Tuning of Multiple Parameters With a BIST System

Sahil Shah, Student Member, IEEE, Jennifer Hasler, Senior Member, IEEE,

Abstract—This paper present a low-power Built-In Self Test (BIST) system to compensate for mismatch and variations in CMOS IC. The system is designed and compiled on a lowpower Field Programmable Analog Array (FPAA) fabricated on a 350nm CMOS process. A second-order band pass filter is used as a Device Under Test (DUT). A set of 12 parallel filter banks are compiled on three different FPAA chips and compensated for local mismatch and variations in the bias current, and mismatch in parasitic and routing capacitances. The band pass filters are biased in subthreshold regime and the system of 12 parallel filter banks consumes power of 7.072μ W. The proposed tuning algorithm reduces variation in center frequency to 5.07% compared to a variation of 10.16% when they are not tuned, variation in quality factor to 8.52% from 13.86%, and variation in the gain at the center frequency to 3.83% from 21.04%. These values are deviation of the parameters from its mean for the data taken from three different FPAA chips fabricated on the same wafer.

Index Terms—FPAA, Continuous-Time Filters, BIST, Mismatch.

I. BUILT-IN SELF TEST ON FPAA

T HE need for a low-power System-On-Chip (SOC) signal processing has increased with the growth in Internet of Things devices for biomedical applications. Biomedical systems for continuous monitoring of the subject require systems and circuits which are low-power and computationally efficient [1]. Such biomedical systems, for processing acoustic signals recorded from knee-joint, see extensive use of lowpower continuous time filters as a front end for performing efficient processing [2] [3].

One of the major issues in implementing an on-chip continuous-time filter is to keep its frequency response stable. It has been shown that due to mismatch and variation in the fabrication process the frequency response could vary by up to 50% [4]. In the case of multiple filter banks or a larger system these variations and mismatch often lead to lower efficiency and higher design margins, thus the need for an automatic Built-In Self Test (BIST) system which could reduce the variation and mismatch while implementing multiple filter banks. Figure 1(a) shows the usual approach for such an automatic tuning system. This approach involves tuning a single band-pass filter and reducing either variation in center frequency or quality factor.

This work proposes a system which could automatically tune multiple filter banks and could reduce the variation/mismatch in center frequency, quality factor, gain at the center frequency, bandwidth of amplitude detectors, time

Frequency tuning

Second Order Filter

Tuning Loop

Tuning 2

parameters

Fig. 1. System diagrams for a BIST. (a) Usual implementation of BIST system for tuning quality factor and center frequency. This approach involves tuning a single parameter for a single band pass filter using an off-chip reference. (b) The proposed system for BIST to tune multiple parameters. The system is implemented on a mixed signal FPAA. The proposed algorithm tunes 66 parameters, the center frequency, quality factor, gain at the center frequency, DC offset, bandwidth of amplitude detectors, and time constant of LPFs.

constant of Low Pass Filters (LPF) and DC offset of the filter bank chain. In general, it could be used to tune multiple parameters on a chip. Figure 1(b) shows such a system where these parameters are tuned using a tuning loop. Here, a set of 12 parallel second-order band-pass filters [5] are used as Devices Under Test (DUT). This system is compiled, using open-source Xcos/Scilab tools [6] onto a large-scale Field Programmable Analog Array (FPAA) [7] and routed using modified version of Versatile Place and Route (VPR) [8]. Figure 1(b) also shows amplitude detectors and LPFs as a part of the filter bank chain. Minimum detector and LPFs are used to track the amplitude of the band-pass filter, thus measuring the center frequency. These parameters, bandwidth in the case of amplitude detectors and time constant in the case of LPFs, are also tuned. The system is then tested on three different FPAA chips to evaluate the accuracy of the algorithm and portability of the code. The programmability and reconfigurability offered by a FPAA allows us to tune and select multiple parameters in a system before using it for a desired application.

Section II gives a brief overview of a FPAA in general and



1

S.Shah and J. Hasler are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (jennifer.hasler@ece.gatech.edu).

