thermal voltage, and κ ("kappa") is the coupling of the gate to the surface potential. A ramp ADC on the drain of the FG device, converting drain voltage (V_{prog}) to 14-bit hex code, is calibrated through external voltage generator. The output hex values have a linear relationship with V_{prog}.

Due to the leakage from the array and the drain decoder (hundreds of pA), the current of FG device during recover injection is measured with V_g at 0 V, which is different than during the operation mode V_g, 0.6 V. Figure 5a shows this gate capacitive coupling effect of the FG device is measured and calculated in the loop of injection and current measurement with two different V_gs. Course injection code refers a pulse width table to calculate the number of injection pulse units (10 μs) to reach the target current. Figure 5b shows the characterization of the table. An automated script characterizes these FG parameters without connecting external devices and saves the tables under dedicated folders to be integrated in the compilation.

IV. Characterization of DACs and ADCs

In Fig. 6, we introduce the calibration of two basic types of DACs and ADCs provided as a default design option in the FG FPAA system, although it is allowed for users to add any different type of compiled DAC and ADC blocks. A signal DAC, a dedicated circuit comprising a 7-bit current bank and a resistor, generates arbitrary waveforms. Once the input vector of 16 signal DACs is defined by a user and compiled into a hex file, it is transmitted to the SRAM. The assembly code for the test mode sets signal DAC output voltage through memory mapped registers at a given frequency, which is also defined by a user.

FG Operational Transconductance Amplifier (OTA) DAC, setting a DC voltage, is a compiled block in a CAB, where FG OTA is in a source follower configuration; $V_{in}(-)$ is connected to V_{out}, $V_{in}(+)$ is connected to V_{dd}. V_{out} is

$$V_{out} = -\frac{Q_{inj}}{C_T}/(\frac{1}{A_v} + \frac{C}{C_T}) \quad (2)$$

where Q_{inj} is the injected charge to the FG node, A_v is the gain of a FG OTA, C_T is total capacitance of the FG. Q_{inj}, controlled by target current ratio of two inputs, sets V_{out}. Signal DACs and FG OTA DAC are connected to external voltmeter through a I/O block in the array and calibrated by an automated script.

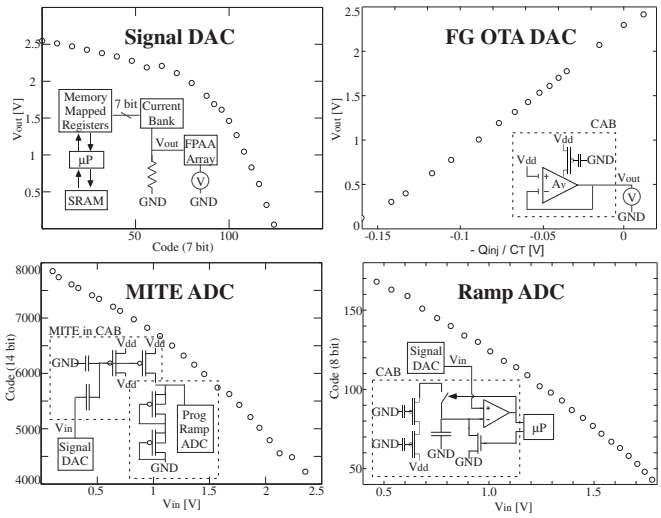

Fig. 6. Calibration of DACs and ADCs on FG FPAA IC. The FG FPAA system provides 16 7-bit signal DACs dedicated in the IC and interfacing with μP, a compiled FG OTA DAC, a compiled MITE ADC, and a compiled ramp ADC. DACs measured by external voltmeters are used in ADC calibration to minimize the use of external devices.