Manuscript received September 3, 2016;Revised November 29, 2016; Accepted Jan 9, 2016



Fig. 2. General architecture of a large-scale FPAA [7] designed and fabricated on a standard 350nm CMOS process. The core FPAA fabric is made of Computational Analog Blocks (CAB) shown as (A) and Computational Logic Blocks (CLB) shown as D. Designs are compiled on to a FPAA via a USB. A $8k \times 16$ SRAM is used as a program memory, where the program to be executed by the microprocessor is stored, and a data memory of $8k \times 16$. Also shown are the contents of a CAB and a shift register, used for scanning multiple inputs/outputs of a CAB, as a part of the routing infrastructure.

the FPAA used in this work. In Section III, the variation and mismatch found in a floating gate based FPAA is described, in particular while designing a larger system is discussed in detail. The proposed algorithm and its tuning capabilities are described in Section IV. The use of this algorithm to tune filters on three different FPAAs is shown in Section V. The deviation of each parameter from its mean is also presented in this section. Section VI summarizes the performance of our system and compares it to other adaptive continuous time filters. This section also discusses the use of such a system in large-scale neuromorphic and biomedical systems where lowpower and efficiency are important factors.

II. OVERVIEW OF FPAA

FPAA are poised to revolutionize analog and neuromorphic systems the same way FPGAs revolutionized digital systems, by making prototyping cost effective and shortening the test cycles [9]–[11]. Also, Floating Gate (FG) based FPAAs [12], due to their reconfigurability, have the potential of operating beyond the energy efficiency wall [13].

In this work the proposed design uses a mixed signal FPAA, shown in Fig. 2, having both analog, Computational Analog Block (CAB), and digital, Computational Logic Block (CLB). The FPAA consists of 98 CABs and 98 CLBs. CABs and CLBs are connected using manhattan style routing composed of Connection (C) and Switch (S) blocks. These interconnects allow analog and digital blocks to interact with each other thus leveraging computational capabilities of analog and digital circuits. Interconnect switches are composed of nonvolatile floating gate transistors which are programmed using hot-electron injection and globally erased using Fowler-Nordheim tunneling. The programming infrastructure, composed of DACs and an ADC, is controlled using a low-power, open source MSP430 microprocessor. The microprocessor has a controllable frequency of 0-50 MHz. A $8K \times 16$ SRAM is used to store the program to be executed by the microprocessor and a separate data memory of $8K \times 16$ is also present. As a part of the infrastructure there are sixteen 7-bit DACs for generating signals. These DACs could be routed to the FPAA fabric to work as an arbitrary waveform generator. The 14bit ramp ADC which is used for measuring the current of a floating gate, during the programming phase of the FPAA, could be routed to the FPAA fabric during run mode. In that case, it would behave as a data acquisition device and store the output on the available data memory.

Figure 2 shows basic elements of a CAB. Inputs and outputs of the CAB elements can be connected via a local routing, instead of a global routing, which has a reduced parasitic capacitance associated with it. A CAB consists of multiple Operational Transconductance Amplifiers (OTA) with the ability to select between wide linear range and high gain amplifier. Current bias of the OTA is set using a FG pFET transistor. The programming infrastructure enables programming the bias current values from 30pA to 10μ A. Thus a FPAA allows for multiple parameters, like linearity, gain, and power to name a few, to be tuned, depending on the application.



Fig. 3. Mismatch in different parameters for a filter bank chain with an example of LPF. The output of the blocks are vectorized, with N=12 for this work. An example of variation in two parameters, in this case f_{-3dB} of the LPFs, is shown. For the continuous-time filter the number of parameters to be tuned are 66. In general these parameters could be more than 1000, especially in a reconfigurable system where available parameters are \approx 50K, and hence the need for an automated tuning system.

The shift register (analog signal scanner) block, shown in Fig 2, is part of the routing infrastructure. It is made of 16

$$\frac{Vout}{Vin} = \frac{s^2 C_1 C_2 - s G_{m2} C_1}{s^2 (C_p C_T - C_2^2) + s (G_{m1} C_p + G_{m2} C_2 - G_{m1} C_2) + (G_{m1} G_{m2})}$$
(1)

serial to parallel D flip-flop. This allows us to measure and observe multiple outputs/inputs without having to increase the routing overhead. The clock and the data bus of the shift register can be routed to a general purpose I/O, which is controlled by the microprocessor. Input and output bus of the shift register can also be routed to DACs and ADCs outside the FPAA fabric. Thus the shift register is controlled digitally but is able to measure multiple analog signals.

III. MISMATCH, VARIATION AND PROGRAMMING THE G_m

As the feature size of a CMOS process scales, mismatch and variation in the threshold voltage (V_{T0}) of the transistor and parasitic capacitance can result in a significant overhead while designing a larger system. One of the hypothesized factors for the energy efficiency wall is due to larger components used for reducing the variation [13]. In the case of an FPAA or an FPGA the variations also depend on placement and routing of the components. The problem of variation and mismatch is usually addressed by using larger transistor sizes, using common centroid methods to layout critical components and various DC offset cancellation techniques [14]. Variation and mismatch in the transistor and parasitics can be easily mitigated by injecting a charge on to the floating gate [15]. Figure 3 shows some of the variations and mismatch found in our system. The parameters shown here are a subset of mismatch and variation found in a large-scale reconfigurable system. In general, the number of parameters could be larger than 1000 in an ASIC and about 50K in a reconfigurable FPAA. Thus for an efficient use of resources, an automatic tuning and a calibration system is necessary which could tune multiple parameters.

In the case of a continuous-time filter implemented on an FPAA, the source of mismatch is due to variations in parasitic capacitance, which is a part of the local routing in the CAB, depicted by C_2 in Fig. 4(a), variations in global routing based on the placement by VPR shown in Fig. 5, and mismatch between two transistors used for indirect programming of the floating gate shown in Fig. 4(b). Indirect programming of the transistor [16] reduces the parasitic capacitance and extra switches for programming at the cost of increased threshold voltage mismatch. The transistor used for biasing the OTA are relatively small (W/L= $\frac{6\mu m}{2\mu m}$), to increase the density of the CAB, and no special layout techniques have been used for reducing the mismatch, since programmability of the floating gate can compensate for this mismatch. The circuit used for the second-order band-pass filter is shown in Fig. 4(a). The transfer function of the band-pass filter is given by (1).

In (1) C_p is a summation of C_L , load parasitic capacitance, and C_2 . The feedback capacitance C_2 is a parasitic routing capacitance to reduce the biasing current and thus the power consumed. C_T is the summation of C_2 , C_W (parasitic capacitance at the input of G_{m2}), and C_1 which is the input capacitance. The transconductance G_{m1} sets the lower frequency pole and feed-forward transconductance G_{m2} sets the high frequency pole. The quality factor and gain at the center frequency of a band-pass filter is given by

$$Q = \frac{\sqrt{C_T * C_p - C_2^2}}{C_L * \sqrt{\frac{G_{m1}}{G_{m2}}} + C_2 * \sqrt{\frac{G_{m2}}{G_{m1}}}} \qquad A = \frac{-C_1}{C_2} * \frac{1}{1 + \frac{G_{m1} * C_L}{G_{m2} * C_2}}$$

The time constant for the lower frequency pole and higher frequency pole is given by

$$F_{low} = \frac{C_2}{G_{m1}}$$
 $F_{high} = \frac{C_T * C_p - C_2^2}{G_{m2} * C_2}$

This allows us to compensate the Q, gain, higher and lower frequency pole of the band pass filter using the transconductance G_{m1} and G_{m2} .

The circuit is built using standard components present in a CAB described in Section II and is fully reconfigurable, as opposed to a custom design. The transconductance used in the filter structure is a general 9-T OTA structure shown in Fig. 4(b) and has a floating gate input to compensate for the input DC offset and also allows for a wider linear range. Thus, transconductance G_{m1} and G_{m2} are smaller compared to a non Floating Gate OTA (FGOTA) because of the presence of a capacitive divider at the input of a FGOTA, with input capacitance of 192fF. The feedback OTA (G_{m1}) provides a unity gain feedback at lower frequencies, and therefore the output DC is set by V_{ref} because of unity gain feedback around the amplifier.

Figure 4(c) shows variation in frequency response of 12 parallel continuous-time filters, placed by VPR tool as shown in Fig. 5(a). Continuous-time filters in Fig. 4(c) were programmed using the same current values, by measuring indirectly the bias current of the programming transistor and not the transistor in circuit. FG transistors are calibrated for the global variation [17]. Calibration allows for compensation of global mismatch in the FPAA fabric, as well as mismatch in the programming infrastructure. Thus the variation, seen in Fig. 4(c), is dominantly because of the mismatch in parasitic capacitance, local threshold voltage mismatch between the programming and the transistor used in the circuit, and to a certain extent the finite resolution of the measurement infrastructure. Figure 4(b) shows two pFET structure used for indirect programming, which is the source of local threshold voltage mismatch. Figure 4(d) shows variation of quality factor, center frequency and gain at the center frequency. Variation values reported here are the difference between the maximum and minimum values. A variation of 107 Hz in center frequency, 5.1 dB in gain at the center frequency, and 0.9 in quality factor of the filters was observed. These variations could lead to significant errors while using them for analog computation. Also, compensating for these mismatches and variation without an automated system would be long



Fig. 4. Typical variation and mismatch measured using an FPAA. (a) Schematic of a second-order band-pass filter (C^4) . (b) The schematic of the OTA used in the G_m -C filter is shown. The OTA has a floating gate input to compensate for the input offset mismatch. The bias current, which controls the G_m of the OTA, is programmed using an indirect floating gate. (c) The measured frequency response of 12 filter banks programmed with the same bias current. d) The variation in Q, f_c and A are shown. These variation are due to mismatch in the current biases of the OTA, local mismatch between the programming transistor and the one used in the circuit, and mismatch in the capacitance.