MITE ADC converts input voltage to 14-bit digital code by using Multiple-Input Translinear Element (MITE) block in a CAB and the FG programming infrastructure. V_{in}, connected to the gate of MITE, couples the surface potential, which is measured by the I-V converter and 14-bit programming ramp ADC. A compiled ramp ADC consists of an OTA, nFET, capacitor, and two FG pFETs in a CAB. The capacitor is charged by the bias current of the FG pFETs and discharged by nFET. μP, connected to the nFET's gate through General Purpose I/O (GPIO), counts clock cycles. When the output of OTA is flipped from V_{dd} to gnd, μP stops counting and stores the output of 8-bit hex value. MITE ADC and ramp ADC are calibrated by connecting to a calibrated signal DAC through the FPAA array in an automated script.

V. Measurement on Calibrated FPAA ICs

Figure 7 shows an example of Winner-Take-All (WTA) circuit experiment and on-chip test results. The example consists of signal DACs, FG devices converting input voltage to current, WTA circuit with FG biased current mirror, and MITE ADC with GPIO. In three different FPAA ICs calibrated with the presented way in this paper, the result shows expected outputs of WTA, which has lower output voltage where the input current is higher than another one.

VI. Conclusion

A calibration flow enabling an analog algorithm implementation with built-in self test in multiple FG FPAA ICs has been presented. We focused on calibration of infrastructure to program FG devices, characterization of FG injection parameters, and the calibration of two types of basic DACs and ADCs. We showed the results of WTA circuit in three different FPAA ICs. Because the compilation of the FG FPAA analog system integrates the calibrated information, analog system designers can pursue more complicated applications, which have previously been possible only in digital systems.

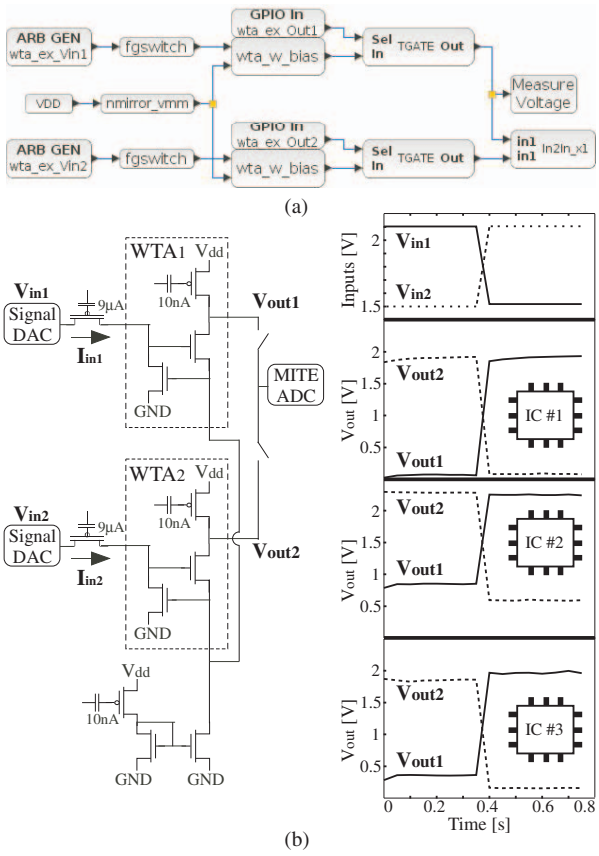

Fig. 7. Analog circuit design and on-chip experiment on multiple ICs. (a) WTA circuit design in XCOS. Two signal DAC blocks and a MITE ADC block with T-gates and GPIO signals are used for inputs and outputs, respectively. (b) Two-input WTA circuit and the results from three calibrated ICs. A WTA circuit has lower output voltage where the input current I_{in} is higher than another WTA's input current, as shown in the results.

References

[1] S. George, S. Kim, S. Shah, J. Hasler, M. Collins, F. Adil, R. Wunderlich, S. Nease, and S. Ramakrishnan, "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.

[2] 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.

[3] 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.

[4] 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.

[5] 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 Journal of Solid-State Circuits*, vol. 43, no. 12, pp. 2759–2768, Dec 2008.

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

[7] 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.