Fig. 5. (a) Output of a modified versatile place and route [8]. Placement of 12 parallel filter banks and their routing to 16-bit shift registers implemented in the routing fabric of the FPAA. There are two shift registers used here for characterizing the frequency response of the filters after tuning. Load capacitance for each filter bank, due to routing length and the number of C and S blocks used, is different and is one of the sources for variation. (b) Die photograph of a SoC FPAA. Twelve CAB used are highlighted on both VPR output and the die photo.

and error prone. As the system scales to a larger design, the number of tuning parameters would increase and thus the need for an automated tuning system which could handle multiple parameters over multiple chips. To demonstrate the effect of injection on frequency, a LPF composed of an OTA in a source follower configuration which is shown as an inset in Fig. 6 was used. Figure 6 shows the change in frequency response with injection for a set drain to source voltage of the biasing transistor. Subthreshold current through a floating gate transistor, used as a biasing transistor of an OTA, in saturation regime is, given by [18], $I_s = I_{th} e^{(\kappa_p (V_{dd} - V_{fg} - V_{T0}))/U_T} e^{(V_{dd} - V_s)/U_T}$ where κ_p is the fractional change in pFET's surface potential due to the change in V_{fg} , and U_T is the thermal voltage. V_s is the source voltage of a floating gate pFET. The source voltage of a floating gate transistor is set to 6 V during injection. Depending on the distance from the target frequency, either a drain voltage of 1.02 V or 0.78 V is used. The drain voltage,

for all the programming floating gate transistors on the FPAA, is set using a DAC controlled by the processor. Higher drain voltage, thus a lower source to drain voltage, allows for a finer control over the frequency, where as a lower drain voltage will allow us to reach the target faster. A simple model for hotelectron injection is given by $I_{inj0} \left(\frac{I_s}{I_{s0}}\right)^{\alpha} e^{-\Delta V_d/V_{inj}}$ where I_{inj0} is the injection current when a floating gate operates with current reference I_{s0}, V_{inj} is a device and bias dependent parameter, and α is $1 - \frac{U_T}{V_{inj}}$. Inset in Fig. 6(a) and (b) shows a change in frequency at -3dB attenuation of the LPF with each injection. A Vdrain of 1.02 V resulted in a change of 2.7 Hz, over 5 injection pulses of $20\mu s$. This allows for a finer control over the frequency. A Vdrain of 0.78 V results in a change of 11 Hz which is used for a coarser control over the frequency. In the case of Fig. 6, tuning was performed in open loop to demonstrate the effects of injection on the bias and frequency.



Fig. 6. Measured changes in the frequency response of a LPF with hot-electron injection. (a) Variation of frequency with injection for a source to drain voltage of 4.98 V. This allows us for a finer control over the change in frequency of the pole. The change in f-3dB frequency with injection is also plotted in the inset (b) Variation of frequency with injection for a source to drain voltage of 5.22 V. This allows us for a coarser control over frequency of the pole.



Fig. 7. Measured output of 12 parallel minimum detectors after tuning their bandwidth are plotted along with its input. The DC Variation is subtracted from the transient response and plotted in the inset. A maximum variation of 0.4V was observed which is tuned when the system is compiled with the band-pass filter and LPF.

In the proposed algorithm, tuning is performed in a closed loop, where after each injection the distance from the target is calculated by measuring the amplitude at the target frequency.

IV. ALGORITHM FOR MISMATCH COMPENSATION

The tuning algorithm measures the amplitude at the output of the filter bank chain to determine the distance from its target frequency. The core of the algorithm is shown in Fig. 8(a). Since an amplitude detector and an LPF are used for measuring the amplitude, they have to be calibrated and tuned before tuning the band-pass filter. Reconfigurability of the FPAA allows us to test them separately and store the tuned parameters on the SRAM. Hence, as seen in Fig. 8(a), after calibrating the minimum detector and LPF, the compiled designs are tunneled, that is, a global erase is done on the FPAA fabric.

Bandwidth of the minimum detector should be above the operating range of the band-pass filter. In this work, all of them are tuned to operate at a maximum input frequency of 5KHz. Figure 7 shows the output of 12 parallel amplitude detectors tuned to work at 5KHz. A max variation of 0.4V was observed in the output DC value, which is tuned when the whole system

is compiled. If the center frequency of each filter is known a priori, the amplitude detectors could be tuned accordingly to save power, that is, individually tune each amplitude detector to operate just a little above the center frequency of the bandpass filter.

The next step involves compiling the tuned amplitude detectors with the LPFs and then tuning their time constants. Here, the time constant of the LPF is of a interest rather than its bandwidth because in this configuration it is used to reduce the ripples at the output of the filter bank chain. Figure 8(b) shows the schematic of a minimum detector and a LPF. Initially, all the LPFs are biased at a very low frequency except for the first filter, which is used as a reference and biased with the target time constant. The response of the LPF follows $|LPF| = \frac{1}{1+\tau s} \approx \frac{1}{\tau s}$. It is easier to consider the amplitude at -15 dB attenuation since the time constant (τ) of the filter is of the interest. The output of the first filter is measured at -15 dB of attenuation, using a 14-bit ADC, and stored on the SRAM. This value is used as a reference for tuning other LPFs. Figure 8(c) shows the frequency response of the LPFs. Figure on top is of LPFs biased with the same current value, to show the variation in frequency response without tuning. The bottom figure is of the results obtained after using the proposed tuning algorithm. The tuning algorithm starts at a lower frequency and programs the biases till the target frequency is achieved, within certain error margin. After each injection, the output amplitude is measured and the distance from the reference filter is calculated. Based on the distance, a Vdrain value is selected for the injection, to reach the target faster or to have a finer control over the frequency/time constant. Fig. 8(d) shows the variation in LPF response which is reduced from 126 Hz to 36 Hz at -15 dB attenuation. Generally the time constant (τ) of the reference filter could be selected according to the application.

The stored parameters are then used while compiling the final design, in general this would be done as a part of a larger system, since a shift register could be used to observe intermediate points. A block diagram of the compiled final design is shown in Fig. 10(a). On-chip processor controls the shift register, a 14-bit ramp ADC and a DAC while



Fig. 8. LPF tuning results. (a) The $Tau(\tau)$ of the LPF is tuned here using I_{bias} of the OTA. (b) The circuit schematic for the LPF and the minimum detector for detecting the minimum amplitude at the output of LPF. (c) Frequency response of LPF banks. The LPFs are characterized with a biasing current of 0.6nA. Tuned response is shown below where the variation is low. (d) Variation in frequency of LPFs, at -15 dB attenuation, before and after the tuning algorithm was applied. Time constant is calculated using the frequency at -15 dB attenuation.



Fig. 9. DC offset, Q, f_c and gain results. (a)Tuning of DC offset by setting the input offset of the FGOTA. Q, f_c and gain are tuned using G_{m1} and G_{m2} . (b) Frequency response of the tuned band-pass filter is plotted. (c) DC offset of the signal chain is characterized here. DC offset is compensated by tuning the FGOTA, having transconductance of G_{m2} , used in the band-pass filter. After compensation the variation in DC offset is reduced to 32 mV. (d) Variation in Q, f_c and gain after tuning. Q varies by 0.6 as opposed to 0.9 without tuning. f_c variation is reduced to 42 Hz compared to 107 Hz and variation in gain at the center frequency is reduced to 1 dB from 5.1 dB.

executing the algorithm. Figure 10(a) also shows the DUT with vectorized outputs, where N is 12 for this work. In general, the system can be scaled as needed to a larger number of filter banks only constrained by the number of CABs in the FPAA. Figure 10(b) shows an example flow chart of the tuning algorithm.

Initially the filters are compiled with a low frequency bias except for the first filter bank, which is used as a reference. Again, the parameters of the reference filter could be selected depending on application. The tuning algorithm reduces the variation of center frequency, quality factor, gain at the center frequency, and the DC offset with respect to the reference filter. The first step after compiling the design is to reduce the DC offset. Without DC offset reduction, the tuning algorithm will have large errors due to the fact that the system is measuring the amplitude and detecting the minimum value of the output to determine the frequency. The DC offset is reduced by programming the input floating gate of the bandpass filter and measuring it at the output of the LPF. Thus the system can control the offset of the whole chain. DC offset of 12 filter banks, scanned using the shifter register, can be



Fig. 10. (a) Block diagram of the compiled design. The DUT used for the experiment is shown with vectorized interconnects, where for this work N is 12. (b) A flow chart of the tuning algorithm. Output of the shift register is measured using a 14-bit ramp ADC and stored on the data memory available. A 7-bit DAC controlled by the microprocessor is used for generating a sine wave, at a desired frequency (F Hz).

seen at the top of Fig. 9(c). The maximum variation can be reduced to 32mV from 0.9 V, after using the tuning algorithm, shown in bottom of Fig. 9(c). After reducing the DC offset, the tuning algorithm reduces the variation in center frequency, quality factor and gain at the center frequency of the filters by measuring them with respect to the reference filter. The reference filter is characterized by measuring its output at F_{low} and F_{high} , by generating a signal at those frequencies with a DAC. Tuning of the rest of the filter bank is done in two steps, first tuning the higher frequency pole and then the lower frequency pole. Injection is performed, as discussed earlier, to vary the biases to change the feed-forward G_{m2} and the feedback G_{m1} . The filters are injected until they are around an acceptable error from the value of the reference filter. Figure 9(b) shows frequency response of tuned filter banks. Variation of filter bank parameters is shown in Fig. 9(d). Specification of the DUT is shown in Table 1. A set of 12 parallel bandpass filters, amplitude detectors and LPFs consumes power of 7.072 μ W. Area is reported in terms of number of CABs used by the compiled design, since that is more relevant while designing on a FPAA.

TABLE 1Specification of the DUT system

Parameter	Values
Area	12 CAB
Power of 12 filter banks	$7.072\mu W$
Technology	350nm
Power Supply	2.5V

V. Algorithm over different FPAA

Open source high level tools [6], the programming algorithm [15] and the calibration of chip-to-chip variation in the programming infrastructure [17] enables compiling the same design on different FPAAs. The design was compiled on three different chips, fabricated on the same wafer, to measure the variation and mismatch. This also allows us to test the portability of the algorithm. Table 2 shows the variation in parameters when compiled and programmed with the same current value, before using the tuning algorithm. They are deviation of each value from its mean, averaged over three chips. Table in Fig. 11 shows the variation in parameters for each FPAA.

The same algorithm was applied to filter banks compiled on all three chips. Procedure discussed in Section IV was followed, with the filter in the first CAB serving as a reference filter. The bandwidth of amplitude detectors and the time constants of LPFs were tuned first and then the DC offsets of DUT chain were tuned. Figure 11 depicts tuning of 66 parameters for three FPAA ICs using the algorithm. The table in Fig. 11 shows percentage deviation of parameters from their mean. In Fig. 11(a),(b) and (c), absolute variation of center frequency, gain at the center frequency and quality factor, after tuning, is plotted. Table 2 shows the deviation of each filter bank value from its mean, as an average over all three chips. A large reduction was observed in variation of center frequency, gain at the center frequency and DC offset. Reduction in variation of Q was nominal due to finite SNR of the ADC and small percentage of errors during adaptation of floating gates. Also, C_L and C_T , vary for different CABs and their route.

 TABLE 2

 Deviation of the values from its mean

Parameter	Untuned	Tuned
Frequency Variation	10.16%	5.07%
Quality Factor Variation	13.86%	8.52%
Gain Variation	21.04%	3.83%
DC offset Variation	0.95 V	29 mV

VI. SUMMARY DISCUSSION AND COMPARISON

An on-chip BIST system was presented on a FPAA fabricated on a 350nm CMOS process. The DUT consisted of 12 filter bank chain, widely used for real time signal processing application and in large neuromorphic systems. The algorithm was tested on three different chips to test its performance and portability. Variation of parameters for each chip is shown in Fig. 11. The proposed system also tunes the bandwidth of the amplitude detectors and the time constant of the LPFs, which are critical parts of the system. The system uses an on-chip DAC and an ADC to generate the necessary signals and to measure the output of the filter bank, via a compiled 16-bit shift register. The compiled system consumes a power of 7.072μ W.

The proposed system automatically tunes multiple parameters to compensate for local mismatch and routing capacitance but does not consider variation in temperature which have been studied in detail elsewhere [23], [24]. Typically one compensates for temperature variation by biasing the FG switches and FG bias of the OTA using a bootstrap current source [24]. Variation in power supply, PSRR, is typically

Ref	Process	Tuning Parameters	Power Consumption (Normalized per filter)	Design
This Work	350nm	66	152.25nW (200 Hz)	Fully reconfigurable
[19]	$0.8 \mu m$	$2 (F_c \text{ and } Q)$	2.5µW (100 Hz)	Custom design with programmable bias
[20]	$1.2\mu m$	$1 (F_c)$	27.86µW (0.83 Hz)	Fully custom design
[21]	350nm	32 (F_c and A_v or F_c and Q)	198nW (200 Hz)	Custom design with programmable bias
[22]	350nm	$3 (F_c, A_v \text{ and } Q)$	290nW (30 Hz)	Fully Custom design

 TABLE 3

 COMPARISON OF LOW-POWER CONTINUOUS-TIME FILTERS



Fig. 11. Tuning for three different FPAA chips using the algorithm proposed in this work. Table shows the values for variation in parameters before and after tuning, for each chip. Variation in center frequency (a), gain (b) and quality factor (c) for 12 filters over three different chips after tuning.

controlled using a precision voltage reference [25]. Previous work has addressed the precision of FG programming [15] and the FG drift and charge leakage [26] [27], and further discussion is beyond the scope of this work.

Comparison of this work with other state of the art lowpower adaptable continuous-time filters is shown in Table 3. The power consumption reported was per filter with its center frequency. The power consumption of the band-pass filter in this work is 152.25nW at 200 Hz. The compiled system and the algorithm tunes 66 parameters, that is, center frequency, quality factor, DC offset, τ of LPFs, bandwidth of amplitude detectors, and gain at the center frequency of 11 parallel filter banks, excluding the reference filter. Reconfigurability of a FPAA enables larger community to adopt such a system and use it remotely using the tool set developed in a single environment [6] and remote system [28]. The proposed algorithm would enable such a remote system to be used widely without using expensive test equipment.

Low-power continuous-time filters, such as the one used in this work, have been extensively used for biomedical systems and in large scale neuromorphic systems. In [2] such a system is used for extracting features from the output of an accelerometer. In case of [3], the system analyzes acoustic signals from the knees. Their use in emulating a silicon cochlea has evolved over the years from the first analog silicon cochlea [29], to an improved version where the silicon cochlea has an increased linearity and stability [30] and using compatible lateral bipolar transistors for reduced mismatch [31], in active 2-D cochlea with nonlinear properties of biological cochlea [32], to their use in more recent large scale binaural spatial audition sensors [33]. They have also seen their use as a Fourier processor, and as a front end for speech detection [34], [35]. Such large scale neuromorphic system would require precise tuning of fewer than 100 parameters.

The number of parameters available for a FPAA, just like in case of an FPGA, is more than an ASIC because of the reconfigurability of the SoC. The density of parameters, in case of a FG based FPAA, is high since the routing infrastructure are also used as computational and tunable elements, which in case of an FPGA is normally considered an overhead. In general, the number of tunable parameters is restricted by available CAB blocks, CLB blocks, precision of routing elements, available measurement, storage infrastructure and mismatch caused by capacitance. Also, number of parameters would be restricted by their orthogonality to each other. The parameters for a custom analog chip is restricted by the density of tunable parameters. If FG are not being used one approach would be to use small non-linear DACs, either current steering or voltage, and calibrating them ahead of time. In [36], such an approach is used for storing the weights of the synapses of a neuron. In case of [37] a 5-bit DAC is used to supply the bias of G_m -C elements. In case of reconfigurable analog systems, such as the one used for speech processing [38], the number of parameters which could be tuned are more than 50K [7].

REFERENCES

- C. N.Teague *et al.*, "Novel methods for sensing acoustical emissions from the knee for wearable joint health assessment," *Transactions on Biomedical Engineering*, 2016.
- [2] S.Shah et al., "Reconfigurable analog classifier for knee-joint rehabilitation," *IEEE Engineering in Medicine and Biology Society*, 2016.
- [3] S.Shah et al., "A proof-of-concept classifier for acoustic signals from the knee joint on a FPAA," *IEEE SENSORS*, 2016.
- [4] Y.Tsividis, "Continuous-time filters in telecommunications chips," *IEEE Communications Magazine*, vol. 39, no. 4, pp. 132–137, Apr 2001.
- [5] D. W.Graham et al., "A low-power programmable bandpass filter section for higher order filter applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 6, pp. 1165–1176, June 2007.
- [6] 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.
- [7] S.George et al., "A programmable and configurable mixed-mode FPAA SoC," *IEEE Transactions on Very Large Scale Integration (VLSI) Sys*tems, vol. 24, no. 6, pp. 2253–2261, June 2016.
- [8] J.Luu et al., "VTR 7.0: Next generation architecture and CAD system for FPGAs," ACM Trans. Reconfigurable Technol. Syst., vol. 7, no. 2, pp. 6:1–6:30, Jul. 2014.
- [9] B.Rumberg et al., "Hibernets: Energy-efficient sensor networks using analog signal processing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 321–334, Sept 2011.
- [10] N.Suda et al., "A 65 nm programmable analog device array (PANDA) for analog circuit emulation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 2, pp. 181–190, Feb 2016.
- [11] N.Guo et al., "Energy-efficient hybrid analog/digital approximate computation in continuous time," *IEEE Journal of Solid-State Circuits*, vol. PP, no. 99, pp. 1–11, 2016.
- [12] A.Basu *et al.*, "RASP 2.8: A new generation of floating-gate based field programmable analog array," in 2008 IEEE Custom Integrated Circuits Conference, Sept 2008, pp. 213–216.
- [13] J.Hasler and H. B.Marr, "Finding a roadmap to achieve large neuromorphic hardware systems," *Frontiers in Neuroscience*, vol. 7, no. 118, 2013.
- [14] P. R.Gray et al., Analysis and Design of Analog Integrated Circuits. Wiley, 2001.
- [15] 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. PP, no. 99, pp. 1–9, 2016.
- [16] 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.
- [17] S.Kim, S.Shah, and J.Hasler, "Calibration of floating-gate FPAA system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Submitted.
- [18] C.Mead, Analog VLSI and Neural Systems. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [19] E.Rodriguez-Villegas, A.Yufera, and A.Rueda, "A 1.25-V micropower gm-c filter based on FGMOS transistors operating in weak inversion," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 100–111, Jan 2004.

- [20] A.Veeravalli, E.Sanchez-Sinencio, and J.Silva-Martinez, "A CMOS transconductance amplifier architecture with wide tuning range for very low frequency applications," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 6, pp. 776–781, Jun 2002.
- [21] B.Rumberg and D. W.Graham, "A low-power and high-precision programmable analog filter bank," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 4, pp. 234–238, April 2012.
- [22] O.Omeni, E.Rodriguez-Villegas, and C.Toumazou, "A micropower CMOS continuous-time filter with on-chip automatic tuning," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 4, pp. 695–705, April 2005.
- [23] C. R.Schlottmann and P. E.Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The fpaa implementation," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 403–411, Sept 2011.
- [24] V.Srinivasan et al., "A floating-gate-based programmable CMOS reference," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 11, pp. 3448–3456, Dec 2008.
- [25] B. K.Ahuja *et al.*, "A very high precision 500-na CMOS floating-gate analog voltage reference," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2364–2372, Dec 2005.
- [26] V.Srinivasan et al., "A precision CMOS amplifier using floating-gate transistors for offset cancellation," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 2, pp. 280–291, Feb 2007.
- [27] R. R.Harrison *et al.*, "A CMOS programmable analog memory-cell array using floating-gate circuits," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 4–11, Jan 2001.
- [28] J.Hasler *et al.*, "Remote FPAA system setup enablingwide accessibility of configurable devices," *Journal of Low Power Electronics and Applications*, vol. Accepted, 2016.
- [29] R. F.Lyon and C.Mead, "An analog electronic cochlea," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1119–1134, Jul 1988.
- [30] L.Watts et al., "Improved implementation of the silicon cochlea," IEEE Journal of Solid-State Circuits, vol. 27, no. 5, pp. 692–700, May 1992.
- [31] A.vanSchaik, E.Fragnière, and E. A.Vittoz, "Improved silicon cochlea using compatible lateral bipolar transistors," in *Advances in Neural Information Processing Systems 8*, D. S.Touretzky and M. E.Hasselmo, Eds. MIT Press, 1996, pp. 671–677.
- [32] T. J.Hamilton et al., "An active 2-D silicon cochlea," IEEE Transactions on Biomedical Circuits and Systems, vol. 2, no. 1, pp. 30–43, March 2008.
- [33] S. C.Liu et al., "Asynchronous binaural spatial audition sensor with 2 x 64 x 4 channel output," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 4, pp. 453–464, Aug 2014.
- [34] M.Kucic et al., "A programmable continuous-time floating-gate Fourier processor," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 90–99, Jan 2001.
- [35] T.Delbruck et al., "Fully integrated 500uW speech detection wakeup circuit," in Proceedings of 2010 IEEE International Symposium on Circuits and Systems, May 2010, pp. 2015–2018.
- [36] S.Moradi and G.Indiveri, "An event-based neural network architecture with an asynchronous programmable synaptic memory," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 1, pp. 98–107, Feb 2014.
- [37] C. D.Salthouse and R.Sarpeshkar, "A practical micropower programmable bandpass filter for use in bionic ears," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 1, pp. 63–70, Jan 2003.
- [38] S.Ramakrishnan et al., "Speech processing on a reconfigurable analog platform," *IEEE Transactions on Very Large Scale Integration (VLSI)* Systems, vol. 22, no. 2, pp. 430–433, Feb 2014.

Sahil Shah and Jennifer Hasler Biography not available at the time of submission